IEC 62680-1

Edition 1.0 2013-09

INTERNATIONAL STANDARD

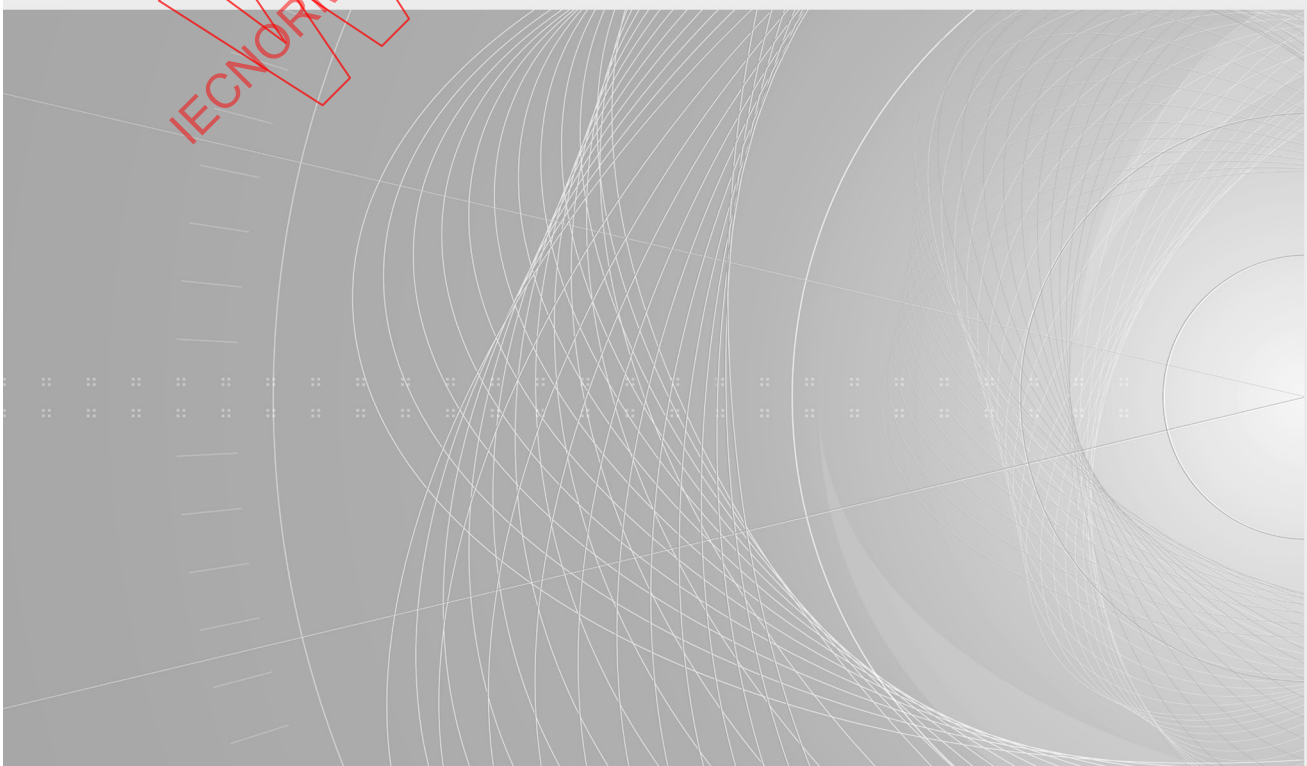
NORME INTERNATIONALE

colour inside

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Universal serial bus interfaces for data and power – Part 1: Universal serial bus specification, revision 2.0

Interfaces de bus universel en série pour les données et l'alimentation électrique – Partie 1: Spécification du bus universel en série, révision 2.0



IEC 62680-1:2013-09(en-fr)

THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2000 USB-IF

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from IEC, or USB-IF at the respective address given below. Any questions about USB-IF copyright should be addressed to the USB-IF. Enquiries about obtaining additional rights to this publication and other information requests should be addressed to the IEC or your local IEC member National Committee.

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

USB Implementers Forum, Inc.
3855 S.W. 153rd Drive
Beaverton, OR 97003
United States of America
Tel. +1 503-619-0426
Admin@usb.org
www.usb.org

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, …). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A propos de l'IEC
La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC
Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC - webstore.iec.ch/advsearchform
La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études, …). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished
Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc
Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org
Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 300 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 19 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

IEC 62680-1

Edition 1.0 2013-09

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Universal serial bus interfaces for data and power – Part 1: Universal serial bus specification, revision 2.0

Interfaces de bus universel en série pour les données et l'alimentation électrique – Partie 1: Spécification du bus universel en série, révision 2.0

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 35.200

ISBN 978-2-8322-1083-3

Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.

® Registered trademark of the International Electrotechnical Commission
Marque déposée de la Commission Electrotechnique Internationale

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

UNIVERSAL SERIAL BUS INTERFACES FOR DATA AND POWER –

Part 1: Universal serial bus specification, revision 2.0

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62680-1 has been prepared by technical area 14: Interfaces and methods of measurement for personal computing equipment, of IEC technical committee 100: Audio, video and multimedia systems and equipment.

The text of this standard is based on documents prepared by the USB Implementers Forum (USB-IF). The structure and editorial rules used in this publication reflect the practice of the organization which submitted it.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

The text of this standard is based on the following documents:

FDIS	Report on voting
100/2156/FDIS	100/2188/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

A list of all the parts in the IEC 62680 series, published under the general title *Universal serial bus interfaces for data and power* can be found on the IEC website.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

INTRODUCTION

The IEC 62680 series is based on a series of specifications that were originally developed by the USB Implementers Forum (USB-IF). These specifications were to submitted to the IEC under the auspices of a special agreement between the IEC and the USB IF.

The USB Implementers Forum, Inc.(USB-IF) is a non-profit corporation founded by the group of companies that developed the Universal Serial Bus specification. The USB-IF was formed to provide a support organization and forum for the advancement and adoption of Universal Serial Bus technology. The Forum facilitates the development of high-quality compatible USB peripherals (devices), and promotes the benefits of USB and the quality of products that have passed compliance testing.

ANY USB SPECIFICATIONS ARE PROVIDED TO YOU "AS IS, "WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE USB IMPLEMENTERS FORUM AND THE AUTHORS OF ANY USB SPECIFICATIONS DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OR INFORMATION IN THIS SPECIFICAITON.

THE PROVISION OF ANY USB SPECIFICATIONS TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

Entering into USB Adopters Agreements may, however, allow a signing company to participate in a reciprocal, royalty-free licensing arrangement for compliant products. For more information, please see:
http://www.usb.org/developers/docs/
http://www.usb.org/developers/devclass_docs#approved

IEC DOES NOT TAKE ANY POSITION AS TO WHETHER IT IS ADVISABLE FOR YOU TO ENTER INTO ANY USB ADOPTERS AGREEMENTS OR TO PARTICIPATE IN THE USB IMPLEMENTERS FORUM."

This series covers the Universal Series Bus interfaces for data and power and consists of the following parts:

IEC 62680-1, *Universal Serial Bus interfaces for data and power - Part 1: Universal Serial Bus Specification, Revision 2.0*

IEC 62680-2, *Universal Serial Bus interfaces for data and power - Part 2: USB Micro-USB Cables and Connectors Specification, Revision 1.01*

IEC 62680-3, *Universal Serial Bus interfaces for data and power - Part 3: USB Battery Charging Specification, Revision 1.2*

IEC 62680-4, *Universal Serial Bus interfaces for data and power - Part 4: Universal Serial Bus Cables and Connectors Class Document Revision. 2.0*

This part of the IEC 62680 series consists of several distinct parts:

- the main body of the text, which consists of the original specification and all ECN and Errata developed by the USB-IF;

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

CONTENTS

1 Chapter 1 Introduction 17
1.1 Motivation 17
1.2 Objective of the Specification 17
1.3 Scope of the Document 18
1.4 USB Product Compliance 18
1.5 Document Organization 18
2 Chapter 2 Terms and Abbreviations 19
3 Chapter 3 Background 25
3.1 Goals for the Universal Serial Bus 25
3.2 Taxonomy of Application Space 25
3.3 Feature List 26
4 Chapter 4 Architectural Overview 28
4.1 USB System Description 28
4.1.1 Bus Topology 28
4.2 Physical Interface 29
4.2.1 Electrical 30
4.2.2 Mechanical 30
4.3 Power 30
4.3.1 Power Distribution 31
4.3.2 Power Management 31
4.4 Bus Protocol 31
4.5 Robustness 31
4.5.1 Error Detection 32
4.5.2 Error Handling 32
4.6 System Configuration 32
4.6.1 Attachment of USB Devices 32
4.6.2 Removal of USB Devices 32
4.6.3 Bus Enumeration 32
4.7 Data Flow Types 33
4.7.1 Control Transfers 33
4.7.2 Bulk Transfers 33
4.7.3 Interrupt Transfers 33
4.7.4 Isochronous Transfers 33
4.7.5 Allocating USB Bandwidth 34
4.8 USB Devices 34
4.8.1 Device Characterizations 34
4.8.2 Device Descriptions 35
4.9 USB Host: Hardware and Software 37
4.10 Architectural Extensions 37
5 Chapter 5 USB Data Flow Model 38
5.1 Implementer Viewpoints 38
5.2 Bus Topology 39
5.2.1 USB Host 40
5.2.2 USB Devices 40

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.2.3 Physical Bus Topology 41
5.2.4 Logical Bus Topology 42
5.2.5 Client Software-to-function Relationship 43
5.3 USB Communication Flow 43
5.3.1 Device Endpoints 45
5.3.2 Pipes 46
5.3.3 Frames and Microframes 48
5.4 Transfer Types 48
5.4.1 Table Calculation Examples 49
5.5 Control Transfers 50
5.5.1 Control Transfer Data Format 50
5.5.2 Control Transfer Direction 51
5.5.3 Control Transfer Packet Size Constraints 51
5.5.4 Control Transfer Bus Access Constraints 52
5.5.5 Control Transfer Data Sequences 54
5.6 Isochronous Transfers 55
5.6.1 Isochronous Transfer Data Format 55
5.6.2 Isochronous Transfer Direction 55
5.6.3 Isochronous Transfer Packet Size Constraints 55
5.6.4 Isochronous Transfer Bus Access Constraints 57
5.6.5 Isochronous Transfer Data Sequences 58
5.7 Interrupt Transfers 58
5.7.1 Interrupt Transfer Data Format 58
5.7.2 Interrupt Transfer Direction 58
5.7.3 Interrupt Transfer Packet Size Constraints 58
5.7.4 Interrupt Transfer Bus Access Constraints 59
5.7.5 Interrupt Transfer Data Sequences 62
5.8 Bulk Transfers 62
5.8.1 Bulk Transfer Data Format 62
5.8.2 Bulk Transfer Direction 62
5.8.3 Bulk Transfer Packet Size Constraints 62
5.8.4 Bulk Transfer Bus Access Constraints 63
5.8.5 Bulk Transfer Data Sequences 64
5.9 High-Speed, High Bandwidth Endpoints 65
5.9.1 High Bandwidth Interrupt Endpoints 65
5.9.2 High Bandwidth Isochronous Endpoints 66
5.10 Split Transactions 67
5.11 Bus Access for Transfers 67
5.11.1 Transfer Management 68
5.11.2 Transaction Tracking 70
5.11.3 Calculating Bus Transaction Times 72
5.11.4 Calculating Buffer Sizes in Functions and Software 74
5.11.5 Bus Bandwidth Reclamation 74
5.12 Special Considerations for Isochronous Transfers 74
5.12.1 Example Non-USB Isochronous Application 75
5.12.2 USB Clock Model 77

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.12.3 Clock Synchronization 79
5.12.4 Isochronous Devices 79
5.12.5 Data Prebuffering 87
5.12.6 SOF Tracking 88
5.12.7 Error Handling 88
5.12.8 Buffering for Rate Matching 89
6 Chapter 6 Mechanical 91
6.1 Architectural Overview 91
6.2 Keyed Connector Protocol 91
6.3 Cable 92
6.4 Cable Assembly 92
6.4.1 Standard Detachable Cable Assemblies 92
6.4.2 High-/full-speed Captive Cable Assemblies 94
6.4.3 Low-speed Captive Cable Assemblies 96
6.4.4 Prohibited Cable Assemblies 98
6.5 Connector Mechanical Configuration and Material Requirements 98
6.5.1 USB Icon Location 99
6.5.2 USB Connector Termination Data 99
6.5.3 Series “A” and Series “B” Receptacles 100
6.5.4 Series “A” and Series “B” Plugs 103
6.6 Cable Mechanical Configuration and Material Requirements 106
6.6.1 Description 107
6.6.2 Construction 107
6.6.3 Electrical Characteristics 110
6.6.4 Cable Environmental Characteristics 110
6.6.5 Listing 110
6.7 Electrical, Mechanical, and Environmental Compliance Standards 110
6.7.1 Applicable Documents 116
6.8 USB Grounding 116
6.9 PCB Reference Drawings 116
7 Chapter 7 Electrical 120
7.1 Signaling 120
7.1.1 USB Driver Characteristics 123
7.1.2 Data Signal Rise and Fall, Eye Patterns 130
7.1.3 Cable Skew 139
7.1.4 Receiver Characteristics 139
7.1.5 Device Speed Identification 141
7.1.6 Input Characteristics 142
7.1.7 Signaling Levels 145
7.1.8 Data Encoding/Decoding 158
7.1.9 Bit Stuffing 158
7.1.10 Sync Pattern 160
7.1.11 Data Signaling Rate 160
7.1.12 Frame Interval 160
7.1.13 Data Source Signaling 161
7.1.14 Hub Signaling Timings 163

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.1.15 Receiver Data Jitter 164
7.1.16 Cable Delay 166
7.1.17 Cable Attenuation 167
7.1.18 Bus Turn-around Time and Inter-packet Delay 168
7.1.19 Maximum End-to-end Signal Delay 169
7.1.20 Test Mode Support 170
7.2 Power Distribution 171
7.2.1 Classes of Devices 171
7.2.2 Voltage Drop Budget 175
7.2.3 Power Control During Suspend/Resume 176
7.2.4 Dynamic Attach and Detach 177
7.3 Physical Layer 178
7.3.1 Regulatory Requirements 178
7.3.2 Bus Timing/Electrical Characteristics 178
7.3.3 Timing Waveforms 188
8 Chapter 8 Protocol Layer 191
8.1 Byte/Bit Ordering 191
8.2 SYNC Field 191
8.3 Packet Field Formats 191
8.3.1 Packet Identifier Field 191
8.3.2 Address Fields 192
8.3.3 Frame Number Field 193
8.3.4 Data Field 193
8.3.5 Cyclic Redundancy Checks 194
8.4 Packet Formats 194
8.4.1 Token Packets 194
8.4.2 Split Transaction Special Token Packets 195
8.4.3 Start-of-Frame Packets 200
8.4.4 Data Packets 201
8.4.5 Handshake Packets 202
8.4.6 Handshake Responses 203
8.5 Transaction Packet Sequences 204
8.5.1 NAK Limiting via Ping Flow Control 213
8.5.2 Bulk Transactions 217
8.5.3 Control Transfers 222
8.5.4 Interrupt Transactions 225
8.5.5 Isochronous Transactions 225
8.6 Data Toggle Synchronization and Retry 230
8.6.1 Initialization via SETUP Token 231
8.6.2 Successful Data Transactions 231
8.6.3 Data Corrupted or Not Accepted 231
8.6.4 Corrupted ACK Handshake 232
8.6.5 Low-speed Transactions 233
8.7 Error Detection and Recovery 234
8.7.1 Packet Error Categories 234
8.7.2 Bus Turn-around Timing 234

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8.7.3 False EOPs 235
8.7.4 Babble and Loss of Activity Recovery 236
9 Chapter 9 USB Device Framework 237
9.1 USB Device States 237
9.1.1 Visible Device States 237
9.1.2 Bus Enumeration 241
9.2 Generic USB Device Operations 241
9.2.1 Dynamic Attachment and Removal 242
9.2.2 Address Assignment 242
9.2.3 Configuration 242
9.2.4 Data Transfer 243
9.2.5 Power Management 243
9.2.6 Request Processing 243
9.2.7 Request Error 245
9.3 USB Device Requests 246
9.3.1 bmRequestType 246
9.3.2 bRequest 246
9.3.3 wValue 246
9.3.4 wIndex 247
9.3.5 wLength 247
9.4 Standard Device Requests 247
9.4.1 Clear Feature 249
9.4.2 Get Configuration 250
9.4.3 Get Descriptor 250
9.4.4 Get Interface 251
9.4.5 Get Status 251
9.4.6 Set Address 253
9.4.7 Set Configuration 253
9.4.8 Set Descriptor 254
9.4.9 Set Feature 254
9.4.10 Set Interface 255
9.4.11 Synch Frame 256
9.5 Descriptors 256
9.6 Standard USB Descriptor Definitions 257
9.6.1 Device 257
9.6.2 Device_Qualifier 259
9.6.3 Configuration 259
9.6.4 Other_Speed_Configuration 261
9.6.5 Interface 261
9.6.6 Endpoint 263
9.6.7 String 266
9.7 Device Class Definitions 267
9.7.1 Descriptors 267
9.7.2 Interface(s) and Endpoint Usage 267
9.7.3 Requests 268
10 Chapter 10 USB Host: Hardware and Software 269

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.1 Overview of the USB Host 269
10.1.1 Overview 269
10.1.2 Control Mechanisms 272
10.1.3 Data Flow 272
10.1.4 Collecting Status and Activity Statistics 273
10.1.5 Electrical Interface Considerations 273
10.2 Host Controller Requirements 273
10.2.1 State Handling 273
10.2.2 Serializer/Deserializer 274
10.2.3 Frame and Microframe Generation 274
10.2.4 Data Processing 275
10.2.5 Protocol Engine 275
10.2.6 Transmission Error Handling 275
10.2.7 Remote Wakeup 276
10.2.8 Root Hub 276
10.2.9 Host System Interface 276
10.3 Overview of Software Mechanisms 276
10.3.1 Device Configuration 276
10.3.2 Resource Management 278
10.3.3 Data Transfers 279
10.3.4 Common Data Definitions 280
10.4 Host Controller Driver 280
10.5 Universal Serial Bus Driver 281
10.5.1 USBD Overview 281
10.5.2 USBD Command Mechanism Requirements 282
10.5.3 USBD Pipe Mechanisms 284
10.5.4 Managing the USB via the USBD Mechanisms 286
10.5.5 Passing USB Preboot Control to the Operating System 288
10.6 Operating System Environment Guides 289
11 Chapter 11 Hub Specification 290
11.1 Overview 290
11.1.1 Hub Architecture 290
11.1.2 Hub Connectivity 291
11.2 Hub Frame/Microframe Timer 293
11.2.1 High-speed Microframe Timer Range 293
11.2.2 Full-speed Frame Timer Range 293
11.2.3 Frame/Microframe Timer Synchronization 294
11.2.4 Microframe Jitter Related to Frame Jitter 296
11.2.5 EOF1 and EOF2 Timing Points 296
11.3 Host Behavior at End-of-Frame 299
11.3.1 Full-/low-speed Latest Host Packet 299
11.3.2 Full-/low-speed Packet Nullification 299
11.3.3 Full-/low-speed Transaction Completion Prediction 299
11.4 Internal Port 300
11.4.1 Inactive 301
11.4.2 Suspend Delay 301

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.4.3 Full Suspend (Fsus) 301
11.4.4 Generate Resume (GResume) 301
11.5 Downstream Facing Ports 301
11.5.1 Downstream Facing Port State Descriptions 303
11.5.2 Disconnect Detect Timer 307
11.5.3 Port Indicator 308
11.6 Upstream Facing Port 309
11.6.1 Full-speed 309
11.6.2 High-speed 310
11.6.3 Receiver 310
11.6.4 Transmitter 313
11.7 Hub Repeater 314
11.7.1 High-speed Packet Connectivity 315
11.7.2 Hub Repeater State Machine 317
11.7.3 Wait for Start of Packet from Upstream Port (WFSOPFU) 319
11.7.4 Wait for End of Packet from Upstream Port (WFEOPFU) 319
11.7.5 Wait for Start of Packet (WFSOP) 319
11.7.6 Wait for End of Packet (WFEOP) 319
11.8 Bus State Evaluation 319
11.8.1 Port Error 320
11.8.2 Speed Detection 320
11.8.3 Collision 320
11.8.4 Low-speed Port Behavior 321
11.9 Suspend and Resume 321
11.10 Hub Reset Behavior 323
11.11 Hub Port Power Control 324
11.11.1 Multiple Gangs 324
11.12 Hub Controller 325
11.12.1 Endpoint Organization 325
11.12.2 Hub Information Architecture and Operation 325
11.12.3 Port Change Information Processing 326
11.12.4 Hub and Port Status Change Bitmap 327
11.12.5 Over-current Reporting and Recovery 328
11.12.6 Enumeration Handling 329
11.13 Hub Configuration 329
11.14 Transaction Translator 330
11.14.1 Overview 331
11.14.2 Transaction Translator Scheduling 333
11.15 Split Transaction Notation Information 335
11.16 Common Split Transaction State Machines 338
11.16.1 Host Controller State Machine 339
11.16.2 Transaction Translator State Machine 343
11.17 Bulk/Control Transaction Translation Overview 348
11.17.1 Bulk/Control Split Transaction Sequences 349
11.17.2 Bulk/Control Split Transaction State Machines 355
11.17.3 Bulk/Control Sequencing 360

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.17.4 Bulk/Control Buffering Requirements 361
11.17.5 Other Bulk/Control Details 361
11.18 Periodic Split Transaction Pipelining and Buffer Management 361
11.18.1 Best Case Full-Speed Budget 362
11.18.2 TT Microframe Pipeline 362
11.18.3 Generation of Full-speed Frames 363
11.18.4 Host Split Transaction Scheduling Requirements 363
11.18.5 TT Response Generation 366
11.18.6 TT Periodic Transaction Handling Requirements 367
11.18.7 TT Transaction Tracking 369
11.18.8 TT Complete-split Transaction State Searching 370
11.19 Approximate TT Buffer Space Required 371
11.20 Interrupt Transaction Translation Overview 371
11.20.1 Interrupt Split Transaction Sequences 371
11.20.2 Interrupt Split Transaction State Machines 375
11.20.3 Interrupt OUT Sequencing 381
11.20.4 Interrupt IN Sequencing 382
11.21 Isochronous Transaction Translation Overview 383
11.21.1 Isochronous Split Transaction Sequences 384
11.21.2 Isochronous Split Transaction State Machines 387
11.21.3 Isochronous OUT Sequencing 391
11.21.4 Isochronous IN Sequencing 392
11.22 TT Error Handling 392
11.22.1 Loss of TT Synchronization With HS SOFs 392
11.22.2 TT Frame and Microframe Timer Synchronization Requirements 393
11.23 Descriptors 395
11.23.1 Standard Descriptors for Hub Class 395
11.23.2 Class-specific Descriptors 403
11.24 Requests 404
11.24.1 Standard Requests 404
11.24.2 Class-specific Requests 405
Appendix A Transaction Examples 421
A.1 Bulk/Control OUT and SETUP Transaction Examples 421
A.2 Bulk/Control IN Transaction Examples 445
A.3 Interrupt OUT Transaction Examples 469
A.4 Interrupt IN Transaction Examples 488
A.5 Isochronous OUT Split-transaction Examples 510
A.6 Isochronous IN Split-transaction Examples 519
Appendix B Example Declarations for State Machines 533
B.1 Global Declarations 533
B.2 Host Controller Declarations 536
B.3 Transaction Translator Declarations 538
Appendix C Reset Protocol State Diagrams 542
C.1 Downstream Facing Port State Diagram 542
C.2 Upstream Facing Port State Diagram 544
Index 549

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Note: All Engineering Change Notice's (ECN) and Errata documents as of September 01, 2012 that pertain to this core specification follow the last page of the specification starting on page 619.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Universal Serial Bus Specification

Compaq
Hewlett-Packard
Intel
Lucent
Microsoft
NEC
Philips

Revision 2.0
April 27, 2000

Scope of this Revision

The 2.0 revision of the specification is intended for product design. Every attempt has been made to ensure a consistent and implementable specification. Implementations should ensure compliance with this revision.

Revision History

Revision	Issue Date	Comments
0.7	November 11, 1994	Supersedes 0.6e.
0.8	December 30, 1994	Revisions to Chapters 3-8, 10, and 11. Added appendixes.
0.9	April 13, 1995	Revisions to all the chapters.
0.99	August 25, 1995	Revisions to all the chapters.
1.0 FDR	November 13, 1995	Revisions to Chapters 1, 2, 5-11.
1.0	January 15, 1996	Edits to Chapters 5, 6, 7, 8, 9, 10, and 11 for consistency.
1.1	September 23, 1998	Updates to all chapters to fix problems identified.
2.0 (draft 0.79)	October 5, 1999	Revisions to chapters 5, 7, 8, 9, 11 to add high speed.
2.0 (draft 0.9)	December 21, 1999	Revisions to all chapters to add high speed.
2.0	April 27, 2000	Revisions for high-speed mode.

Universal Serial Bus Specification
Copyright © 2000, Compaq Computer Corporation,
Hewlett-Packard Company,
Intel Corporation,
Lucent Technologies Inc,
Microsoft Corporation,
NEC Corporation,
Koninklijke Philips Electronics N.V.
All rights reserved.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

INTELLECTUAL PROPERTY DISCLAIMER

THIS SPECIFICATION IS PROVIDED TO YOU “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE. THE AUTHORS OF THIS SPECIFICATION DISCLAIM ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OR IMPLEMENTATION OF INFORMATION IN THIS SPECIFICATION. THE PROVISION OF THIS SPECIFICATION TO YOU DOES NOT PROVIDE YOU WITH ANY LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS.

All product names are trademarks, registered trademarks, or servicemarks of their respective owners.

Please send comments via electronic mail to techsup@usb.org

For industry information, refer to the USB Implementers Forum web page at http://www.usb.org

Acknowledgement of USB 2.0 Technical Contribution

The authors of this specification would like to recognize the following people who participated in the USB 2.0 Promoter Group technical working groups. We would also like to thank others in the USB 2.0 Promoter companies and throughout the industry who contributed to the development of this specification.

Hub Working Group

John Garney	Intel Corporation (Chair/Editor)
Ken Stufflebeam	Compaq Computer Corporation
David Wooten	Compaq Computer Corporation
Matt Nieberger	Hewlett-Packard Company
John Howard	Intel Corporation
Venkat Iyer	Intel Corporation
Steve McGowan	Intel Corporation
Geert Knapen	Royal Philips Electronics
Zong Liang Wu	Royal Philips Electronics
Jim Clee	Lucent Technologies Inc
Jim Guziak	Lucent Technologies Inc
Dave Thompson	Lucent Technologies Inc
John Fuller	Microsoft Corporation
Nathan Sherman	Microsoft Corporation
Mark Williams	Microsoft Corporation
Nobuo Furuya	NEC Corporation
Toshimi Sakurai	NEC Corporation
Moto Sato	NEC Corporation
Katsuya Suzuki	NEC Corporation

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Electrical Working Group

Jon Lueker	Intel Corporation (Chair/Editor)
David Wooten	Compaq Computer Corporation
Matt Nieberger	Hewlett-Packard Company
Larry Taugher	Hewlett-Packard Company
Venkat Iyer	Intel Corporation
Steve McGowan	Intel Corporation
Mike Pennell	Intel Corporation
Todd West	Intel Corporation
Gerrit den Besten	Royal Philips Electronics
Marq Kole	Royal Philips Electronics
Zong Liang Wu	Royal Philips Electronics
Jim Clee	Lucent Technologies Inc
Jim Guziak	Lucent Technologies Inc
Par Parikh	Lucent Technologies Inc
Dave Thompson	Lucent Technologies Inc
Ed Giaimo	Microsoft Corporation
Mark Williams	Microsoft Corporation
Toshihiko Ohtani	NEC Corporation
Kugao Ouchi	NEC Corporation
Katsuya Suzuki	NEC Corporation
Toshio Tasaki	NEC Corporation

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

1 Chapter 1 Introduction

1.1 Motivation

The original motivation for the Universal Serial Bus (USB) came from three interrelated considerations:

- Connection of the PC to the telephone

It is well understood that the merge of computing and communication will be the basis for the next generation of productivity applications. The movement of machine-oriented and human-oriented data types from one location or environment to another depends on ubiquitous and cheap connectivity. Unfortunately, the computing and communication industries have evolved independently. The USB provides a ubiquitous link that can be used across a wide range of PC-to-telephone interconnects.

- Ease-of-use

The lack of flexibility in reconfiguring the PC has been acknowledged as the Achilles' heel to its further deployment. The combination of user-friendly graphical interfaces and the hardware and software mechanisms associated with new-generation bus architectures have made computers less confrontational and easier to reconfigure. However, from the end user's point of view, the PC's I/O interfaces, such as serial/parallel ports, keyboard/mouse/joystick interfaces, etc., do not have the attributes of plug-and-play.

- Port expansion

The addition of external peripherals continues to be constrained by port availability. The lack of a bi-directional, low-cost, low-to-mid speed peripheral bus has held back the creative proliferation of peripherals such as telephone/fax/modem adapters, answering machines, scanners, PDA's, keyboards, mice, etc. Existing interconnects are optimized for one or two point products. As each new function or capability is added to the PC, a new interface has been defined to address this need.

The more recent motivation for USB 2.0 stems from the fact that PCs have increasingly higher performance and are capable of processing vast amounts of data. At the same time, PC peripherals have added more performance and functionality. User applications such as digital imaging demand a high performance connection between the PC and these increasingly sophisticated peripherals. USB 2.0 addresses this need by adding a third transfer rate of 480 Mb/s to the 12 Mb/s and 1.5 Mb/s originally defined for USB. USB 2.0 is a natural evolution of USB, delivering the desired bandwidth increase while preserving the original motivations for USB and maintaining full compatibility with existing peripherals.

Thus, USB continues to be the answer to connectivity for the PC architecture. It is a fast, bi-directional, isochronous, low-cost, dynamically attachable serial interface that is consistent with the requirements of the PC platform of today and tomorrow.

1.2 Objective of the Specification

This document defines an industry-standard USB. The specification describes the bus attributes, the protocol definition, types of transactions, bus management, and the programming interface required to design and build systems and peripherals that are compliant with this standard.

The goal is to enable such devices from different vendors to interoperate in an open architecture. The specification is intended as an enhancement to the PC architecture, spanning portable, business desktop, and home environments. It is intended that the specification allow system OEMs and peripheral developers adequate room for product versatility and market differentiation without the burden of carrying obsolete interfaces or losing compatibility.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

1.3 Scope of the Document

The specification is primarily targeted to peripheral developers and system OEMs, but provides valuable information for platform operating system/ BIOS/ device driver, adapter IHVs/ISVs, and platform/adapter controller vendors. This specification can be used for developing new products and associated software.

1.4 USB Product Compliance

Adopters of the USB 2.0 specification have signed the USB 2.0 Adopters Agreement, which provides them access to a reciprocal royalty-free license from the Promoters and other Adopters to certain intellectual property contained in products that are compliant with the USB 2.0 specification. Adopters can demonstrate compliance with the specification through the testing program as defined by the USB Implementers Forum. Products that demonstrate compliance with the specification will be granted certain rights to use the USB Implementers Forum logo as defined in the logo license.

1.5 Document Organization

Chapters 1 through 5 provide an overview for all readers, while Chapters 6 through 11 contain detailed technical information defining the USB.

- Peripheral implementers should particularly read Chapters 5 through 11.
- USB Host Controller implementers should particularly read Chapters 5 through 8, 10, and 11.
- USB device driver implementers should particularly read Chapters 5, 9, and 10.

This document is complemented and referenced by the Universal Serial Bus Device Class Specifications. Device class specifications exist for a wide variety of devices. Please contact the USB Implementers Forum for further details.

Readers are also requested to contact operating system vendors for operating system bindings specific to the USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

2 Chapter 2
Terms and Abbreviations

This chapter lists and defines terms and abbreviations used throughout this specification.

ACK	Handshake packet indicating a positive acknowledgment.
Active Device	A device that is powered and is not in the Suspend state.
Asynchronous Data	Data transferred at irregular intervals with relaxed latency requirements.
Asynchronous RA	The incoming data rate, Fs_i, and the outgoing data rate, Fs_o, of the RA process are independent (i.e., there is no shared master clock). See also rate adaptation.
Asynchronous SRC	The incoming sample rate, Fs_i, and outgoing sample rate, Fs_o, of the SRC process are independent (i.e., there is no shared master clock). See also sample rate conversion.
Audio Device	A device that sources or sinks sampled analog data.
AWG#	The measurement of a wire's cross section, as defined by the American Wire Gauge standard.
Babble	Unexpected bus activity that persists beyond a specified point in a (micro)frame.
Bandwidth	The amount of data transmitted per unit of time, typically bits per second (b/s) or bytes per second (B/s).
Big Endian	A method of storing data that places the most significant byte of multiple-byte values at a lower storage address. For example, a 16-bit integer stored in big endian format places the least significant byte at the higher address and the most significant byte at the lower address. See also little endian.
Bit	A unit of information used by digital computers. Represents the smallest piece of addressable memory within a computer. A bit expresses the choice between two possibilities and is typically represented by a logical one (1) or zero (0).
Bit Stuffing	Insertion of a "0" bit into a data stream to cause an electrical transition on the data wires, allowing a PLL to remain locked.
b/s	Transmission rate expressed in bits per second.
B/s	Transmission rate expressed in bytes per second.
Buffer	Storage used to compensate for a difference in data rates or time of occurrence of events, when transmitting data from one device to another.
Bulk Transfer	One of the four USB transfer types. Bulk transfers are non-periodic, large bursty communication typically used for a transfer that can use any available bandwidth and can also be delayed until bandwidth is available. See also transfer type.
Bus Enumeration	Detecting and identifying USB devices.
Byte	A data element that is eight bits in size.
Capabilities	Those attributes of a USB device that are administrated by the host.
Characteristics	Those qualities of a USB device that are unchangeable; for example, the device class is a device characteristic.
Client	Software resident on the host that interacts with the USB System Software to arrange data transfer between a function and the host. The client is often the data provider and consumer for transferred data.
Configuring Software	Software resident on the host software that is responsible for configuring a USB device. This may be a system configurator or software specific to the device.
Control Endpoint	A pair of device endpoints with the same endpoint number that are used by a control pipe. Control endpoints transfer data in both directions and, therefore, use both endpoint directions of a device address and endpoint number combination. Thus, each control endpoint consumes two endpoint addresses.
Control Pipe	Same as a message pipe.
Control Transfer	One of the four USB transfer types. Control transfers support configuration/command/status type communications between client and function. See also transfer type.

IECNORM.COM : Click to view the full PDF of iec 62680-1:2013
Withdrawn

CRC	See Cyclic Redundancy Check.
CTI	Computer Telephony Integration.
Cyclic Redundancy Check (CRC)	A check performed on data to see if an error has occurred in transmitting, reading, or writing the data. The result of a CRC is typically stored or transmitted with the checked data. The stored or transmitted result is compared to a CRC calculated for the data to determine if an error has occurred.
Default Address	An address defined by the USB Specification and used by a USB device when it is first powered or reset. The default address is 00H.
Default Pipe	The message pipe created by the USB System Software to pass control and status information between the host and a USB device's endpoint zero.
Device	A logical or physical entity that performs a function. The actual entity described depends on the context of the reference. At the lowest level, device may refer to a single hardware component, as in a memory device. At a higher level, it may refer to a collection of hardware components that perform a particular function, such as a USB interface device. At an even higher level, device may refer to the function performed by an entity attached to the USB; for example, a data/FAX modem device. Devices may be physical, electrical, addressable, and logical. When used as a non-specific reference, a USB device is either a hub or a function.
Device Address	A seven-bit value representing the address of a device on the USB. The device address is the default address (00H) when the USB device is first powered or the device is reset. Devices are assigned a unique device address by the USB System Software.
Device Endpoint	A uniquely addressable portion of a USB device that is the source or sink of information in a communication flow between the host and device. See also endpoint address.
Device Resources	Resources provided by USB devices, such as buffer space and endpoints. See also Host Resources and Universal Serial Bus Resources.
Device Software	Software that is responsible for using a USB device. This software may or may not also be responsible for configuring the device for use.
Downstream	The direction of data flow from the host or away from the host. A downstream port is the port on a hub electrically farthest from the host that generates downstream data traffic from the hub. Downstream ports receive upstream data traffic.
Driver	When referring to hardware, an I/O pad that drives an external load. When referring to software, a program responsible for interfacing to a hardware device, that is, a device driver.
DWORD	Double word. A data element that is two words (i.e., four bytes or 32 bits) in size.
Dynamic Insertion and Removal	The ability to attach and remove devices while the host is in operation.
E^2PROM	See Electrically Erasable Programmable Read Only Memory.
EEPROM	See Electrically Erasable Programmable Read Only Memory.
Electrically Erasable Programmable Read Only Memory (EEPROM)	Non-volatile rewritable memory storage technology.
End User	The user of a host.
Endpoint	See device endpoint.
Endpoint Address	The combination of an endpoint number and an endpoint direction on a USB device. Each endpoint address supports data transfer in one direction.
Endpoint Direction	The direction of data transfer on the USB. The direction can be either IN or OUT. IN refers to transfers to the host; OUT refers to transfers from the host.
Endpoint Number	A four-bit value between 0H and FH, inclusive, associated with an endpoint on a USB device.
Envelope detector	An electronic circuit inside a USB device that monitors the USB data lines and detects certain voltage related signal characteristics.
EOF	End-of-(micro)Frame.
EOP	End-of-Packet.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

External Port	See port.
Eye pattern	A representation of USB signaling that provides minimum and maximum voltage levels as well as signal jitter.
False EOP	A spurious, usually noise-induced event that is interpreted by a packet receiver as an EOP.
Frame	A 1 millisecond time base established on full-/low-speed buses.
Frame Pattern	A sequence of frames that exhibit a repeating pattern in the number of samples transmitted per frame. For a 44.1 kHz audio transfer, the frame pattern could be nine frames containing 44 samples followed by one frame containing 45 samples.
Fs	See sample rate.
Full-duplex	Computer data transmission occurring in both directions simultaneously.
Full-speed	USB operation at 12 Mb/s. See also low-speed and high-speed.
Function	A USB device that provides a capability to the host, such as an ISDN connection, a digital microphone, or speakers.
Handshake Packet	A packet that acknowledges or rejects a specific condition. For examples, see ACK and NAK.
High-bandwidth endpoint	A high-speed device endpoint that transfers more than 1024 bytes and less than 3073 bytes per microframe.
High-speed	USB operation at 480 Mb/s. See also low-speed and full-speed.
Host	The host computer system where the USB Host Controller is installed. This includes the host hardware platform (CPU, bus, etc.) and the operating system in use.
Host Controller	The host's USB interface.
Host Controller Driver (HCD)	The USB software layer that abstracts the Host Controller hardware. The Host Controller Driver provides an SPI for interaction with a Host Controller. The Host Controller Driver hides the specifics of the Host Controller hardware implementation.
Host Resources	Resources provided by the host, such as buffer space and interrupts. See also Device Resources and Universal Serial Bus Resources.
Hub	A USB device that provides additional connections to the USB.
Hub Tier	One plus the number of USB links in a communication path between the host and a function. See Figure 4-1.
Interrupt Request (IRQ)	A hardware signal that allows a device to request attention from a host. The host typically invokes an interrupt service routine to handle the condition that caused the request.
Interrupt Transfer	One of the four USB transfer types. Interrupt transfer characteristics are small data, non-periodic, low-frequency, and bounded-latency. Interrupt transfers are typically used to handle service needs. See also transfer type.
I/O Request Packet	An identifiable request by a software client to move data between itself (on the host) and an endpoint of a device in an appropriate direction.
IRP	See I/O Request Packet.
IRQ	See Interrupt Request.
Isochronous Data	A stream of data whose timing is implied by its delivery rate.
Isochronous Device	An entity with isochronous endpoints, as defined in the USB Specification, that sources or sinks sampled analog streams or synchronous data streams.
Isochronous Sink Endpoint	An endpoint that is capable of consuming an isochronous data stream that is sent by the host.
Isochronous Source Endpoint	An endpoint that is capable of producing an isochronous data stream and sending it to the host.
Isochronous Transfer	One of the four USB transfer types. Isochronous transfers are used when working with isochronous data. Isochronous transfers provide periodic, continuous communication between host and device. See also transfer type.
Jitter	A tendency toward lack of synchronization caused by mechanical or electrical changes. More specifically, the phase shift of digital pulses over a transmission medium.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

kb/s	Transmission rate expressed in kilobits per second.
kB/s	Transmission rate expressed in kilobytes per second.
Little Endian	Method of storing data that places the least significant byte of multiple-byte values at lower storage addresses. For example, a 16-bit integer stored in little endian format places the least significant byte at the lower address and the most significant byte at the next address. See also big endian.
LOA	Loss of bus activity characterized by an SOP without a corresponding EOP.
Low-speed	USB operation at 1.5 Mb/s. See also full-speed and high-speed.
LSb	Least significant bit.
LSB	Least significant byte.
Mb/s	Transmission rate expressed in megabits per second.
MB/s	Transmission rate expressed in megabytes per second.
Message Pipe	A bi-directional pipe that transfers data using a request/data/status paradigm. The data has an imposed structure that allows requests to be reliably identified and communicated.
Microframe	A 125 microsecond time base established on high-speed buses.
MSb	Most significant bit.
MSB	Most significant byte.
NAK	Handshake packet indicating a negative acknowledgment.
Non Return to Zero Invert (NRZI)	A method of encoding serial data in which ones and zeroes are represented by opposite and alternating high and low voltages where there is no return to zero (reference) voltage between encoded bits. Eliminates the need for clock pulses.
NRZI	See Non Return to Zero Invert.
Object	Host software or data structure representing a USB entity.
Packet	A bundle of data organized in a group for transmission. Packets typically contain three elements: control information (e.g., source, destination, and length), the data to be transferred, and error detection and correction bits.
Packet Buffer	The logical buffer used by a USB device for sending or receiving a single packet. This determines the maximum packet size the device can send or receive.
Packet ID (PID)	A field in a USB packet that indicates the type of packet, and by inference, the format of the packet and the type of error detection applied to the packet.
Phase	A token, data, or handshake packet. A transaction has three phases.
Phase Locked Loop (PLL)	A circuit that acts as a phase detector to keep an oscillator in phase with an incoming frequency.
Physical Device	A device that has a physical implementation; e.g., speakers, microphones, and CD players.
PID	See Packet ID.
Pipe	A logical abstraction representing the association between an endpoint on a device and software on the host. A pipe has several attributes; for example, a pipe may transfer data as streams (stream pipe) or messages (message pipe). See also stream pipe and message pipe.
PLL	See Phase Locked Loop.
Polling	Asking multiple devices, one at a time, if they have any data to transmit.
POR	See Power On Reset.
Port	Point of access to or from a system or circuit. For the USB, the point where a USB device is attached.
Power On Reset (POR)	Restoring a storage device, register, or memory to a predetermined state when power is applied.
Programmable Data Rate	Either a fixed data rate (single-frequency endpoints), a limited number of data rates (32 kHz, 44.1 kHz, 48 kHz, …), or a continuously programmable data rate. The exact programming capabilities of an endpoint must be reported in the appropriate class-specific endpoint descriptors.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Protocol	A specific set of rules, procedures, or conventions relating to format and timing of data transmission between two devices.
RA	See rate adaptation.
Rate Adaptation	The process by which an incoming data stream, sampled at Fs_i, is converted to an outgoing data stream, sampled at Fs_o,with a certain loss of quality, determined by the rate adaptation algorithm. Error control mechanisms are required for the process. Fs_i and Fs_o can be different and asynchronous. Fs_i is the input data rate of the RA; Fs_o is the output data rate of the RA.
Request	A request made to a USB device contained within the data portion of a SETUP packet.
Retire	The action of completing service for a transfer and notifying the appropriate software client of the completion.
Root Hub	A USB hub directly attached to the Host Controller. This hub (tier 1) is attached to the host.
Root Port	The downstream port on a Root Hub.
Sample	The smallest unit of data on which an endpoint operates; a property of an endpoint.
Sample Rate (Fs)	The number of samples per second, expressed in Hertz (Hz).
Sample Rate Conversion (SRC)	A dedicated implementation of the RA process for use on sampled analog data streams. The error control mechanism is replaced by interpolating techniques.
Service	A procedure provided by a System Programming Interface (SPI).
Service Interval	The period between consecutive requests to a USB endpoint to send or receive data.
Service Jitter	The deviation of service delivery from its scheduled delivery time.
Service Rate	The number of services to a given endpoint per unit time.
SOF	See Start-of-Frame.
SOP	Start-of-Packet.
SPI	See System Programming Interface.
Split transaction	A transaction type supported by host controllers and hubs. This transaction type allows full- and low-speed devices to be attached to hubs operating at high-speed.
SRC	See Sample Rate Conversion.
Stage	One part of the sequence composing a control transfer; stages include the Setup stage, the Data stage, and the Status stage.
Start-of-Frame (SOF)	The first transaction in each (micro)frame. An SOF allows endpoints to identify the start of the (micro)frame and synchronize internal endpoint clocks to the host.
Stream Pipe	A pipe that transfers data as a stream of samples with no defined USB structure.
Synchronization Type	A classification that characterizes an isochronous endpoint's capability to connect to other isochronous endpoints.
Synchronous RA	The incoming data rate, Fs_i, and the outgoing data rate, Fs_o, of the RA process are derived from the same master clock. There is a fixed relation between Fs_i and Fs_o.
Synchronous SRC	The incoming sample rate, Fs_i, and outgoing sample rate, Fs_o, of the SRC process are derived from the same master clock. There is a fixed relation between Fs_i and Fs_o.
System Programming Interface (SPI)	A defined interface to services provided by system software.
TDM	See Time Division Multiplexing.
TDR	See Time Domain Reflectometer.
Termination	Passive components attached at the end of cables to prevent signals from being reflected or echoed.
Time Division Multiplexing (TDM)	A method of transmitting multiple signals (data, voice, and/or video) simultaneously over one communications medium by interleaving a piece of each signal one after another.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Time Domain Reflectometer (TDR)	An instrument capable of measuring impedance characteristics of the USB signal lines.
Timeout	The detection of a lack of bus activity for some predetermined interval.
Token Packet	A type of packet that identifies what transaction is to be performed on the bus.
Transaction	The delivery of service to an endpoint; consists of a token packet, optional data packet, and optional handshake packet. Specific packets are allowed/required based on the transaction type.
Transaction translator	A functional component of a USB hub. The Transaction Translator responds to special high-speed transactions and translates them to full/low-speed transactions with full/low-speed devices attached on downstream facing ports.
Transfer	One or more bus transactions to move information between a software client and its function.
Transfer Type	Determines the characteristics of the data flow between a software client and its function. Four standard transfer types are defined: control, interrupt, bulk, and isochronous.
Turn-around Time	The time a device needs to wait to begin transmitting a packet after a packet has been received to prevent collisions on the USB. This time is based on the length and propagation delay characteristics of the cable and the location of the transmitting device in relation to other devices on the USB.
Universal Serial Bus Driver (USBD)	The host resident software entity responsible for providing common services to clients that are manipulating one or more functions on one or more Host Controllers.
Universal Serial Bus Resources	Resources provided by the USB, such as bandwidth and power. See also Device Resources and Host Resources.
Upstream	The direction of data flow towards the host. An upstream port is the port on a device electrically closest to the host that generates upstream data traffic from the hub. Upstream ports receive downstream data traffic.
USBD	See Universal Serial Bus Driver.
USB-IF	USB Implementers Forum, Inc. is a nonprofit corporation formed to facilitate the development of USB compliant products and promote the technology.
Virtual Device	A device that is represented by a software interface layer. An example of a virtual device is a hard disk with its associated device driver and client software that makes it able to reproduce an audio .WAV file.
Word	A data element that is two bytes (16 bits) in size.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

3 Chapter 3 Background

This chapter presents a brief description of the background of the Universal Serial Bus (USB), including design goals, features of the bus, and existing technologies.

3.1 Goals for the Universal Serial Bus

The USB is specified to be an industry-standard extension to the PC architecture with a focus on PC peripherals that enable consumer and business applications. The following criteria were applied in defining the architecture for the USB:

- Ease-of-use for PC peripheral expansion
- Low-cost solution that supports transfer rates up to 480 Mb/s
- Full support for real-time data for voice, audio, and video
- Protocol flexibility for mixed-mode isochronous data transfers and asynchronous messaging
- Integration in commodity device technology
- Comprehension of various PC configurations and form factors
- Provision of a standard interface capable of quick diffusion into product
- Enabling new classes of devices that augment the PC's capability
- Full backward compatibility of USB 2.0 for devices built to previous versions of the specification

3.2 Taxonomy of Application Space

Figure 3-1 describes a taxonomy for the range of data traffic workloads that can be serviced over a USB. As can be seen, a 480 Mb/s bus comprehends the high-speed, full-speed, and low-speed data ranges. Typically, high-speed and full-speed data types may be isochronous, while low-speed data comes from interactive devices. The USB is primarily a PC bus but can be readily applied to other host-centric computing devices. The software architecture allows for future extension of the USB by providing support for multiple USB Host Controllers.

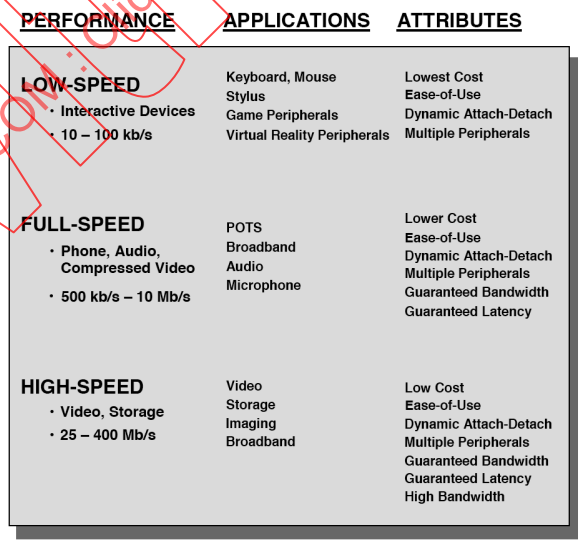


Figure 3-1. Application Space Taxonomy

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

3.3 Feature List

The USB Specification provides a selection of attributes that can achieve multiple price/performance integration points and can enable functions that allow differentiation at the system and component level. Features are categorized by the following benefits:

Easy to use for end user

- Single model for cabling and connectors
- Electrical details isolated from end user (e.g., bus terminations)
- Self-identifying peripherals, automatic mapping of function to driver and configuration
- Dynamically attachable and reconfigurable peripherals

Wide range of workloads and applications

- Suitable for device bandwidths ranging from a few kb/s to several hundred Mb/s
- Supports isochronous as well as asynchronous transfer types over the same set of wires
- Supports concurrent operation of many devices (multiple connections)
- Supports up to 127 physical devices
- Supports transfer of multiple data and message streams between the host and devices
- Allows compound devices (i.e., peripherals composed of many functions)
- Lower protocol overhead, resulting in high bus utilization

Isochronous bandwidth

- Guaranteed bandwidth and low latencies appropriate for telephony, audio, video, etc.

Flexibility

- Supports a wide range of packet sizes, which allows a range of device buffering options
- Allows a wide range of device data rates by accommodating packet buffer size and latencies
- Flow control for buffer handling is built into the protocol

Robustness

- Error handling/fault recovery mechanism is built into the protocol
- Dynamic insertion and removal of devices is identified in user-perceived real-time
- Supports identification of faulty devices

Synergy with PC industry

- Protocol is simple to implement and integrate
- Consistent with the PC plug-and-play architecture
- Leverages existing operating system interfaces

Low-cost implementation

- Low-cost subchannel at 1.5 Mb/s
- Optimized for integration in peripheral and host hardware
- Suitable for development of low-cost peripherals
- Low-cost cables and connectors
- Uses commodity technologies

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Upgrade path

- Architecture upgradeable to support multiple USB Host Controllers in a system

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

4 Chapter 4
Architectural Overview

This chapter presents an overview of the Universal Serial Bus (USB) architecture and key concepts. The USB is a cable bus that supports data exchange between a host computer and a wide range of simultaneously accessible peripherals. The attached peripherals share USB bandwidth through a host-scheduled, token-based protocol. The bus allows peripherals to be attached, configured, used, and detached while the host and other peripherals are in operation.

Later chapters describe the various components of the USB in greater detail.

4.1 USB System Description

A USB system is described by three definitional areas:

- USB interconnect
- USB devices
- USB host

The USB interconnect is the manner in which USB devices are connected to and communicate with the host. This includes the following:

- Bus Topology: Connection model between USB devices and the host.
- Inter-layer Relationships: In terms of a capability stack, the USB tasks that are performed at each layer in the system.
- Data Flow Models: The manner in which data moves in the system over the USB between producers and consumers.
- USB Schedule: The USB provides a shared interconnect. Access to the interconnect is scheduled in order to support isochronous data transfers and to eliminate arbitration overhead.

USB devices and the USB host are described in detail in subsequent sections.

4.1.1 Bus Topology

The USB connects USB devices with the USB host. The USB physical interconnect is a tiered star topology. A hub is at the center of each star. Each wire segment is a point-to-point connection between the host and a hub or function, or a hub connected to another hub or function. Figure 4-1 illustrates the topology of the USB.

Due to timing constraints allowed for hub and cable propagation times, the maximum number of tiers allowed is seven (including the root tier). Note that in seven tiers, five non-root hubs maximum can be supported in a communication path between the host and any device. A compound device (see Figure 4-1) occupies two tiers; therefore, it cannot be enabled if attached at tier level seven. Only functions can be enabled in tier seven.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

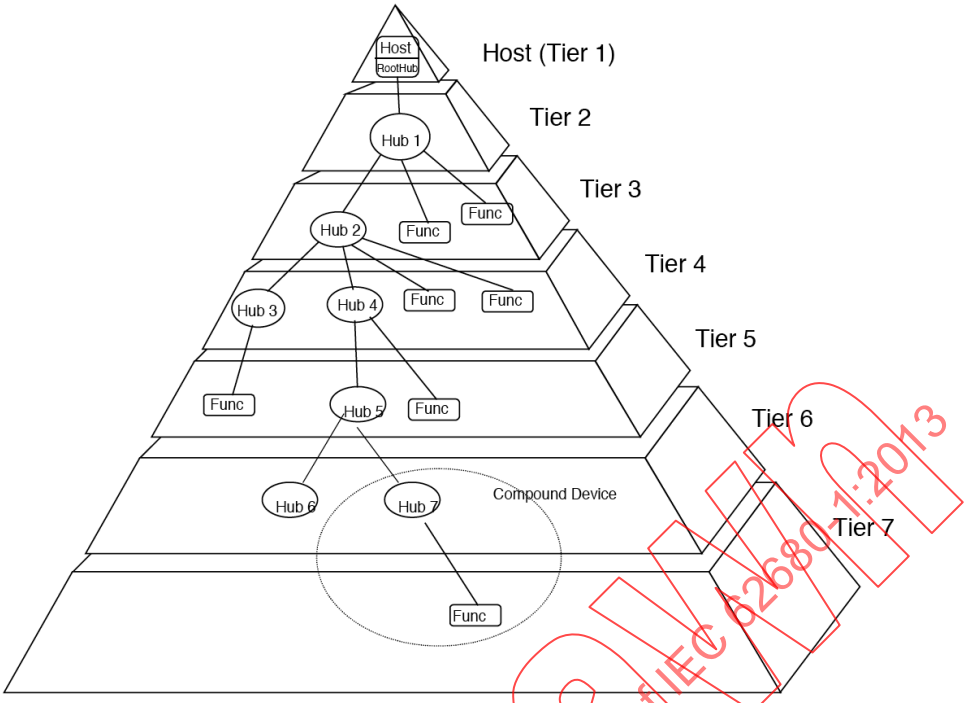


Figure 4-1. Bus Topology

4.1.1.1 USB Host

There is only one host in any USB system. The USB interface to the host computer system is referred to as the Host Controller. The Host Controller may be implemented in a combination of hardware, firmware, or software. A root hub is integrated within the host system to provide one or more attachment points.

Additional information concerning the host may be found in Section 4.9 and in Chapter 10.

4.1.1.2 USB Devices

USB devices are one of the following:

- Hubs, which provide additional attachment points to the USB
- Functions, which provide capabilities to the system, such as an ISDN connection, a digital joystick, or speakers

USB devices present a standard USB interface in terms of the following:

- Their comprehension of the USB protocol
- Their response to standard USB operations, such as configuration and reset
- Their standard capability descriptive information

Additional information concerning USB devices may be found in Section 4.8 and in Chapter 9.

4.2 Physical Interface

The physical interface of the USB is described in the electrical (Chapter 7) and mechanical (Chapter 6) specifications for the bus.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4.2.1 Electrical

The USB transfers signal and power over a four-wire cable, shown in Figure 4-2. The signaling occurs over two wires on each point-to-point segment.

There are three data rates:

- The USB high-speed signaling bit rate is 480 Mb/s.
- The USB full-speed signaling bit rate is 12 Mb/s.
- A limited capability low-speed signaling mode is also defined at 1.5 Mb/s.

USB 2.0 host controllers and hubs provide capabilities so that full-speed and low-speed data can be transmitted at high-speed between the host controller and the hub, but transmitted between the hub and the device at full-speed or low-speed. This capability minimizes the impact that full-speed and low-speed devices have upon the bandwidth available for high-speed devices.

The low-speed mode is defined to support a limited number of low-bandwidth devices, such as mice, because more general use would degrade bus utilization.

The clock is transmitted, encoded along with the differential data. The clock encoding scheme is NRZI with bit stuffing to ensure adequate transitions. A SYNC field precedes each packet to allow the receiver(s) to synchronize their bit recovery clocks.

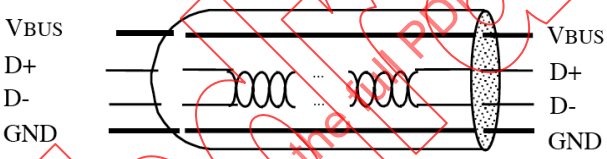


Figure 4-2. USB Cable

The cable also carries VBUS and GND wires on each segment to deliver power to devices. VBUS is nominally +5 V at the source. The USB allows cable segments of variable lengths, up to several meters, by choosing the appropriate conductor gauge to match the specified IR drop and other attributes such as device power budget and cable flexibility. In order to provide guaranteed input voltage levels and proper termination impedance, biased terminations are used at each end of the cable. The terminations also permit the detection of attach and detach at each port and differentiate between high/full-speed and low-speed devices.

4.2.2 Mechanical

The mechanical specifications for cables and connectors are provided in Chapter 6. All devices have an upstream connection. Upstream and downstream connectors are not mechanically interchangeable, thus eliminating illegal loopback connections at hubs. The cable has four conductors: a twisted signal pair of standard gauge and a power pair in a range of permitted gauges. The connector is four-position, with shielded housing, specified robustness, and ease of attach-detach characteristics.

4.3 Power

The specification covers two aspects of power:

- Power distribution over the USB deals with the issues of how USB devices consume power provided by the host over the USB.
- Power management deals with how the USB System Software and devices fit into the host-based power management system.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4.3.1 Power Distribution

Each USB segment provides a limited amount of power over the cable. The host supplies power for use by USB devices that are directly connected. In addition, any USB device may have its own power supply. USB devices that rely totally on power from the cable are called bus-powered devices. In contrast, those that have an alternate source of power are called self-powered devices. A hub also supplies power for its connected USB devices. The architecture permits bus-powered hubs within certain constraints of topology that are discussed later in Chapter 11.

4.3.2 Power Management

A USB host may have a power management system that is independent of the USB. The USB System Software interacts with the host's power management system to handle system power events such as suspend or resume. Additionally, USB devices typically implement additional power management features that allow them to be power managed by system software.

The power distribution and power management features of the USB allow it to be designed into power-sensitive systems such as battery-based notebook computers.

4.4 Bus Protocol

The USB is a polled bus. The Host Controller initiates all data transfers.

Most bus transactions involve the transmission of up to three packets. Each transaction begins when the Host Controller, on a scheduled basis, sends a USB packet describing the type and direction of transaction, the USB device address, and endpoint number. This packet is referred to as the "token packet." The USB device that is addressed selects itself by decoding the appropriate address fields. In a given transaction, data is transferred either from the host to a device or from a device to the host. The direction of data transfer is specified in the token packet. The source of the transaction then sends a data packet or indicates it has no data to transfer. The destination, in general, responds with a handshake packet indicating whether the transfer was successful.

Some bus transactions between host controllers and hubs involve the transmission of four packets. These types of transactions are used to manage the data transfers between the host and full-/low- speed devices.

The USB data transfer model between a source or destination on the host and an endpoint on a device is referred to as a pipe. There are two types of pipes: stream and message. Stream data has no USB-defined structure, while message data does. Additionally, pipes have associations of data bandwidth, transfer service type, and endpoint characteristics like directionality and buffer sizes. Most pipes come into existence when a USB device is configured. One message pipe, the Default Control Pipe, always exists once a device is powered, in order to provide access to the device's configuration, status, and control information.

The transaction schedule allows flow control for some stream pipes. At the hardware level, this prevents buffers from underrun or overrun situations by using a NAK handshake to throttle the data rate. When NAKed, a transaction is retried when bus time is available. The flow control mechanism permits the construction of flexible schedules that accommodate concurrent servicing of a heterogeneous mix of stream pipes. Thus, multiple stream pipes can be serviced at different intervals and with packets of different sizes.

4.5 Robustness

There are several attributes of the USB that contribute to its robustness:

- Signal integrity using differential drivers, receivers, and shielding
- CRC protection over control and data fields

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Detection of attach and detach and system-level configuration of resources
- Self-recovery in protocol, using timeouts for lost or corrupted packets
- Flow control for streaming data to ensure isochrony and hardware buffer management
- Data and control pipe constructs for ensuring independence from adverse interactions between functions

4.5.1 Error Detection

The core bit error rate of the USB medium is expected to be close to that of a backplane and any glitches will very likely be transient in nature. To provide protection against such transients, each packet includes error protection fields. When data integrity is required, such as with lossless data devices, an error recovery procedure may be invoked in hardware or software.

The protocol includes separate CRCs for control and data fields of each packet. A failed CRC is considered to indicate a corrupted packet. The CRC gives 100% coverage on single- and double-bit errors.

4.5.2 Error Handling

The protocol allows for error handling in hardware or software. Hardware error handling includes reporting and retry of failed transfers. A USB Host Controller will try a transmission that encounters errors up to three times before informing the client software of the failure. The client software can recover in an implementation-specific way.

4.6 System Configuration

The USB supports USB devices attaching to and detaching from the USB at any time. Consequently, system software must accommodate dynamic changes in the physical bus topology.

4.6.1 Attachment of USB Devices

All USB devices attach to the USB through ports on specialized USB devices known as hubs. Hubs have status bits that are used to report the attachment or removal of a USB device on one of its ports. The host queries the hub to retrieve these bits. In the case of an attachment, the host enables the port and addresses the USB device through the device's control pipe at the default address.

The host assigns a unique USB address to the device and then determines if the newly attached USB device is a hub or a function. The host establishes its end of the control pipe for the USB device using the assigned USB address and endpoint number zero.

If the attached USB device is a hub and USB devices are attached to its ports, then the above procedure is followed for each of the attached USB devices.

If the attached USB device is a function, then attachment notifications will be handled by host software that is appropriate for the function.

4.6.2 Removal of USB Devices

When a USB device has been removed from one of a hub's ports, the hub disables the port and provides an indication of device removal to the host. The removal indication is then handled by appropriate USB System Software. If the removed USB device is a hub, the USB System Software must handle the removal of both the hub and of all of the USB devices that were previously attached to the system through the hub.

4.6.3 Bus Enumeration

Bus enumeration is the activity that identifies and assigns unique addresses to devices attached to a bus. Because the USB allows USB devices to attach to or detach from the USB at any time, bus enumeration is an on-going activity for the USB System Software.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Additionally, bus enumeration for the USB also includes the detection and processing of removals.

4.7 Data Flow Types

The USB supports functional data and control exchange between the USB host and a USB device as a set of either uni-directional or bi-directional pipes. USB data transfers take place between host software and a particular endpoint on a USB device. Such associations between the host software and a USB device endpoint are called pipes. In general, data movement though one pipe is independent from the data flow in any other pipe. A given USB device may have many pipes. As an example, a given USB device could have an endpoint that supports a pipe for transporting data to the USB device and another endpoint that supports a pipe for transporting data from the USB device.

The USB architecture comprehends four basic types of data transfers:

- Control Transfers: Used to configure a device at attach time and can be used for other device-specific purposes, including control of other pipes on the device.
- Bulk Data Transfers: Generated or consumed in relatively large and bursty quantities and have wide dynamic latitude in transmission constraints.
- Interrupt Data Transfers: Used for timely but reliable delivery of data, for example, characters or coordinates with human-perceptible echo or feedback response characteristics.
- Isochronous Data Transfers: Occupy a prenegotiated amount of USB bandwidth with a prenegotiated delivery latency. (Also called streaming real time transfers).

A pipe supports only one of the types of transfers described above for any given device configuration. The USB data flow model is described in more detail in Chapter 5.

4.7.1 Control Transfers

Control data is used by the USB System Software to configure devices when they are first attached. Other driver software can choose to use control transfers in implementation-specific ways. Data delivery is lossless.

4.7.2 Bulk Transfers

Bulk data typically consists of larger amounts of data, such as that used for printers or scanners. Bulk data is sequential. Reliable exchange of data is ensured at the hardware level by using error detection in hardware and invoking a limited number of retries in hardware. Also, the bandwidth taken up by bulk data can vary, depending on other bus activities.

4.7.3 Interrupt Transfers

A limited-latency transfer to or from a device is referred to as interrupt data. Such data may be presented for transfer by a device at any time and is delivered by the USB at a rate no slower than is specified by the device.

Interrupt data typically consists of event notification, characters, or coordinates that are organized as one or more bytes. An example of interrupt data is the coordinates from a pointing device. Although an explicit timing rate is not required, interactive data may have response time bounds that the USB must support.

4.7.4 Isochronous Transfers

Isochronous data is continuous and real-time in creation, delivery, and consumption. Timing-related information is implied by the steady rate at which isochronous data is received and transferred. Isochronous data must be delivered at the rate received to maintain its timing. In addition to delivery rate, isochronous data may also be sensitive to delivery delays. For isochronous pipes, the bandwidth required is typically based upon the sampling

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

characteristics of the associated function. The latency required is related to the buffering available at each endpoint.

A typical example of isochronous data is voice. If the delivery rate of these data streams is not maintained, drop-outs in the data stream will occur due to buffer or frame underruns or overruns. Even if data is delivered at the appropriate rate by USB hardware, delivery delays introduced by software may degrade applications requiring real-time turn-around, such as telephony-based audio conferencing.

The timely delivery of isochronous data is ensured at the expense of potential transient losses in the data stream. In other words, any error in electrical transmission is not corrected by hardware mechanisms such as retries. In practice, the core bit error rate of the USB is expected to be small enough not to be an issue. USB isochronous data streams are allocated a dedicated portion of USB bandwidth to ensure that data can be delivered at the desired rate. The USB is also designed for minimal delay of isochronous data transfers.

4.7.5 Allocating USB Bandwidth

USB bandwidth is allocated among pipes. The USB allocates bandwidth for some pipes when a pipe is established. USB devices are required to provide some buffering of data. It is assumed that USB devices requiring more bandwidth are capable of providing larger buffers. The goal for the USB architecture is to ensure that buffering-induced hardware delay is bounded to within a few milliseconds.

The USB's bandwidth capacity can be allocated among many different data streams. This allows a wide range of devices to be attached to the USB. Further, different device bit rates, with a wide dynamic range, can be concurrently supported.

The USB Specification defines the rules for how each transfer type is allowed access to the bus.

4.8 USB Devices

USB devices are divided into device classes such as hub, human interface, printer, imaging, or mass storage device. The hub device class indicates a specially designated USB device that provides additional USB attachment points (refer to Chapter 11). USB devices are required to carry information for self-identification and generic configuration. They are also required at all times to display behavior consistent with defined USB device states.

4.8.1 Device Characterizations

All USB devices are accessed by a USB address that is assigned when the device is attached and enumerated. Each USB device additionally supports one or more pipes through which the host may communicate with the device. All USB devices must support a specially designated pipe at endpoint zero to which the USB device's USB control pipe will be attached. All USB devices support a common access mechanism for accessing information through this control pipe.

Associated with the control pipe at endpoint zero is the information required to completely describe the USB device. This information falls into the following categories:

- Standard: This is information whose definition is common to all USB devices and includes items such as vendor identification, device class, and power management capability. Device, configuration, interface, and endpoint descriptions carry configuration-related information about the device. Detailed information about these descriptors can be found in Chapter 9.
- Class: The definition of this information varies, depending on the device class of the USB device.
- USB Vendor: The vendor of the USB device is free to put any information desired here. The format, however, is not determined by this specification.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Additionally, each USB device carries USB control and status information.

4.8.2 Device Descriptions

Two major divisions of device classes exist: hubs and functions. Only hubs have the ability to provide additional USB attachment points. Functions provide additional capabilities to the host.

4.8.2.1 Hubs

Hubs are a key element in the plug-and-play architecture of the USB. Figure 4-3 shows a typical hub. Hubs serve to simplify USB connectivity from the user's perspective and provide robustness at relatively low cost and complexity.

Hubs are wiring concentrators and enable the multiple attachment characteristics of the USB. Attachment points are referred to as ports. Each hub converts a single attachment point into multiple attachment points. The architecture supports concatenation of multiple hubs.

The upstream port of a hub connects the hub towards the host. Each of the downstream ports of a hub allows connection to another hub or function. Hubs can detect attach and detach at each downstream port and enable the distribution of power to downstream devices. Each downstream port can be individually enabled and attached to either high-, full- or low-speed devices.

A USB 2.0 hub consists of three portions: the Hub Controller, the Hub Repeater, and the Transaction Translator. The Hub Repeater is a protocol-controlled switch between the upstream port and downstream ports. It also has hardware support for reset and suspend/resume signaling. The Host Controller provides the communication to/from the host. Hub-specific status and control commands permit the host to configure a hub and to monitor and control its ports. The Transaction Translator provides the mechanisms that support full-/low-speed devices behind the hub, while transmitting all device data between the host and the hub at high-speed.

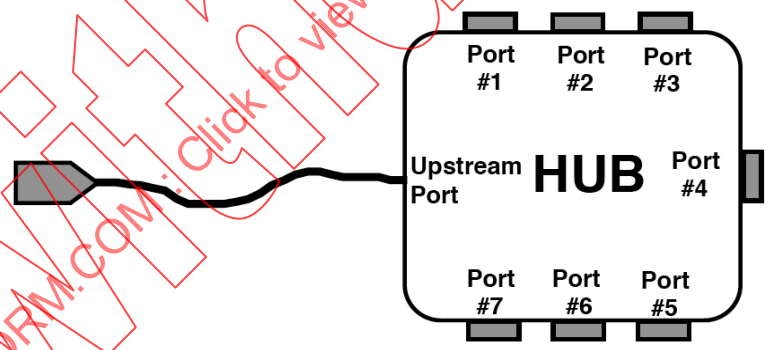


Figure 4-3. A Typical Hub

Figure 4-4 illustrates how hubs provide connectivity in a typical desktop computer environment.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

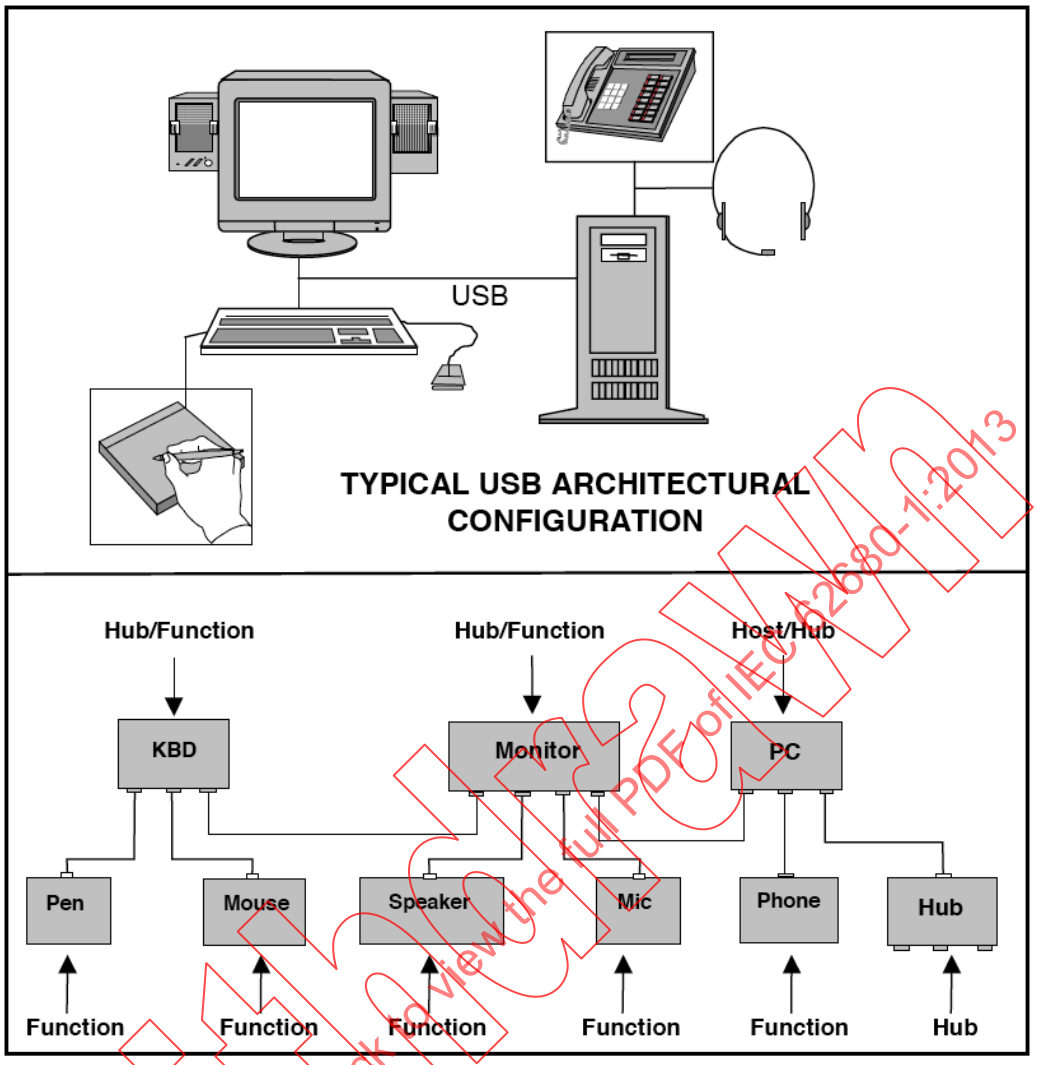


Figure 4-4. Hubs in a Desktop Computer Environment

4.8.2.2 Functions

A function is a USB device that is able to transmit or receive data or control information over the bus. A function is typically implemented as a separate peripheral device with a cable that plugs into a port on a hub. However, a physical package may implement multiple functions and an embedded hub with a single USB cable. This is known as a compound device. A compound device appears to the host as a hub with one or more non-removable USB devices.

Each function contains configuration information that describes its capabilities and resource requirements. Before a function can be used, it must be configured by the host. This configuration includes allocating USB bandwidth and selecting function-specific configuration options.

Examples of functions include the following:

- A human interface device such as a mouse, keyboard, tablet, or game controller
- An imaging device such as a scanner, printer, or camera
- A mass storage device such as a CD-ROM drive, floppy drive, or DVD drive

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4.9 USB Host: Hardware and Software

The USB host interacts with USB devices through the Host Controller. The host is responsible for the following:

- Detecting the attachment and removal of USB devices
- Managing control flow between the host and USB devices
- Managing data flow between the host and USB devices
- Collecting status and activity statistics
- Providing power to attached USB devices

The USB System Software on the host manages interactions between USB devices and host-based device software. There are five areas of interactions between the USB System Software and device software:

- Device enumeration and configuration
- Isochronous data transfers
- Asynchronous data transfers
- Power management
- Device and bus management information

4.10 Architectural Extensions

The USB architecture comprehends extensibility at the interface between the Host Controller Driver and USB Driver. Implementations with multiple Host Controllers, and associated Host Controller Drivers, are possible.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5 Chapter 5
USB Data Flow Model

This chapter presents information about how data is moved across the USB. The information in this chapter affects all implementers. The information presented is at a level above the signaling and protocol definitions of the system. Consult Chapter 7 and Chapter 8 for more details about their respective parts of the USB system. This chapter provides framework information that is further expanded in Chapters 9 through 11. All implementers should read this chapter so they understand the key concepts of the USB.

5.1 Implementer Viewpoints

The USB provides communication services between a host and attached USB devices. However, the simple view an end user sees of attaching one or more USB devices to a host, as in Figure 5-1, is in fact a little more complicated to implement than is indicated by the figure. Different views of the system are required to explain specific USB requirements from the perspective of different implementers. Several important concepts and features must be supported to provide the end user with the reliable operation demanded from today's personal computers. The USB is presented in a layered fashion to ease explanation and allow implementers of particular USB products to focus on the details related to their product.



Figure 5-1. Simple USB Host/Device View

Figure 5-2 shows a deeper overview of the USB, identifying the different layers of the system that will be described in more detail in the remainder of the specification. In particular, there are four focus implementation areas:

- USB Physical Device: A piece of hardware on the end of a USB cable that performs some useful end user function.
- Client Software: Software that executes on the host, corresponding to a USB device. This client software is typically supplied with the operating system or provided along with the USB device.
- USB System Software: Software that supports the USB in a particular operating system. The USB System Software is typically supplied with the operating system, independently of particular USB devices or client software.
- USB Host Controller (Host Side Bus Interface): The hardware and software that allows USB devices to be attached to a host.

There are shared rights and responsibilities between the four USB system components. The remainder of this specification describes the details required to support robust, reliable communication flows between a function and its client.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

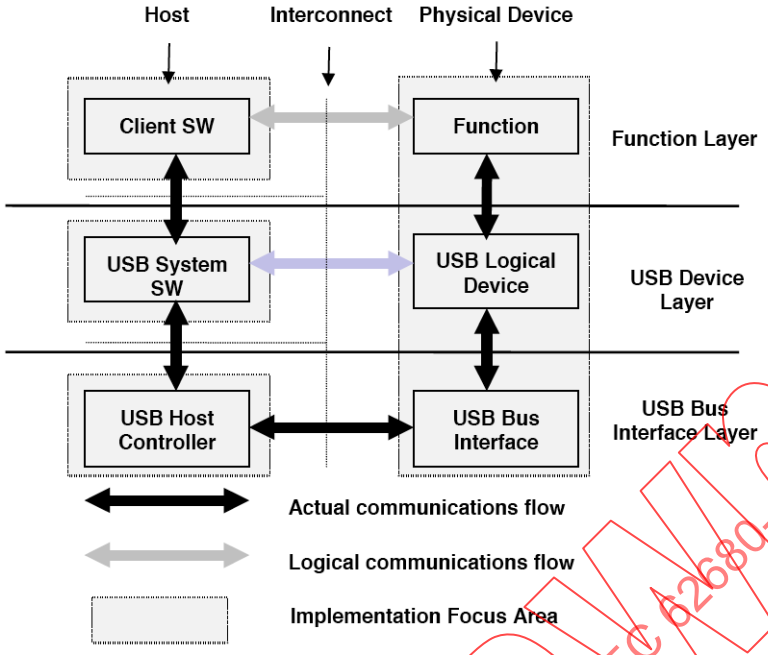


Figure 5-2. USB Implementation Areas

As shown in Figure 5-2, the simple connection of a host to a device requires interaction between a number of layers and entities. The USB Bus Interface layer provides physical/signaling/packet connectivity between the host and a device. The USB Device layer is the view the USB System Software has for performing generic USB operations with a device. The Function layer provides additional capabilities to the host via an appropriate matched client software layer. The USB Device and Function layers each have a view of logical communication within their layer that actually uses the USB Bus Interface layer to accomplish data transfer.

The physical view of USB communication as described in Chapters 6, 7, and 8 is related to the logical communication view presented in Chapters 9 and 10. This chapter describes those key concepts that affect USB implementers and should be read by all before proceeding to the remainder of the specification to find those details most relevant to their product.

To describe and manage USB communication, the following concepts are important:

- Bus Topology: Section 5.2 presents the primary physical and logical components of the USB and how they interrelate.
- Communication Flow Models: Sections 5.3 through 5.8 describe how communication flows between the host and devices through the USB and defines the four USB transfer types.
- Bus Access Management: Section 5.11 describes how bus access is managed within the host to support a broad range of communication flows by USB devices.
- Special Consideration for Isochronous Transfers: Section 5.12 presents features of the USB specific to devices requiring isochronous data transfers. Device implementers for non-isochronous devices do not need to read Section 5.12.

5.2 Bus Topology

There are four main parts to USB topology:

- Host and Devices: The primary components of a USB system
- Physical Topology: How USB elements are connected

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Logical Topology: The roles and responsibilities of the various USB elements and how the USB appears from the perspective of the host and a device
- Client Software-to-function Relationships: How client software and its related function interfaces on a USB device view each other

5.2.1 USB Host

The host's logical composition is shown in Figure 5-3 and includes the following:

- USB Host Controller
- Aggregate USB System Software (USB Driver, Host Controller Driver, and host software)
- Client

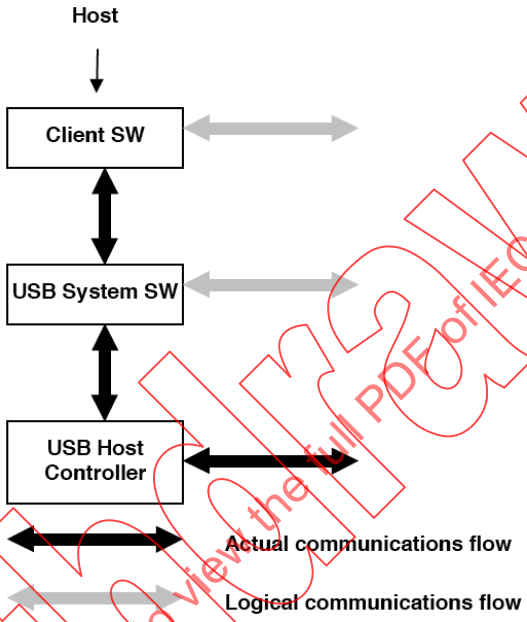


Figure 5-3. Host Composition

The USB host occupies a unique position as the coordinating entity for the USB. In addition to its special physical position, the host has specific responsibilities with regard to the USB and its attached devices. The host controls all access to the USB. A USB device gains access to the bus only by being granted access by the host. The host is also responsible for monitoring the topology of the USB.

For a complete discussion of the host and its duties, refer to Chapter 10.

5.2.2 USB Devices

A USB physical device's logical composition is shown in Figure 5-4 and includes the following:

- USB bus interface
- USB logical device
- Function

USB physical devices provide additional functionality to the host. The types of functionality provided by USB devices vary widely. However, all USB logical devices present the same basic interface to the host. This allows the host to manage the USB-relevant aspects of different USB devices in the same manner.

To assist the host in identifying and configuring USB devices, each device carries and reports configuration-related information. Some of the information reported is common among all

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

logical devices. Other information is specific to the functionality provided by the device. The detailed format of this information varies, depending on the device class of the device.

For a complete discussion of USB devices, refer to Chapter 9.

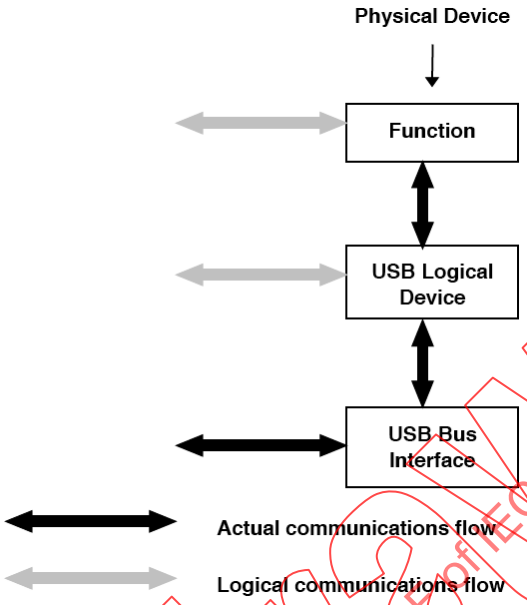


Figure 5-4. Physical Device Composition

5.2.3 Physical Bus Topology

Devices on the USB are physically connected to the host via a tiered star topology, as illustrated in Figure 5-5. USB attachment points are provided by a special class of USB device known as a hub. The additional attachment points provided by a hub are called ports. A host includes an embedded hub called the root hub. The host provides one or more attachment points via the root hub. USB devices that provide additional functionality to the host are known as functions. To prevent circular attachments, a tiered ordering is imposed on the star topology of the USB. This results in the tree-like configuration illustrated in Figure 5-5.

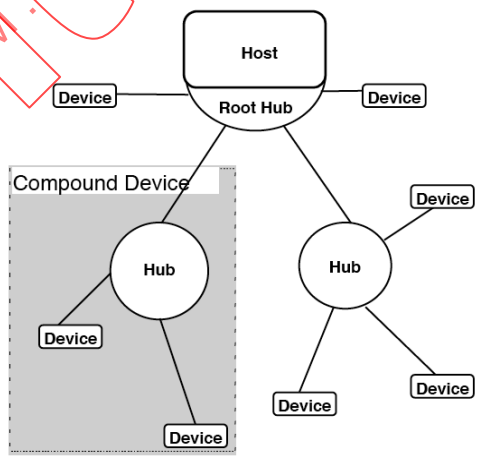


Figure 5-5. USB Physical Bus Topology

Multiple functions may be packaged together in what appears to be a single physical device. For example, a keyboard and a trackball might be combined in a single package. Inside the package, the individual functions are permanently attached to a hub and it is the internal hub that is connected to the USB. When multiple functions are combined with a hub in a single

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

package, they are referred to as a compound device. The hub and each function attached to the hub within the compound device is assigned its own device address. A device that has multiple interfaces controlled independently of each other is referred to as a composite device. A composite device has only a single device address. From the host's perspective, a compound device is the same as a separate hub with multiple functions attached. Figure 5-5 also illustrates a compound device.

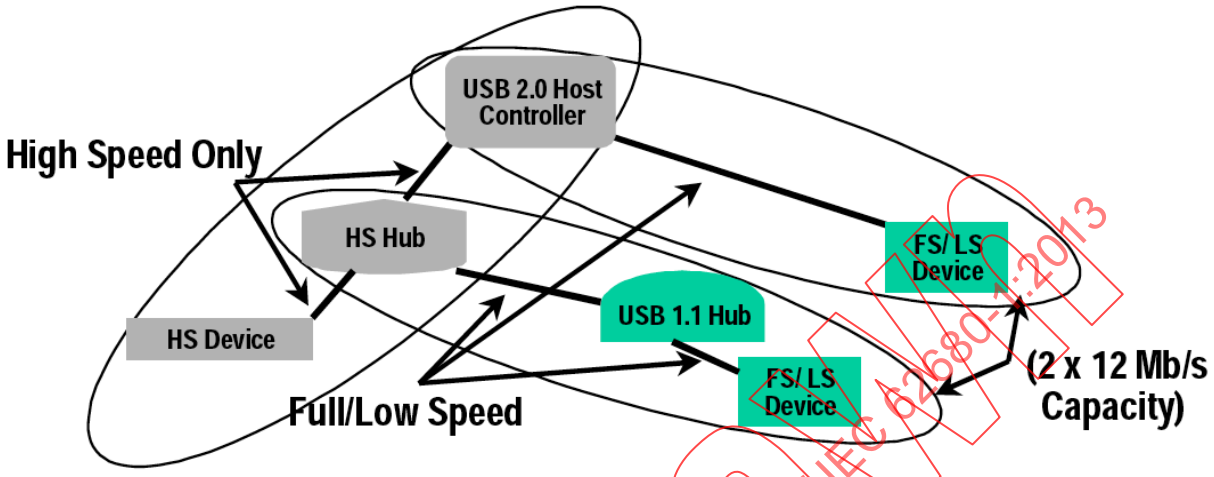


Figure 5-6. Multiple Full-speed Buses in a High-speed System

The hub plays a special role in a high-speed system. The hub isolates the full-/low-speed signaling environment from the high-speed signaling environment. Figure 5-6 shows a hub operating in high speed supporting a high-speed attached device. The hub also allows USB1.1 hubs to attach and operate at full-/low-speed along with other full-/low-speed only devices. The host controller also directly supports attaching full-/low-speed only devices. Chapter 11 describes the details of how the hub accomplishes the isolation of the two signaling environments.

Each high-speed operating hub essentially adds one (or more) additional full-/low-speed buses; i.e., each hub supports additional (optionally multiple) 12 Mb/s of USB full-/low-speed bandwidth. This allows more full-/low-speed buses to be attached without requiring additional host controllers in a system. Even though there can be several 12 Mb/s full-/low-speed buses, there are only at most 127 USB devices attached to any single host controller.

5.2.4 Logical Bus Topology

While devices physically attach to the USB in a tiered, star topology, the host communicates with each logical device as if it were directly connected to the root port. This creates the logical view illustrated in Figure 5-7 that corresponds to the physical topology shown in Figure 5-5. Hubs are logical devices also but are not shown in Figure 5-7 to simplify the picture. Even though most host/logical device activities use this logical perspective, the host maintains an awareness of the physical topology to support processing the removal of hubs. When a hub is removed, all of the devices attached to the hub must be removed from the host's view of the logical topology. A more complete discussion of hubs can be found in Chapter 11.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

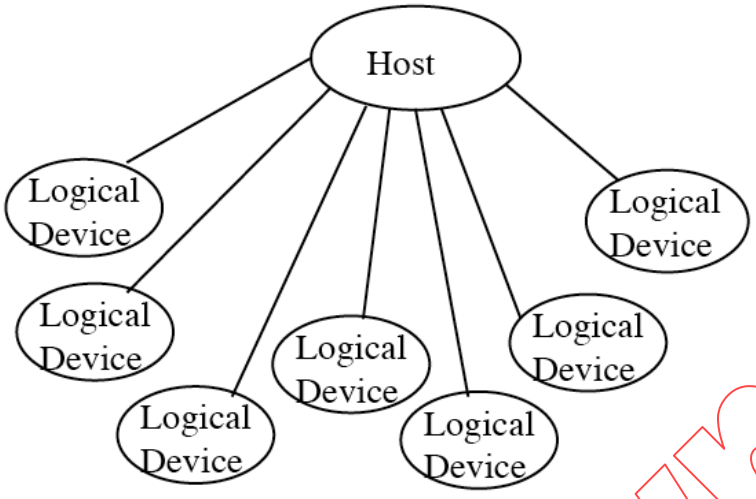


Figure 5-7. USB Logical Bus Topology

5.2.5 Client Software-to-function Relationship

Even though the physical and logical topology of the USB reflects the shared nature of the bus, client software (CSw) manipulating a USB function interface is presented with the view that it deals only with its interface(s) of interest. Client software for USB functions must use USB software programming interfaces to manipulate their functions as opposed to directly manipulating their functions via memory or I/O accesses as with other buses (e.g., PCI, EISA, PCMCIA, etc.). During operation, client software should be independent of other devices that may be connected to the USB. This allows the designer of the device and client software to focus on the hardware/software interaction design details. Figure 5-8 illustrates a device designer's perspective of the relationships of client software and USB functions with respect to the USB logical topology of Figure 5-7.

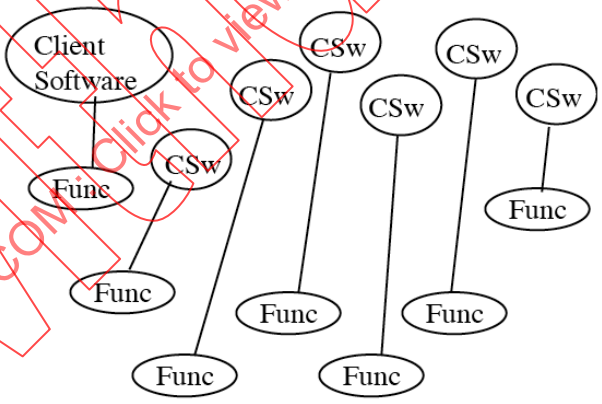


Figure 5-8. Client Software-to-function Relationships

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.3 USB Communication Flow

The USB provides a communication service between software on the host and its USB function. Functions can have different communication flow requirements for different client-to-function interactions. The USB provides better overall bus utilization by allowing the separation of the different communication flows to a USB function. Each communication flow makes use of some bus access to accomplish communication between client and function. Each communication flow is terminated at an endpoint on a device. Device endpoints are used to identify aspects of each communication flow.

Figure 5-9 shows a more detailed view of Figure 5-2. The complete definition of the actual communication flows of Figure 5-2 supports the logical device and function layer

communication flows. These actual communication flows cross several interface boundaries. Chapters 6 through 8 describe the mechanical, electrical, and protocol interface definitions of the USB "wire." Chapter 9 describes the USB device programming interface that allows a USB device to be manipulated from the host side of the wire. Chapter 10 describes two host side software interfaces:

- Host Controller Driver (HCD): The software interface between the USB Host Controller and USB System Software. This interface allows a range of Host Controller implementations without requiring all host software to be dependent on any particular implementation. One USB Driver can support different Host Controllers without requiring specific knowledge of a Host Controller implementation. A Host Controller implementer provides an HCD implementation that supports the Host Controller.
- USB Driver (USBD): The interface between the USB System Software and the client software. This interface provides clients with convenient functions for manipulating USB devices.

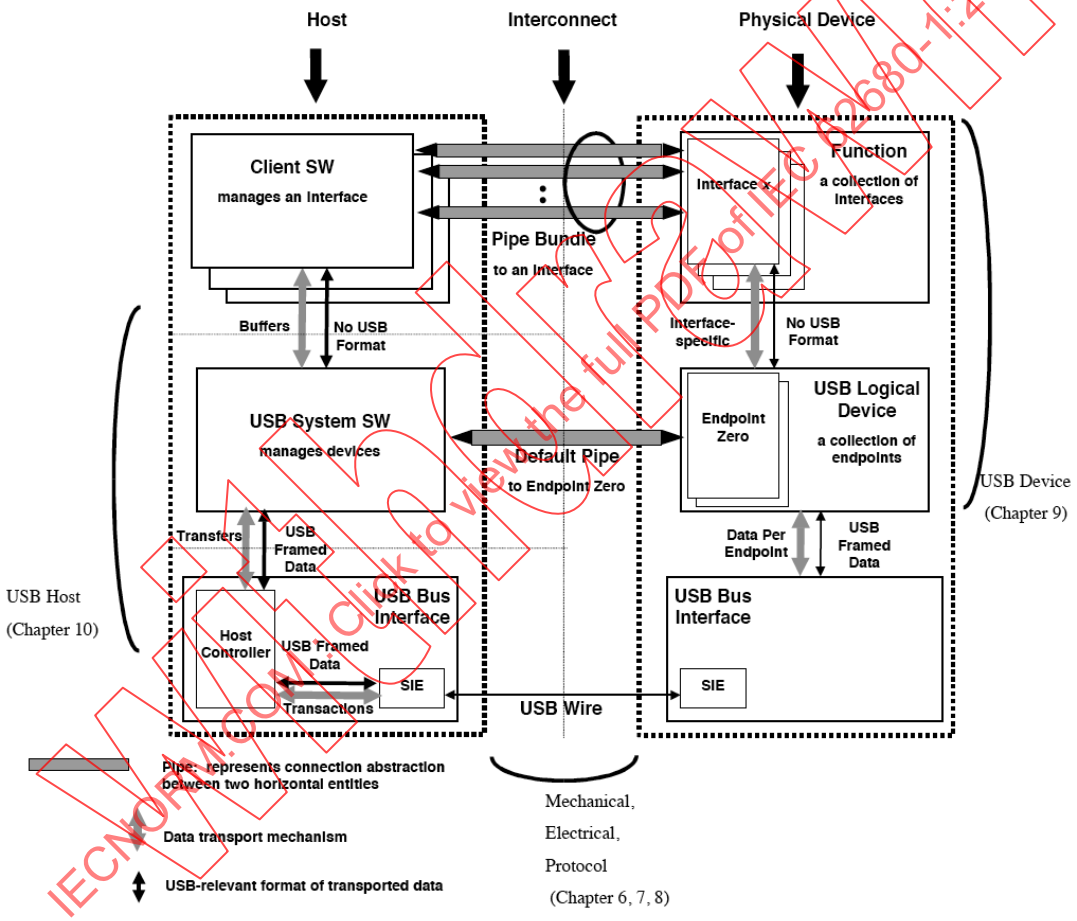


Figure 5-9. USB Host/Device Detailed View

A USB logical device appears to the USB system as a collection of endpoints. Endpoints are grouped into endpoint sets that implement an interface. Interfaces are views to the function. The USB System Software manages the device using the Default Control Pipe. Client software manages an interface using pipe bundles (associated with an endpoint set). Client software requests that data be moved across the USB between a buffer on the host and an endpoint on the USB device. The Host Controller (or USB device, depending on transfer direction) packetizes the data to move it over the USB. The Host Controller also coordinates when bus access is used to move the packet of data over the USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 5-10 illustrates how communication flows are carried over pipes between endpoints and host side memory buffers. The following sections describe endpoints, pipes, and communication flows in more detail.

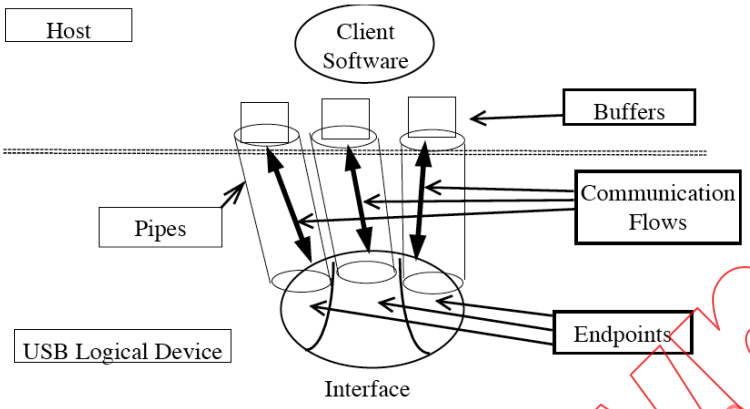


Figure 5-10. USB Communication Flow

Software on the host communicates with a logical device via a set of communication flows. The set of communication flows are selected by the device software/hardware designer(s) to efficiently match the communication requirements of the device to the transfer characteristics provided by the USB.

5.3.1 Device Endpoints

An endpoint is a uniquely identifiable portion of a USB device that is the terminus of a communication flow between the host and device. Each USB logical device is composed of a collection of independent endpoints. Each logical device has a unique address assigned by the system at device attachment time. Each endpoint on a device is given at design time a unique device-determined identifier called the endpoint number. Each endpoint has a device-determined direction of data flow. The combination of the device address, endpoint number, and direction allows each endpoint to be uniquely referenced. Each endpoint is a simplex connection that supports data flow in one direction: either input (from device to host) or output (from host to device).

An endpoint has characteristics that determine the type of transfer service required between the endpoint and the client software. An endpoint describes itself by:

- Bus access frequency/latency requirement
- Bandwidth requirement
- Endpoint number
- Error handling behavior requirements
- Maximum packet size that the endpoint is capable of sending or receiving
- The transfer type for the endpoint (refer to Section 5.4 for details)
- The direction in which data is transferred between the endpoint and the host

Endpoints other than those with endpoint number zero are in an unknown state before being configured and may not be accessed by the host before being configured.

5.3.1.1 Endpoint Zero Requirements

All USB devices are required to implement a default control method that uses both the input and output endpoints with endpoint number zero. The USB System Software uses this default control method to initialize and generically manipulate the logical device (e.g., to configure the logical device) as the Default Control Pipe (see Section 5.3.2). The Default Control Pipe

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

provides access to the device's configuration information and allows generic USB status and control access. The Default Control Pipe supports control transfers as defined in Section 5.5. The endpoints with endpoint number zero are always accessible once a device is attached, powered, and has received a bus reset.

A USB device that is capable of operating at high-speed must have a minimum level of support for operating at full-speed. When the device is attached to a hub operating in full-speed, the device must:

- Be able to reset successfully at full-speed
- Respond successfully to standard requests: set_address, set_configuration, get_descriptor for device and configuration descriptors, and return appropriate information

The high-speed device may or may not be able to support its intended functionality when operating at full-speed.

5.3.1.2 Non-endpoint Zero Requirements

Functions can have additional endpoints as required for their implementation. Low-speed functions are limited to two optional endpoints beyond the two required to implement the Default Control Pipe. Full-speed devices can have additional endpoints only limited by the protocol definition (i.e., a maximum of 15 additional input endpoints and 15 additional output endpoints).

Endpoints other than those for the Default Control Pipe cannot be used until the device is configured as a normal part of the device configuration process (refer to Chapter 9).

5.3.2 Pipes

A USB pipe is an association between an endpoint on a device and software on the host. Pipes represent the ability to move data between software on the host via a memory buffer and an endpoint on a device. There are two mutually exclusive pipe communication modes:

- Stream: Data moving through a pipe has no USB-defined structure
- Message: Data moving through a pipe has some USB-defined structure

The USB does not interpret the content of data it delivers through a pipe. Even though a message pipe requires that data be structured according to USB definitions, the content of the data is not interpreted by the USB.

Additionally, pipes have the following associated with them:

- A claim on USB bus access and bandwidth usage.
- A transfer type.
- The associated endpoint's characteristics, such as directionality and maximum data payload sizes. The data payload is the data that is carried in the data field of a data packet within a bus transaction (as defined in Chapter 8).

The pipe that consists of the two endpoints with endpoint number zero is called the Default Control Pipe. This pipe is always available once a device is powered and has received a bus reset. Other pipes come into existence when a USB device is configured. The Default Control Pipe is used by the USB System Software to determine device identification and configuration requirements and to configure the device. The Default Control Pipe can also be used by device-specific software after the device is configured. The USB System Software retains "ownership" of the Default Control Pipe and mediates use of the pipe by other client software.

A software client normally requests data transfers via I/O Request Packets (IRPs) to a pipe and then either waits or is notified when they are completed. Details about IRPs are defined in an operating system-specific manner. This specification uses the term to simply refer to an

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

identifiable request by a software client to move data between itself (on the host) and an endpoint of a device in an appropriate direction. A software client can cause a pipe to return all outstanding IRPs if it desires. The software client is notified that an IRP has completed when the bus transactions associated with it have completed either successfully or due to errors.

If there are no IRPs pending or in progress for a pipe, the pipe is idle and the Host Controller will take no action with regard to the pipe; i.e., the endpoint for such a pipe will not see any bus transactions directed to it. The only time bus activity is present for a pipe is when IRPs are pending for that pipe.

If a non-isochronous pipe encounters a condition that causes it to send a STALL to the host (refer to Chapter 8) or three bus errors are encountered on any packet of an IRP, the IRP is aborted/retired, all outstanding IRPs are also retired, and no further IRPs are accepted until the software client recovers from the condition (in an implementation-dependent way) and acknowledges the halt or error condition via a USBD call. An appropriate status informs the software client of the specific IRP result for error versus halt (refer to Chapter 10). Isochronous pipe behavior is described in Section 5.6.

An IRP may require multiple data payloads to move the client data over the bus. The data payloads for such a multiple data payload IRP are expected to be of the maximum packet size until the last data payload that contains the remainder of the overall IRP. See the description of each transfer type for more details. For such an IRP, short packets (i.e., less than maximum-sized data payloads) on input that do not completely fill an IRP data buffer can have one of two possible meanings, depending upon the expectations of a client:

- A client can expect a variable-sized amount of data in an IRP. In this case, a short packet that does not fill an IRP data buffer can be used simply as an in-band delimiter to indicate “end of unit of data.” The IRP should be retired without error and the Host Controller should advance to the next IRP.
- A client can expect a specific-sized amount of data. In this case, a short packet that does not fill an IRP data buffer is an indication of an error. The IRP should be retired, the pipe should be stalled, and any pending IRPs associated with the pipe should also be retired.

Because the Host Controller must behave differently in the two cases and cannot know on its own which way to behave for a given IRP; it is possible to indicate per IRP which behavior the client desires.

An endpoint can inform the host that it is busy by responding with NAK. NAKs are not used as a retire condition for returning an IRP to a software client. Any number of NAKs can be encountered during the processing of a given IRP. A NAK response to a transaction does not constitute an error and is not counted as one of the three errors described above.

5.3.2.1 Stream Pipes

Stream pipes deliver data in the data packet portion of bus transactions with no USB-required structure on the data content. Data flows in at one end of a stream pipe and out the other end in the same order. Stream pipes are always uni-directional in their communication flow.

Data flowing through a stream pipe is expected to interact with what the USB believes is a single client. The USB System Software is not required to provide synchronization between multiple clients that may be using the same stream pipe. Data presented to a stream pipe is moved through the pipe in sequential order: first-in, first-out.

A stream pipe to a device is bound to a single device endpoint number in the appropriate direction (i.e., corresponding to an IN or OUT token as defined by the protocol layer). The device endpoint number for the opposite direction can be used for some other stream pipe to the device.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Stream pipes support bulk, isochronous, and interrupt transfer types, which are explained in later sections.

5.3.2.2 Message Pipes

Message pipes interact with the endpoint in a different manner than stream pipes. First, a request is sent to the USB device from the host. This request is followed by data transfer(s) in the appropriate direction. Finally, a Status stage follows at some later time. In order to accommodate the request/data/status paradigm, message pipes impose a structure on the communication flow that allows commands to be reliably identified and communicated. Message pipes allow communication flow in both directions, although the communication flow may be predominately one way. The Default Control Pipe is always a message pipe.

The USB System Software ensures that multiple requests are not sent to a message pipe concurrently. A device is required to service only a single message request at a time per message pipe. Multiple software clients on the host can make requests via the Default Control Pipe, but they are sent to the device in a first-in, first-out order. A device can control the flow of information during the Data and Status stages based on its ability to respond to the host transactions (refer to Chapter 8 for more details).

A message pipe will not normally be sent the next message from the host until the current message's processing at the device has been completed. However, there are error conditions whereby a message transfer can be aborted by the host and the message pipe can be sent a new message transfer prematurely (from the device's perspective). From the perspective of the software manipulating a message pipe, an error on some part of an IRP retires the current IRP and all queued IRPs. The software client that requested the IRP is notified of the IRP completion with an appropriate error indication.

A message pipe to a device requires a single device endpoint number in both directions (IN and OUT tokens). The USB does not allow a message pipe to be associated with different endpoint numbers for each direction.

Message pipes support the control transfer type, which is explained in Section 5.5.

5.3.3 Frames and Microframes

USB establishes a 1 millisecond time base called a frame on a full-/low-speed bus and a 125 μs time base called a microframe on a high-speed bus. A (micro)frame can contain several transactions. Each transfer type defines what transactions are allowed within a (micro)frame for an endpoint. Isochronous and interrupt endpoints are given opportunities to the bus every N (micro)frames. The values of N and other details about isochronous and interrupt transfers are described in Sections 5.6 and 5.7.

5.4 Transfer Types

The USB transports data through a pipe between a memory buffer associated with a software client on the host and an endpoint on the USB device. Data transported by message pipes is carried in a USB-defined structure, but the USB allows device-specific structured data to be transported within the USB-defined message data payload. The USB also defines that data moved over the bus is packetized for any pipe (stream or message), but ultimately the formatting and interpretation of the data transported in the data payload of a bus transaction is the responsibility of the client software and function using the pipe. However, the USB provides different transfer types that are optimized to more closely match the service requirements of the client software and function using the pipe. An IRP uses one or more bus transactions to move information between a software client and its function.

Each transfer type determines various characteristics of the communication flow including the following:

- Data format imposed by the USB

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Direction of communication flow
- Packet size constraints
- Bus access constraints
- Latency constraints
- Required data sequences
- Error handling

The designers of a USB device choose the capabilities for the device's endpoints. When a pipe is established for an endpoint, most of the pipe's transfer characteristics are determined and remain fixed for the lifetime of the pipe. Transfer characteristics that can be modified are described for each transfer type.

The USB defines four transfer types:

- Control Transfers: Bursty, non-periodic, host software-initiated request/response communication, typically used for command/status operations.
- Isochronous Transfers: Periodic, continuous communication between host and device, typically used for time-relevant information. This transfer type also preserves the concept of time encapsulated in the data. This does not imply, however, that the delivery needs of such data is always time-critical.
- Interrupt Transfers: Low-frequency, bounded-latency communication.
- Bulk Transfers: Non-periodic, large-packet bursty communication, typically used for data that can use any available bandwidth and can also be delayed until bandwidth is available.

Each transfer type is described in detail in the following four major sections. The data for any IRP is carried by the data field of the data packet as described in Section 8.3.4. Chapter 8 also describes details of the protocol that are affected by use of each particular transfer type.

5.4.1 Table Calculation Examples

The following sections describe each of the USB transfer types. In these sections, there are tables that illustrate the maximum number of transactions that can be expected to be contained in a (micro)frame. These tables can be used to determine the maximum performance behavior possible for a specific transfer type. Actual performance may vary with specific system implementation details.

Each table shows:

- The protocol overhead required for the specific transfer type (and speed)
- For some sample data payload sizes:
  - The maximum sustained bandwidth possible for this case
  - The percentage of a (micro)frame that each transaction requires
  - The maximum number of transactions in a (micro)frame for the specific case
  - The remaining bytes in a (micro)frame that would not be required for the specific case
  - The total number of data bytes transported in a single (micro)frame for the specific case

A transaction of a particular transfer type typically requires multiple packets. The protocol overhead for each transaction includes:

- A SYNC field for each packet: either 8 bits (full-/low-speed) or 32 bits (high-speed)
- A PID byte for each packet: includes PID and PID invert (check) bits
- An EOP for each packet: 3 bits (full-/low-speed) or 8 bits (high-speed)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- In a token packet, the endpoint number, device address, and CRC5 fields (16 bits total)
- In a data packet, CRC16 fields (16 bits total)
- In a data packet, any data field (8 bits per byte)
- For transaction with multiple packets, the inter packet gap or bus turnaround time required.

For these calculations, there is assumed to be no bit-stuffing required.

Using the low speed interrupt OUT as an example, there are 5 packets in the transaction:

- A PRE special packet
- A token packet
- A PRE special packet
- A data packet
- A handshake packet

There is one bus turnaround between the data and handshake packets. The protocol overhead is therefore:

5 SYNC, 5 PID, Endpoint + CRC5, CRC16, 5 EOPs and interpacket delay (one bus turnaround, 1 delay between packets, and 2 hub setup times).

5.5 Control Transfers

Control transfers allow access to different parts of a device. Control transfers are intended to support configuration/command/status type communication flows between client software and its function. A control transfer is composed of a Setup bus transaction moving request information from host to function, zero or more Data transactions sending data in the direction indicated by the Setup transaction, and a Status transaction returning status information from function to host. The Status transaction returns "success" when the endpoint has successfully completed processing the requested operation. Section 8.5.3 describes the details of what packets, bus transactions, and transaction sequences are used to accomplish a control transfer. Chapter 9 describes the details of the defined USB command codes.

Each USB device is required to implement the Default Control Pipe as a message pipe. This pipe is used by the USB System Software. The Default Control Pipe provides access to the USB device's configuration, status, and control information. A function can, but is not required to, provide endpoints for additional control pipes for its own implementation needs.

The USB device framework (refer to Chapter 9) defines standard, device class, or vendor-specific requests that can be used to manipulate a device's state. Descriptors are also defined that can be used to contain different information on the device. Control transfers provide the transport mechanism to access device descriptors and make requests of a device to manipulate its behavior.

Control transfers are carried only through message pipes. Consequently, data flows using control transfers must adhere to USB data structure definitions as described in Section 5.5.1.

The USB system will make a "best effort" to support delivery of control transfers between the host and devices. A function and its client software cannot request specific bus access frequency or bandwidth for control transfers. The USB System Software may restrict the bus access and bandwidth that a device may desire for control transfers. These restrictions are defined in Section 5.5.3 and Section 5.5.4.

5.5.1 Control Transfer Data Format

The Setup packet has a USB-defined structure that accommodates the minimum set of commands required to enable communication between the host and a device. The structure definition allows vendor-specific extensions for device specific commands. The Data

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

transactions following Setup have a USB-defined structure except when carrying vendor-specific information. The Status transaction also has a USB-defined structure. Specific control transfer Setup/Data definitions are described in Section 8.5.3 and Chapter 9.

5.5.2 Control Transfer Direction

Control transfers are supported via bi-directional communication flow over message pipes. As a consequence, when a control pipe is configured, it uses both the input and output endpoint with the specified endpoint number.

5.5.3 Control Transfer Packet Size Constraints

An endpoint for control transfers specifies the maximum data payload size that the endpoint can accept from or transmit to the bus. The allowable maximum control transfer data payload sizes for full-speed devices is 8, 16, 32, or 64 bytes; for high-speed devices, it is 64 bytes and for low-speed devices, it is 8 bytes. This maximum applies to the data payloads of the Data packets following a Setup; i.e., the size specified is for the data field of the packet as defined in Chapter 8, not including other information that is required by the protocol. A Setup packet is always eight bytes. A control pipe (including the Default Control Pipe) always uses its *wMaxPacketSize* value for data payloads.

An endpoint reports in its configuration information the value for its maximum data payload size. The USB does not require that data payloads transmitted be exactly the maximum size; i.e., if a data payload is less than the maximum, it does not need to be padded to the maximum size.

All Host Controllers are required to have support for 8-, 16-, 32-, and 64-byte maximum data payload sizes for full-speed control endpoints, only 8-byte maximum data payload sizes for low-speed control endpoints, and only 64-byte maximum data payload size for high-speed control endpoints. No Host Controller is required to support larger or smaller maximum data payload sizes.

In order to determine the maximum packet size for the Default Control Pipe, the USB System Software reads the device descriptor. The host will read the first eight bytes of the device descriptor. The device always responds with at least these initial bytes in a single packet. After the host reads the initial part of the device descriptor, it is guaranteed to have read this default pipe's *wMaxPacketSize* field (byte 7 of the device descriptor). It will then allow the correct size for all subsequent transactions. For all other control endpoints, the maximum data payload size is known after configuration so that the USB System Software can ensure that no data payload will be sent to the endpoint that is larger than the supported size.

An endpoint must always transmit data payloads with a data field less than or equal to the endpoint's *w*MaxPacketSize (refer to Chapter 9). When a control transfer involves more data than can fit in one data payload of the currently established maximum size, all data payloads are required to be maximum-sized except for the last data payload, which will contain the remaining data.

The Data stage of a control transfer from an endpoint to the host is complete when the endpoint does one of the following:

- Has transferred exactly the amount of data specified during the Setup stage
- Transfers a packet with a payload size less than *wMaxPacketSize* or transfers a zero-length packet

When a Data stage is complete, the Host Controller advances to the Status stage instead of continuing on with another data transaction. If the Host Controller does not advance to the Status stage when the Data stage is complete, the endpoint halts the pipe as was outlined in Section 5.3.2. If a larger-than-expected data payload is received from the endpoint, the IRP for the control transfer will be aborted/retired.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The Data stage of a control transfer from the host to an endpoint is complete when all of the data has been transferred. If the endpoint receives a larger-than-expected data payload from the host, it halts the pipe.

5.5.4 Control Transfer Bus Access Constraints

Control transfers can be used by high-speed, full-speed, and low-speed USB devices.

An endpoint has no way to indicate a desired bus access frequency for a control pipe. The USB balances the bus access requirements of all control pipes and the specific IRPs that are pending to provide “best effort” delivery of data between client software and functions.

The USB requires that part of each (micro)frame be reserved to be available for use by control transfers as follows:

- If the control transfers that are attempted (in an implementation-dependent fashion) consume less than 10% of the frame time for full-/low-speed endpoints or less than 20% of a microframe for high-speed endpoints, the remaining time can be used to support bulk transfers (refer to Section 5.8).
- A control transfer that has been attempted and needs to be retried can be retried in the current or a future (micro)frame; i.e., it is not required to be retried in the same (micro)frame.
- If there are more control transfers than reserved time, but there is additional (micro)frame time that is not being used for isochronous or interrupt transfers, a Host Controller may move additional control transfers as they are available.
- If there are too many pending control transfers for the available (micro)frame time, control transfers are selected to be moved over the bus as appropriate.
- If there are control transfers pending for multiple endpoints, control transfers for the different endpoints are selected according to a fair access policy that is Host Controller implementation-dependent.
- A transaction of a control transfer that is frequently being retried should not be expected to consume an unfair share of the bus time.

High-speed control endpoints must support the PING flow control protocol for OUT transactions. The details of this protocol are described in Section 8.5.1.

These requirements allow control transfers between host and devices to be regularly moved over the bus with “best effort.”

The USB System Software can, at its discretion, vary the rate of control transfers to a particular endpoint. An endpoint and its client software cannot assume a specific rate of service for control transfers. A control endpoint may see zero or more transfers in a single (micro)frame. Bus time made available to a software client and its endpoint can be changed as other devices are inserted into and removed from the system or also as control transfers are requested for other device endpoints.

The bus frequency and (micro)frame timing limit the maximum number of successful control transfers within a (micro)frame for any USB system. For full-/low-speed buses, the number of successful control transfers per frame is limited to less than 29 full-speed eight-byte data payloads or less than four low-speed eight-byte data payloads. For high-speed buses, the number of control transfers is limited to less than 32 high-speed 64-byte data payloads per microframe.

Table 5-1 lists information about different-sized low-speed packets and the maximum number of packets possible in a frame. The table does not include the overhead associated with bit stuffing.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 5-1. Low-speed Control Transfer Limits

	Protocol Overhead (63 bytes)		(15 SYNC bytes, 15 PID bytes, 6 Endpoint + CRC bytes, 6 CRC bytes, 8 Setup data bytes, and a 13-byte interpacket delay (EOP, etc.))			
	Data Payload	Max Bandwidth (bytes/second)	Frame Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/Frame Useful Data
	1	3000	26%	3	40	3
	2	6000	27%	3	37	6
	4	12000	28%	3	31	12
	8	24000	30%	3	19	24
Max		187500				187

For all speeds, because a control transfer is composed of several packets, the packets can be spread over several (micro)frames to spread the bus time required across several (micro)frames.

The 10% frame reservation for full-/low-speed non-periodic transfers means that in a system with bus time fully allocated, all full-speed control transfers in the system contend for a nominal three control transfers per frame. Because the USB system uses control transfers for configuration purposes in addition to whatever other control transfers other client software may be requesting, a given software client and its function should not expect to be able to make use of this full bandwidth for its own control purposes. Host Controllers are also free to determine how the individual bus transactions for specific control transfers are moved over the bus within and across frames. An endpoint could see all bus transactions for a control transfer within the same frame or spread across several noncontiguous frames. A Host Controller, for various implementation reasons, may not be able to provide the theoretical maximum number of control transfers per frame.

For high-speed endpoints, the 20% microframe reservation for non-periodic transfers means that all high speed control transfers are contending for nominally six control transfers per microframe. High-speed control transfers contend for microframe time along with split-transactions (see Sections 11.15-11.21 for more information about split transactions) for full- and low-speed control transfers. Both full-speed and low-speed control transfers contend for the same available frame time. However, high-speed control transfers for some endpoints can occur simultaneously with full- and low-speed control transfers for other endpoints. Low-speed control transfers simply take longer to transfer.

Table 5-2 lists information about different-sized full-speed control transfers and the maximum number of transfers possible in a frame. This table was generated assuming that there is one Data stage transaction and that the Data stage has a zero-length status phase. The table illustrates the possible power of two data payloads less than or equal to the allowable maximum data payload sizes. The table does not include the overhead associated with bit stuffing.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 5-2. Full-speed Control Transfer Limits

	Protocol Overhead (45 bytes)		(9 SYNC bytes, 9 PID bytes, 6 Endpoint + CRC bytes, 6 CRC bytes, 8 Setup data bytes, and a 7-byte interpacket delay (EOP, etc.))			
	Data Payload	Max Bandwidth (bytes/second)	Frame Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/Frame Useful Data
	1	32000	3%	32	23	32
	2	62000	3%	31	43	62
	4	120000	3%	30	30	120
	8	224000	4%	28	16	224
	16	384000	4%	24	36	384
	32	608000	5%	19	37	608
	64	832000	7%	13	83	832
Max		1500000				1500

Table 5-3 lists information about different-sized high-speed control transfers and the maximum number of transfers possible in a microframe. This table was generated assuming that there is one Data stage transaction and that the Data stage has a zero-length status stage. The table illustrates the possible power of two data payloads less than or equal to the allowable maximum data payload size. The table does not include the overhead associated with bit stuffing.

Table 5-3. High-speed Control Transfer Limits

	Protocol Overhead (173 bytes)		(Based on 480Mb/s and 8 bit interpacket gap, 88 bit min bus turnaround, 32 bit sync, 8 bit EOP: (9x4 SYNC bytes, 9 PID bytes, 6 EP/ADDR+CRC,6 CRC16, 8 Setup data, 9x(1+11) byte interpacket delay (EOP, etc.))			
	Data Payload	Max Bandwidth (bytes/second)	Microframe Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/ Microframe Useful Data
	1	344000	2%	43	18	43
	2	672000	2%	42	150	84
	4	1344000	2%	42	66	168
	8	2624000	2%	41	79	328
	16	4992000	3%	39	129	624
	32	9216000	3%	36	120	1152
	64	15872000	3%	31	153	1984
Max		60000000				7500

5.5.5 Control Transfer Data Sequences

Control transfers require that a Setup bus transaction be sent from the host to a device to describe the type of control access that the device should perform. The Setup transaction is followed by zero or more control Data transactions that carry the specific information for the requested access. Finally, a Status transaction completes the control transfer and allows the endpoint to return the status of the control transfer to the client software. After the Status transaction for a control transfer is completed, the host can advance to the next control transfer for the endpoint. As described in Section 5.5.4, each control transaction and the next

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

control transfer will be moved over the bus at some Host Controller implementation-defined time.

The endpoint can be busy for a device-specific time during the Data and Status transactions of the control transfer. During these times when the endpoint indicates it is busy (refer to Chapter 8 and Chapter 9 for details), the host will retry the transaction at a later time.

If a Setup transaction is received by an endpoint before a previously initiated control transfer is completed, the device must abort the current transfer/operation and handle the new control Setup transaction. A Setup transaction should not normally be sent before the completion of a previous control transfer. However, if a transfer is aborted, for example, due to errors on the bus, the host can send the next Setup transaction prematurely from the endpoint's perspective.

After a halt condition is encountered or an error is detected by the host, a control endpoint is allowed to recover by accepting the next Setup PID; i.e., recovery actions via some other pipe are not required for control endpoints. For the Default Control Pipe, a device reset will ultimately be required to clear the halt or error condition if the next Setup PID is not accepted.

The USB provides robust error detection and recovery/retransmission for errors that occur during control transfers. Transmitters and receivers can remain synchronized with regard to where they are in a control transfer and recover with minimum effort. Retransmission of Data and Status packets can be detected by a receiver via data retry indicators in the packet. A transmitter can reliably determine that its corresponding receiver has successfully accepted a transmitted packet by information returned in a handshake to the packet. The protocol allows for distinguishing a retransmitted packet from its original packet except for a control Setup packet. Setup packets may be retransmitted due to a transmission error; however, Setup packets cannot indicate that a packet is an original or a retried transmission.

5.6 Isochronous Transfers

In non-USB environments, isochronous transfers have the general implication of constant-rate, error-tolerant transfers. In the USB environment, requesting an isochronous transfer type provides the requester with the following:

- Guaranteed access to USB bandwidth with bounded latency
- Guaranteed constant data rate through the pipe as long as data is provided to the pipe
- In the case of a delivery failure due to error, no retrying of the attempt to deliver the data

While the USB isochronous transfer type is designed to support isochronous sources and destinations, it is not required that software using this transfer type actually be isochronous in order to use the transfer type. Section 5.12 presents more detail on special considerations for handling isochronous data on the USB.

5.6.1 Isochronous Transfer Data Format

The USB imposes no data content structure on communication flows for isochronous pipes.

5.6.2 Isochronous Transfer Direction

An isochronous pipe is a stream pipe and is, therefore, always uni-directional. An endpoint description identifies whether a given isochronous pipe's communication flow is into or out of the host. If a device requires bi-directional isochronous communication flow, two isochronous pipes must be used, one in each direction.

5.6.3 Isochronous Transfer Packet Size Constraints

An endpoint in a given configuration for an isochronous pipe specifies the maximum size data payload that it can transmit or receive. The USB System Software uses this information during configuration to ensure that there is sufficient bus time to accommodate this maximum

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

data payload in each (micro)frame. If there is sufficient bus time for the maximum data payload, the configuration is established; if not, the configuration is not established.

The USB limits the maximum data payload size to 1,023 bytes for each full-speed isochronous endpoint. High-speed endpoints are allowed up to 1024-byte data payloads. A high speed, high bandwidth endpoint specifies whether it requires two or three transactions per microframe. Table 5-4 lists information about different-sized full-speed isochronous transactions and the maximum number of transactions possible in a frame. The table is shaded to indicate that a full-speed isochronous endpoint (with a non-zero *wMaxpacket* size) must not be part of a default interface setting. The table does not include the overhead associated with bit stuffing.

Table 5-4. Full-speed Isochronous Transaction Limits

	Protocol Overhead (9 bytes)		(2 SYNC bytes, 2 PID bytes, 2 Endpoint + CRC bytes, 2 CRC bytes, and a 1-byte interpacket delay)			
	Data Payload	Max Bandwidth(bytes/second)	Frame Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/Frame Useful Data
	1	150000	1%	150	0	150
	2	272000	1%	136	4	272
	4	460000	1%	115	5	460
	8	704000	1%	88	4	704
	16	960000	2%	60	0	960
	32	1152000	3%	36	24	1152
	64	1280000	5%	20	40	1280
	128	1280000	9%	10	130	1280
	256	1280000	18%	5	175	1280
	512	1024000	35%	2	458	1024
	1023	1023000	69%	1	468	1023
Max		1500000				1500

Table 5-5 lists information about different-sized high-speed isochronous transactions and the maximum number of transactions possible in a microframe. The table is shaded to indicate that a high-speed isochronous endpoint must not be part of a default interface setting. The table does not include the overhead associated with bit stuffing.

Any given transaction for an isochronous pipe need not be exactly the maximum size specified for the endpoint. The size of a data payload is determined by the transmitter (client software or function) and can vary as required from transaction to transaction. The USB ensures that whatever size is presented to the Host Controller is delivered on the bus. The actual size of a data payload is determined by the data transmitter and may be less than the prenegotiated maximum size. Bus errors can change the actual packet size seen by the receiver. However, these errors can be detected by either CRC on the data or by knowledge the receiver has about the expected size for any transaction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 5-5. High-speed Isochronous Transaction Limits

	Protocol Overhead		(Based on 480Mb/s and 8 bit interpacket gap, 88 bit min bus turnaround, 32 bit sync, 8 bit EOP: (2x4 SYNC bytes, 2 PID bytes, 2 EP/ADDR+addr+CRC5, 2 CRC16, and a 2x(1+11)) byte interpacket delay (EOP, etc.))			
	Data Payload	Max Bandwidth (bytes/second)	Microframe Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/ MicroFrame Useful Data
	1	1536000	1%	192	12	192
	2	2992000	1%	187	20	374
	4	5696000	1%	178	24	712
	8	10432000	1%	163	2	1304
	16	17664000	1%	138	48	2208
	32	27392000	1%	107	10	3424
	64	37376000	1%	73	54	4672
	128	46080000	2%	45	30	5760
	256	51200000	4%	25	150	6400
	512	53248000	7%	13	350	6656
	1024	57344000	14%	7	66	7168
	2048	49152000	28%	3	1242	6144
	3072	49152000	41%	2	1280	6144
Max		60000000				7500

All device default interface settings must not include any isochronous endpoints with non-zero data payload sizes (specified via *wMaxPacketSize* in the endpoint descriptor). Alternate interface settings may specify non-zero data payload sizes for isochronous endpoints. If the isochronous endpoints have a large data payload size, it is recommended that additional alternate configurations or interface settings be used to specify a range of data payload sizes. This increases the chance that the device can be used successfully in combination with other USB devices.

5.6.4 Isochronous Transfer Bus Access Constraints

Isochronous transfers can only be used by full-speed and high-speed devices.

The USB requires that no more than 90% of any frame be allocated for periodic (isochronous and interrupt) transfers for full-speed endpoints. High-speed endpoints can allocate at most 80% of a microframe for periodic transfers.

An isochronous endpoint must specify its required bus access period. Full-/high-speed endpoints must specify a desired period as ($2^{bInterval-1}$) x F, where bInterval is in the range one to (and including) 16 and F is 125 μs for high-speed and 1ms for full-speed. This allows full-/high-speed isochronous transfers to have rates slower than one transaction per (micro)frame. However, an isochronous endpoint must be prepared to handle poll rates faster than the one specified. A host must not issue more than 1 transaction in a (micro)frame for an isochronous endpoint unless the endpoint is high-speed, high-bandwidth (see below). An isochronous IN endpoint must return a zero-length packet whenever data is requested at a faster interval than the specified interval and data is not available.

A high-speed endpoint can move up to 3072 bytes per microframe (or 192 Mb/s). A high-speed isochronous endpoint that requires more than 1024 bytes per period is called a high-bandwidth endpoint. A high-bandwidth endpoint uses multiple transactions per microframe. A

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

high-bandwidth endpoint must specify a period of 1x125 μs (i.e., a *bInterval* value of 1). See Section 5.9 for more information about the details of multiple transactions per microframe for high-bandwidth high-speed endpoints.

Errors on the bus or delays in operating system scheduling of client software can result in no packet being transferred for a (micro)frame. An error indication should be returned as status to the client software in such a case. A device can also detect this situation by tracking SOF tokens and noticing a disturbance in the specified bus access period pattern.

The bus frequency and (micro)frame timing limit the maximum number of successful isochronous transactions within a (micro)frame for any USB system to less than 151 full-speed one-byte data payloads and less than 193 high-speed one-byte data payloads. A Host Controller, for various implementation reasons, may not be able to provide the theoretical maximum number of isochronous transactions per (micro)frame.

5.6.5 Isochronous Transfer Data Sequences

Isochronous transfers do not support data retransmission in response to errors on the bus. A receiver can determine that a transmission error occurred. The low-level USB protocol does not allow handshakes to be returned to the transmitter of an isochronous pipe. Normally, handshakes would be returned to tell the transmitter whether a packet was successfully received or not. For isochronous transfers, timeliness is more important than correctness/retransmission, and, given the low error rates expected on the bus, the protocol is optimized by assuming transfers normally succeed. Isochronous receivers can determine whether they missed data during a (micro)frame. Also, a receiver can determine how much data was lost. Section 5.12 describes these USB mechanisms in more detail.

An endpoint for isochronous transfers never halts because there is no handshake to report a halt condition. Errors are reported as status associated with the IRP for an isochronous transfer, but the isochronous pipe is not halted in an error case. If an error is detected, the host continues to process the data associated with the next (micro)frame of the transfer. Only limited error detection is possible because the protocol for isochronous transactions does not allow per-transaction handshakes.

5.7 Interrupt Transfers

The interrupt transfer type is designed to support those devices that need to send or receive data infrequently but with bounded service periods. Requesting a pipe with an interrupt transfer type provides the requester with the following:

- Guaranteed maximum service period for the pipe
- Retry of transfer attempts at the next period, in the case of occasional delivery failure due to error on the bus

5.7.1 Interrupt Transfer Data Format

The USB imposes no data content structure on communication flows for interrupt pipes.

5.7.2 Interrupt Transfer Direction

An interrupt pipe is a stream pipe and is therefore always uni-directional. An endpoint description identifies whether a given interrupt pipe's communication flow is into or out of the host.

5.7.3 Interrupt Transfer Packet Size Constraints

An endpoint for an interrupt pipe specifies the maximum size data payload that it will transmit or receive. The maximum allowable interrupt data payload size is 64 bytes or less for full-speed. High-speed endpoints are allowed maximum data payload sizes up to 1024 bytes. A high speed, high bandwidth endpoint specifies whether it requires two or three transactions per microframe. Low-speed devices are limited to eight bytes or less maximum data payload size. This maximum applies to the data payloads of the data packets; i.e., the size specified

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

is for the data field of the packet as defined in Chapter 8, not including other protocol-required information. The USB does not require that data packets be exactly the maximum size; i.e., if a data packet is less than the maximum, it does not need to be padded to the maximum size.

All Host Controllers are required to support maximum data payload sizes from 0 to 64 bytes for full-speed interrupt endpoints, from 0 to 8 bytes for low-speed interrupt endpoints, and from 0 to 1024 bytes for high-speed interrupt endpoints. See Section 5.9 for more information about the details of multiple transactions per microframe for high bandwidth high-speed endpoints. No Host Controller is required to support larger maximum data payload sizes.

The USB System Software determines the maximum data payload size that will be used for an interrupt pipe during device configuration. This size remains constant for the lifetime of a device's configuration. The USB System Software uses the maximum data payload size determined during configuration to ensure that there is sufficient bus time to accommodate this maximum data payload in its assigned period. If there is sufficient bus time, the pipe is established; if not, the pipe is not established. However, the actual size of a data payload is still determined by the data transmitter and may be less than the maximum size.

An endpoint must always transmit data payloads with a data field less than or equal to the endpoint's *w*MaxPacketSize value. A device can move data via an interrupt pipe that is larger than *wMaxPacketSize*. A software client can accept this data via an IRP for the interrupt transfer that requires multiple bus transactions without requiring an IRP-complete notification per transaction. This can be achieved by specifying a buffer that can hold the desired data size. The size of the buffer is a multiple of *w*MaxPacketSize with some remainder. The endpoint must transfer each transaction except the last as *w*MaxPacketSize and the last transaction is the remainder. The multiple data transactions are moved over the bus at the period established for the pipe.

When an interrupt transfer involves more data than can fit in one data payload of the currently established maximum size, all data payloads are required to be maximum-sized except for the last data payload, which will contain the remaining data. An interrupt transfer is complete when the endpoint does one of the following:

- Has transferred exactly the amount of data expected
- Transfers a packet with a payload size less than *wMaxPacketSize* or transfers a zero-length packet

When an interrupt transfer is complete, the Host Controller retires the current IRP and advances to the next IRP. If a data payload is received that is larger than expected, the interrupt IRP will be aborted/retired and the pipe will stall future IRPs until the condition is corrected and acknowledged.

All high-speed device default interface settings must not include any interrupt endpoints with a data payload size (specified via *wMaxPacketSize* in the endpoint descriptor) greater than 64 bytes. Alternate interface settings may specify larger data payload sizes for interrupt endpoints. If the interrupt endpoints have a large data payload size, it is recommended that additional configurations or alternate interface settings be used to specify a range of data payload sizes. This increases the chances that the device can be used successfully in combination with other USB devices.

5.7.4 Interrupt Transfer Bus Access Constraints

Interrupt transfers can be used by low-speed, full-speed, and high-speed devices. High-speed endpoints can be allocated at most 80% of a microframe for periodic transfers. The USB requires that no more than 90% of any frame be allocated for periodic (isochronous and interrupt) full-/low-speed transfers.

The bus frequency and (micro)frame timing limit the maximum number of successful interrupt transactions within a (micro)frame for any USB system to less than 108 full-speed one-byte data payloads, or less than 10 low-speed one-byte data payloads, or to less than 134 high-

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

speed one-byte data payloads. A Host Controller, for various implementation reasons, may not be able to provide the above maximum number of interrupt transactions per (micro)frame.

Table 5-6 lists information about different low-speed interrupt transactions and the maximum number of transactions possible in a frame. Table 5-7 lists similar information for full-speed interrupt transactions. Table 5-8 lists similar information for high-speed interrupt transactions. The shaded portion of Table 5-8 indicates the data payload sizes of a high-speed interrupt endpoint that must not be part of a default interface setting. The tables do not include the overhead associated with bit stuffing.

Table 5-6. Low-speed Interrupt Transaction Limits

	Protocol Overhead (19 bytes)		(5 SYNC bytes, 5 PID bytes, 2 Endpoint + CRC bytes, 2 CRC bytes, and a 5-byte interpacket delay)			
	Data Payload	**Max Bandwidth (bytes/second)**	**Frame Bandwidth per Transfer**	**Max Transfers**	**Bytes Remaining**	**Bytes/Frame Useful Data**
	1	9000	11%	9	7	9
	2	16000	11%	8	19	16
	4	32000	12%	8	3	32
	8	48000	14%	6	25	48
Max		187500				187

Table 5-7. Full-speed Interrupt Transaction Limits

	Protocol Overhead (13 bytes)		(3 SYNC bytes, 3 PID bytes, 2 Endpoint + CRC bytes, 2 CRC bytes, and a 3-byte interpacket delay)			
	Data Payload	**Max Bandwidth (bytes/second)**	**Frame Bandwidth per Transfer**	**Max Transfers**	**Bytes Remaining**	**Bytes/Frame Useful Data**
	1	107000	1%	107	2	107
	2	200000	1%	100	0	200
	4	352000	1%	88	4	352
	8	568000	1%	71	9	568
	16	816000	2%	51	21	816
	32	1056000	3%	33	15	1056
	64	1216000	5%	19	37	1216
Max		1500000				1500

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 5-8. High-speed Interrupt Transaction Limits

	Protocol Overhead	(Based on 480Mb/s and 8 bit interpacket gap, 88 bit min bus turnaround, 32 bit sync, 8 bit EOP: (3x4 SYNC bytes, 3 PID bytes, 2 EP/ADDR+CRC bytes, 2 CRC16 and a 3x(1+11) byte interpacket delay(EOP, etc.))				
	Data Payload	**Max Bandwidth (bytes/second)**	**Microframe Bandwidth per Transfer**	**Max Transfers**	**Bytes Remaining**	**Bytes/ Microframe Useful Data**
	1	1064000	1%	133	52	133
	2	2096000	1%	131	33	262
	4	4064000	1%	127	7	508
	8	7616000	1%	119	3	952
	16	13440000	1%	105	45	1680
	32	22016000	1%	86	18	2752
	64	32256000	2%	63	3	4032
	128	40960000	2%	40	180	5120
	256	49152000	4%	24	36	6144
	512	53248000	8%	13	129	6656
	1024	49152000	14%	6	1026	6144
	2048	49152000	28%	3	1191	6144
	3072	49152000	42%	2	1246	6144
Max		60000000				7500

An endpoint for an interrupt pipe specifies its desired bus access period. A full-speed endpoint can specify a desired period from 1 ms to 255 ms. Low-speed endpoints are limited to specifying only 10 ms to 255 ms. High-speed endpoints can specify a desired period $(2^{bInterval-1})$x125 µs, where *bInterval* is in the range 1 to (including) 16. The USB System Software will use this information during configuration to determine a period that can be sustained. The period provided by the system may be shorter than that desired by the device up to the shortest period defined by the USB (125 µs microframe or 1 ms frame). The client software and device can depend only on the fact that the host will ensure that the time duration between two transaction attempts with the endpoint will be no longer than the desired period. Note that errors on the bus can prevent an interrupt transaction from being successfully delivered over the bus and consequently exceed the desired period. Also, the endpoint is only polled when the software client has an IRP for an interrupt transfer pending. If the bus time for performing an interrupt transfer arrives and there is no IRP pending, the endpoint will not be given an opportunity to transfer data at that time. Once an IRP is available, its data will be transferred at the next allocated period.

A high-speed endpoint can move up to 3072 bytes per microframe (or 192 Mb/s). A high-speed interrupt endpoint that requires more than 1024 bytes per period is called a high-bandwidth endpoint. A high-bandwidth endpoint uses multiple transactions per microframe. A high-bandwidth endpoint must specify a period of 1x125 µs (i.e., a *bInterval* value of 1). See Section 5.9 for more information about the details of multiple transactions per microframe for high-bandwidth high-speed endpoints.

Interrupt transfers are moved over the USB by accessing an interrupt endpoint every specified period. For input interrupt endpoints, the host has no way to determine whether an endpoint will source an interrupt without accessing the endpoint and requesting an interrupt transfer. If the endpoint has no interrupt data to transmit when accessed by the host, it responds with NAK. An endpoint should only provide interrupt data when it has an interrupt pending to avoid

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

having a software client erroneously notified of IRP complete. A zero-length data payload is a valid transfer and may be useful for some implementations.

5.7.5 Interrupt Transfer Data Sequences

Interrupt transactions may use either alternating data toggle bits, such that the bits are toggled only upon successful transfer completion or a continuously toggling of data toggle bits. The host in any case must assume that the device is obeying full handshake/retry rules as defined in Chapter 8. A device may choose to always toggle DATA0/DATA1 PIDs so that it can ignore handshakes from the host. However, in this case, the client software can miss some data packets when an error occurs, because the Host Controller interprets the next packet as a retry of a missed packet.

If a halt condition is detected on an interrupt pipe due to transmission errors or a STALL handshake being returned from the endpoint, all pending IRPs are retired. Removal of the halt condition is achieved via software intervention through a separate control pipe. This recovery will reset the data toggle bit to DATA0 for the endpoint on both the host and the device. Interrupt transactions are retried due to errors detected on the bus that affect a given transfer.

5.8 Bulk Transfers

The bulk transfer type is designed to support devices that need to communicate relatively large amounts of data at highly variable times where the transfer can use any available bandwidth. Requesting a pipe with a bulk transfer type provides the requester with the following:

- Access to the USB on a bandwidth-available basis
- Retry of transfers, in the case of occasional delivery failure due to errors on the bus
- Guaranteed delivery of data but no guarantee of bandwidth or latency

Bulk transfers occur only on a bandwidth-available basis. For a USB with large amounts of free bandwidth, bulk transfers may happen relatively quickly; for a USB with little bandwidth available, bulk transfers may trickle out over a relatively long period of time.

5.8.1 Bulk Transfer Data Format

The USB imposes no data content structure on communication flows for bulk pipes.

5.8.2 Bulk Transfer Direction

A bulk pipe is a stream pipe and, therefore, always has communication flowing either into or out of the host for a given pipe. If a device requires bi-directional bulk communication flow, two bulk pipes must be used, one in each direction.

5.8.3 Bulk Transfer Packet Size Constraints

An endpoint for bulk transfers specifies the maximum data payload size that the endpoint can accept from or transmit to the bus. The USB defines the allowable maximum bulk data payload sizes to be only 8, 16, 32, or 64 bytes for full-speed endpoints and 512 bytes for high-speed endpoints. A low-speed device must not have bulk endpoints. This maximum applies to the data payloads of the data packets; i.e., the size specified is for the data field of the packet as defined in Chapter 8, not including other protocol-required information.

A bulk endpoint is designed to support a maximum data payload size. A bulk endpoint reports in its configuration information the value for its maximum data payload size. The USB does not require that data payloads transmitted be exactly the maximum size; i.e., if a data payload is less than the maximum, it does not need to be padded to the maximum size.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

All Host Controllers are required to have support for 8-, 16-, 32-, and 64-byte maximum packet sizes for full-speed bulk endpoints and 512 bytes for high-speed bulk endpoints. No Host Controller is required to support larger or smaller maximum packet sizes.

During configuration, the USB System Software reads the endpoint's maximum data payload size and ensures that no data payload will be sent to the endpoint that is larger than the supported size.

An endpoint must always transmit data payloads with a data field less than or equal to the endpoint's reported *w*MaxPacketSize value. When a bulk IRP involves more data than can fit in one maximum-sized data payload, all data payloads are required to be maximum size except for the last data payload, which will contain the remaining data. A bulk transfer is complete when the endpoint does one of the following:

- Has transferred exactly the amount of data expected
- Transfers a packet with a payload size less than *wMaxPacketSize* or transfers a zero-length packet

When a bulk transfer is complete, the Host Controller retires the current IRP and advances to the next IRP. If a data payload is received that is larger than expected, all pending bulk IRPs for that endpoint will be aborted/retired.

5.8.4 Bulk Transfer Bus Access Constraints

Only full-speed and high-speed devices can use bulk transfers.

An endpoint has no way to indicate a desired bus access frequency for a bulk pipe. The USB balances the bus access requirements of all bulk pipes and the specific IRPs that are pending to provide "good effort" delivery of data between client software and functions. Moving control transfers over the bus has priority over moving bulk transfers.

There is no time guaranteed to be available for bulk transfers as there is for control transfers. Bulk transfers are moved over the bus only on a bandwidth-available basis. If there is bus time that is not being used for other purposes, bulk transfers will be moved over the bus. If there are bulk transfers pending for multiple endpoints, bulk transfers for the different endpoints are selected according to a fair access policy that is Host Controller implementation-dependent.

All bulk transfers pending in a system contend for the same available bus time. Because of this, the USB System Software at its discretion can vary the bus time made available for bulk transfers to a particular endpoint. An endpoint and its client software cannot assume a specific rate of service for bulk transfers. Bus time made available to a software client and its endpoint can be changed as other devices are inserted into and removed from the system or also as bulk transfers are requested for other device endpoints. Client software cannot assume ordering between bulk and control transfers; i.e., in some situations, bulk transfers can be delivered ahead of control transfers.

High-speed bulk OUT endpoints must support the PING flow control protocol. The details of this protocol are described in Section 8.5.1.

The bus frequency and (micro)frame timing limit the maximum number of successful bulk transactions within a (micro)frame for any USB system to less than 72 full-speed eight-byte data payloads or less than 14 high-speed 512-byte data payloads. Table 5-9 lists information about different-sized full-speed bulk transactions and the maximum number of transactions possible in a frame. The table does not include the overhead associated with bit stuffing. Table 5-10 lists similar information for high-speed bulk transactions.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 5-9. Full-speed Bulk Transaction Limits

	Protocol Overhead (13 bytes)		(3 SYNC bytes, 3 PID bytes, 2 Endpoint + CRC bytes, 2 CRC bytes, and a 3-byte interpacket delay)			
	Data Payload	Max Bandwidth (bytes/second)	Frame Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/Frame Useful Data
	1	107000	1%	107	2	107
	2	200000	1%	100	0	200
	4	352000	1%	88	4	352
	8	568000	1%	71	9	568
	16	816000	2%	51	21	816
	32	1056000	3%	33	15	1056
	64	1216000	5%	19	37	1216
Max		1500000				1500

Table 5-10. High-speed Bulk Transaction Limits

	Protocol Overhead (55 bytes)		(3x4 SYNC bytes, 3 PID bytes, 2 EP/ADDR+CRC bytes, 2 CRC16, and a 3x(1+11) byte interpacket delay (EOP, etc.))			
	Data Payload	Max Bandwidth (bytes/second)	Microframe Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/ Microframe Useful Data
	1	1064000	1%	133	52	133
	2	2096000	1%	131	33	262
	4	4064000	1%	127	7	508
	8	7616000	1%	119	3	952
	16	13440000	1%	105	45	1680
	32	22016000	1%	86	18	2752
	64	32256000	2%	63	3	4032
	128	40960000	2%	40	180	5120
	256	49152000	4%	24	36	6144
	512	53248000	8%	13	129	6656
Max		60000000				7500

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Host Controllers are free to determine how the individual bus transactions for specific bulk transfers are moved over the bus within and across (micro)frames. An endpoint could see all bus transactions for a bulk transfer within the same (micro)frame or spread across several (micro)frames. A Host Controller, for various implementation reasons, may not be able to provide the above maximum number of transactions per (micro)frame.

5.8.5 Bulk Transfer Data Sequences

Bulk transactions use data toggle bits that are toggled only upon successful transaction completion to preserve synchronization between transmitter and receiver when transactions are retried due to errors. Bulk transactions are initialized to DATA0 when the endpoint is configured by an appropriate control transfer. The host will also start the first bulk transaction with DATA0. If a halt condition is detected on a bulk pipe due to transmission errors or a STALL handshake being returned from the endpoint, all pending IRPs are retired. Removal of

the halt condition is achieved via software intervention through a separate control pipe. This recovery will reset the data toggle bit to DATA0 for the endpoint on both the host and the device.

Bulk transactions are retried due to errors detected on the bus that affect a given transaction.

5.9 High-Speed, High Bandwidth Endpoints

USB supports individual high-speed interrupt or isochronous endpoints that require data rates up to 192 Mb/s (i.e., 3072 data bytes per microframe). One, two, or three high-speed transactions are allowed in a single microframe to support high-bandwidth endpoints.

A high-speed interrupt or isochronous endpoint indicates that it requires more than 1024 bytes per microframe when bits 12..11 of the *wMaxPacketSize* field of the endpoint descriptor (see Table 5-11) are non-zero. The lower 11 bits of *wMaxPacketSize* indicate the size of the data payload for each individual transaction while bits 12..11 indicate the maximum number of required transactions possible. See Section 9.6.6 for restrictions on the allowed combinations of values for bits 12..11 and bits 10..0.

Table 5-11. *wMaxPacketSize* Field of Endpoint Descriptor

Bits	15..13	12..11	10..0
Field	Reserved, must be set to zero	Number of transactions per microframe	Maximum size of data payload in bytes

Note: This representation means that endpoints requesting two transactions per microframe will specify a total data payload size in the microframe that is a multiple of two bytes. Also endpoints requesting three transactions per microframe will specify a total data payload size that is a multiple of three bytes. In any case, any number of bytes can actually be transferred in a microframe.

The host controller must issue an appropriate number of high-speed transactions per microframe. Errors in the host or on the bus can result in the host controller issuing fewer transactions than requested for the endpoint. The first transaction(s) must have a data payload(s) as specified by the lower 11 bits of *wMaxPacketSize* if enough data is available, while the last transaction has any remaining data less than or equal to the maximum size specified. The host controller may issue transactions for the same endpoint one immediately after the other (as required for the actual data provided) or may issue transactions for other endpoints in between the transactions for a high bandwidth endpoint.

5.9.1 High Bandwidth Interrupt Endpoints

For interrupt transactions, if the endpoint NAKs a transaction during a microframe, the host controller must not issue further transactions for that endpoint until the next period.

If the endpoint times-out a transaction, the host controller must retry the transaction. The endpoint specifies the maximum number of desired transactions per microframe. If the maximum number of transactions per microframe has not been reached, the host controller may immediately retry the transaction during the current microframe. Host controllers are recommended to do an immediate retry since this minimizes impact on devices that are bandwidth sensitive. If the maximum number of transactions per microframe has been reached, the host controller must retry the transaction at the next period for the endpoint.

A host controller is allowed to issue less than the maximum number of transactions to an endpoint per microframe only if more than a single memory buffer is required for the transactions within the microframe.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Normal DATA0/DATA1 data toggle sequencing is used for each interrupt transaction during a microframe.

5.9.2 High Bandwidth Isochronous Endpoints

For isochronous transactions, if an IN endpoint provides less than a maximum data payload as specified by its endpoint descriptor, the host must not issue further transactions for that endpoint for that microframe.

For an isochronous OUT endpoint, the host controller must issue the number of transactions as required for the actual data provided, not exceeding the maximum number specified by the endpoint descriptor. The transactions issued must adhere to the maximum payload sizes as specified in the endpoint descriptor.

No retries are ever done for isochronous endpoints.

High bandwidth isochronous endpoints (IN and OUT) must support data PID sequencing. Data PID sequencing provides the required support for the data receiver to detect one or more lost/damaged packets per microframe.

Data PID sequencing for a high-speed, high bandwidth isochronous IN endpoint uses a repeating sequence of DATA2, DATA1, DATA0 PIDs for the data packet of each transaction in a microframe. If there is only a single transaction in the microframe, only a DATA0 data packet PID is used. If there are two transactions per microframe, DATA1 is used for the first transaction data packet and DATA0 is used for the second transaction data packet. If there are three transactions per microframe, DATA2 is used for the first transaction data packet, DATA1 is used for the second, and DATA0 is used for the third. In all cases, the data PID sequence starts over again the next microframe. Figure 5-11 shows the order of data packet PIDs that are used in subsequent transactions within a microframe for high-bandwidth isochronous IN endpoints.

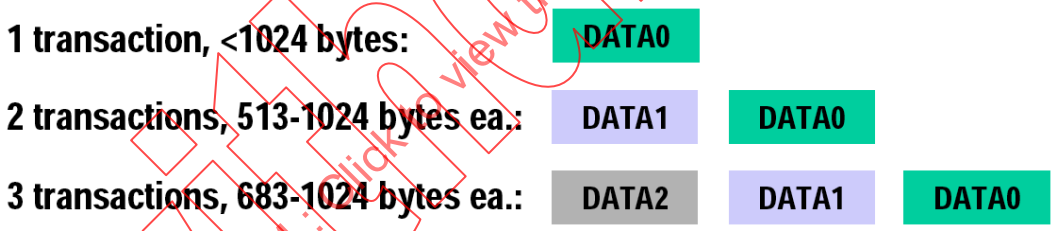


Figure 5-11. Data Phase PID Sequence for Isochronous IN High Bandwidth Endpoints

An endpoint must respond to an IN token for the first transaction with a DATA2 when it requires three transactions of data to be moved. It must respond with a DATA1 for the first transaction when it requires two transactions and with a DATA0 when it requires only a single transaction. After the first transaction, the endpoint follows the data PID sequence as described above.

The host knows the maximum number of allowed transactions per microframe for the IN endpoint. The host expects the response to the first transaction to encode (via the data packet PID) how many transactions are required by the endpoint for this microframe. If the host doesn't receive an error-free, appropriate response to any transaction, the host must not issue any further transactions to the endpoint for that microframe. When the host receives a DATA0 data packet from the endpoint, it must not issue any further transactions to the endpoint for that microframe.

Data PID sequencing for a high-speed, high bandwidth isochronous OUT endpoint uses a different sequence than that used for an IN endpoint. The host must issue a DATA0 data packet when there is a single transaction. The host must issue an MDATA for the first transaction and a DATA1 for the second transaction when there are two transactions per microframe. The host must issue two MDATA transactions and a DATA2 for the third

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

transaction when there are three transactions per microframe. These sequences allow the endpoint to detect if there was a lost/damaged transaction during a microframe. Figure 5-12 shows the order of data packet PIDs that are used in subsequent transactions within a microframe for high-bandwidth isochronous OUT.

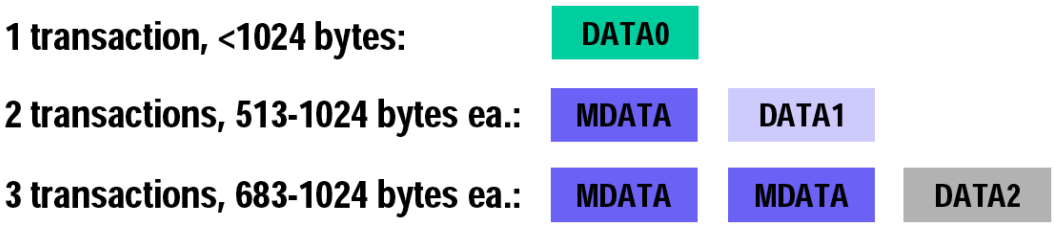


Figure 5-12. Data Phase PID Sequence for Isochronous OUT High Bandwidth Endpoints

If the wrong OUT transactions are detected by the endpoint, all of the data transferred during the microframe must be treated as if it had encountered an error. Note that for the three transactions per microframe case with a missing MDATA transaction, USB provides no way for the endpoint to determine which of the two MDATA transactions was lost. There may be application specific methods to more precisely determine which data was lost, but USB provides no method to do so at the bus level.

5.10 Split Transactions

Host controllers and hubs support one additional transaction type called split transactions. This transaction type allows full- and low-speed devices to be attached to hubs operating at high-speed. These transactions involve only host controllers and hubs and are not visible to devices. High-speed split transactions for interrupt and isochronous transfers must be allocated by the host from the 80% periodic portion of a microframe. More information on split transactions can be found in Chapter 8 and Chapter 11.

5.11 Bus Access for Transfers

Accomplishing any data transfer between the host and a USB device requires some use of the USB bandwidth. Supporting a wide variety of isochronous and asynchronous devices requires that each device's transfer requirements are accommodated. The process of assigning bus bandwidth to devices is called transfer management. There are several entities on the host that coordinate the information flowing over the USB: client software, the USB Driver (USBD), and the Host Controller Driver (HCD). Implementers of these entities need to know the key concepts related to bus access:

- Transfer Management: The entities and the objects that support communication flow over the USB
- Transaction Tracking: The USB mechanisms that are used to track transactions as they move through the USB system
- Bus Time: The time it takes to move a packet of information over the bus
- Device/Software Buffer Size: The space required to support a bus transaction
- Bus Bandwidth Reclamation: Conditions where bandwidth that was allocated to other transfers but was not used and can now be possibly reused by control and bulk transfers

The previous sections focused on how client software relates to a function and what the logical flows are over a pipe between the two entities. This section focuses on the different parts of the host and how they must interact to support moving data over the USB. This information may also be of interest to device implementers so they understand aspects of what the host is doing when a client requests a transfer and how that transfer is presented to the device.

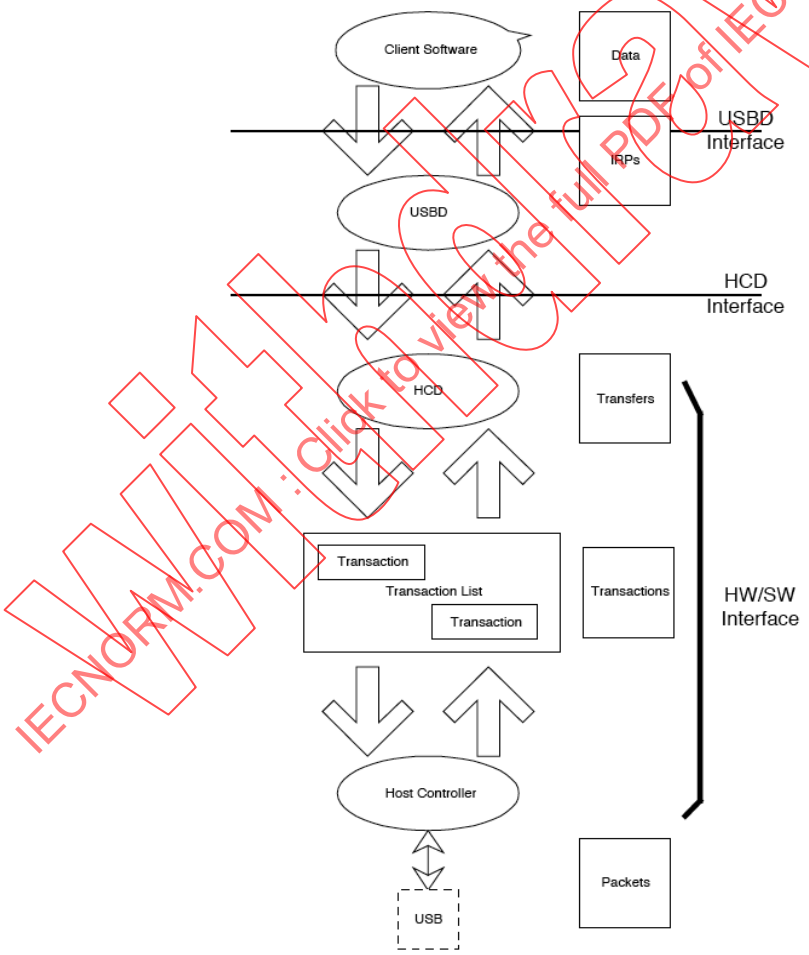
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.11.1 Transfer Management

Transfer management involves several entities that operate on different objects in order to move transactions over the bus:

- Client Software: Consumes/generates function-specific data to/from a function endpoint via calls and callbacks requesting IRPs with the USBD interface.
- USB Driver (USBD): Converts data in client IRPs to/from device endpoint via calls/callbacks with the appropriate HCD. A single client IRP may involve one or more transfers.
- Host Controller Driver (HCD): Converts IRPs to/from transactions (as required by a Host Controller implementation) and organizes them for manipulation by the Host Controller. Interactions between the HCD and its hardware is implementation-dependent and is outside the scope of the USB Specification.
- Host Controller: Takes transactions and generates bus activity via packets to move function-specific data across the bus for each transaction.

Figure 5-13 shows how the entities are organized as information flows between client software and the USB. The objects of primary interest to each entity are shown at the interfaces between entities.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 5-13. USB Information Conversion From Client Software to Bus

5.11.1.1 Client Software

Client software determines what transfers need to be made with a function. It uses appropriate operating system-specific interfaces to request IRPs. Client software is aware

only of the set of pipes (i.e., the interface) it needs to manipulate its function. The client is aware of and adheres to all bus access and bandwidth constraints as described previously for each transfer type. The requests made by the client software are presented via the USBD interface.

Some clients may manipulate USB functions via other device class interfaces defined by the operating system and may themselves not make direct USBD calls. However, there is always some lowest level client that makes USBD calls to pass IRPs to the USBD. All IRPs presented are required to adhere to the prenegotiated bandwidth constraints set when the pipe was established. If a function is moved from a non-USB environment to the USB, the driver that would have directly manipulated the function hardware via memory or I/O accesses is the lowest client software in the USB environment that now interacts with the USBD to manipulate the driver's USB function.

After client software has requested a transfer of its function and the request has been serviced, the client software receives notification of the completion status of the IRP. If the transfer involved function-to-host data transfer, the client software can access the data in the data buffer associated with the completed IRP.

The USBD interface is defined in Chapter 10.

5.11.1.2 USB Driver

The Universal Serial Bus Driver (USBD) is involved in mediating bus access at two general times:

- While a device is attached to the bus during configuration
- During normal transfers

When a device is attached and configured, the USBD is involved to ensure that the desired device configuration can be accommodated on the bus. The USBD receives configuration requests from the configuring software that describe the desired device configuration: endpoint(s), transfer type(s), transfer period(s), data size(s), etc. The USBD either accepts or rejects a configuration request based on bandwidth availability and the ability to accommodate that request type on the bus. If it accepts the request, the USBD creates a pipe for the requester of the desired type and with appropriate constraints as defined for the transfer type. Bandwidth allocation for periodic endpoints does not have to be made when the device is configured and, once made, a bandwidth allocation can be released without changing the device configuration.

The configuration aspects of the USBD are typically operating system-specific and heavily leverage the configuration features of the operating system to avoid defining additional (redundant) interfaces.

Once a device is configured, the software client can request IRPs to move data between it and its function endpoints.

5.11.1.3 Host Controller Driver

The Host Controller Driver (HCD) is responsible for tracking the IRPs in progress and ensuring that USB bandwidth and (micro)frame time maximums are never exceeded. When IRPs are made for a pipe, the HCD adds them to the transaction list. When an IRP is complete, the HCD notifies the requesting software client of the completion status for the IRP. If the IRP involved data transfer from the function to the software client, the data was placed in the client-indicated data buffer.

IRPs are defined in an operating system-dependent manner.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.11.1.4 Transaction List

The transaction list is a Host Controller implementation-dependent description of the current outstanding set of bus transactions that need to be run on the bus. Only the HCD and its Host Controller have access to the specific representation. Each description contains transaction descriptions in which parameters, such as data size in bytes, the device address and endpoint number, and the memory area to which data is to be sent or received, are identified.

A transaction list and the interface between the HCD and its Host Controller is typically represented in an implementation-dependent fashion and is not defined explicitly as part of the USB Specification.

5.11.1.5 Host Controller

The Host Controller has access to the transaction list and translates it into bus activity. In addition, the Host Controller provides a reporting mechanism whereby the status of a transaction (done, pending, halted, etc.) can be obtained. The Host Controller converts transactions into appropriate implementation-dependent activities that result in USB packets moving over the bus topology rooted in the root hub.

The Host Controller ensures that the bus access rules defined by the protocol are obeyed, such as inter-packet timings, timeouts, babble, etc. The HCD interface provides a way for the Host Controller to participate in deciding whether a new pipe is allowed access to the bus. This is done because Host Controller implementations can have restrictions/constraints on the minimum inter-transaction times they may support for combinations of bus transactions.

The interface between the transaction list and the Host Controller is hidden within an HCD and Host Controller implementation.

5.11.2 Transaction Tracking

A USB function sees data flowing across the bus in packets as described in Chapter 8. The Host Controller uses some implementation-dependent representation to track what packets to transfer to/from what endpoints at what time or in what order. Most client software does not want to deal with packetized communication flows because this involves a degree of complexity and interconnect dependency that limits the implementation. The USB System Software (USBD and HCD) provides support for matching data movement requirements of a client to packets on the bus. The Host Controller hardware and software uses IRPs to track information about one or more transactions that combine to deliver a transfer of information between the client software and the function. Figure 5-14 summarizes how transactions are organized into IRPs for the four transfer types. Detailed protocol information for each transfer type can be found in Chapter 8. More information about client software views of IRPs can be found in Chapter 10 and in the operating system specific-information for a particular operating system.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

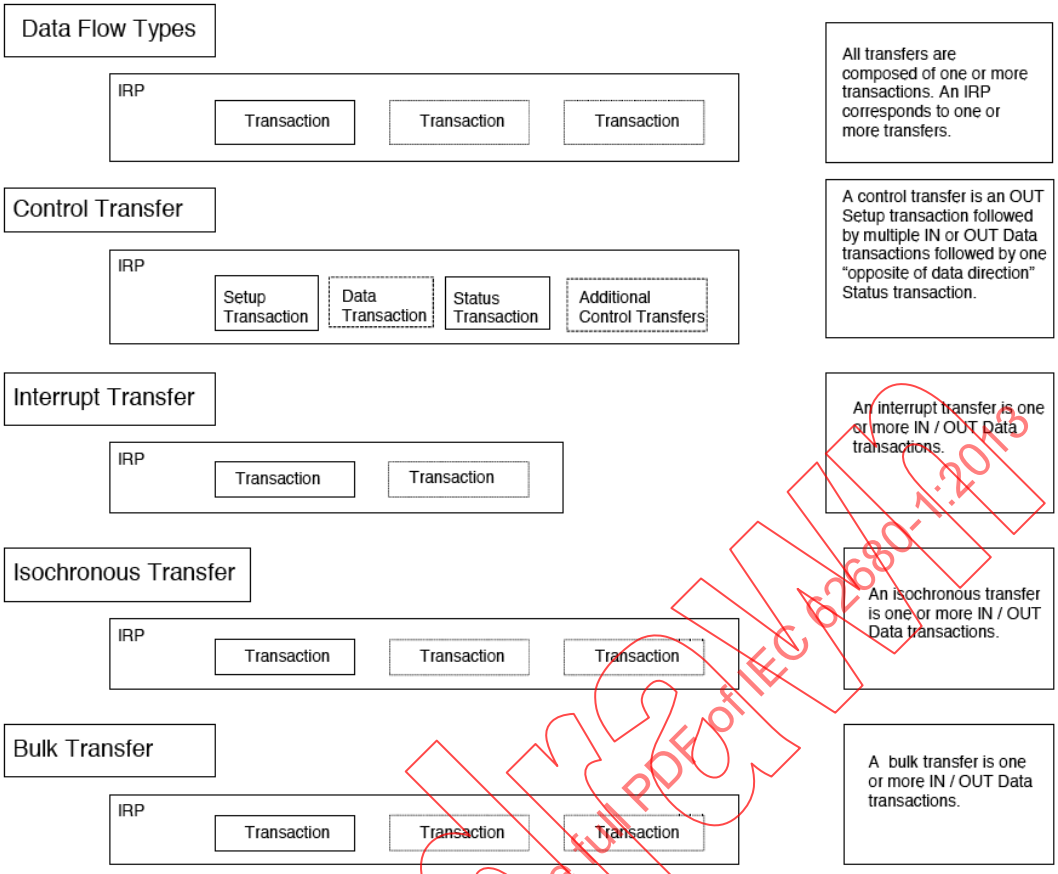


Figure 5-14. Transfers for Communication Flows

Even though IRPs track the bus transactions that need to occur to move a specific data flow over the USB, Host Controllers are free to choose how the particular bus transactions are moved over the bus subject to the USB-defined constraints (e.g., exactly one transaction per (micro)frame for isochronous transfers). In any case, an endpoint will see transactions in the order they appear within an IRP unless errors occur. For example, Figure 5-15 shows two IRPs, one each for two pipes where each IRP contains three transactions. For any transfer type, a Host Controller is free to move the first transaction of the first IRP followed by the first transaction of the second IRP somewhere in (micro)Frame 1, while moving the second transaction of each IRP in opposite order somewhere in (micro)Frame 2. If these are isochronous transfer types, that is the only degree of freedom a Host Controller has. If these are control or bulk transfers, a Host Controller could further move more or less transactions from either IRP within either (micro)frame. Functions cannot depend on seeing transactions within an IRP back-to-back within a (micro)frame nor should they depend on not seeing transactions back-to-back within a (micro)frame.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

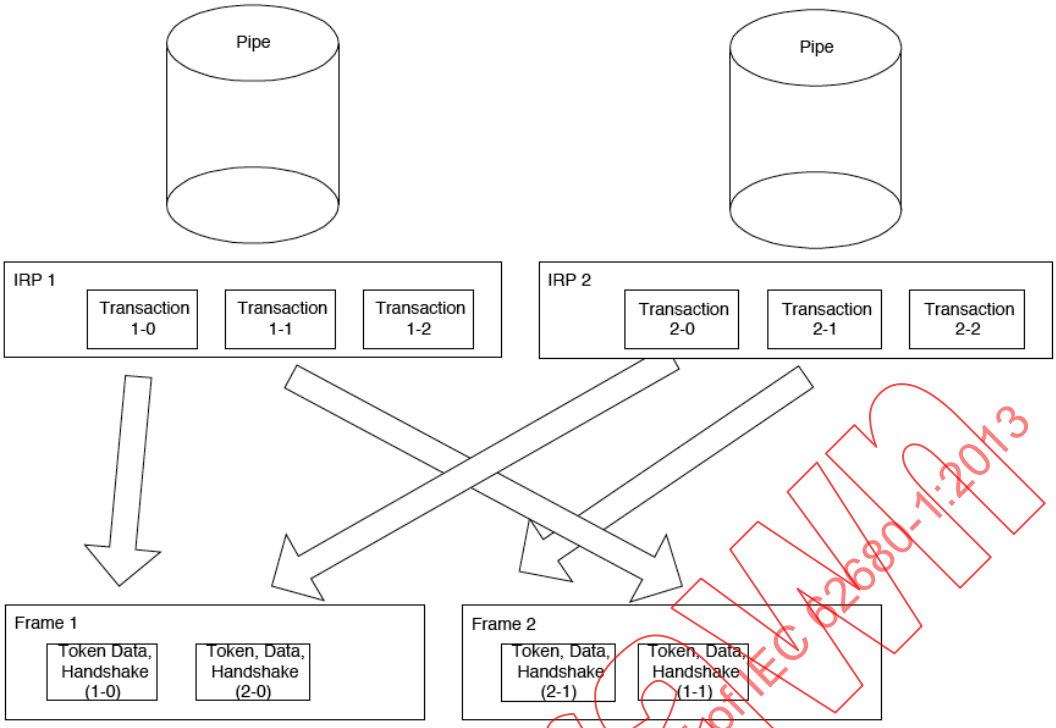


Figure 5-15. Arrangement of IRPs to Transactions/(Micro)frames

5.11.3 Calculating Bus Transaction Times

When the USB System Software allows a new pipe to be created for the bus, it must calculate how much bus time is required for a given transaction. That bus time is based on the maximum packet size information reported for an endpoint, the protocol overhead for the specific transaction type request, the overhead due to signaling imposed bit stuffing, inter-packet timings required by the protocol, inter-transaction timings, etc. These calculations are required to ensure that the time available in a (micro)frame is not exceeded. The equations used to determine transaction bus time are:

KEY:

Data_bc	The byte count of data payload
Host_Delay	The time required for the host or transaction translator to prepare for or recover from the transmission; Host Controller implementation-specific
Floor()	The integer portion of argument
Hub_LS_Setup	The time provided by the Host Controller for hubs to enable low-speed ports; measured as the delay from the end of the PRE PID to the start of the low-speed SYNC; minimum of four full-speed bit times
BitStuffTime	Function that calculates theoretical additional time required due to bit stuffing in signaling; worst case is (1.1667*8*Data_bc)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

```
High-speed (Input)

    Non-Isochronous Transfer (Handshake Included)
    = (55 * 8 * 2.083) + (2.083 * Floor(3.167  + BitStuffTime(Data_bc))) +
    Host_Delay

    Isochronous Transfer (No Handshake)
    = (38 * 8 * 2.083)  + (2.083 * Floor(3.167  + BitStuffTime(Data_bc))) +
    Host_Delay


High-speed (Output)

    Non-Isochronous Transfer (Handshake Included)
    = (55 * 8 * 2.083)  + (2.083 * Floor(3.167  + BitStuffTime(Data_bc))) +
    Host_Delay


    Isochronous Transfer (No Handshake)
    = (38 * 8 * 2.083)  + (2.083 * Floor(3.167  + BitStuffTime(Data_bc))) +
    Host_Delay


Full-speed (Input)

    Non-Isochronous Transfer (Handshake Included)
    = 9107 + (83.54 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay

    Isochronous Transfer (No Handshake)
    = 7268 + (83.54 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay


Full-speed (Output)

    Non-Isochronous Transfer (Handshake Included)
           = 9107 + (83.54 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay


    Isochronous Transfer (No Handshake)
           = 6265 + (83.54 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay


Low-speed (Input)

           = 64060 + (2 * Hub_LS_Setup) +
                  (676.67 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay


Low-speed (Output)

           = 64107 + (2 * Hub_LS_Setup) +
                  (667.0 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay
```

The bus times in the above equations are in nanoseconds and take into account propagation delays due to the distance the device is from the host. These are typical equations that can be used to calculate bus time; however, different implementations may choose to use coarser approximations of these times.

The actual bus time taken for a given transaction will almost always be less than that calculated because bit stuffing overhead is data-dependent. Worst case bit stuffing is calculated as 1.1667 (7/6) times the raw time (i.e., the BitStuffTime function multiplies the Data_bc by 8*1.1667 in the equations). This means that there will almost always be time unused on the bus (subject to data pattern specifics) after all regularly scheduled transactions have completed. The bus time made available due to less bit stuffing can be reused as discussed in Section 5.11.5.

The Host_Delay term in the equations is Host Controller-, Transaction Translator(TT)-, and system-dependent and allows for additional time a Host Controller (or TT) may require due to

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

delays in gaining access to memory or other implementation dependencies. This term is incorporated into an implementation of these equations by using the transfer management functions provided by the HCD interface. These equations are typically implemented by a combination of USBD and HCD software working in cooperation. The results of these calculations are used to determine whether a transfer or pipe creation can be supported in a given USB configuration.

5.11.4 Calculating Buffer Sizes in Functions and Software

Client software and functions both need to provide buffer space for pending data transactions awaiting their turn on the bus. For non-isochronous pipes, this buffer space needs to be just large enough to hold the next data packet. If more than one transaction request is pending for a given endpoint, the buffering for each transaction must be supplied. Methods to calculate the precise absolute minimum buffering a function may require because of specific interactions defined between its client software and the function are outside the scope of this specification.

The Host Controller is expected to be able to support an unlimited number of transactions pending for the bus subject to available system memory for buffer and descriptor space, etc. Host Controllers are allowed to limit how many (micro)frames into the future they allow a transaction to be requested.

For isochronous pipes, Section 5.12.4 describes details affecting host side and device side buffering requirements. In general, buffers need to be provided to hold approximately twice the amount of data that can be transferred in 1ms for full-speed endpoints or 125 μs for high-speed endpoints.

5.11.5 Bus Bandwidth Reclamation

The USB bandwidth and bus access are granted based on a calculation of worst-case bus transmission time and required latencies. However, due to the constraints placed on different transfer types and the fact that the bit stuffing bus time contribution is calculated as a constant but is data-dependent, there will frequently be bus time remaining in each (micro)frame time versus what the (micro)frame transmission time was calculated to be. In order to support the most efficient use of the bus bandwidth, control and bulk transfers are candidates to be moved over the bus as bus time becomes available. Exactly how a Host Controller supports this is implementation-dependent. A Host Controller can take into account the transfer types of pending IRPs and implementation-specific knowledge of remaining (micro)frame time to reuse reclaimed bandwidth.

5.12 Special Considerations for Isochronous Transfers

Support for isochronous data movement between the host and a device is one of the system capabilities supported by the USB. Delivering isochronous data reliably over the USB requires careful attention to detail. The responsibility for reliable delivery is shared by several USB entities:

- The device/function
- The bus
- The Host Controller
- One or more software agents

Because time is a key part of an isochronous transfer, it is important for USB designers to understand how time is dealt with within the USB by these different entities.

Note: The examples in this section describe USB for an example involving full-speed endpoints. The general example details are also appropriate for high-speed endpoints when corresponding changes are made; for example, frame replaced with microframe, 1 ms replaced with 125 μs, rate adjustments made between full-speed and high-speed, etc.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

All isochronous devices must report their capabilities in the form of device-specific descriptors. The capabilities should also be provided in a form that the potential customer can use to decide whether the device offers a solution to his problem(s). The specific capabilities of a device can justify price differences.

In any communication system, the transmitter and receiver must be synchronized enough to deliver data robustly. In an asynchronous communication system, data can be delivered robustly by allowing the transmitter to detect that the receiver has not received a data item correctly and simply retrying transmission of the data.

In an isochronous communication system, the transmitter and receiver must remain time- and data-synchronized to deliver data robustly. The USB does not support transmission retry of isochronous data so that minimal bandwidth can be allocated to isochronous transfers and time synchronization is not lost due to a retry delay. However, it is critical that a USB isochronous transmitter/receiver pair still remain synchronized both in normal data transmission cases and in cases where errors occur on the bus.

In many systems that deal with isochronous data, a single global clock is used to which all entities in the system synchronize. An example of such a system is the PSTN (Public Switched Telephone Network). Given that a broad variety of devices with different natural frequencies may be attached to the USB, no single clock can provide all the features required to satisfy the synchronization requirements of all devices and software while still supporting the cost targets of mass-market PC products. The USB defines a clock model that allows a broad range of devices to coexist on the bus and have reasonable cost implementations.

This section presents options or features that can be used by isochronous endpoints to minimize behavior differences between a non-USB implemented function and a USB version of the function. An example is included to illustrate the similarities and differences between the non-USB and USB versions of a function.

The remainder of the section presents the following key concepts:

- USB Clock Model: What clocks are present in a USB system that have impact on isochronous data transfers
- USB (micro)frame Clock-to-function Clock Synchronization Options: How the USB (micro)frame clock can relate to a function clock
- SOF Tracking: Responsibilities and opportunities of isochronous endpoints with respect to the SOF token and USB (micro)frames
- Data Prebuffering: Requirements for accumulating data before generation, transmission, and consumption
- Error Handling: Isochronous-specific details for error handling
- Buffering for Rate Matching: Equations that can be used to calculate buffer space required for isochronous endpoints

5.12.1 Example Non-USB Isochronous Application

The example used is a reasonably generalized example. Other simpler or more complex cases are possible and the relevant USB features identified can be used or not as appropriate.

The example consists of an 8 kHz mono microphone connected through a mixer driver that sends the input data stream to 44 kHz stereo speakers. The mixer expects the data to be received and transmitted at some sample rate and encoding. A rate matcher driver on input and output converts the sample rate and encoding from the natural rate and encoding of the device to the rate and encoding expected by the mixer. Figure 5-16 illustrates this example.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

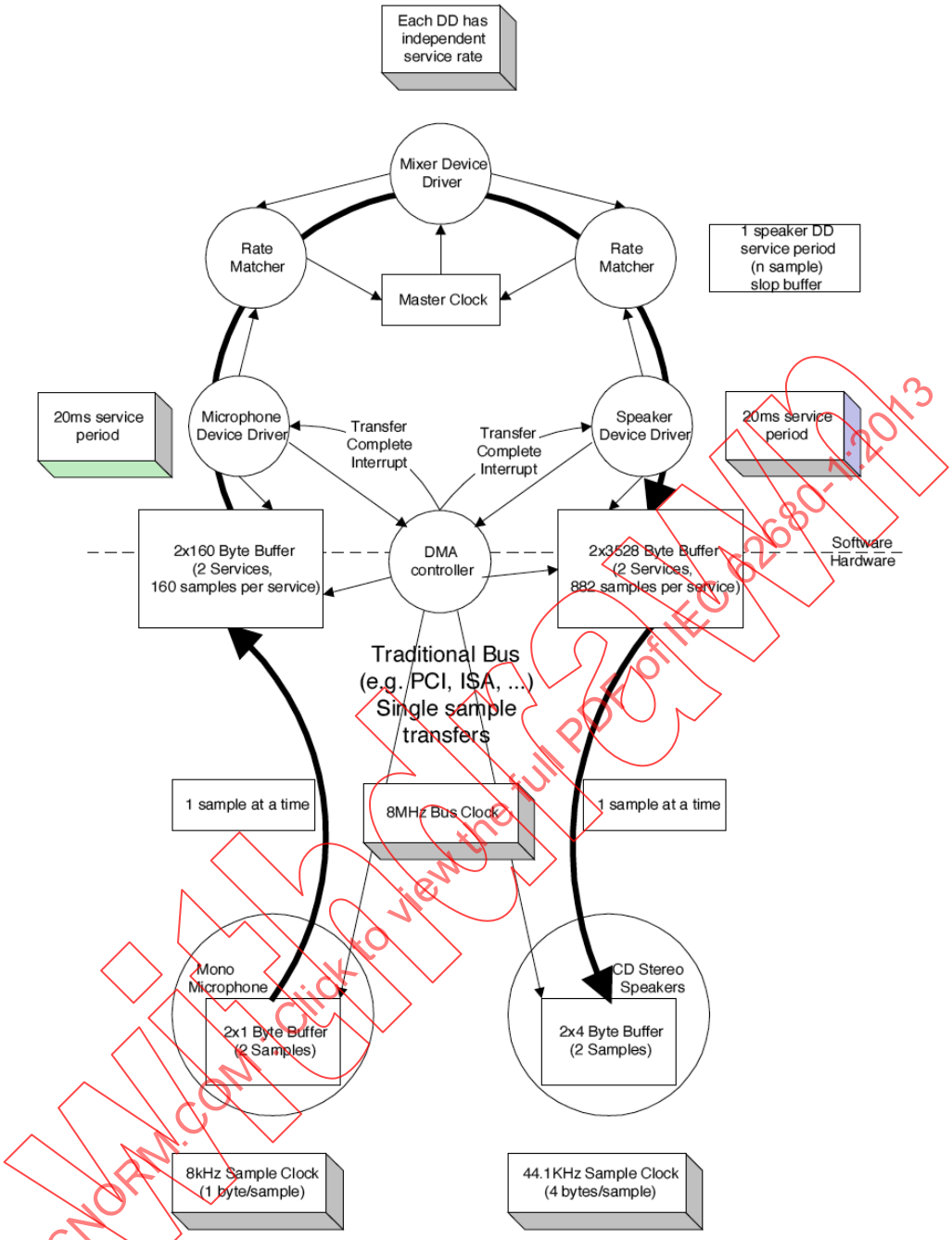


Figure 5-16. Non-USB Isochronous Example

A master clock (which can be provided by software driven from the real time clock) in the PC is used to awaken the mixer to ask the input source for input data and to provide output data to the output sink. In this example, assume it awakens every 20 ms. The microphone and speakers each have their own sample clocks that are unsynchronized with respect to each other or the master mixer clock. The microphone produces data at its natural rate (one-byte samples, 8,000 times a second) and the speakers consume data at their natural rate (four-byte samples, 44,100 times a second). The three clocks in the system can drift and jitter with respect to each other. Each rate matcher may also be running at a different natural rate than either the mixer driver, the input source/driver, or output sink/driver.

The rate matchers also monitor the long-term data rate of their device compared to the master mixer clock and interpolate an additional sample or merge two samples to adjust the data rate

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

of their device to the data rate of the mixer. This adjustment may be required every couple of seconds, but typically occurs infrequently. The rate matchers provide some additional buffering to carry through a rate match.

Note: Some other application might not be able to tolerate sample adjustment and would need some other means of accommodating master clock-to-device clock drift or else would require some means of synchronizing the clocks to ensure that no drift could occur.

The mixer always expects to receive exactly a service period of data (20 ms service period) from its input device and produce exactly a service period of data for its output device. The mixer can be delayed up to less than a service period if data or space is not available from its input/output device. The mixer assumes that such delays do not accumulate.

The input and output devices and their drivers expect to be able to put/get data in response to a hardware interrupt from the DMA controller when their transducer has processed one service period of data. They expect to get/put exactly one service period of data. The input device produces 160 bytes (ten samples) every service period of 20 ms. The output device consumes 3,528 bytes (882 samples) every 20 ms service period. The DMA controller can move a single sample between the device and the host buffer at a rate much faster than the sample rate of either device.

The input and output device drivers provide two service periods of system buffering. One buffer is always being processed by the DMA controller. The other buffer is guaranteed to be ready before the current buffer is exhausted. When the current buffer is emptied, the hardware interrupt awakens the device driver and it calls the rate matcher to give it the buffer. The device driver requests a new IRP with the buffer before the current buffer is exhausted.

The devices can provide two samples of data buffering to ensure that they always have a sample to process for the next sample period while the system is reacting to the previous/next sample.

The service periods of the drivers are chosen to survive interrupt latency variabilities that may be present in the operating system environment. Different operating system environments will require different service periods for reliable operation. The service periods are also selected to place a minimum interrupt load on the system, because there may be other software in the system that requires processing time.

5.12.2 USB Clock Model

Time is present in the USB system via clocks. In fact, there are multiple clocks in a USB system that must be understood:

- Sample Clock: This clock determines the natural data rate of samples moving between client software on the host and the function. This clock does not need to be different between non-USB and USB implementations.
- Bus Clock: This clock runs at a 1.000 ms period (1 kHz frequency) on full-speed segments and 125.000 μs (8 kHz frequency) on high-speed segments of the bus and is indicated by the rate of SOF packets on the bus. This clock is somewhat equivalent to the 8 MHz clock in the non-USB example. In the USB case, the bus clock is often a lower-frequency clock than the sample clock, whereas the bus clock is almost always a higher-frequency clock than the sample clock in a non-USB case.
- Service Clock: This clock is determined by the rate at which client software runs to service IRPs that may have accumulated between executions. This clock also can be the same in the USB and non-USB cases.

In most existing operating systems, it is not possible to support a broad range of isochronous communication flows if each device driver must be interrupted for each sample for fast sample rates. Therefore, multiple samples, if not multiple packets, will be processed by client software and then given to the Host Controller to sequence over the bus according to the

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

prenegotiated bus access requirements. Figure 5-17 presents an example for a reasonable USB clock environment equivalent to the non-USB example in Figure 5-16.

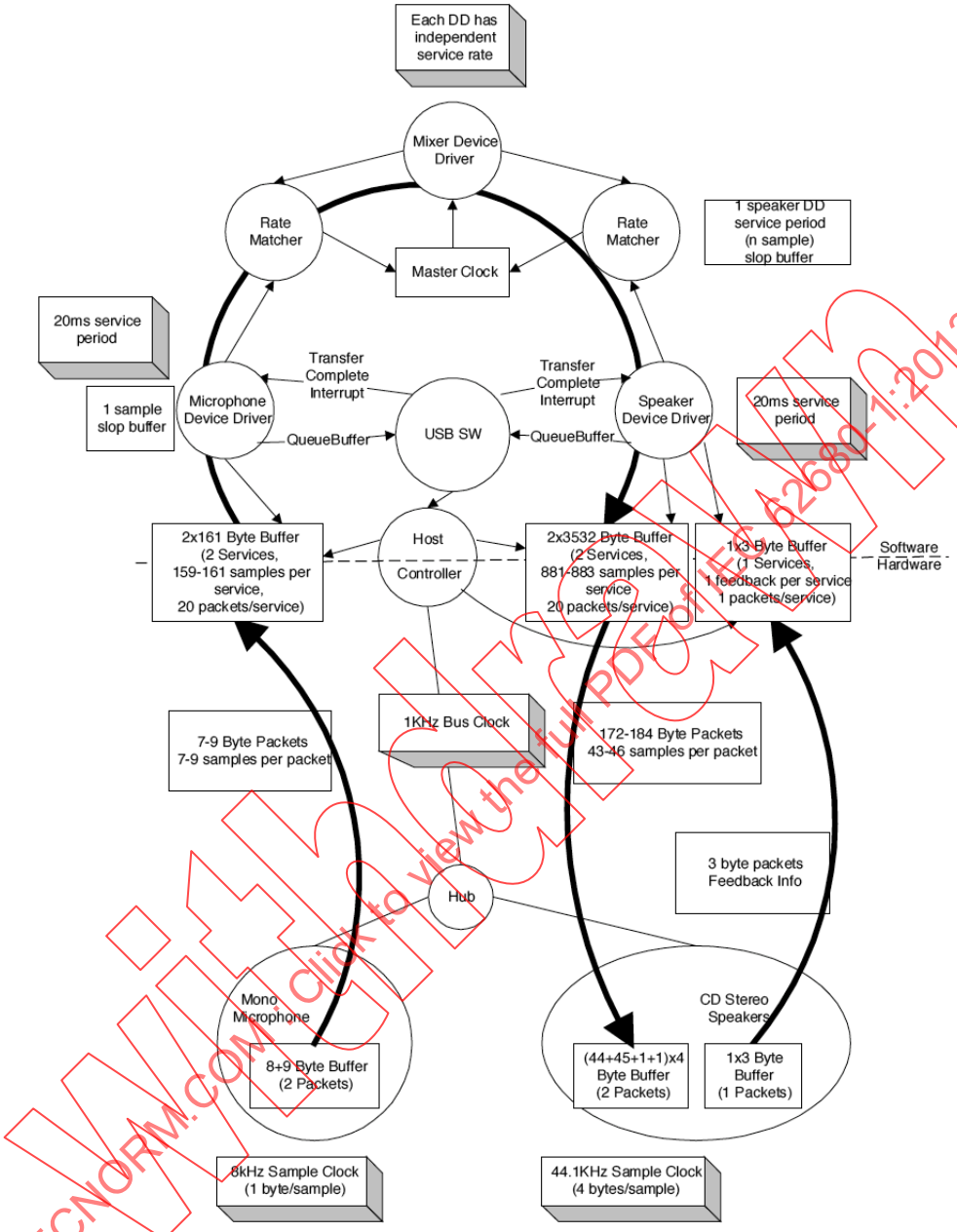


Figure 5-17. USB Full-speed Isochronous Application

Figure 5-17 shows a typical round trip path of information from a microphone as an input device to a speaker as an output device. The clocks, packets, and buffering involved are also shown. Figure 5-17 will be explored in more detail in the following sections.

The focus of this example is to identify the differences introduced by the USB compared to the previous non-USB example. The differences are in the areas of buffering, synchronization given the existence of a USB bus clock, and delay. The client software above the device drivers can be unaffected in most cases.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

5.12.3 Clock Synchronization

In order for isochronous data to be manipulated reliably, the three clocks identified above must be synchronized in some fashion. If the clocks are not synchronized, several clock-to-clock attributes can be present that can be undesirable:

- Clock Drift: Two clocks that are nominally running at the same rate can, in fact, have implementation differences that result in one clock running faster or slower than the other over long periods of time. If uncorrected, this variation of one clock compared to the other can lead to having too much or too little data when data is expected to always be present at the time required.
- Clock Jitter: A clock may vary its frequency over time due to changes in temperature, etc. This may also alter when data is actually delivered compared to when it is expected to be delivered.
- Clock-to-clock Phase Differences: If two clocks are not phase locked, different amounts of data may be available at different points in time as the beat frequency of the clocks cycle out over time. This can lead to quantization/sampling related artifacts.

The bus clock provides a central clock with which USB hardware devices and software can synchronize to one degree or another. However, the software will, in general, not be able to phase- or frequency-lock precisely to the bus clock given the current support for “real time-like” operating system scheduling support in most PC operating systems. Software running in the host can, however, know that data moved over the USB is packetized. For isochronous transfer types, a unit of data is moved exactly once per (micro)frame and the (micro)frame clock is reasonably precise. Providing the software with this information allows it to adjust the amount of data it processes to the actual (micro)frame time that has passed.

Note: For high-speed high-bandwidth endpoints, the data exchanged in the two or three transactions per microframe is still considered to belong to the same “single packet.” The large amount of data per packet is split into two or three transactions only for bus efficiency reasons.

5.12.4 Isochronous Devices

The USB includes a framework for isochronous devices that defines synchronization types, how isochronous endpoints provide data rate feedback, and how they can be connected together. Isochronous devices include sampled analog devices (for example, audio and telephony devices) and synchronous data devices. Synchronization type classifies an endpoint according to its capability to synchronize its data rate to the data rate of the endpoint to which it is connected. Feedback is provided by indicating accurately what the required data rate is, relative to the SOF frequency. The ability to make connections depends on the quality of connection that is required, the endpoint synchronization type, and the capabilities of the host application that is making the connection. Additional device class-specific information may be required, depending on the application.

Note: The term “data” is used very generally, and may refer to data that represents sampled analog information (like audio), or it may be more abstract information. “Data rate” refers to the rate at which analog information is sampled, or the rate at which data is clocked.

The following information is required in order to determine how to connect isochronous endpoints:

- Synchronization type:
  - Asynchronous: Unsynchronized, although sinks provide data rate feedback
  - Synchronous: Synchronized to the USB’s SOF
  - Adaptive: Synchronized using feedback or feedforward data rate information
- Available data rates

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Available data formats

Synchronization type and data rate information are needed to determine if an exact data rate match exists between source and sink, or if an acceptable conversion process exists that would allow the source to be connected to the sink. It is the responsibility of the application to determine whether the connection can be supported within available processing resources and other constraints (like delay). Specific USB device classes define how to describe synchronization type and data rate information.

Data format matching and conversion is also required for a connection, but it is not a unique requirement for isochronous connections. Details about format conversion can be found in other documents related to specific formats.

5.12.4.1 Synchronization Type

Three distinct synchronization types are defined. Table 5-12 presents an overview of endpoint synchronization characteristics for both source and sink endpoints. The types are presented in order of increasing capability.

Table 5-12. Synchronization Characteristics

	Source	Sink
Asynchronous	Free running Fs Provides implicit feedforward (data stream)	Free running Fs Provides explicit feedback (isochronous pipe)
Synchronous	Fs locked to SOF Uses implicit feedback (SOF)	Fs locked to SOF Uses implicit feedback (SOF)
Adaptive	Fs locked to sink Uses explicit feedback (isochronous pipe)	Fs locked to data flow Uses implicit feedforward (data stream)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.12.4.1.1 Asynchronous

Asynchronous endpoints cannot synchronize to SOF or any other clock in the USB domain. They source or sink an isochronous data stream at either a fixed data rate (single-frequency endpoints), a limited number of data rates (32 kHz, 44.1 kHz, 48 kHz, …), or a continuously programmable data rate. If the data rate is programmable, it is set during initialization of the isochronous endpoint. Asynchronous devices must report their programming capabilities in the class-specific endpoint descriptor as described in their device class specification. The data rate is locked to a clock external to the USB or to a free-running internal clock. These devices place the burden of data rate matching elsewhere in the USB environment. Asynchronous source endpoints carry their data rate information implicitly in the number of samples they produce per (micro)frame. Asynchronous sink endpoints must provide explicit feedback information to an adaptive driver (refer to Section 5.12.4.2).

An example of an asynchronous source is a CD-audio player that provides its data based on an internal clock or resonator. Another example is a Digital Audio Broadcast (DAB) receiver or a Digital Satellite Receiver (DSR). Here, too, the sample rate is fixed at the broadcasting side and is beyond USB control.

Asynchronous sink endpoints could be low-cost speakers running off of their internal sample clock.

5.12.4.1.2 Synchronous

Synchronous endpoints can have their clock system (their notion of time) controlled externally through SOF synchronization. These endpoints must slave their sample clock to the 1 ms SOF tick (by means of a programmable PLL). For high-speed endpoints, the presence of the microframe SOF can be used for tighter frame clock tracking.

Synchronous endpoints may source or sink isochronous data streams at either a fixed data rate (single-frequency endpoints), a limited number of data rates (32 kHz, 44.1 kHz, 48 kHz, …), or a continuously programmable data rate. If programmable, the operating data rate is set during initialization of the isochronous endpoint. The number of samples or data units generated in a series of USB (micro)frames is deterministic and periodic. Synchronous devices must report their programming capabilities in the class-specific endpoint descriptor as described in their device class specification.

An example of a synchronous source is a digital microphone that synthesizes its sample clock from SOF and produces a fixed number of audio samples every USB (micro)frame. Likewise, a synchronous sink derives its sample clock from SOF and consumes a fixed number of samples every USB (micro)frame.

5.12.4.1.3 Adaptive

Adaptive endpoints are the most capable endpoints possible. They are able to source or sink data at any rate within their operating range. Adaptive source endpoints produce data at a rate that is controlled by the data sink. The sink provides feedback (refer to Section 5.12.4.2) to the source, which allows the source to know the desired data rate of the sink. For adaptive sink endpoints, the data rate information is embedded in the data stream. The average number of samples received during a certain averaging time determines the instantaneous data rate. If this number changes during operation, the data rate is adjusted accordingly.

The data rate operating range may center around one rate (e.g., 8 kHz), select between several programmable or auto-detecting data rates (32 kHz, 44.1 kHz, 48 kHz, …), or may be within one or more ranges (e.g., 5 kHz to 12 kHz or 44 kHz to 49 kHz). Adaptive devices must report their programming capabilities in the class-specific endpoint descriptor as described in their device class specification.

An example of an adaptive source is a CD player that contains a fully adaptive sample rate converter (SRC) so that the output sample frequency no longer needs to be 44.1 kHz but can be anything within the operating range of the SRC. Adaptive sinks include such endpoints as high-end digital speakers, headsets, etc.

5.12.4.2 Feedback

An asynchronous sink must provide explicit feedback to the host by indicating accurately what its desired data rate (*Ff*) is, relative to the USB (micro)frame frequency. This allows the host to continuously adjust the number of samples sent to the sink so that neither underflow or overflow of the data buffer occurs. Likewise, an adaptive source must receive explicit feedback from the host so that it can accurately generate the number of samples required by the host. Feedback endpoints can be specified as described in Section 9.6.6 for the *bmAttributes* field of the endpoint descriptor.

To generate the desired data rate F_f, the device must measure its actual sampling rate F_s, referenced to the USB notion of time, i.e., the USB (micro)frame frequency. This specification requires the data rate F_f to be resolved to better than one sample per second (1Hz) in order to allow a high-quality source rate to be created and to tolerate delays and errors in the feedback loop. To achieve this accuracy, the measurement time T_{meas} must be at least 1 second. Therefore:

$$T_{meas} = 2^K$$

where T_{meas} is now expressed in USB (micro)frames and K=10 for full-speed devices (1 ms frames) and K=13 for high-speed devices (125 µs microframes). However, in most devices, the actual sampling rate F_s is derived from a master clock F_m through a binary divider. Therefore:

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

$$F_m = F_s * 2^P$$

where *P* is a positive integer (including 0 if no higher-frequency master clock is available). The measurement time T_{meas} can now be decreased by measuring F_m instead of F_s and:

$$T_{meas} = \frac{2^K}{2^P} = 2^{(K-P)}$$

In this way, a new estimate for F_f becomes available every $2^{(K-P)}$ (micro)frames. *P* is practically bound to be in the range [0,K] because there is no point in using a clock slower than F_s (*P*=0), and no point in trying to update F_f more than once per (micro)frame (*P*=K). A sink can determine F_f by counting cycles of the master clock F_m for a period of $2^{(K-P)}$(micro)frames. The counter is read into F_f and reset every $2^{(K-P)}$(micro)frames. As long as no clock cycles are skipped, the count will be accurate over the long term.

Each (micro)frame, an adaptive source adds F_f to any remaining fractional sample count from the previous (micro)frame, sources the number of samples in the integer part of the sum, and retains the fractional sample count for the next (micro)frame. The source can look at the behavior of F_f over many (micro)frames to determine an even more accurate rate, if it needs to.

F_f is expressed in number of samples per (micro)frame. The F_f value consists of an integer part that represents the (integer) number of samples per (micro)frame and a fractional part that represents the “fraction” of a sample that would be needed to match the sampling frequency F_s to a resolution of 1 Hz or better. The fractional part requires at least K bits to represent the “fraction” of a sample to a resolution of 1 Hz or better. The integer part must have enough bits to represent the maximum number of samples that can ever occur in a single (micro)frame. Assuming that the minimum sample size is one byte, then this number is limited to 1,023 for full-speed endpoints. Ten bits are therefore sufficient to encode this value. For high-speed endpoints, this number is limited to 3*1,024=3,072 and twelve bits are needed.

In summary, for full-speed endpoints, the F_f value shall be encoded in an unsigned 10.10 (*K*=10) format which fits into three bytes. Because the maximum integer value is fixed to 1,023, the 10.10 number will be left-justified in the 24 bits, so that it has a 10.14 format. Only the first ten bits behind the binary point are required. The lower four bits may be optionally used to extend the precision of F_f, otherwise, they shall be reported as zero. For high-speed endpoints, the F_f value shall be encoded in an unsigned 12.13 (*K*=13) format which fits into four bytes. The value shall be aligned into these four bytes so that the binary point is located between the second and the third byte so that it has a 16.16 format. The most significant four bits shall be reported zero. Only the first 13 bits behind the binary point are required. The lower three bits may be optionally used to extend the precision of F_f, otherwise, they shall be reported as zero.

An endpoint needs to implement only the number of bits that it effectively requires for its maximum F_f.

The choice of *P* is endpoint-specific. Use the following guidelines when choosing *P*:

- *P* must be in the range [0,K].

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- Larger values of *P* are preferred, because they reduce the size of the frame counter and increase the rate at which F_f is updated. More frequent updates result in a tighter control of the source data rate, which reduces the buffer space required to handle F_f changes.
- *P* should be less than *K* so that F_f is averaged across at least two frames in order to reduce SOF jitter effects.
- *P* should not be zero in order to keep the deviation in the number of samples sourced to less than 1 in the event of a lost F_f value.

Isochronous transfers are used to read F_f from the feedback register. The desired reporting rate for the feedback should be $2^{(K-P)}$frames. F_f will be reported at most once per update period. There is nothing to be gained by reporting the same F_f value more than once per update period. The endpoint may choose to report F_f only if the updated value has changed from the previous F_f value. If the value has not changed, the endpoint may report the current F_f value or a zero length data payload. It is strongly recommended that an endpoint always report the current F_f value any time it is polled.

It is possible that the source will deliver one too many or one too few samples over a long period due to errors or accumulated inaccuracies in measuring F_f. The sink must have sufficient buffer capability to accommodate this. When the sink recognizes this condition, it should adjust the reported F_f value to correct it. This may also be necessary to compensate for relative clock drifts. The implementation of this correction process is endpoint-specific and is not specified.

5.12.4.3 Implicit Feedback

In some cases, implementing a separate explicit feedback endpoint can be avoided. If a device implements a group of isochronous data endpoints that are closely related and if:

- All the endpoints in the group are synchronized (i.e. use sample clocks that are derived from a common master clock)
- The group contains one or more isochronous data endpoints in one direction that normally would need explicit feedback
- The group contains at least one isochronous data endpoint in the opposite direction

Under these circumstances, the device may elect not to implement a separate isochronous explicit feedback endpoint. Instead, feedback information can be derived from the data endpoint in the opposite direction by observing its data rate.

Two cases can arise:

- One or more asynchronous sink endpoints are accompanied by an asynchronous source endpoint. The data rate on the source endpoint can be used as implicit feedback information to adjust the data rate on the sink endpoint(s).
- One or more adaptive source endpoints are accompanied by an adaptive sink endpoint. The source endpoint can adjust its data rate based on the data rate received by the sink endpoint.

This specification provides the necessary framework to implement synchronization as described above (see Chapter 9). However, exactly how the desired data rate F_f is derived from the data rate of the implied feedback endpoint is implementation-dependent.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.12.4.4 Connectivity

In order to fully describe the source-to-sink connectivity process, an interconnect model is presented. The model indicates the different components involved and how they interact to establish the connection.

The model provides for multi-source/multi-sink situations. Figure 5-18 illustrates a typical situation (highly condensed and incomplete). A physical device is connected to the host application software through different hardware and software layers as described in this specification. At the client interface level, a virtual device is presented to the application. From the application standpoint, only virtual devices exist. It is up to the device driver and client software to decide what the exact relation is between physical and virtual device.

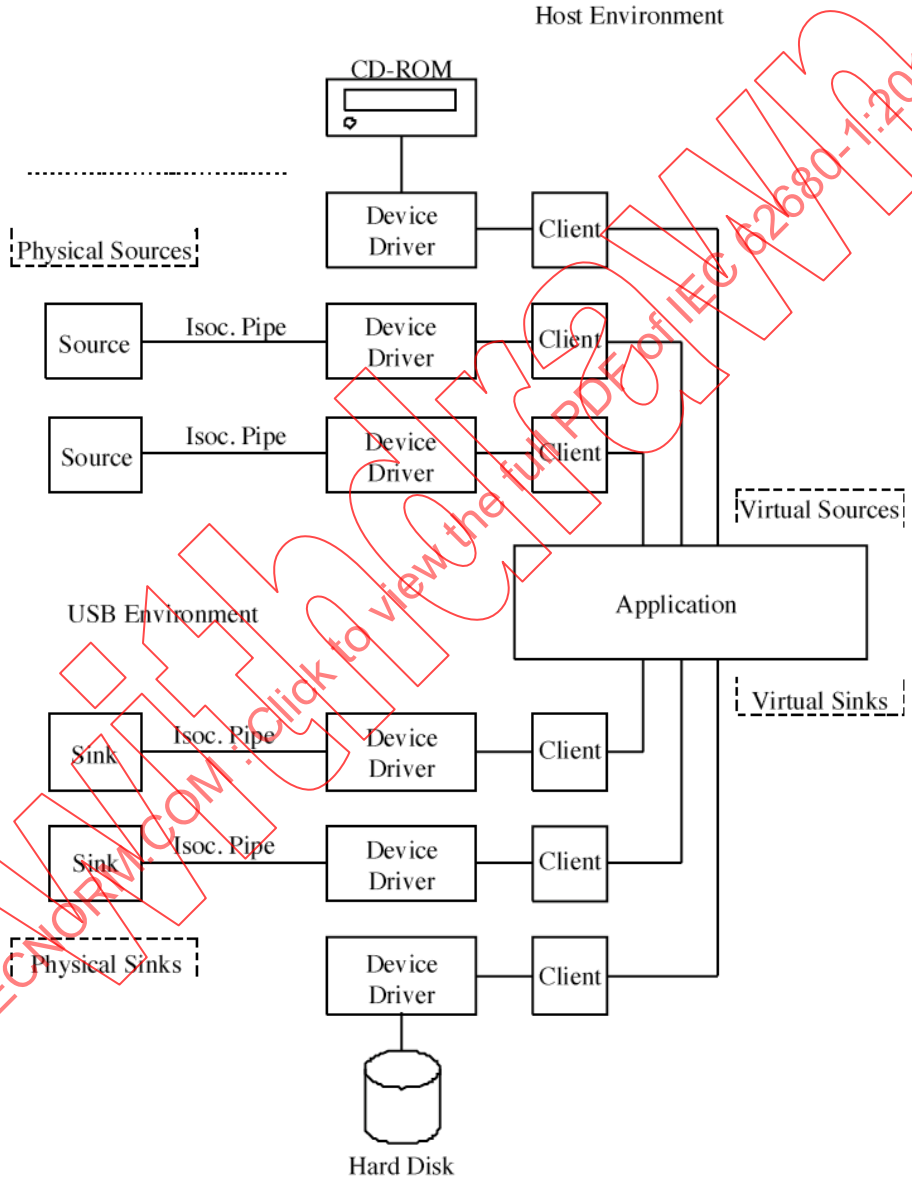


Figure 5-18. Example Source/Sink Connectivity

Device manufacturers (or operating system vendors) must provide the necessary device driver software and client interface software to convert their device from the physical implementation to a USB-compliant software implementation (the virtual device). As stated before, depending on the capabilities built into this software, the virtual device can exhibit different synchronization behavior from the physical device. However, the synchronization

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

classification applies equally to both physical and virtual devices. All physical devices belong to one of the three possible synchronization types. Therefore, the capabilities that have to be built into the device driver and/or client software are the same as the capabilities of a physical device. The word “application” must be replaced by “device driver/client software.” In the case of a physical source to virtual source connection, “virtual source device” must be replaced by “physical source device” and “virtual sink device” must be replaced by “virtual source device.” In the case of a virtual sink to physical sink connection, “virtual source device” must be replaced by “virtual sink device” and “virtual sink device” must be replaced by “physical sink device.”

Placing the rate adaptation (RA) functionality into the device driver/client software layer has the distinct advantage of isolating all applications, relieving the device from the specifics and problems associated with rate adaptation. Applications that would otherwise be multi-rate degenerate to simpler mono-rate systems.

Note: The model is not limited to only USB devices. For example, a CD-ROM drive containing 44.1 kHz audio can appear as either an asynchronous, synchronous, or adaptive source. Asynchronous operation means that the CD-ROM fills its buffer at the rate that it reads data from the disk, and the driver empties the buffer according to its USB service interval. Synchronous operation means that the driver uses the USB service interval (e.g., 10 ms) and nominal sample rate of the data (44.1 kHz) to determine to put out 441 samples every USB service interval. Adaptive operation would build in a sample rate converter to match the CD-ROM output rate to different sink sampling rates.

Using this reference model, it is possible to define what operations are necessary to establish connections between various sources and sinks. Furthermore, the model indicates at what level these operations must or can take place. First, there is the stage where physical devices are mapped onto virtual devices and vice versa. This is accomplished by the driver and/or client software. Depending on the capabilities included in this software, a physical device can be transformed into a virtual device of an entirely different synchronization type. The second stage is the application that uses the virtual devices. Placing rate matching capabilities at the driver/client level of the software stack relieves applications communicating with virtual devices from the burden of performing rate matching for every device that is attached to them. Once the virtual device characteristics are decided, the actual device characteristics are not any more interesting than the actual physical device characteristics of another driver.

As an example, consider a mixer application that connects at the source side to different sources, each running at their own frequencies and clocks. Before mixing can take place, all streams must be converted to a common frequency and locked to a common clock reference. This action can be performed in the physical-to-virtual mapping layer or it can be handled by the application itself for each source device independently. Similar actions must be performed at the sink side. If the application sends the mixed data stream out to different sink devices, it can either do the rate matching for each device itself or it can rely on the driver/client software to do that, if possible.

Table 5-13 indicates at the intersections what actions the application must perform to connect a source endpoint to a sink endpoint.

Table 5-13. Connection Requirements

	Source Endpoint		
Sink Endpoint	**Asynchronous**	**Synchronous**	**Adaptive**
Asynchronous	Async Source/Sink RA See Note 1.	Async SOF/Sink RA See Note 2.	Data + Feedback Feedthrough See Note 3.
Synchronous	Async Source/SOF RA See Note 4.	Sync RA See Note 5.	Data Feedthrough + Application Feedback

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

	Source Endpoint		
Sink Endpoint	**Asynchronous**	**Synchronous**	**Adaptive**
			See Note 6.
Adaptive	Data Feedthrough See Note 7.	Data Feedthrough See Note 8.	Data Feedthrough See Note 9.

Notes:

1. Asynchronous RA in the application. Fs_i is determined by the source, using the feedforward information embedded in the data stream. Fs_o is determined by the sink, based on feedback information from the sink. If nominally $Fs_i = Fs_o$, the process degenerates to a feedthrough connection if slips/stuffs due to lack of synchronization are tolerable. Such slips/stuffs will cause audible degradation in audio applications.
2. Asynchronous RA in the application. Fs_i is determined by the source but locked to SOF. Fs_o is determined by the sink, based on feedback information from the sink. If nominally $Fs_i = Fs_o$, the process degenerates to a feedthrough connection if slips/stuffs due to lack of synchronization are tolerable. Such slips/stuffs will cause audible degradation in audio applications.
3. If Fs_o falls within the locking range of the adaptive source, a feedthrough connection can be established. $Fs_i = Fs_o$ and both are determined by the asynchronous sink, based on feedback information from the sink. If Fs_o falls outside the locking range of the adaptive source, the adaptive source is switched to synchronous mode and Note 2 applies.
4. Asynchronous RA in the application. Fs_i is determined by the source. Fs_o is determined by the sink and locked to SOF. If nominally $Fs_i = Fs_o$, the process degenerates to a feedthrough connection if slips/stuffs due to lack of synchronization are tolerable. Such slips/stuffs will cause audible degradation in audio applications.
5. Synchronous RA in the application. Fs_i is determined by the source and locked to SOF. Fs_o is determined by the sink and locked to SOF. If $Fs_i = Fs_o$, the process degenerates to a loss-free feedthrough connection.
6. The application will provide feedback to synchronize the source to SOF. The adaptive source appears to be a synchronous endpoint and Note 5 applies.
7. If Fs_i falls within the locking range of the adaptive sink, a feedthrough connection can be established. $Fs_i = Fs_o$ and both are determined by and locked to the source. If Fs_i falls outside the locking range of the adaptive sink, synchronous RA is done in the host to provide an Fs_o that is within the locking range of the adaptive sink.
8. If Fs_i falls within the locking range of the adaptive sink, a feedthrough connection can be established. $Fs_o = Fs_i$ and both are determined by the source and locked to SOF. If Fs_i falls outside the locking range of the adaptive sink, synchronous RA is done in the host to provide an Fs_o that is within the locking range of the adaptive sink.
9. The application will use feedback control to set Fs_o of the adaptive source when the connection is set up. The adaptive source operates as an asynchronous source in the absence of ongoing feedback information and Note 7 applies.

In cases where RA is needed but not available, the rate adaptation process could be mimicked by sample dropping/stuffing. The connection could then still be made, possibly with a warning about poor quality, otherwise, the connection cannot be made.

5.12.4.4.1 Audio Connectivity

When the above is applied to audio data streams, the RA process is replaced by sample rate conversion, which is a specialized form of rate adaptation. Instead of error control, some

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

form of sample interpolation is used to match incoming and outgoing sample rates. Depending on the interpolation techniques used, the audio quality (distortion, signal to noise ratio, etc.) of the conversion can vary significantly. In general, higher quality requires more processing power.

5.12.4.4.2 Synchronous Data Connectivity

For the synchronous data case, RA is used. Occasional slips/stuffs may be acceptable to many applications that implement some form of error control. Error control includes error detection and discard, error detection and retransmit, or forward error correction. The rate of slips/stuffs will depend on the clock mismatch between the source and sink and may be the dominant error source of the channel. If the error control is sufficient, then the connection can still be made.

5.12.5 Data Prebuffering

The USB requires that devices prebuffer data before processing/transmission to allow the host more flexibility in managing when each pipe's transaction is moved over the bus from (micro)frame to (micro)frame.

For transfers from function to host, the endpoint must accumulate samples during (micro)frame X until it receives the SOF token for (micro)frame X+1. It "latches" the data from (micro)frame X into its packet buffer and is now ready to send the packet containing those samples during (micro)frame X+1. When it will send that data during the (micro)frame is determined solely by the Host Controller and can vary from (micro)frame to (micro)frame.

For transfers from host to function, the endpoint will accept a packet from the host sometime during (micro)frame Y. When it receives the SOF for (micro)frame Y+1, it can then start processing the data received in (micro)frame Y.

This approach allows an endpoint to use the SOF token as a stable clock with very little jitter and/or drift when the Host Controller moves the packet over the bus. This approach also allows the Host Controller to vary within a (micro)frame precisely when the packet is actually moved over the bus. This prebuffering introduces some additional delay between when a sample is available at an endpoint and when it moves over the bus compared to an environment where the bus access is at exactly the same time offset from SOF from (micro)frame to (micro)frame.

Figure 5-19 shows the time sequence for a function-to-host transfer (IN process). Data D_0 is accumulated during (micro)frame F_i at time T_i and transmitted to the host during (micro)frame F_{i+1}. Similarly, for a host-to-function transfer (OUT process), data D_0 is received by the endpoint during (micro)frame F_{i+1} and processed during (micro)frame F_{i+2}.

Time:	T_i	T_{i+1}	T_{i+2}	T_{i+3}	------	------	T_m	T_{m+1}	------
(Micro)Frame:	F_i	F_{i+1}	F_{i+2}	M_{i+3}	------	------	F_m	F_{m+1}	------
Data on Bus:	------	D_0	D_1	D_2	------	------	D_0	D_1	------
OUT Process:	------	------	D_0	D_1	------	------	------	D_0	------
IN Process:	D_0	D_1	------	------	------	D_0	------	------	------

Figure 5-19. Data Prebuffering

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.12.6 SOF Tracking

Functions supporting isochronous pipes must receive and comprehend the SOF token to support prebuffering as previously described. Given that SOFs can be corrupted, a device must be prepared to recover from a corrupted SOF. These requirements limit isochronous transfers to full-speed and high-speed devices only, because low-speed devices do not see SOFs on the bus. Also, because SOF packets can be damaged in transmission, devices that support isochronous transfers need to be able to synthesize the existence of an SOF that they may not see due to a bus error.

Isochronous transfers require the appropriate data to be transmitted in the corresponding (micro)frame. The USB requires that when an isochronous transfer is presented to the Host Controller, it identifies the (micro)frame number for the first (micro)frame. The Host Controller must not transmit the first transaction before the indicated (micro)frame number. Each subsequent transaction in the IRP must be transmitted in succeeding (micro)frames (except for high-speed high-bandwidth transfers where up to three transactions may occur in the same microframe). If there are no transactions pending for the current (micro)frame, then the Host Controller must not transmit anything for an isochronous pipe. If the indicated (micro)frame number has passed, the Host Controller must skip (i.e. not transmit) all transactions until the one corresponding to the current (micro)frame is reached.

5.12.7 Error Handling

Isochronous transfers provide no data packet retries (i.e., no handshakes are returned to a transmitter by a receiver) so that timeliness of data delivery is not perturbed. However, it is still important for the agents responsible for data transport to know when an error occurs and how the error affects the communication flow. In particular, for a sequence of data packets (A, B, C, D), the USB allows sufficient information such that a missing packet (A, _, C, D) can be detected and will not unknowingly be turned into an incorrect data or time sequence (A, C, D or A, _, B, C, D). The protocol provides four mechanisms that support this: a strictly defined periodicity for the transmission of packets and data PID sequencing mechanisms for high-speed high-bandwidth endpoints, SOF, CRC and bus transaction timeout.

- Isochronous transfers require periodic occurrence of data transactions for normal operation. The period must be an exact power of two (micro)frames. The USB does not dictate what data is transmitted in each frame. The data transmitter/source determines specifically what data to provide. This regular periodic data delivery provides a framework that is fundamental to detecting missing data errors. For high-speed high-bandwidth endpoints, data PID sequencing allows the detection of missing or damaged transactions during a microframe. Any phase of a transaction can be damaged during transmission on the bus. Chapter 8 describes how each error case affects the protocol.
- Because every (micro)frame is preceded by an SOF and a receiver can see SOFs on the bus, a receiver can determine that its expected transaction for that (micro)frame did not occur between two SOFs. Additionally, because even an SOF can be damaged, a device must be able to reconstruct the existence of a missed SOF as described in Section 5.12.6.
- A data packet may be corrupted on the bus; therefore, CRC protection allows a receiver to determine that the data packet it received was corrupted.
- The protocol defines the details that allow a receiver to determine via bus transaction timeout that it is not going to receive its data packet after it has successfully seen its token packet.

Once a receiver has determined that a data packet was not received, it may need to know the size of the data that was missed in order to recover from the error with regard to its functional behavior. If the communication flow is always the same data size per (micro)frame, then the size is always a known constant. However, in some cases, the data size can vary from (micro)frame to (micro)frame. In this case, the receiver and transmitter have an implementation-dependent mechanism to determine the size of the lost packet.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

In summary, whether a transaction is actually moved successfully over the bus or not, the transmitter and receiver always advance their data/buffer streams as indicated by the bus access period to keep data-per-time synchronization. The detailed mechanisms described above allow detection, tracking, and reporting of damaged transactions so that a function or its client software can react to the damage in a function-appropriate fashion. The details of that function- or application-specific reaction are outside the scope of the USB Specification.

5.12.8 Buffering for Rate Matching

Given that there are multiple clocks that affect isochronous communication flows in the USB, buffering is required to rate match the communication flow across the USB. There must be buffer space available both in the device per endpoint and on the host side on behalf of the client software. These buffers provide space for data to accumulate until it is time for a transfer to move over the USB. Given the natural data rates of the device, the maximum size of the data packets that move over the bus can also be calculated.

Figure 5-20 shows the equations used to determine buffer size on the device and host and maximum packet size that must be requested to support a desired data rate. These equations are a function of the service clock rate (FX), bus clock rate (F_{SOF}), sample clock rate (FS), bus access period (I), and sample size (S). These equations should provide design information for selecting the appropriate packet size that an endpoint will report in its characteristic information and the appropriate buffer requirements for the device/endpoint and its client software. Figure 5-17 shows actual buffer, packet, and clock values for a typical full-speed isochronous example.

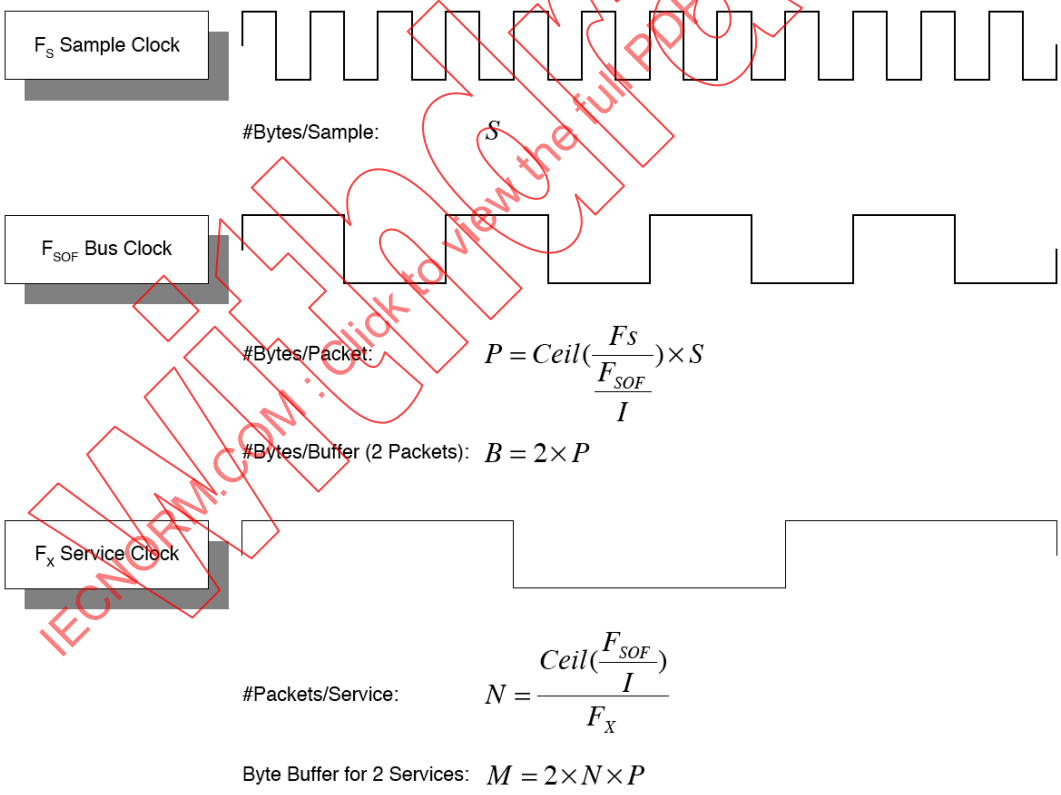


Figure 5-20. Packet and Buffer Size Formulas for Rate-matched Isochronous Transfers

The USB data model assumes that devices have some natural sample size and rate. The USB supports the transmission of packets that are multiples of sample size to make error recovery handling easier when isochronous transactions are damaged on the bus. If a device has no natural sample size or if its samples are larger than a packet, it should describe its sample size as being one byte. If a sample is split across a data packet, the error recovery can be harder when an arbitrary transaction is lost. In some cases, data synchronization can

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

be lost unless the receiver knows in what (micro)frame number each partial sample is transmitted. Furthermore, if the number of samples can vary due to clock correction (e.g., for a non-derived device clock), it may be difficult or inefficient to know when a partial sample is transmitted. Therefore, the USB does not split samples across packets.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

6 Chapter 6 Mechanical

This chapter provides the mechanical and electrical specifications for the cables, connectors, and cable assemblies used to interconnect USB devices. The specification includes the dimensions, materials, electrical, and reliability requirements. This chapter documents minimum requirements for the external USB interconnect. Substitute material may be used as long as it meets these minimums.

6.1 Architectural Overview

The USB physical topology consists of connecting the downstream hub port to the upstream port of another hub or to a device. The USB can operate at three speeds. High-speed (480 Mb/s) and full-speed (12 Mb/s) require the use of a shielded cable with two power conductors and twisted pair signal conductors. Low-speed (1.5 Mb/s) recommends, but does not require the use of a cable with twisted pair signal conductors.

The connectors are designed to be hot plugged. The USB Icon on the plugs provides tactile feedback making it easy to obtain proper orientation.

6.2 Keyed Connector Protocol

To minimize end user termination problems, USB uses a "keyed connector" protocol. The physical difference in the Series "A" and "B" connectors insures proper end user connectivity. The "A" connector is the principle means of connecting USB devices directly to a host or to the downstream port of a hub. All USB devices must have the standard Series "A" connector specified in this chapter. The "B" connector allows device vendors to provide a standard detachable cable. This facilitates end user cable replacement. Figure 6-1 illustrates the keyed connector protocol.

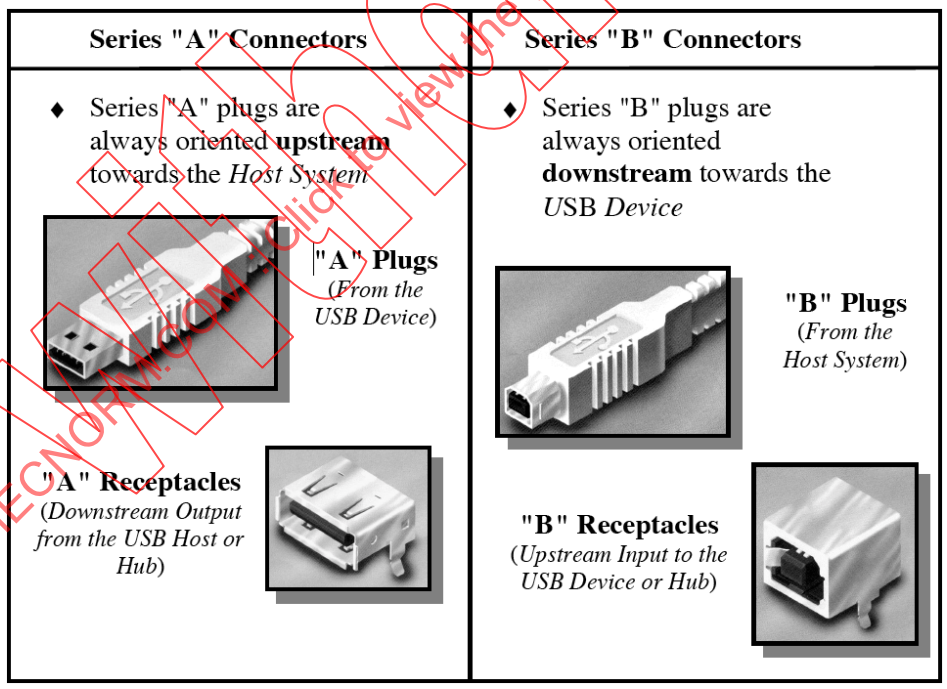


Figure 6-1. Keyed Connector Protocol

The following list explains how the plugs and receptacles can be mated:

- Series "A" receptacle mates with a Series "A" plug. Electrically, Series "A" receptacles function as outputs from host systems and/or hubs.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- Series “A” plug mates with a Series “A” receptacle. The Series “A” plug always is oriented towards the host system.
- Series “B” receptacle mates with a Series “B” plug (male). Electrically, Series “B” receptacles function as inputs to hubs or devices.
- Series “B” plug mates with a Series “B” receptacle. The Series “B” plug is always oriented towards the USB hub or device.

6.3 Cable

USB cable consists of four conductors, two power conductors, and two signal conductors.

High-/full-speed cable consists of a signaling twisted pair, VBUS, GND, and an overall shield. High-/full-speed cable must be marked to indicate suitability for USB usage (see Section 6.6.2). High-/full-speed cable may be used with either low-speed, full-speed, or high-speed devices. When high-/full-speed cable is used with low-speed devices, the cable must meet all low-speed requirements.

Low-speed recommends, but does not require the use of a cable with twisted signaling conductors.

6.4 Cable Assembly

This specification describes three USB cable assemblies: standard detachable cable, high-/full-speed captive cable, and low-speed captive cable.

A standard detachable cable is a high-/full-speed cable that is terminated on one end with a Series “A” plug and terminated on the opposite end with a series “B” plug. A high-/full-speed captive cable is terminated on one end with a Series “A” plug and has a vendor-specific connect means (hardwired or custom detachable) on the opposite end for the high-/full-speed peripheral. The low-speed captive cable is terminated on one end with a Series “A” plug and has a vendor-specific connect means (hardwired or custom detachable) on the opposite end for the low-speed peripheral. Any other cable assemblies are prohibited.

The color used for the cable assembly is vendor specific; recommended colors are white, grey, or black.

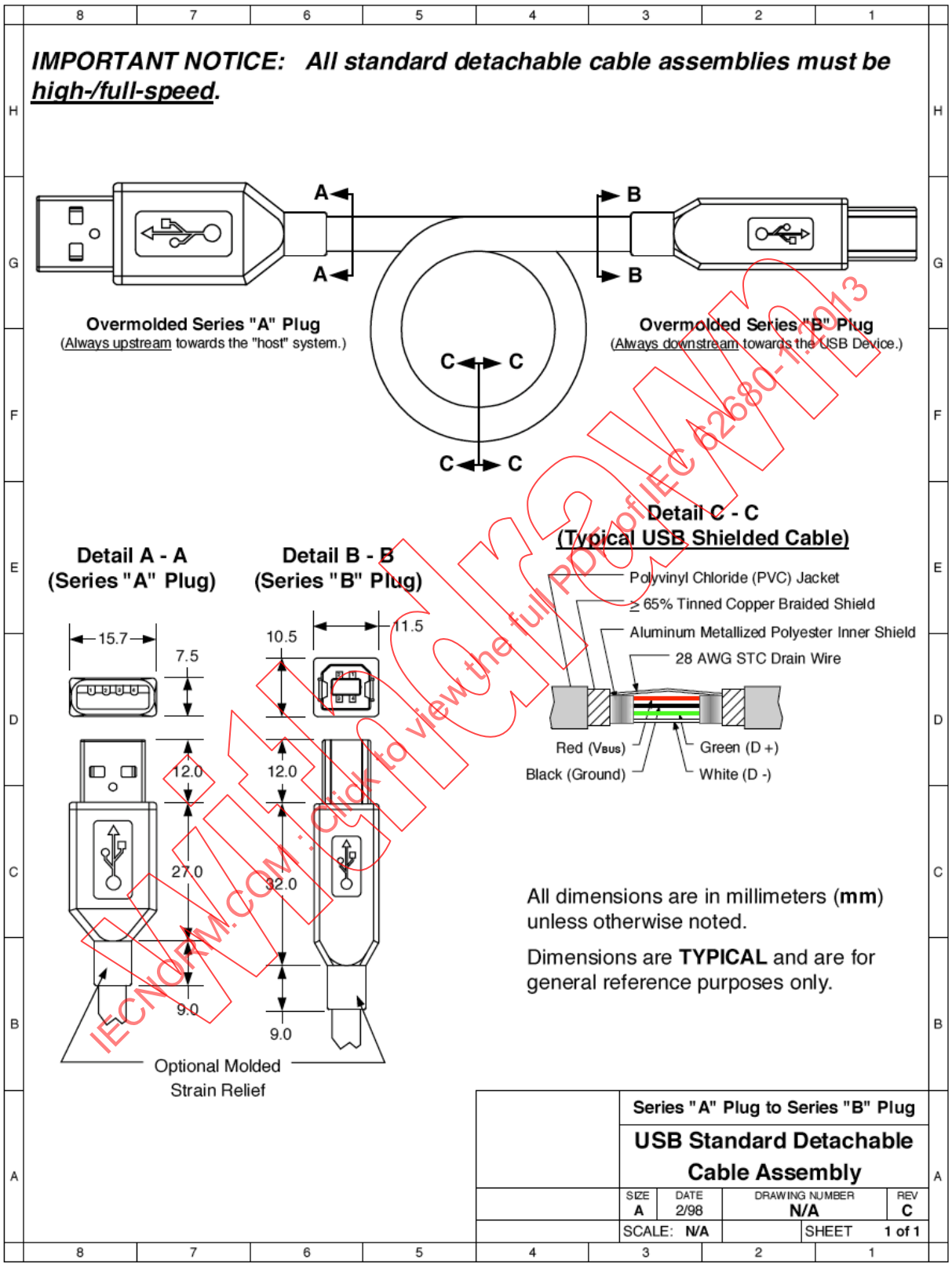
6.4.1 Standard Detachable Cable Assemblies

High-speed and full-speed devices can utilize the “B” connector. This allows the device to have a standard detachable USB cable. This eliminates the need to build the device with a hardwired cable and minimizes end user problems if cable replacement is necessary.

Devices utilizing the “B” connector must be designed to work with worst case maximum length detachable cable. Standard detachable cable assemblies may be used only on high-speed and full-speed devices. Using a high-/full-speed standard detachable cable on a low-speed device may exceed the maximum low-speed cable length.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-2 illustrates a standard detachable cable assembly.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-2. USB Standard Detachable Cable Assembly

Standard detachable cable assemblies must meet the following electrical requirements:

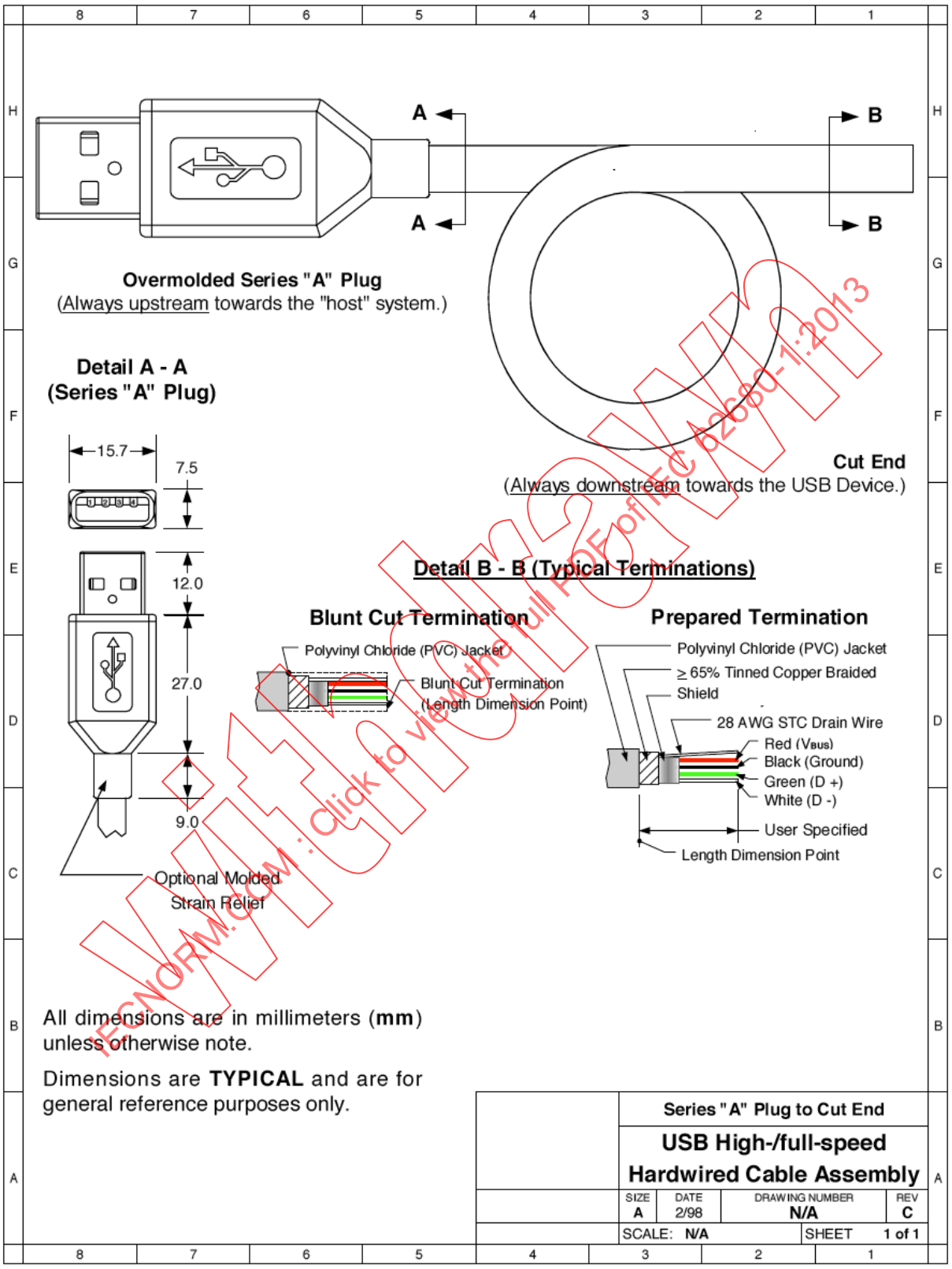
- The cable must be terminated on one end with an overmolded Series “A” plug and the opposite end is terminated with an overmolded Series “B” plug.
- The cable must be rated for high-speed and full-speed.
- The cable impedance must match the impedance of the high-speed and full-speed drivers. The drivers are characterized to drive specific cable impedance. Refer to Section 7.1.1 for details.
- The maximum allowable cable length is determined by signal pair attenuation and propagation delay. Refer to Sections 7.1.14 and 7.1.17 for details.
- Differences in propagation delay between the two signal conductors must be minimized. Refer to Section 7.1.3 for details.
- The GND lead provides a common ground reference between the upstream and downstream ports. The maximum cable length is limited by the voltage drop across the GND lead. Refer to Section 7.2.2 for details. The minimum acceptable wire gauge is calculated assuming the attached device is high power.
- The VBUS lead provides power to the connected device. For standard detachable cables, the VBUS requirement is the same as the GND lead.

6.4.2 High-/full-speed Captive Cable Assemblies

Assemblies are considered captive if they are provided with a vendor-specific connect means (hardwired or custom detachable) to the peripheral. High-/full-speed hardwired cable assemblies may be used with either high-speed, full-speed, or low-speed devices. When using a high-/full-speed hardwired cable on a low-speed device, the cable must meet all low-speed requirements.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-3 illustrates a high-/full-speed hardwired cable assembly.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-3. USB High-/full-speed Hardwired Cable Assembly

High-/full-speed captive cable assemblies must meet the following electrical requirements:

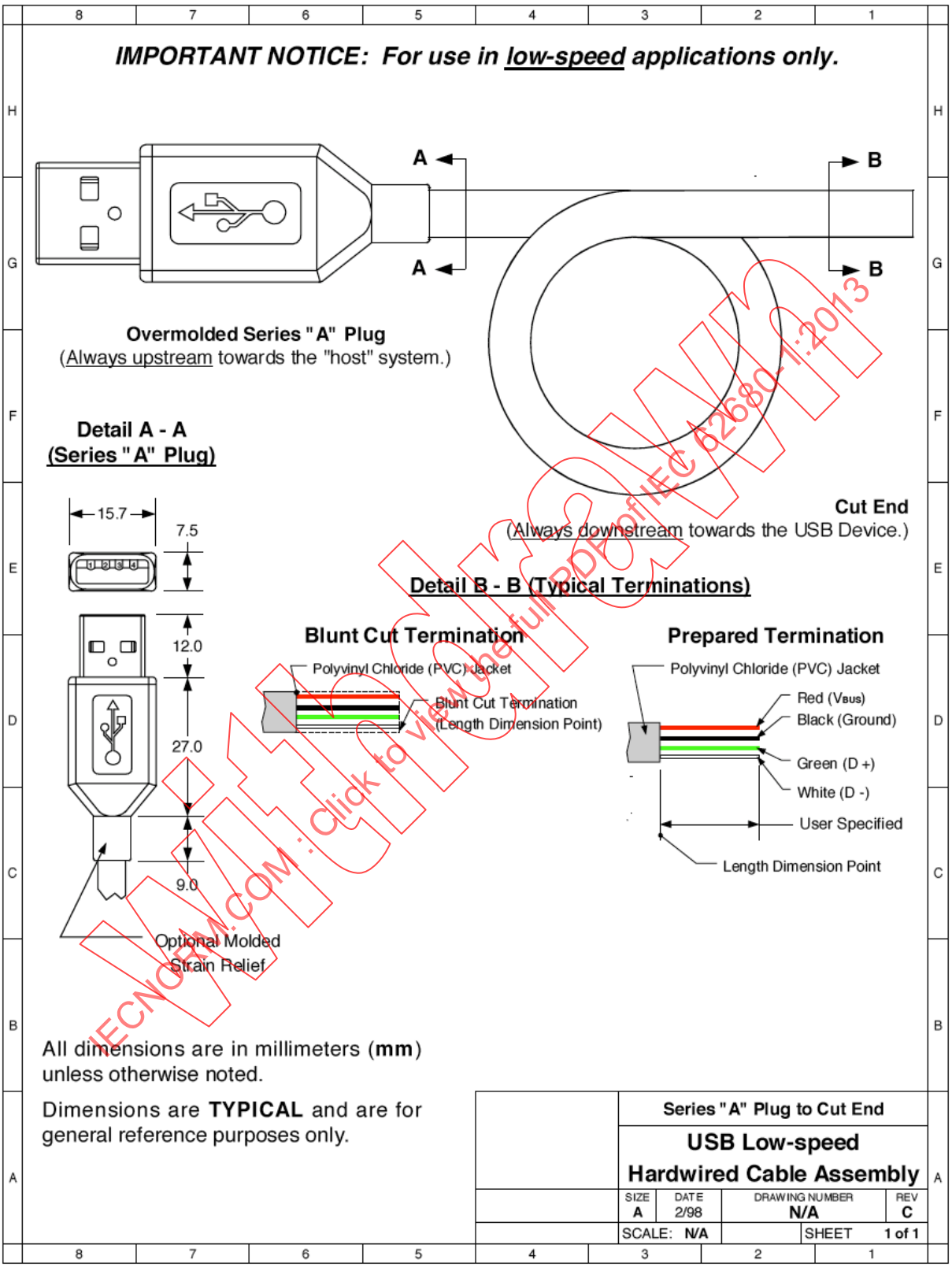
- The cable must be terminated on one end with an overmolded Series “A” plug and the opposite end is vendor specific. If the vendor specific interconnect is to be hot plugged, it must meet the same performance requirements as the USB “B” connector.
- The cable must be rated for high-speed and full-speed.
- The cable impedance must match the impedance of the high-speed and full-speed drivers. The drivers are characterized to drive specific cable impedance. Refer to Section 7.1.1 for details.
- The maximum allowable cable length is determined by signal pair attenuation and propagation delay. Refer to Sections 7.1.14 and 7.1.17 for details.
- Differences in propagation delay between the two signal conductors must be minimized. Refer to Section 7.1.3 for details.
- The GND lead provides a common reference between the upstream and downstream ports. The maximum cable length is determined by the voltage drop across the GND lead. Refer to Section 7.2.2 for details. The minimum wire gauge is calculated using the worst case current consumption.
- The VBUS lead provides power to the connected device. The minimum wire gauge is vendor specific.

6.4.3 Low-speed Captive Cable Assemblies

Assemblies are considered captive if they are provided with a vendor-specific connect means (hardwired or custom detachable) to the peripheral. Low-speed cables may only be used on low-speed devices.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-4 illustrates a low-speed hardwired cable assembly.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-4. USB Low-speed Hardwired Cable Assembly

Low-speed captive cable assemblies must meet the following electrical requirements:

- The cable must be terminated on one end with an overmolded Series “A” plug and the opposite end is vendor specific. If the vendor specific interconnect is to be hot plugged, it must meet the same performance requirements as the USB “B” connector.
- Low-speed drivers are characterized for operation over a range of capacitive loads. This value includes all sources of capacitance on the D+ and D-lines, not just the cable. Cable selection must insure that total load capacitance falls between specified minimum and maximum values. If the desired implementation does not meet the minimum requirement, additional capacitance needs to be added to the device. Refer to Section 7.1.1.2 for details.
- The maximum low-speed cable length is determined by the rise and fall times of low-speed signaling. This forces low-speed cable to be significantly shorter than high-/full-speed. Refer to Section 7.1.1.2 for details.
- Differences in propagation delay between the two signal conductors must be minimized. Refer to Section 7.1.3 for details.
- The GND lead provides a common reference between the upstream and downstream ports. The maximum cable length is determined by the voltage drop across the GND lead. Refer to Section 7.2.2 for details. The minimum wire gauge is calculated using the worst case current consumption.
- The VBUS lead provides power to the connected device. The minimum wire gauge is vendor specific.

6.4.4 Prohibited Cable Assemblies

USB is optimized for ease of use. The expectation is that if the device can be plugged in, it will work. By specification, the only conditions that prevent a USB device from being successfully utilized are lack of power, lack of bandwidth, and excessive topology depth. These conditions are well understood by the system software.

Prohibited cable assemblies may work in some situations, but they cannot be guaranteed to work in all instances.

- **Extension cable assembly**
  A cable assembly that provides a Series “A” plug with a series “A” receptacle or a Series “B” plug with a Series “B” receptacle. This allows multiple cable segments to be connected together, possibly exceeding the maximum permissible cable length.
- **Cable assembly that violates USB topology rules**
  A cable assembly with both ends terminated in either Series “A” plugs or Series “B” receptacles. This allows two downstream ports to be directly connected. Note: This prohibition does not prevent using a USB device to provide a bridge between two USB buses.
- **Standard detachable cables for low-speed devices**
  Low-speed devices are prohibited from using standard detachable cables. A standard detachable cable assembly must be high-/full-speed. Since a standard detachable cable assembly is high-/full-speed rated, using a long high-/full-speed cable exceeds the capacitive load of low-speed.

6.5 Connector Mechanical Configuration and Material Requirements

The USB Icon is used to identify USB plugs and the receptacles. Figure 6-5 illustrates the USB Icon.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

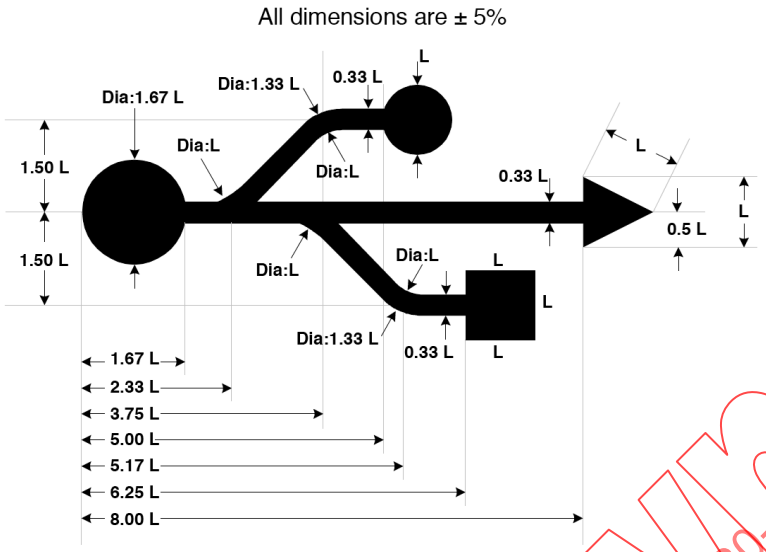


Figure 6-5. USB Icon

6.5.1 USB Icon Location

The USB Icon is embossed, in a recessed area, on the topside of the USB plug. This provides easy user recognition and facilitates alignment during the mating process. The USB Icon and Manufacturer's logo should not project beyond the overmold surface. The USB Icon is required, while the Manufacturer's logo is recommended, for both Series "A" and "B" plug assemblies. The USB Icon is also located adjacent to each receptacle. Receptacles should be oriented to allow the Icon on the plug to be visible during the mating process. Figure 6-6 illustrates the typical plug orientation.

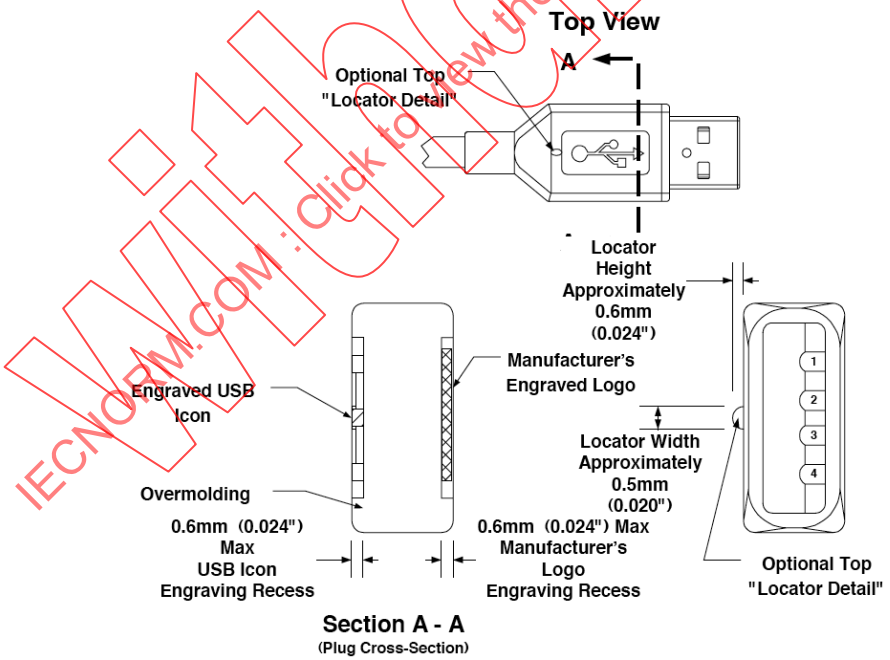


Figure 6-6. Typical USB Plug Orientation

6.5.2 USB Connector Termination Data

Table 6-1 provides the standardized contact terminating assignments by number and electrical value for Series "A" and Series "B" connectors.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

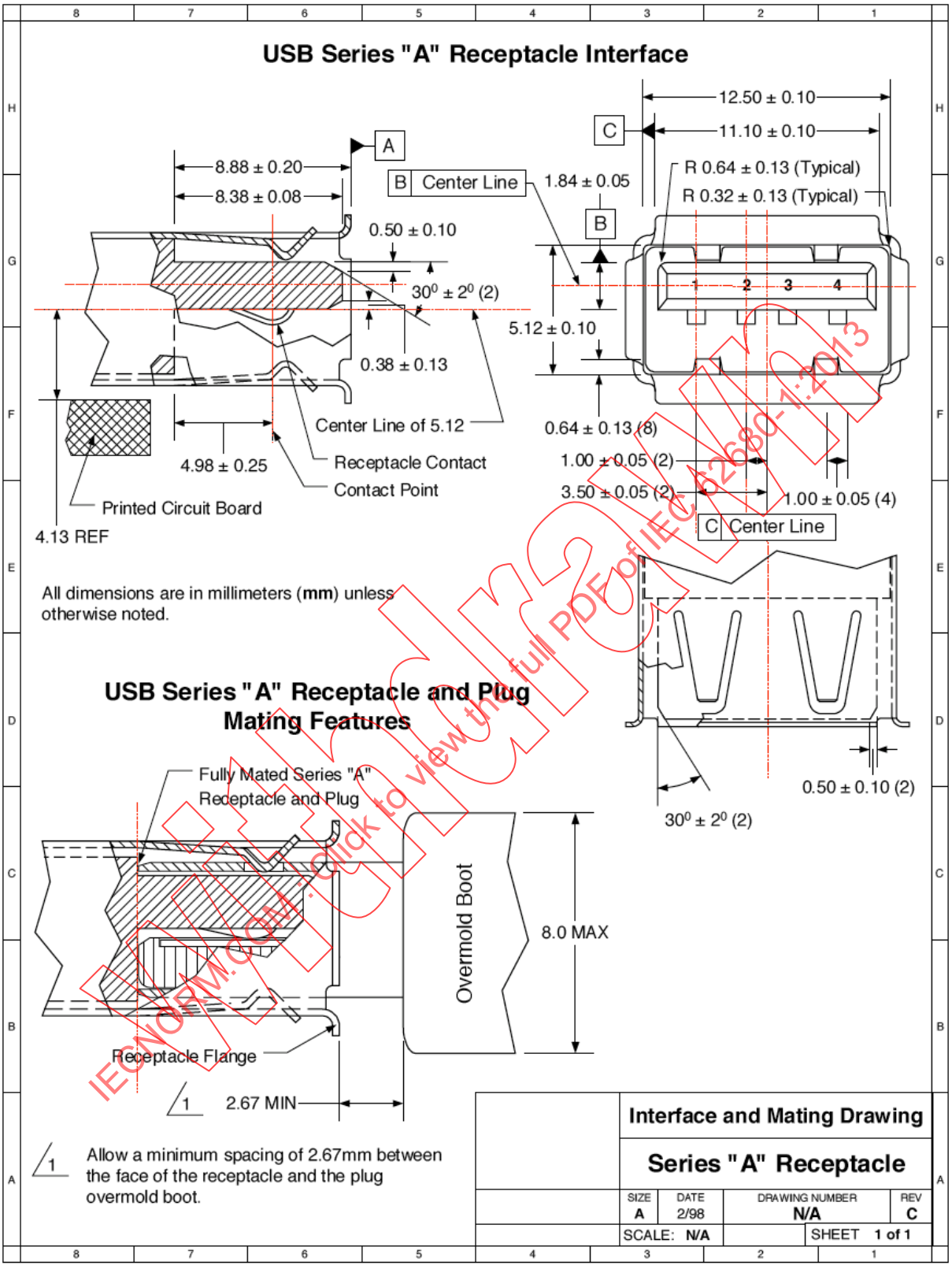
Table 6-1. USB Connector Termination Assignment

Contact Number	Signal Name	Typical Wiring Assignment
1	VBUS	Red
2	D-	White
3	D+	Green
4	GND	Black
Shell	Shield	Drain Wire

6.5.3 Series “A” and Series “B” Receptacles

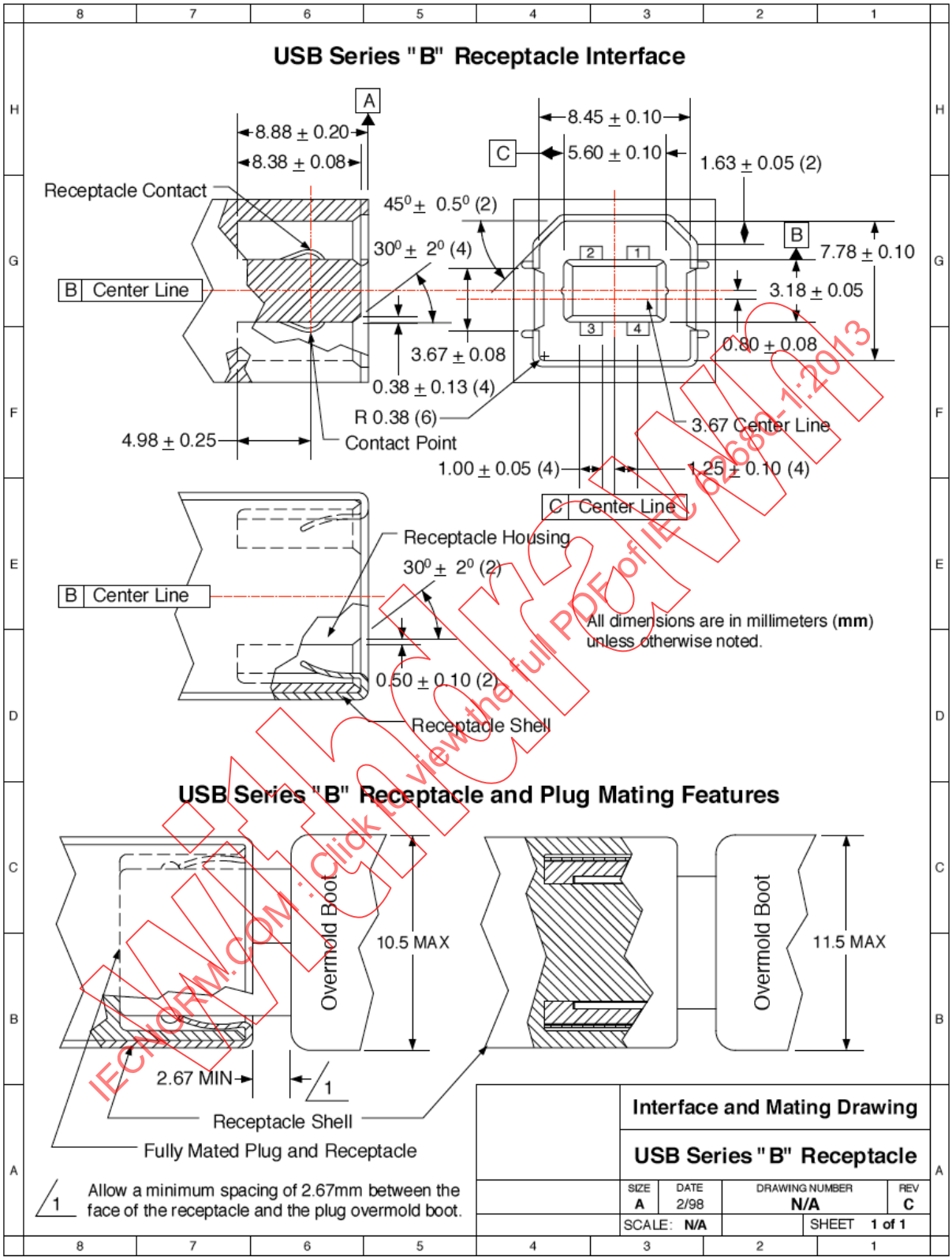
Electrical and mechanical interface configuration data for Series "A" and Series "B" receptacles are shown in Figure 6-7 and Figure 6-8. Also, refer to Figure 6-12, Figure 6-13, and Figure 6-14 at the end of this chapter for typical PCB receptacle layouts.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 6-7. USB Series "A" Receptacle Interface and Mating Drawing



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-8. USB Series "B" Receptacle Interface and Mating Drawing

6.5.3.1 Receptacle Injection Molded Thermoplastic Insulator Material

Minimum UL 94-V0 rated, thirty percent (30%) glass-filled polybutylene terephthalate (PBT) or polyethylene terephthalate (PET) or better.

Typical Colors: Black, gray, and natural.

Flammability Characteristics: UL 94-V0 rated.

Flame Retardant Package must meet or exceed the requirements for UL, CSA, VDE, etc.

Oxygen Index (LOI): Greater than 21%. ASTM D 2863.

6.5.3.2 Receptacle Shell Materials

Substrate Material: 0.30 + 0.05 mm phosphor bronze, nickel silver, or other copper based high strength materials.

Plating:

1. Underplate: Optional. Minimum 1.00 micrometers (40 microinches) nickel. In addition, manufacturer may use a copper underplate beneath the nickel.
2. Outside: Minimum 2.5 micrometers (100 microinches) bright tin or bright tin-lead.

6.5.3.3 Receptacle Contact Materials

Substrate Material: 0.30 ± 0.05 mm minimum half-hard phosphor bronze or other high strength copper based material.

Plating: Contacts are to be selectively plated.

A. Option I
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.70 micrometers (28 microinches) palladium.
   3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

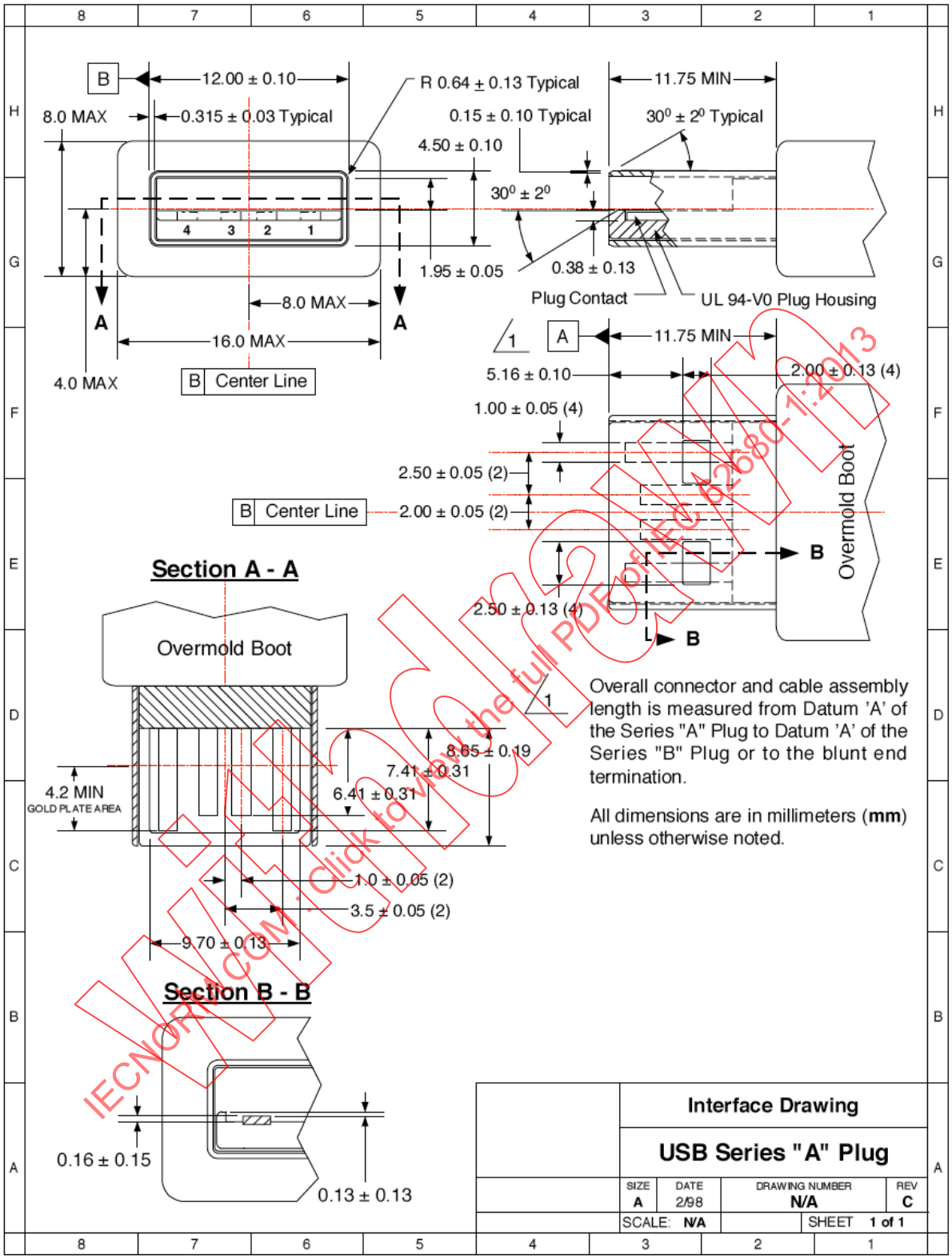
B. Option II
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.75 micrometers (30 microinches) palladium-nickel.
   3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

C. Option III
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.75 micrometers (30 microinches) gold.
   3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

6.5.4 Series “A” and Series “B” Plugs

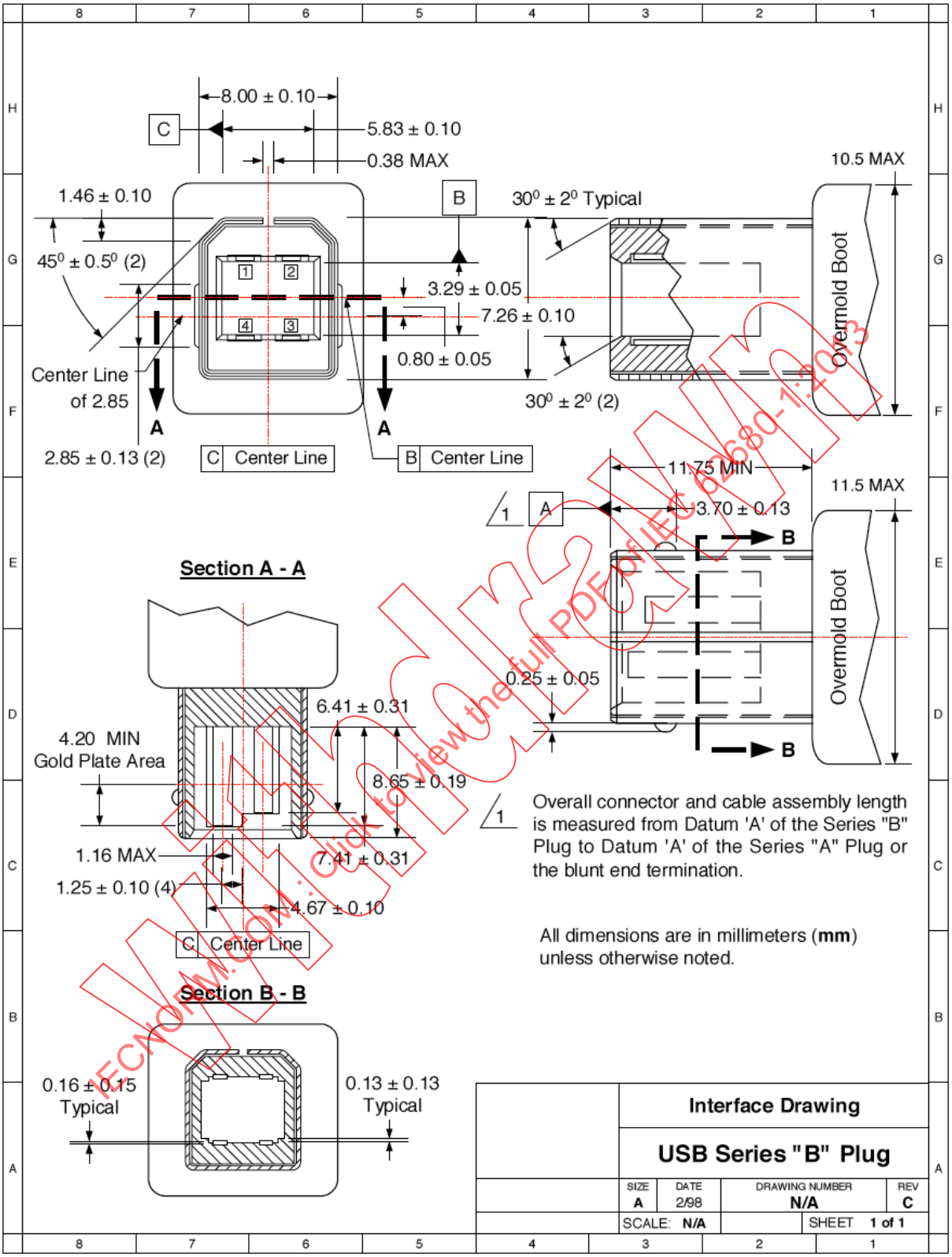
Electrical and mechanical interface configuration data for Series "A" and Series "B" plugs are shown in Figure 6-9 and Figure 6-10.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-9. USB Series "A" Plug Interface Drawing



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-10. USB Series "B" Plug Interface Drawing

6.5.4.1 Plug Injection Molded Thermoplastic Insulator Material

Minimum UL 94-V0 rated, thirty percent (30%) glass-filled polybutylene terephthalate (PBT) or polyethylene terephthalate (PET) or better.

Typical Colors: Black, gray, and natural.

Flammability Characteristics: UL 94-V0 rated.

Flame Retardant Package must meet or exceed the requirements for UL, CSA, and VDE.

Oxygen Index (LOI): 21%. ASTM D 2863.

6.5.4.2 Plug Shell Materials

Substrate Material: 0.30 ± 0.05 mm phosphor bronze, nickel silver, or other suitable material.

Plating:

A. Underplate: Optional. Minimum 1.00 micrometers (40 microinches) nickel. In addition, manufacturer may use a copper underplate beneath the nickel.
B. Outside: Minimum 2.5 micrometers (100 microinches) bright tin or bright tin-lead.

6.5.4.3 Plug (Male) Contact Materials

Substrate Material: 0.30 ± 0.05 mm half-hard phosphor bronze.

Plating: Contacts are to be selectively plated.

A. Option I
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.70 micrometers (28 microinches) palladium.
   3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.
B. Option II
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.75 micrometers (30 microinches) palladium-nickel.
   3. Wire Crimp/Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.
C. Option III
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.75 micrometers (30 microinches) gold.
   3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

6.6 Cable Mechanical Configuration and Material Requirements

High-/full-speed and low-speed cables differ in data conductor arrangement and shielding. Low-speed recommends, but does not require, use of a cable with twisted data conductors. Low speed recommends, but does not require, use of a cable with a braided outer shield. Figure 6-11 shows the typical high-/full-speed cable construction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

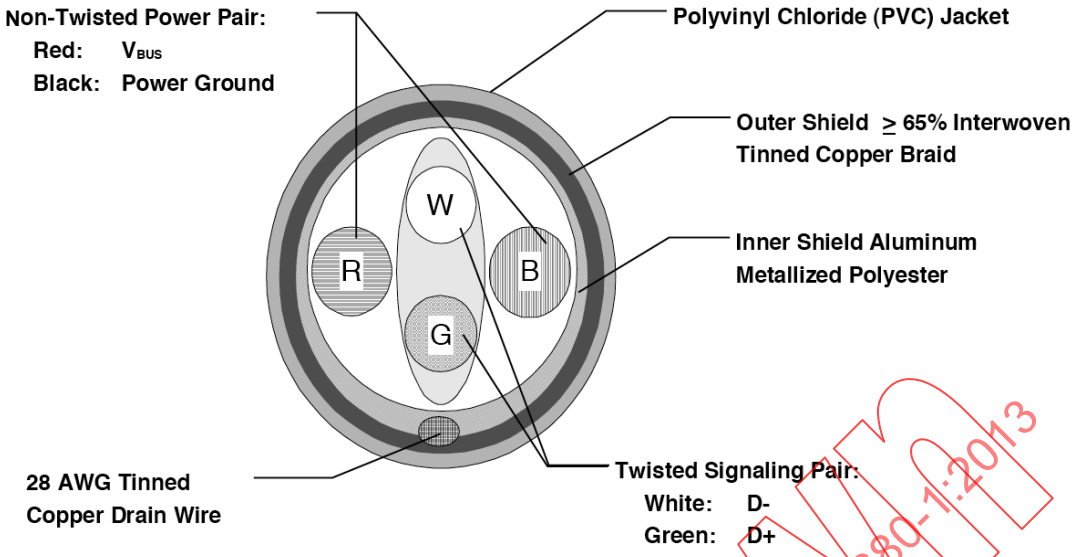


Figure 6-11. Typical High-/full-speed Cable Construction

6.6.1 Description

High-/full-speed cable consists of one 28 to 20 AWG non-twisted power pair and one 28 AWG twisted data pair with an aluminum metallized polyester inner shield, 28 AWG stranded tinned copper drain wire, ≥ 65% tinned copper wire interwoven (braided) outer shield, and PVC outer jacket.

Low-speed cable consists of one 28 to 20 AWG non-twisted power pair and one 28 AWG data pair (a twist is recommended) with an aluminum metallized polyester inner shield, 28 AWG stranded tinned copper drain wire and PVC outer jacket. A ≥ 65% tinned copper wire interwoven (braided) outer shield is recommended.

6.6.2 Construction

Raw materials used in the fabrication of this cable must be of such quality that the fabricated cable is capable of meeting or exceeding the mechanical and electrical performance criteria of the most current USB Specification revision and all applicable domestic and international safety/testing agency requirements; e.g., UL, CSA, BSA, NEC, etc., for electronic signaling and power distribution cables in its category.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 6-2. Power Pair

American Wire Gauge (AWG)	Nominal Conductor Outer Diameter	Stranded Tinned Conductors
28	0.381 mm (0.015”) 0.406 mm (0.016”)	7 x 36 19 x 40
26	0.483 mm (0.019”) 0.508 mm (0.020”)	7 x 34 19 x 38
24	0.610 mm (0.024”) 0.610 mm (0.024”)	7 x 32 19 x 36
22	0.762 mm (0.030”) 0.787 mm (0.031”)	7 x 30 19 x 34
20	0.890 mm (0.035”) 0.931 mm (0.037”)	7 x 28 19 x 32

Note: Minimum conductor construction must be stranded tinned copper.

Non-Twisted Power Pair:

A. Wire Gauge: Minimum 28 AWG or as specified by the user contingent upon the specified cable length. Refer to Table 6-2.

B. Wire Insulation: Semirigid polyvinyl chloride (PVC).

1. Nominal Insulation Wall Thickness: 0.25 mm (0.010”)
2. Typical Power (V_{BUS}) Conductor: Red Insulation
3. Typical Ground Conductor: Black Insulation

Signal Pair:

A. Wire Gauge: 28 AWG minimum. Refer to Table 6-3.

Table 6-3. Signal Pair

American Wire Gauge (AWG)	Nominal Conductor Outer Diameter	Stranded Tinned Conductors
28	0.381 mm (0.015”) 0.406 mm (0.016”)	7 x 36 19 x 40

Note: Minimum conductor construction must be stranded tinned copper.

B. Wire Insulation: High-density polyethylene (HDPE), alternately foamed polyethylene or foamed polypropylene

1. Nominal Insulation Wall Thickness: 0.31 mm (0.012”)
2. Typical Data Plus (+) Conductor: Green Insulation
3. Typical Data Minus (-) Conductor: White Insulation

C. Nominal Twist Ratio (not required for low-speed): One full twist every 60 mm (2.36”) to 80 mm (3.15”)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Aluminum Metallized Polyester Inner Shield (required for low-speed):

A. Substrate Material: Polyethylene terephthalate (PET) or equivalent material

B. Metallizing: Vacuum deposited aluminum

C. Assembly:

1. The aluminum metallized side of the inner shield must be positioned facing out to ensure direct contact with the drain wire.

2. The aluminum metallized inner shield must overlap by approximately one-quarter turn.

Drain Wire (required for low-speed):

A. Wire Gauge: Minimum 28 AWG stranded tinned copper (STC) non-insulated. Refer to Table 6-4.

Table 6-4. Drain Wire Signal Pair

American Wire Gauge (AWG)	Nominal Conductor Outer Diameter	Stranded Tinned Conductors
28	0.381 mm (0.015”) 0.406 mm (0.016”)	7 x 36 19 x 40

Interwoven (Braided) Tinned Copper Wire (ITCW) Outer Shield (recommended but not required for low-speed):

A. Coverage Area: Minimum 65%.

B. Assembly: The interwoven (braided) tinned copper wire outer shield must encase the aluminum metallized PET shielded power and signal pairs and must be in direct contact with the drain wire.

Outer Polyvinyl Chloride (PVC) Jacket:

A. Assembly: The outer PVC jacket must encase the fully shielded power and signal pairs and must be in direct contact with the tinned copper outer shield.

B. Nominal Wall Thickness: 0.64 mm (0.025”).

Marking: The cable must be legibly marked using contrasting color permanent ink.

A. Minimum marking information for high-/full-speed cable must include:
USB SHIELDED <Gauge/2C + Gauge/2C> UL CM 75 °C — UL Vendor ID.

B. Minimum marking information for low-speed cable shall include:
USB specific marking is not required for low-speed cable.

Nominal Fabricated Cable Outer Diameter:

This is a nominal value and may vary slightly from manufacturer to manufacturer as a function of the conductor insulating materials and conductor specified. Refer to Table 6-5.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 6-5. Nominal Cable Diameter

Shielded USB Cable Configuration	Nominal Outer Cable Diameter
28/28	4.06 mm (0.160”)
28/26	4.32 mm (0.170”)
28/24	4.57 mm (0.180”)
28/22	4.83 mm (0.190”)
28/20	5.21 mm (0.205”)

6.6.3 Electrical Characteristics

All electrical characteristics must be measured at or referenced to +20 °C (68 °F).

Voltage Rating: 30 V rms maximum.

Conductor Resistance: Conductor resistance must be measured in accordance with ASTM-D-4566 Section 13. Refer to Table 6-6.

Conductor Resistance Unbalance (Pairs): Conductor resistance unbalance between two (2) conductors of any pair must not exceed five percent (5%) when measured in accordance with ASTM-D-4566 Section 15.

The DC resistance from plug shell to plug shell (or end of integrated cable) must be less than 0.6 ohms.

Table 6-6. Conductor Resistance

American Wire Gauge (AWG)	Ohms (Ω) / 100 Meters Maximum
28	23.20
26	14.60
24	9.09
22	5.74
20	3.58

6.6.4 Cable Environmental Characteristics

Temperature Range:

A. Operating Temperature Range: 0 °C to +50 °C
B. Storage Temperature Range: -20 °C to +60 °C
C. Nominal Temperature Rating: +20 °C

Flammability: All plastic materials used in the fabrication of this product shall meet or exceed the requirements of NEC Article 800 for communications cables Type CM (Commercial).

6.6.5 Listing

The product shall be UL listed per UL Subject 444, Class 2, Type CM for Communications Cable Requirements.

6.7 Electrical, Mechanical, and Environmental Compliance Standards

Table 6-7 lists the minimum test criteria for all USB cable, cable assemblies, and connectors.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 6-7. USB Electrical, Mechanical, and Environmental Compliance Standards

Test Description	Test Procedure	Performance Requirement
Visual and Dimensional Inspection	EIA 364-18 Visual, dimensional, and functional inspection in accordance with the USB quality inspection plans.	Must meet or exceed the requirements specified by the most current version of Chapter 6 of the USB Specification.
Insulation Resistance	EIA 364-21 The object of this test procedure is to detail a standard method to assess the insulation resistance of USB connectors. This test procedure is used to determine the resistance offered by the insulation materials and the various seals of a connector to a DC potential tending to produce a leakage of current through or on the surface of these members.	1,000 MΩ minimum.
Dielectric Withstanding Voltage	EIA 364-20 The object of this test procedure is to detail a test method to prove that a USB connector can operate safely at its rated voltage and withstand momentary over-potentials due to switching, surges, and/or other similar phenomena.	The dielectric must withstand 500 V AC for one minute at sea level.
Low Level Contact Resistance	EIA 364-23 The object of this test is to detail a standard method to measure the electrical resistance across a pair of mated contacts such that the insulating films, if present, will not be broken or asperity melting will not occur.	30 mΩ maximum when measured at 20 mV maximum open circuit at 100 mA. Mated test contacts must be in a connector housing.
Contact Current Rating	EIA 364-70 — Method B The object of this test procedure is to detail a standard method to assess the current carrying capacity of mated USB connector contacts.	1.5 A at 250 V AC minimum when measured at an ambient temperature of 25 ˚C. With power applied to the contacts, the Δ T must not exceed +30 ˚C at any point in the USB connector under test.
Contact Capacitance	EIA 364-30 The object of this test is to detail a standard method to determine the capacitance between conductive elements of a USB connector.	2 pF maximum unmated per contact.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 6-7. USB Electrical, Mechanical, and Environmental Compliance Standards (Continued)

Test Description	Test Procedure	Performance Requirement
Insertion Force	EIA 364-13 The object of this test is to detail a standard method for determining the mechanical forces required for inserting a USB connector.	35 Newtons maximum at a maximum rate of 12.5 mm (0.492”) per minute.
Extraction Force	EIA 364-13 The object of this test is to detail a standard method for determining the mechanical forces required for extracting a USB connector.	10 Newtons minimum at a maximum rate of 12.5 mm (0.492”) per minute.
Durability	EIA 364-09 The object of this test procedure is to detail a uniform test method for determining the effects caused by subjecting a USB connector to the conditioning action of insertion and extraction, simulating the expected life of the connectors. Durability cycling with a gauge is intended only to produce mechanical stress. Durability performed with mating components is intended to produce both mechanical and wear stress.	1,500 insertion/extraction cycles at a maximum rate of 200 cycles per hour.
Cable Pull-Out	EIA 364-38 Test Condition A The object of this test procedure is to detail a standard method for determining the holding effect of a USB plug cable clamp without causing any detrimental effects upon the cable or connector components when the cable is subjected to inadvertent axial tensile loads.	After the application of a steady state axial load of 40 Newtons for one minute.
Physical Shock	EIA 364-27 Test Condition H The object of this test procedure is to detail a standard method to assess the ability of a USB connector to withstand specified severity of mechanical shock.	No discontinuities of 1 μs or longer duration when mated USB connectors are subjected to 11 ms duration 30 Gs half-sine shock pulses. Three shocks in each direction applied along three mutually perpendicular planes for a total of 18 shocks.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 6-7. USB Electrical, Mechanical, and Environmental Compliance Standards (Continued)

Test Description	Test Procedure	Performance Requirement
Random Vibration	EIA 364-28 Test Condition V Test Letter A This test procedure is applicable to USB connectors that may, in service, be subjected to conditions involving vibration. Whether a USB connector has to function during vibration or merely to survive conditions of vibration should be clearly stated by the detailed product specification. In either case, the relevant specification should always prescribe the acceptable performance tolerances.	No discontinuities of 1 μs or longer duration when mated USB connectors are subjected to 5.35 Gs RMS. 15 minutes in each of three mutually perpendicular planes.
Thermal Shock	EIA 364-32 Test Condition I The object of this test is to determine the resistance of a USB connector to exposure at extremes of high and low temperatures and to the shock of alternate exposures to these extremes, simulating the worst case conditions for storage, transportation, and application.	10 cycles –55 °C and +85 °C. The USB connectors under test must be mated.
Humidity Life	EIA 364-31 Test Condition A Method III The object of this test procedure is to detail a standard test method for the evaluation of the properties of materials used in USB connectors as they are influenced by the effects of high humidity and heat.	168 hours minimum (seven complete cycles). The USB connectors under test must be tested in accordance with EIA 364-31.
Solderability	EIA 364-52 The object of this test procedure is to detail a uniform test method for determining USB connector solderability. The test procedure contained herein utilizes the solder dip technique. It is not intended to test or evaluate solder cup, solder eyelet, other hand-soldered type, or SMT type terminations.	USB contact solder tails must pass 95% coverage after one hour steam aging as specified in Category 2.
Flammability	UL 94 V-0 This procedure is to ensure thermoplastic resin compliance to UL flammability standards.	The manufacturer will require its thermoplastic resin vendor to supply a detailed C of C with each resin shipment. The C of C shall clearly show the resin's UL listing number, lot number, date code, etc.
Flammability	UL 94 V-0 This procedure is to ensure thermoplastic resin compliance to UL flammability standards.	The manufacturer will require its thermoplastic resin vendor to supply a detailed C of C with each resin shipment. The C of C shall clearly show the resin's UL listing number, lot number, date code, etc.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 6-7. USB Electrical, Mechanical, and Environmental Compliance Standards (Continued)

Test Description	Test Procedure	Performance Requirement
Cable Impedance (Only required for high-/full-speed)	The object of this test is to insure the signal conductors have the proper impedance. Connect the Time Domain Reflectometer (TDR) outputs to the impedance/delay/skew test fixture (Note 1). Use separate 50 Ω cables for the plus (or true) and minus (or complement) outputs. Set the TDR head to differential TDR mode. Connect the Series "A" plug of the cable to be tested to the text fixture, leaving the other end open-circuited. Define a waveform composed of the difference between the true and complement waveforms, to allow measurement of differential impedance. Measure the minimum and maximum impedances found between the connector and the open circuited far end of the cable.	Impedance must be in the range specified in Table 7-9 (ZO).
Signal Pair Attenuation (Only required for high-/full-speed)	The object of this test is to insure that adequate signal strength is presented to the receiver to maintain a low error rate. 1. Connect the Network Analyzer output port (port 1) to the input connector on the attenuation test fixture (Note 2). 2. Connect the Series “A” plug of the cable to be tested to the test fixture, leaving the other end open-circuited. 3. Calibrate the network analyzer and fixture using the appropriate calibration standards over the desired frequency range. 4. Follow the method listed in Hewlett Packard Application Note 380-2 to measure the open-ended response of the cable. 5. Short circuit the Series “B” end (or bare leads end, if a captive cable) and measure the short-circuit response. 6. Using the software in H-P App. Note 380-2 or equivalent, calculate the cable attenuation accounting for resonance effects in the cable as needed.	Refer to Section 7.1.17 for frequency range and allowable attenuation.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 6-7. USB Electrical, Mechanical, and Environmental Compliance Standards (Continued)

Test Description	Test Procedure	Performance Requirement
Propagation Delay	The purpose of the test is to verify the end to end propagation of the cable. 1. Connect one output of the TDR sampling head to the D+ and D- inputs of the impedance/delay/skew test fixture (Note 1). Use one 50 Ω cable for each signal and set the TDR head to differential TDR mode. 2. Connect the cable to be tested to the test fixture. If detachable, plug both connectors in to the matching fixture connectors. If captive, plug the series “A” plug into the matching fixture connector and solder the stripped leads on the other end to the test fixture. 3. Measure the propagation delay of the test fixture by connecting a short piece of wire across the fixture from input to output and recording the delay. 4. Remove the short piece of wire and remeasure the propagation delay. Subtract from it the delay of the test fixture measured in the previous step.	High-/full-speed. See Section 7.1.1.1, Section 7.1.4, Section 7.1.16, and Table 7-9 (TFSCBL). Low-speed. See Section 7.1.1.2, Section 7.1.16, and Table 7-9 (TLSCBL).
Propagation Delay Skew	This test insures that the signal on both the D+ and D- lines arrive at the receiver at the same time. 1. Connect the TDR to the fixture with test sample cable, as in the previous section. 2. Measure the difference in delay for the two conductors in the test cable. Use the TDR cursors to find the open-circuited end of each conductor (where the impedance goes infinite) and subtract the time difference between the two values.	Propagation skew must meet the requirements as listed in Section 7.1.3.
Capacitive Load Only required for low-speed	The purpose of this test is to insure the distributed inter-wire capacitance is less than the lumped capacitance specified by the low-speed transmit driver. 1. Connect the one lead of the Impedance Analyzer to the D+ pin on the impedance/delay/skew fixture (Note 1) and the other lead to the D- pin. 2. Connect the series "A" plug to the fixture, with the series “B” end leads open-circuited. 3. Set the Impedance Analyzer to a frequency of 100 kHz, to measure the capacitance.	See Section 7.1.1.2 and Table 7-7 (CLINUA).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Note1: Impedance, propagation delay, and skew test fixture
This fixture will be used with the TDR for measuring the time domain performance of the cable under test. The fixture impedance should be matched to the equipment, typically 50 Ω. Coaxial connectors should be provided on the fixture for connection from the TDR.

Note 2: Attenuation text fixture
This fixture provides a means of connection from the network analyzer to the Series "A" plug. Since USB signals are differential in nature and operate over balanced cable, a transformer or balun (North Hills NH13734 or equivalent) is ideally used. The transformer converts the unbalanced (also known as single-ended) signal from the signal generator which is typically a 50 Ω output to the balanced (also known as differential) and likely different impedance loaded presented by the cable. A second transformer or balun should be used on the other end of the cable under test to convert the signal back to unbalanced form of the correct impedance to match the network analyzer.

6.7.1 Applicable Documents

American National Standard/Electronic Industries Association

ANSI/EIA-364-C (12/94)	Electrical Connector/Socket Test Procedures Including Environmental Classifications

American Standard Test Materials

ASTM-D-4565	Physical and Environmental Performance Properties of Insulation and Jacket for Telecommunication Wire and Cable, Test Standard Method
ASTM-D-4566	Electrical Performance Properties of Insulation and Jacket for Telecommunication Wire and Cable, Test Standard Method

Underwriters' Laboratory, Inc.

UL STD-94	Test for Flammability of Plastic materials for Parts in Devices and Appliances
UL Subject-444	Communication Cables

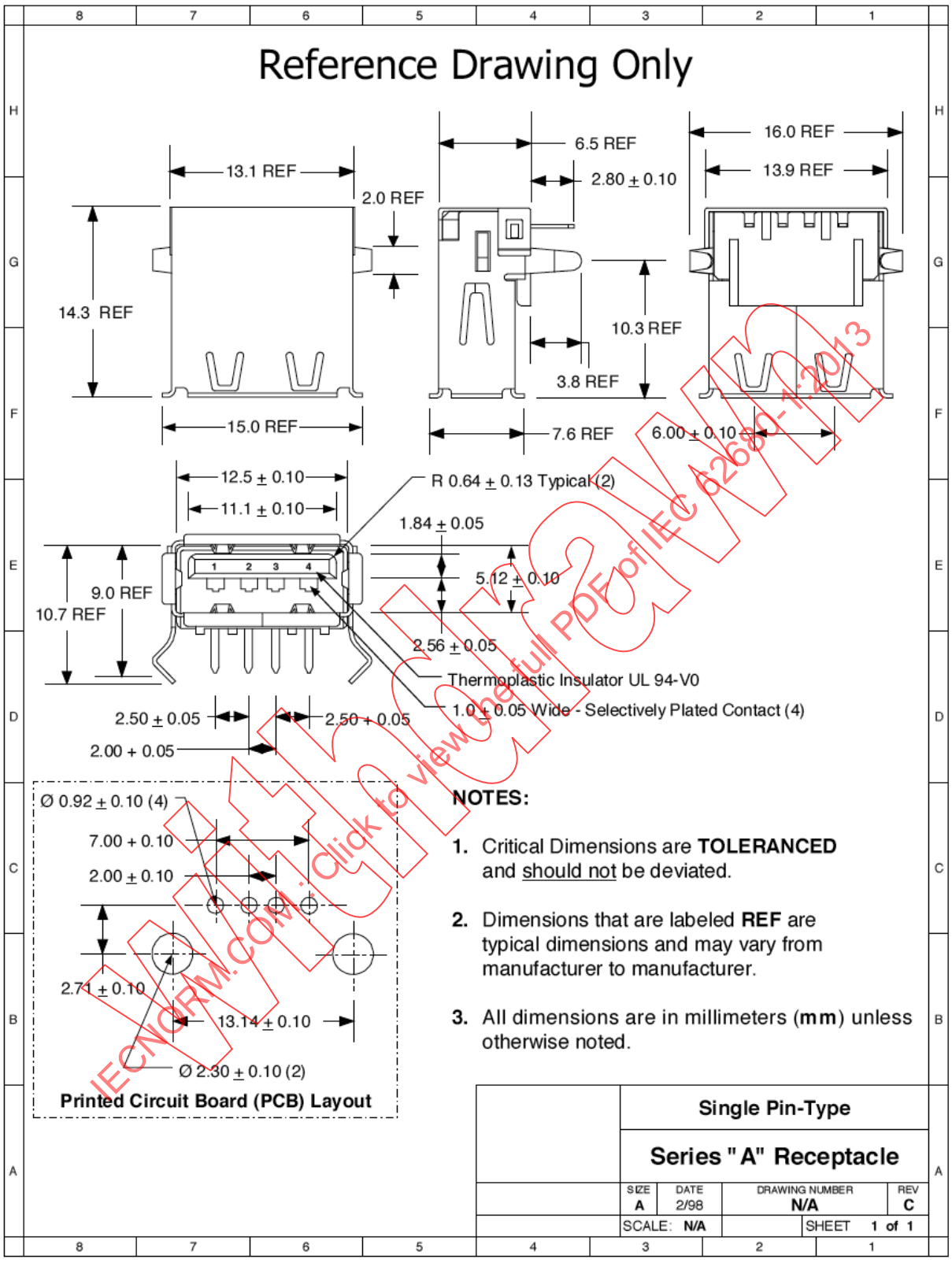
6.8 USB Grounding

The shield must be terminated to the connector plug for completed assemblies. The shield and chassis are bonded together. The user selected grounding scheme for USB devices, and cables must be consistent with accepted industry practices and regulatory agency standards for safety and EMI/ESD/RFI.

6.9 PCB Reference Drawings

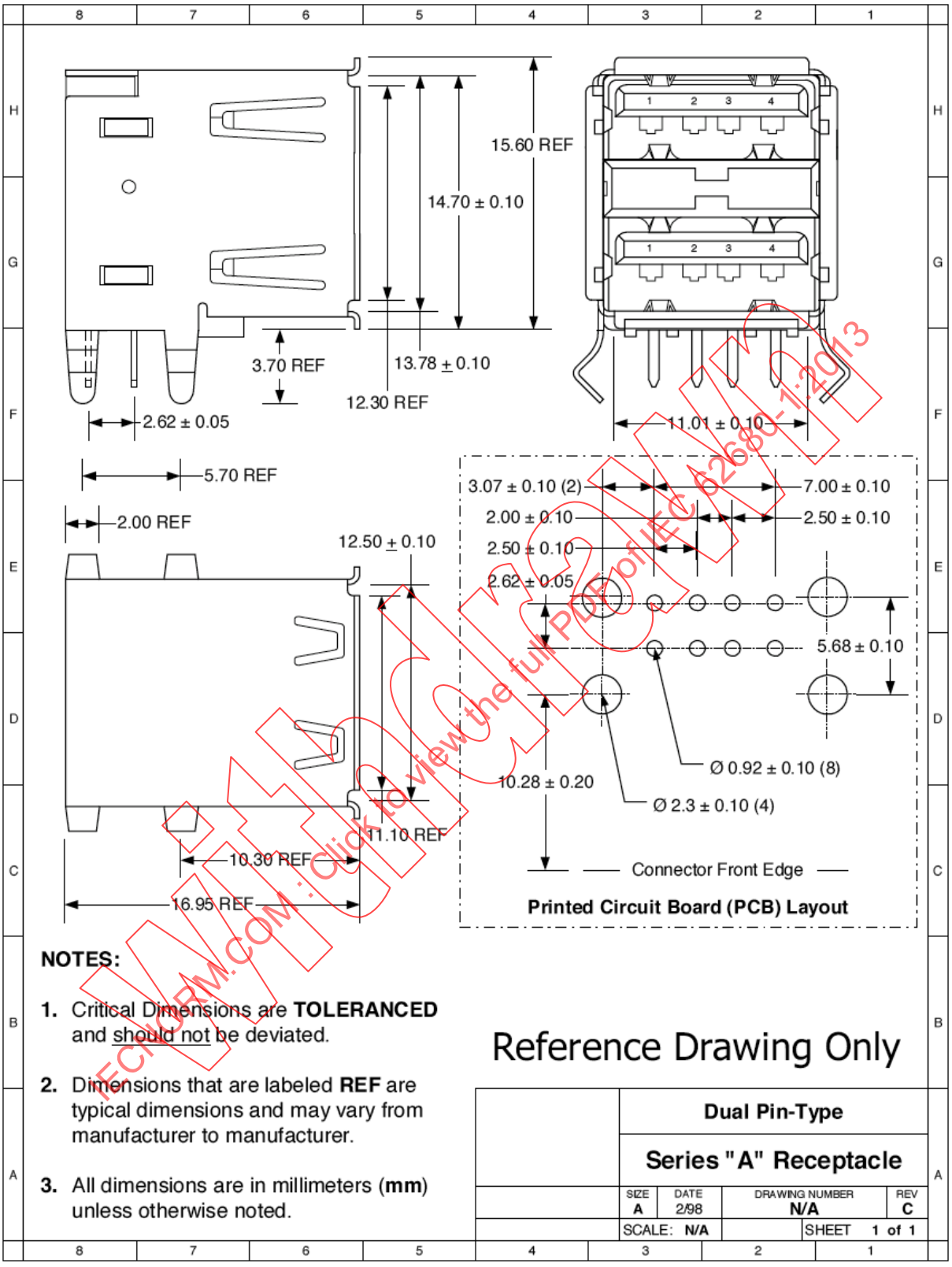
The drawings in Figure 6-12, Figure 6-13, and Figure 6-14 describe typical receptacle PCB interfaces. These drawings are included for informational purposes only.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



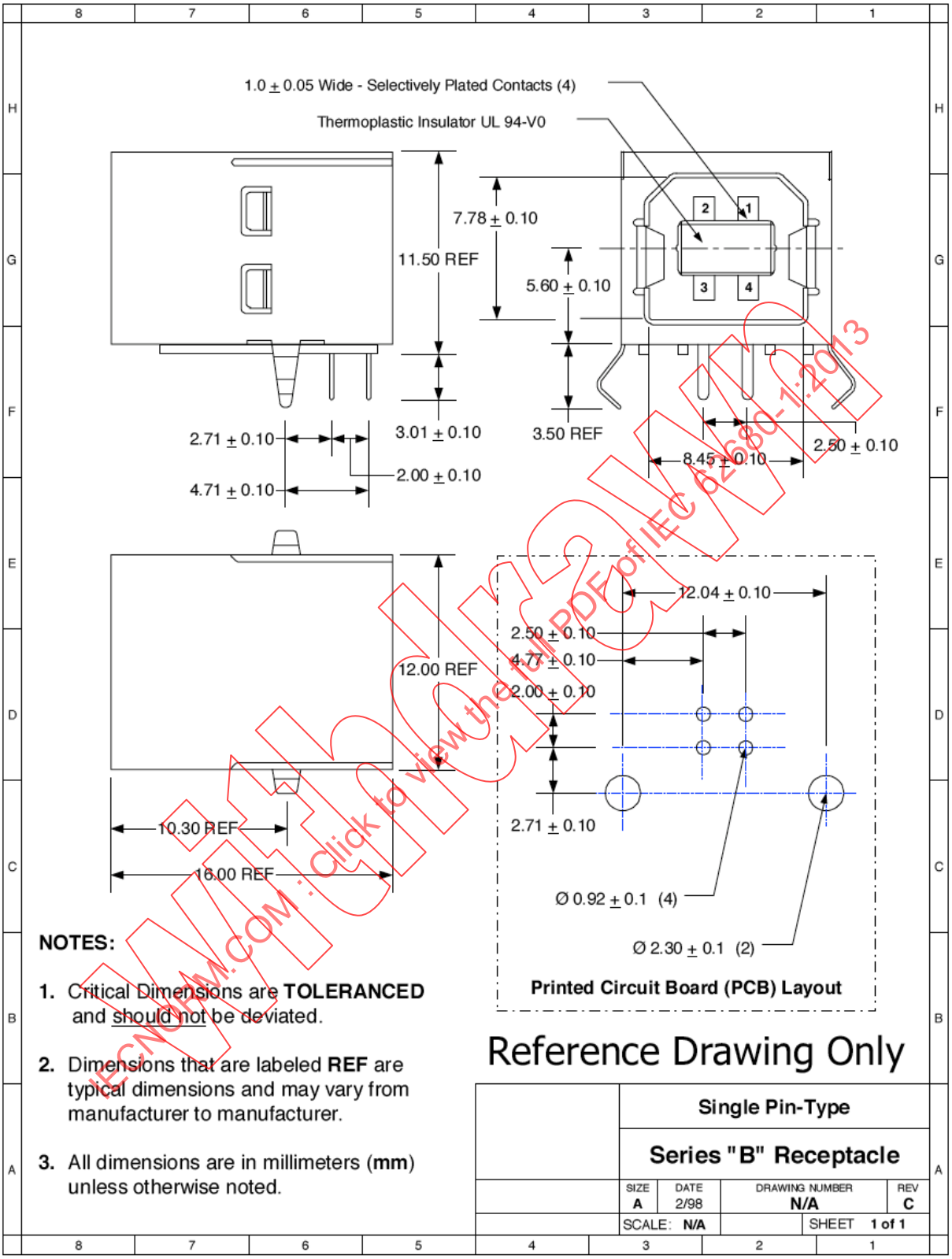
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-12. Single Pin-type Series "A" Receptacle



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-13. Dual Pin-type Series "A" Receptacle



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-14. Single Pin-type Series "B" Receptacle

7 Chapter 7 Electrical

This chapter describes the electrical specification for the USB. It contains signaling, power distribution, and physical layer specifications. This specification does not address regulatory compliance. It is the responsibility of product designers to make sure that their designs comply with all applicable regulatory requirements.

The USB 2.0 specification requires hubs to support high-speed mode. USB 2.0 devices are not required to support high-speed mode. A high-speed capable upstream facing transceiver must not support low-speed signaling mode. A USB 2.0 downstream facing transceiver must support high-speed, full-speed, and low-speed modes.

To assure reliable operation at high-speed data rates, this specification requires the use of cables that conform to all current cable specifications.

In this chapter, there are numerous references to strings of J's and K's, or to strings of 1's and 0's. In each of these instances, the leftmost symbol is transmitted/received first, and the rightmost is transmitted/received last.

7.1 Signaling

The signaling specification for the USB is described in the following subsections.

Overview of High-speed Signaling

A high-speed USB connection is made through a shielded, twisted pair cable that conforms to all current USB cable specifications.

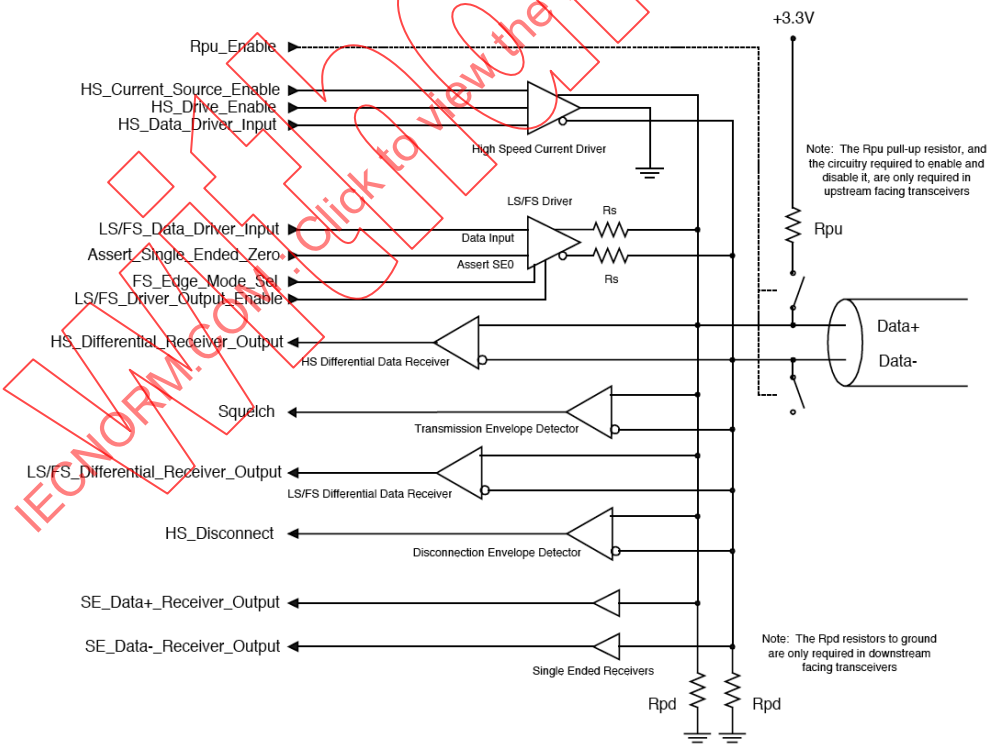


Figure 7-1. Example High-speed Capable Transceiver Circuit

Figure 7-1 depicts an example implementation which largely utilizes USB 1.1 transceiver elements and adds the new elements required for high-speed operation.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

High-speed operation supports signaling at 480 Mb/s. To achieve reliable signaling at this rate, the cable is terminated at each end with a resistance from each wire to ground. The value of this resistance (on each wire) is nominally set to 1/2 the specified differential impedance of the cable, or 45 Ω. This presents a differential termination of 90 Ω.

For a link operating in high-speed mode, the high-speed idle state occurs when the transceivers at both ends of the cable present high-speed terminations to ground, and when neither transceiver drives signaling current into the D+ or D- lines. This state is achieved by using the low-/full-speed driver to assert a single ended zero, and to closely control the combined total of the intrinsic driver output impedance and the Rs resistance (to 45 Ω, nominal). The recommended practice is to make the intrinsic driver impedance as low as possible, and to let Rs contribute as much of the 45 Ω as possible. This will generally lead to the best termination accuracy with the least parasitic loading.

In order to transmit in high-speed mode, a transceiver activates an internal current source which is derived from its positive supply voltage and directs this current into one of the two data lines via a high speed current steering switch. In this way, the transceiver generates the high-speed J or K state on the cable.

The dynamic switching of this current into the D+ or D- line follows the same NRZI data encoding scheme used in low-speed or full-speed operation and also in the bit stuffing behavior. To signal a J, the current is directed into the D+ line, and to signal a K, the current is directed into the D- line. The SYNC field and the EOP delimiters have been modified for high-speed mode.

The magnitude of the current source and the value of the termination resistors are controlled to specified tolerances, and together they determine the actual voltage drive levels. The DC resistance from D+ or D- to the device ground is required to be 45 Ω ±10% when measured without a load, and the differential output voltage measured across the lines (in either the J or K state) must be ±400 mV ±10% when D+ and D- are terminated with precision 45 Ω resistors to ground.

The differential voltage developed across the lines is used for three purposes:

- A differential receiver at the receiving end of the cable receives the differential data signal.
- A differential envelope detector at the receiving end of the cable determines when the link is in the Squelch state. A receiver uses squelch detection as indication that the signal at its connector is not valid.
- In the case of a downstream facing hub transceiver, a differential envelope detector monitors whether the signal at its connector is in the high-speed state. A downstream facing transceiver operating in high-speed mode is required to test for this state at a particular point in time when it is transmitting a SOF packet, as described in Section 7.1.7.3. This is used to detect device disconnection. In the absence of the far end terminations, the differential voltage will nominally double (as compared to when a high-speed device is present) when a high-speed J or K are continuously driven for a period exceeding the round-trip delay for the cable and board-traces between the two transceivers.

USB 2.0 requires that a downstream facing transceiver must be able to operate in low-speed, full-speed, and high-speed signaling modes. An upstream facing high-speed capable transceiver must not operate in low-speed signaling mode, but must be able to operate in full-speed signaling mode. Therefore, a 1.5 kΩ pull-up on the D- line is not allowed for a high-speed capable device, since a high-speed capable transceiver must never signal low-speed operation to the hub port to which it is attached.

Table 7-1 describes the required functional elements of a high-speed capable transceiver, using the diagram shown in Figure 7-1 as an example.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 7-1. Description of Functional Elements in the Example Shown in Figure 7-1

Element	Description
Low-/full-speed Driver	The low-/full-speed driver is used for low-speed and full-speed transmission. It is required to meet all specifications called out in USB 1.1 for low-speed and full-speed operation, with one exception. The exception is that in high-speed capable transceivers, the impedance of each output, including the contribution of RS, must be 45 Ω ±10%. The line terminations for high-speed operation are created by having this driver drive D+ and D- to ground. (This is equivalent to driving SE0 in the full-speed or low-speed mode.) Because of the output impedance requirement described above, this provides a well-controlled high-speed termination on each data line to ground. This is equivalent to a 90 Ω differential termination.
Low-/full-speed Differential Receiver	The low-/full-speed differential receiver is used for receiving low-speed and full-speed data.
Single Ended Receivers	The single ended receivers are used for low-speed and full-speed signaling.
High-speed Current Driver	The high-speed current driver is used for high-speed data transmission. A current source derived from a positive supply is switched into either the D+ or D- lines to signal a J or a K, respectively. The nominal value of the current source is 17.78 mA. When this current is applied to a data line with a 45 Ω termination to ground at each end, the nominal high level voltage (VHSOH) is +400 mV. The nominal differential high-speed voltage (D+ - D-) is thus 400 mV for a J and -400 mV for a K. The current source must comply with the Transmit Eye Pattern Templates specified in Section 7.1.2.2, starting with the first symbol of a packet. One means of achieving this is to leave the current source on continuously when a transceiver is operating in high-speed mode. If this approach is used, the current can be directed to the port ground when the transceiver is not transmitting (the example design in Figure 7-1 shows a control line called HS_Current_Source_Enable to turn the current on, and another called HS_Drive_Enable to direct the current into the data lines.) The penalty of this approach is the 17.78 mA of standing current for every such enabled transceiver in the system. The preferred design is to fully turn the current source off when the transceiver is not transmitting.
High-speed Differential Data Receiver	The high-speed differential data receiver is used to receive high-speed data. It is left to transceiver designers to choose between incorporating separate high-speed and low-/full-speed receivers, as shown in Figure 7-1, or combining both functions into a single receiver.
Transmission Envelope Detector	This envelope detector is used to indicate that data is invalid when the amplitude of the differential signal at a receiver's inputs falls below the squelch threshold (VHSSQ). It must indicate Squelch when the signal drops below 100 mV differential amplitude, and it must indicate that the line is not in the Squelch state when the signal exceeds 150 mV differential amplitude. The response time of the detector must be fast enough to allow a receiver to detect data transmission, to achieve DLL lock, and to detect the end of the SYNC field within 12 bit times, the minimum number of SYNC bits that a receiver is guaranteed to see. This envelope detector must incorporate a filtering mechanism that prevents indication of squelch during the longest differential data transitions allowed by the receiver eye pattern specifications.
Disconnection Envelope Detector	This envelope detector is required in downstream facing ports to detect the high-speed Disconnect state on the line (VHSDSC). Disconnection must be indicated when the amplitude of the differential signal at the downstream facing driver's connector ≥625 mV, and it must not be indicated when the signal amplitude is ≤525 mV. The output of this detector is sampled at a specific time during the transmission of the high-speed SOF EOP, as described in Section 7.1.7.3.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 7-1. Description of Functional Elements in the Example Shown in Figure 7-1 (Continued)

Element	Description
Pull-up Resistor (RPU)	This resistor is required only in upstream facing transceivers and is used to indicate signaling speed capability. A high-speed capable device is required to initially attach as a full-speed device and must transition to high-speed as described in this specification. Once operating in high-speed, the 1.5 kΩ resistor must be electrically removed from the circuit. In Figure 7-1, a control line called RPU_Enable is indicated for this purpose. The preferred embodiment is to attach matched switching devices to both the D+ and D- lines so as to keep the lines' parasitic loading balanced, even though a pull-up resistor must never be used on the D- line of an upstream facing high-speed capable transceiver. When connected, this pull-up must meet all the specifications called out for full-speed operation.
Pull-down Resistors (RPD)	These resistors are required only in downstream facing transceivers and must conform to the same specifications called out for low-speed and full-speed operation.

7.1.1 USB Driver Characteristics

The USB uses a differential output driver to drive the USB data signal onto the USB cable.

For low-speed and full-speed operation, the static output swing of the driver in its low state must be below VOL (max) of 0.3 V with a 1.5 kΩ load to 3.6 V, and in its high state must be above the VOH (min) of 2.8 V with a 15 kΩ load to ground as listed in Table 7-7. Full-speed drivers have more stringent requirements, as described in Section 7.1.1.1. The output swings between the differential high and low state must be well-balanced to minimize signal skew. Slew rate control on the driver is required to minimize the radiated noise and cross talk. The driver's outputs must support three-state operation to achieve bi-directional half-duplex operation.

Low-speed and full-speed USB drivers must never "intentionally" generate an SE1 on the bus. SE1 is a state in which both the D+ and D- lines are at a voltage above VOSE1 (min), which is 0.8 V.

High-speed drivers use substantially different signaling levels, as described in Section 7.1.1.3.

USB ports must be capable of withstanding continuous exposure to the waveforms shown in Figure 7-2 while in any drive state. These waveforms are applied directly into each USB data pin from a voltage source with an output impedance of 39 Ω. The open-circuit voltage of the source shown in Figure 7-2 is based on the expected worst-case overshoot and undershoot.

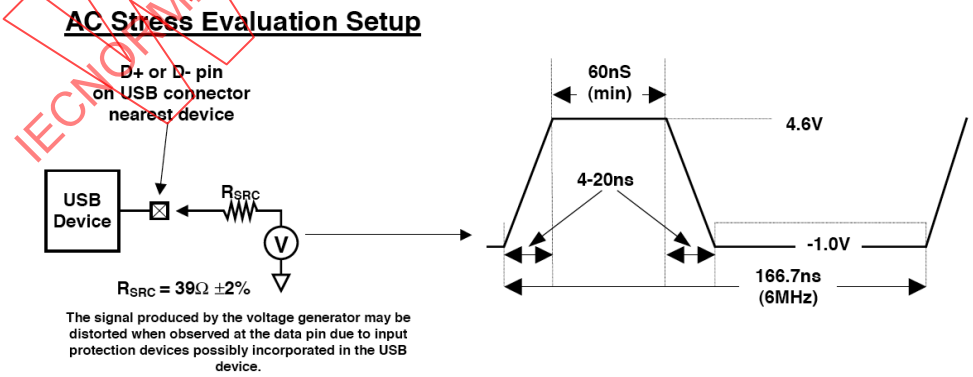


Figure 7-2. Maximum Input Waveforms for USB Signaling

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Short Circuit Withstand

A USB transceiver is required to withstand a continuous short circuit of D+ and/or D- to V_{BUS}, GND, other data line, or the cable shield at the connector, for a minimum of 24 hours without degradation. It is recommended that transceivers be designed so as to withstand such short circuits indefinitely. The device must not be damaged under this short circuit condition when transmitting 50% of the time and receiving 50% of the time (in all supported speeds). The transmit phase consists of a symmetrical signal that toggles between drive high and drive low. This requirement must be met for max value of V_{BUS} (5.25 V).

It is recommended that these AC and short circuit stresses be used as qualification criteria against which the long-term reliability of each device is evaluated.

7.1.1.1 Full-speed (12 Mb/s) Driver Characteristics

A full-speed USB connection is made through a shielded, twisted pair cable with a differential characteristic impedance (Z_0) of 90 Ω ±15%, a common mode impedance (Z_{CM}) of 30 Ω ±30%, and a maximum one-way delay (T_{FSCBL}) of 26 ns. When the full-speed driver is not part of a high-speed capable transceiver, the impedance of each of the drivers (Z_{DRV}) must be between 28 Ω and 44 Ω, i.e., within the gray area in Figure 7-4. When the full-speed driver is part of a high-speed capable transceiver, the impedance of each of the drivers (Z_{HSDRV}) must be between 40.5 Ω and 49.5 Ω, i.e., within the gray area in Figure 7-5.

For a CMOS implementation, the driver impedance will typically be realized by a CMOS driver with an impedance significantly less than this resistance with a discrete series resistor making up the balance as shown in Figure 7-3. The series resistor Rs is included in the buffer impedance requirement shown in Figure 7-4 and Figure 7-5. In the rest of the chapter, references to the buffer assume a buffer with the series impedance unless stated otherwise.

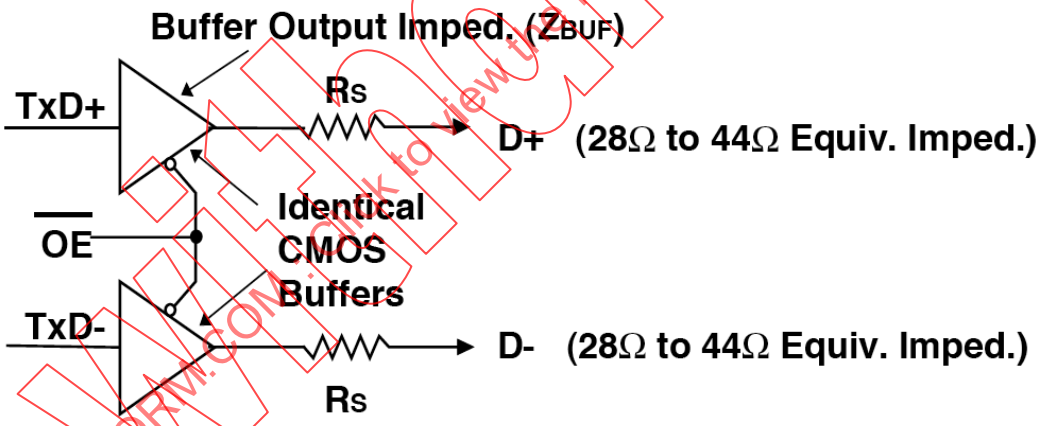


Figure 7-3. Example Full-speed CMOS Driver Circuit (non High-speed capable)

Full-speed Buffers in Transceivers Which are Not High-speed Capable

The buffer impedance must be measured for driving high as well as driving low. Figure 7-4 shows the composite V/I characteristics for the full-speed drivers with included series damping resistor (Rs). The characteristics are normalized to the steady-state, unloaded output swing of the driver. The normalized driver characteristics are found by dividing the measured voltages and currents by the actual swing of the driver under test. The normalized V/I curve for the driver must fall entirely inside the shaded region. The V/I region is bounded by the minimum driver impedance above and the maximum driver impedance below. The minimum drive region is intersected by a constant current region of |6.1V_{OH}| mA when driving low and -|6.1V_{OH}| mA when driving high. In the special case of a full-speed driver which is driving low, and which is part of a high-speed capable transceiver, the low drive region is intersected by a constant current region of 22.0 mA. This is the minimum current drive level

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

necessary to ensure that the waveform at the receiver crosses the opposite single-ended switching level on the first reflection.

When testing, the current into or out of the device need not exceed ±10.71*V_{OH} mA and the voltage applied to D+/D- need not exceed 0.3*V_{OH} for the drive low case and need not drop below 0.7*V_{OH} for the drive high case.

Full-speed Buffers in High-speed Capable Transceivers

Figure 7-5 shows the V/I characteristics for a Full-speed buffer which is part of a high-speed capable transceiver. The output impedance, Z_{HSDRV} (including the contribution of R_S), is required to be between 40.5 Ω and 49.5 Ω. Additionally, the output voltage must be within 10mV of ground when no current is flowing in or out of the pin (V_{HSTERM}).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

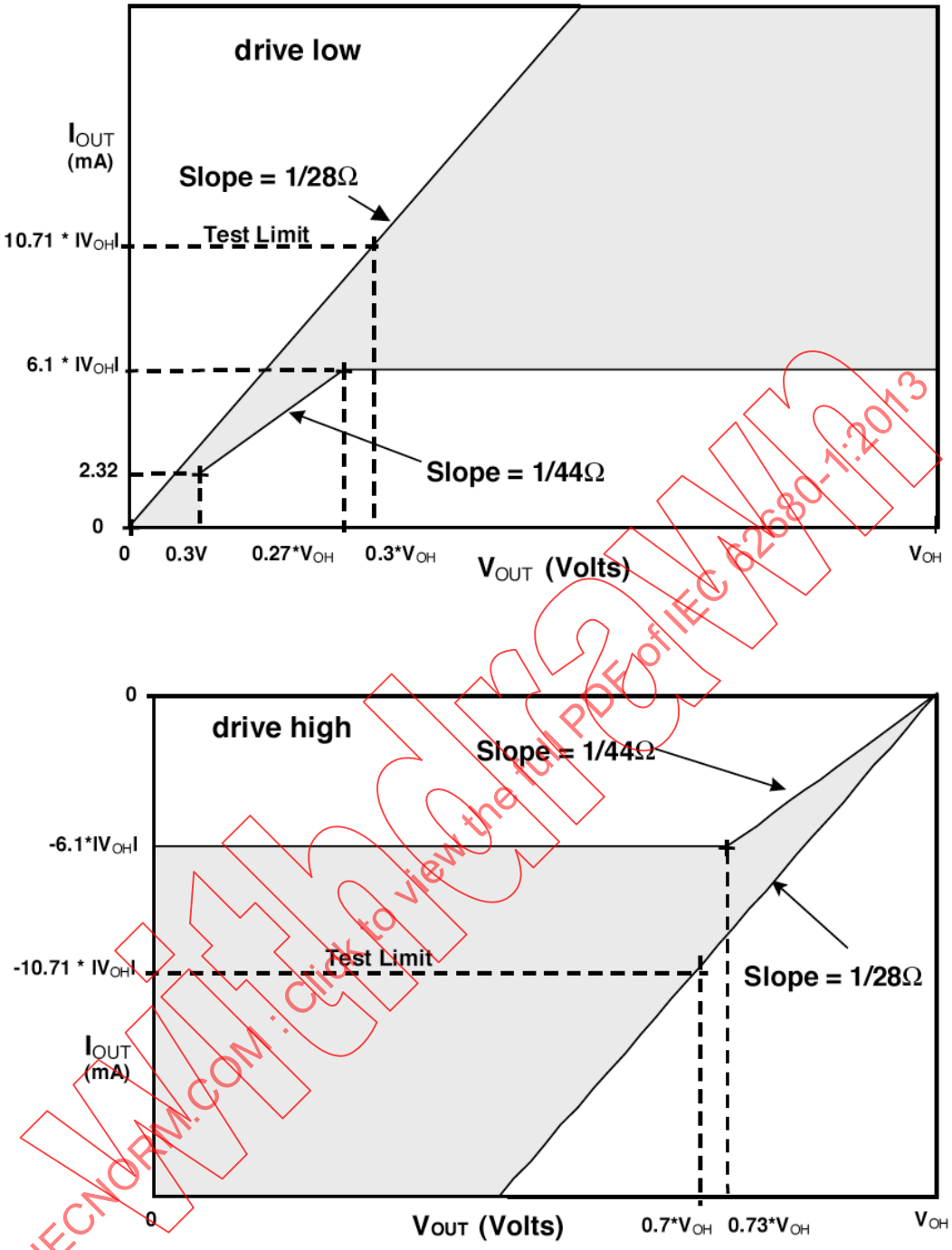


Figure 7-4. Full-speed Buffer V/I Characteristics

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

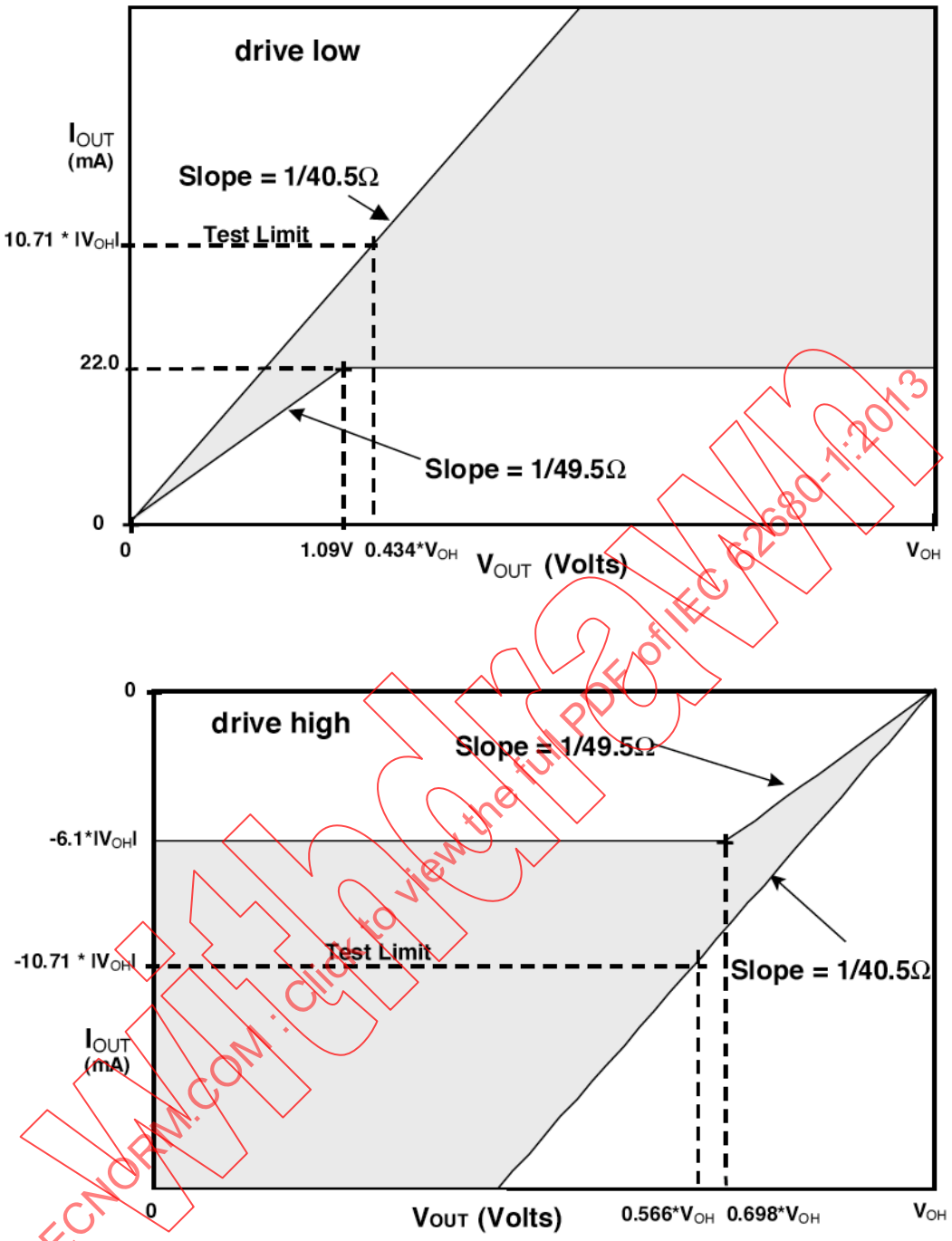


Figure 7-5. Full-speed Buffer V/I Characteristics for High-speed Capable Transceiver

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 7-6 shows the full-speed driver signal waveforms.

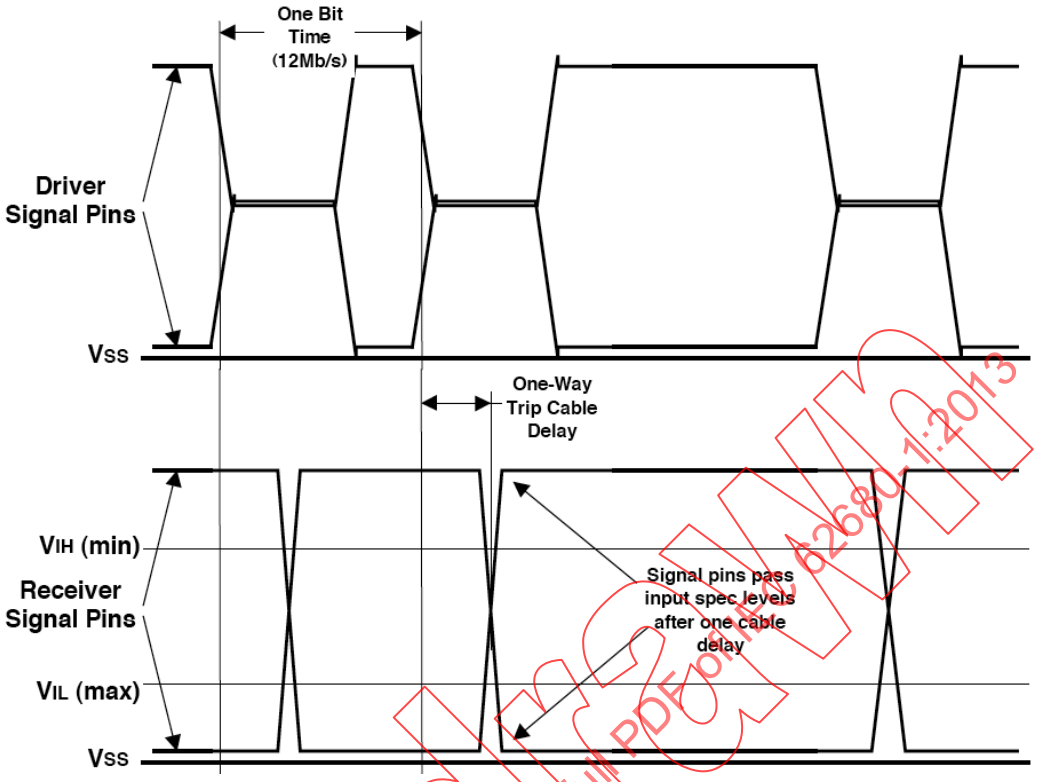


Figure 7-6. Full-speed Signal Waveforms

7.1.1.2 Low-speed (1.5 Mb/s) Driver Characteristics

A low-speed device must have a captive cable with the Series A connector on the plug end. The combination of the cable and the device must have a single-ended capacitance of no less than 200 pF and no more than 450 pF on the D+ or D- lines.

The propagation delay (T_{LSCBL}) of a low-speed cable must be less than 18 ns. This is to ensure that the reflection occurs during the first half of the signal rise/fall, which allows the cable to be approximated by a lumped capacitance.

Figure 7-7 shows the low-speed driver signal waveforms.

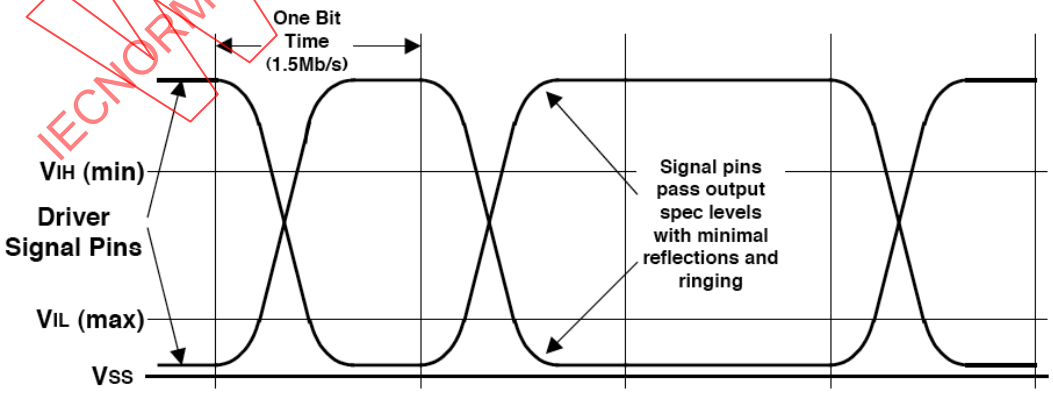


Figure 7-7. Low-speed Driver Signal Waveforms

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.1.1.3 High-speed (480 Mb/s) Driver Characteristics

A high-speed USB connection is made through a shielded, twisted pair cable with a differential characteristic impedance (Z_0) of 90 Ω ±15%, a common mode impedance (Z_{CM}) of 30 Ω ±30%, and a maximum one-way delay of 26 ns (T_{FSCBL}). The D+ and D- circuit board traces which run between a transceiver and its associated connector should also have a nominal differential impedance of 90 Ω, and together they may add an additional 4 ns of delay between the transceivers. (See Section 7.1.6 for details on impedance specifications of boards and transceivers.) The differential output impedance of a high-speed capable driver is required to be 90 Ω ±10%. When either the D+ or D- lines are driven high, V_{HSOH} (the high-speed mode high-level output voltage driven on a data line with a precision 45 Ω load to GND) must be 400 mV ±10%. On a line which is not driven, either because the transceiver is not transmitting or because the opposite line is being driven high, V_{HSOL} (the high-speed mode low-level output voltage driven on a data line with a 45 Ω load to GND) must be 0 V ± 10 mV.

Note: Unless indicated otherwise, all voltage measurements are to be made with respect to the local circuit ground.

Note: This specification requires that a high-speed capable transceiver operating in full-speed or low-speed mode must have a driver impedance (Z_{HSDRV}) of 45 Ω ±10%. It is recommended that the driver impedances be matched to within 5 Ω within a transceiver. For upstream facing transceivers which do not support high-speed mode, the driver output impedance (Z_{DRV}) must fall within the range of 28 Ω to 44 Ω.

On downstream facing ports, R_{PD} resistors (15 kΩ ±5%) must be connected from D+ and D- to ground.

When a high-speed capable transceiver transitions to high-speed mode, the high-speed idle state is achieved by driving SE0 with the low-/full-speed drivers at each end of the link (so as to provide the required terminations), and by disconnecting the D+ pull-up resistor in the upstream facing transceiver.

In the preferred embodiment, a transceiver activates its high-speed current driver only when transmitting high-speed signals. This is a potential design challenge, however, since the signal amplitude and timing specifications must be met even on the first symbol within a packet. As a less efficient alternative, a transceiver may cause its high-speed current source to be continually active while in high-speed mode. When the transceiver is not transmitting, the current may be directed into the device ground rather than through the current steering switch which is used for data signaling. In the example circuit, steering the current to ground is accomplished by setting HS_Drive_Enable low.

In CMOS implementations, the driver impedance will typically be realized by the combination of the driver's intrinsic output impedance and R_S. To optimally control Z_{HSDRV} and to minimize parasitics, it is preferred the driver impedance be minimized (under 5 Ω) and the balance of the 45 Ω should be contributed by the R_S component.

When a transceiver operating in high-speed mode transmits, the transmit current is directed into either the D+ or D- data line. A J is asserted by directing the current to the D+ line, a K by directing it to the D- line.

When each of the data lines is terminated with a 45 Ω resistor to the device ground, the effective load resistance on each side is 22.5 Ω. Therefore, the line into which the drive current is being directed rises to 17.78 ma * 22.5 Ω or 400 mV (nominal). The other line remains at the device ground voltage. When the current is directed to the opposite line, these voltages are reversed.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.1.2 Data Signal Rise and Fall, Eye Patterns

The following sections specify the data signal rise and fall times for full-speed and low-speed signaling, and the rise time and eye patterns for high-speed signaling.

7.1.2.1 Low-speed and Full-speed Data Signal Rise and Fall

For low-speed and full-speed, the output rise time and fall times are measured between 10% and 90% of the signal (Figure 7-8). Rise and fall time requirements apply to differential transitions as well as to transitions between differential and single-ended signaling.

The rise and fall times for full-speed buffers are measured with the load shown in Figure 7-9. The rise and fall times must be between 4 ns and 20 ns and matched to within ±10% to minimize RFI emissions and signal skew. The transitions must be monotonic.

The rise and fall times for low-speed buffers are measured with the load shown in Figure 7-10. The capacitive load shown in Figure 7-10 is representative of the worst-case load allowed by the specification. A downstream facing transceiver is allowed 150 pF of input/output capacitance (C_{IND}). A low-speed device (including cable) may have a capacitance of as little as 200 pF and as much as 450 pF. This gives a range of 200 pF to 600 pF as the capacitive load that a downstream facing low-speed buffer might encounter. Upstream facing buffers on low-speed devices must be designed to drive the capacitance of the attached cable plus an additional 150 pF. If a low-speed buffer is designed for an application where the load capacitance is known to fall in a different range, the test load can be adjusted to match the actual application. Low-speed buffers on hosts and hubs that are attached to USB receptacles must be designed for the 200 pF to 600 pF range. The rise and fall time must be between 75 ns and 300 ns for any balanced, capacitive test load. In all cases, the edges must be matched to within ±20% to minimize RFI emissions and signal skew. The transitions must be monotonic.

For both full-speed and low-speed signaling, the crossover voltage (V_{CRS}) must be between 1.3 V and 2.0 V.

For low-speed and full-speed, this specification does not require matching signal swing matching to any greater degree than described above. However, when signaling, it is preferred that the average voltage on the D+ and D- lines should be constant. This means that the amplitude of the signal swing on both D+ and D- should be the same; the low and high going transition should begin at the same time and change at the same rate; and the crossover voltage should be the same when switching to a J or K. Deviations from signal matching will result in common-mode noise that will radiate and affect the ability of devices and systems to pass tests that are mandated by government agencies.

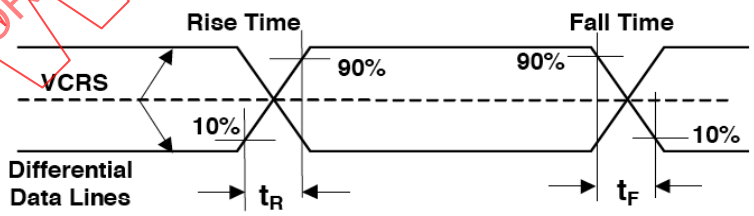


Figure 7-8. Data Signal Rise and Fall Time

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

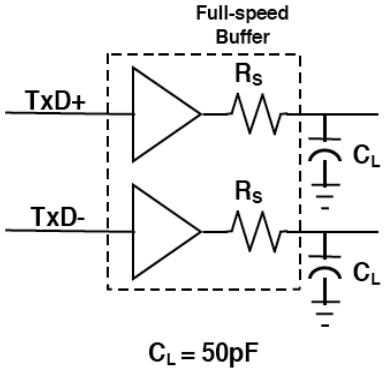


Figure 7-9. Full-speed Load

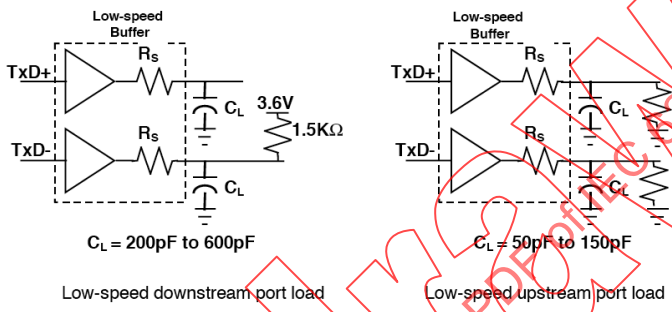


Figure 7-10. Low-speed Port Loads

Note: The CL for low-speed port load only represents the range of loading that might be added when the low-speed device is attached to a hub. The low-speed buffer must be designed to drive the load of its attached cable plus CL. A low-speed buffer design that can drive the downstream test load would be capable of driving any legitimate upstream load.

7.1.2.2 High-speed Signaling Eye Patterns and Rise and Fall Time

The following specifications apply to high-speed mode signaling. All bits, including the first and last bit of a packet, must meet the following eye pattern requirements for timing and amplitude.

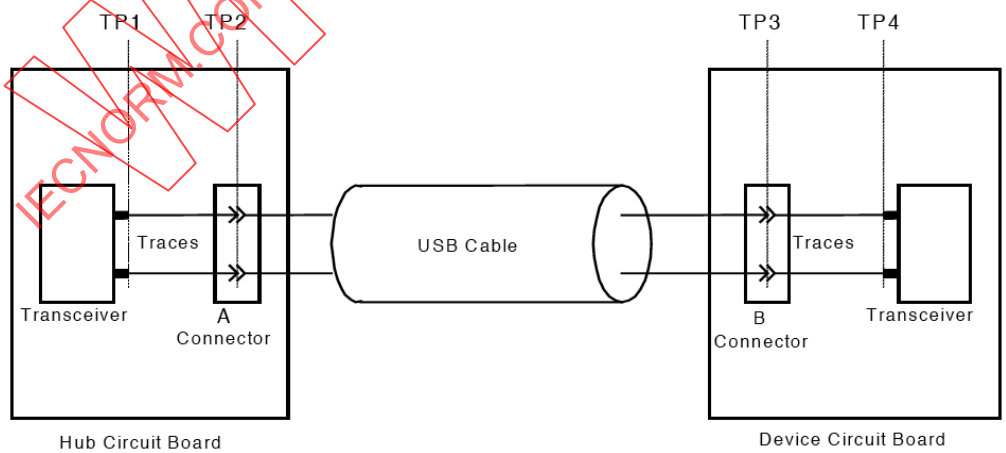


Figure 7-11. Measurement Planes

Figure 7-11 defines four test planes which will be referenced in this section. TP1 and TP4 are the points where the transceiver IC pins are soldered to the hub and device circuit boards, respectively. TP2 is at the mated pins of the A connector, and TP3 is at the mated pins of the

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

B connector (or, in the case of a captive cable, where the cable is attached to the circuit board). The following differential eye pattern templates specify transmit waveform and receive sensitivity requirements at various points and under various conditions.

When testing high-speed transmitters and receivers, measurements are made with the Transmitter/Receiver Test Fixture shown in Figure 7-12. In either case, the fixture is attached to the USB connector closest to the transceiver being tested.

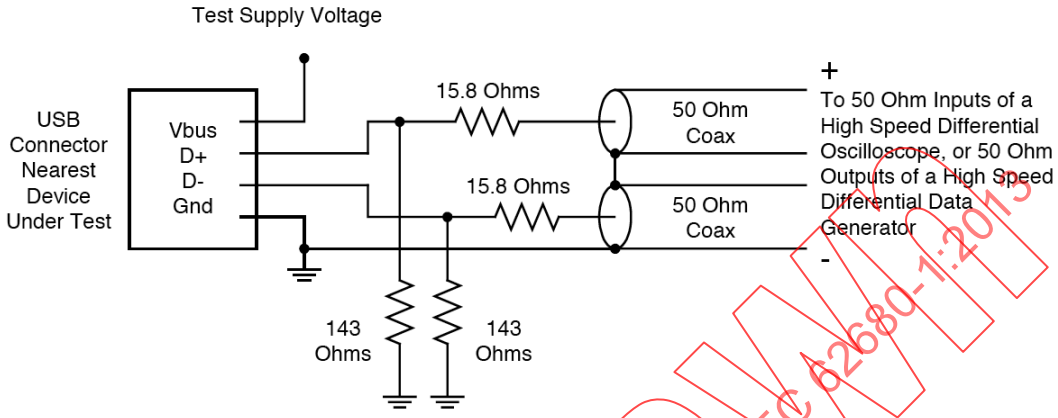


Figure 7-12. Transmitter/Receiver Test Fixture

Note: When testing the upstream facing port of a device, VBUS must be provided from the time the device is placed in the appropriate test mode until the test is completed. This requirement will likely necessitate additional switching functionality in the test fixture (for example, to switch the D+ and D- lines between the host controller and the test instrument). Such additions must have minimal impact on the high frequency measurement results.

Transmit eye patterns specify the minimum and maximum limits, as well as limits on timing jitter, within which a driver must drive signals at each of the specified test planes. Receive eye patterns specify the minimum and maximum limits, as well as limits on timing jitter, within which a receiver must recover data.

Conformance to Templates 1, 2, 3, and 4 is required for USB 2.0 hubs and devices:

Template 1: Transmit waveform requirements for hub measured at TP2, and for device (without a captive cable) measured at TP3

Template 2: Transmit waveform requirements for device (with a captive cable) measured at TP2

Template 3: Receiver sensitivity requirements for device (with a captive cable) when signal is applied at TP2

Template 4: Receiver sensitivity requirements for device (without a captive cable) when signal is applied at TP3, and for hub when signal is applied at TP2

Templates 5 and 6 are recommended guidelines for designers:

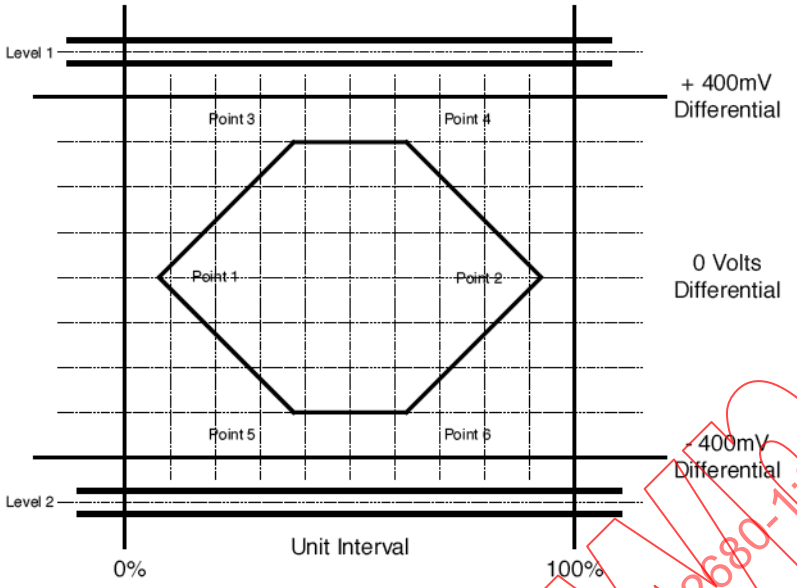
Template 5: Transmit waveform requirements for hub transceiver measured at TP1, and for device transceiver measured at TP4

Template 6: Receiver sensitivity requirements for device transceiver when signal is applied at TP4, and for hub transceiver at when signal is applied at TP1

Template 1

Figure 7-13 shows the transmit waveform requirements for a hub measured at TP2, and for a device (without a captive cable) measured at TP3.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



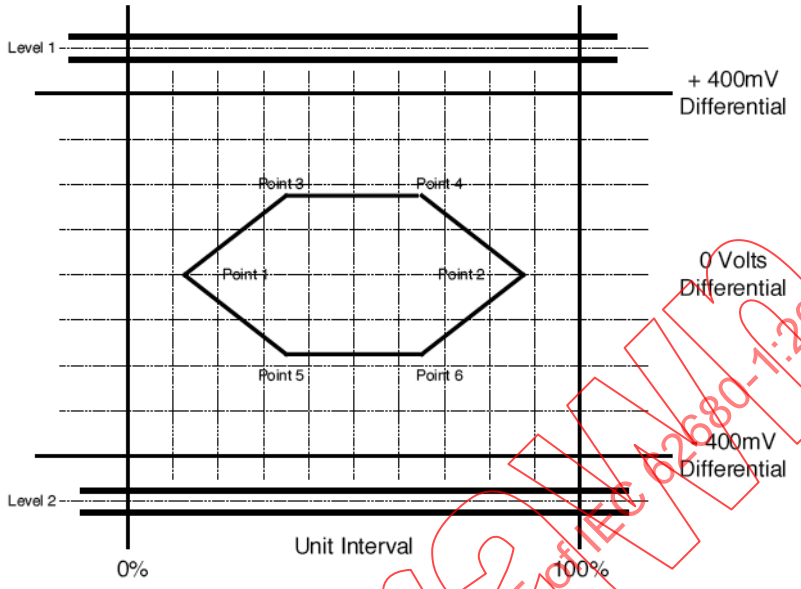
	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition, -475 in all others	N/A
Point 1	0 V	7.5% UI
Point 2	0 V	92.5% UI
Point 3	300 mV	37.5% UI
Point 4	300 mV	62.5% UI
Point 5	-300 mV	37.5% UI
Point 6	-300 mV	62.5% UI

Figure 7-13. Template 1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Template 2

Figure 7-14 shows transmit waveform requirements for a device (with a captive cable) measured at TP2.



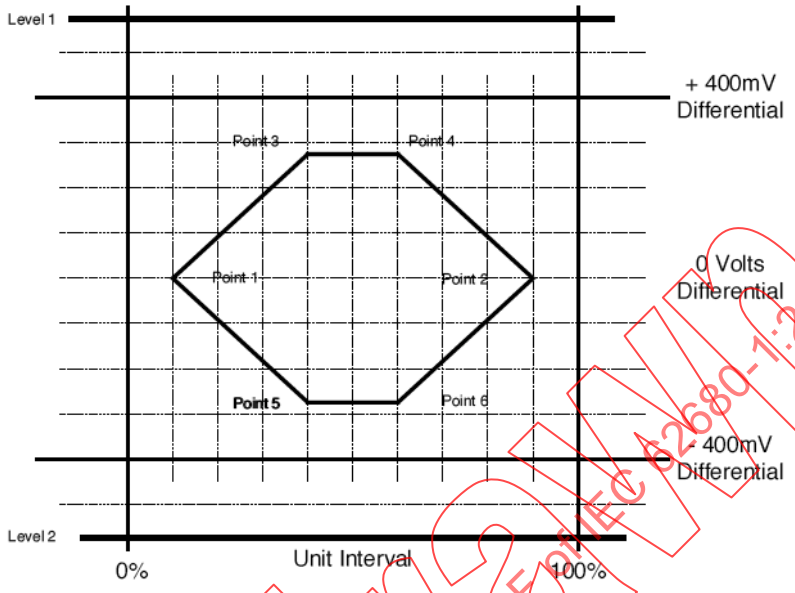
	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition, -475 in all others	N/A
Point 1	0 V	12.5% UI
Point 2	0 V	87.5% UI
Point 3	175 mV	35% UI
Point 4	175 mV	65% UI
Point 5	-175 mV	35% UI
Point 6	-175 mV	65% UI

Figure 7-14. Template 2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Template 3

Figure 7-15 shows receiver sensitivity requirements for a device (with a captive cable) when a signal is applied at TP2.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	575 mV	N/A
Level 2	-575 mV	N/A
Point 1	0 V	10% UI
Point 2	0 V	90% UI
Point 3	275 mV	40% UI
Point 4	275 mV	60% UI
Point 5	-275 mV	40% UI
Point 6	-275 mV	60% UI

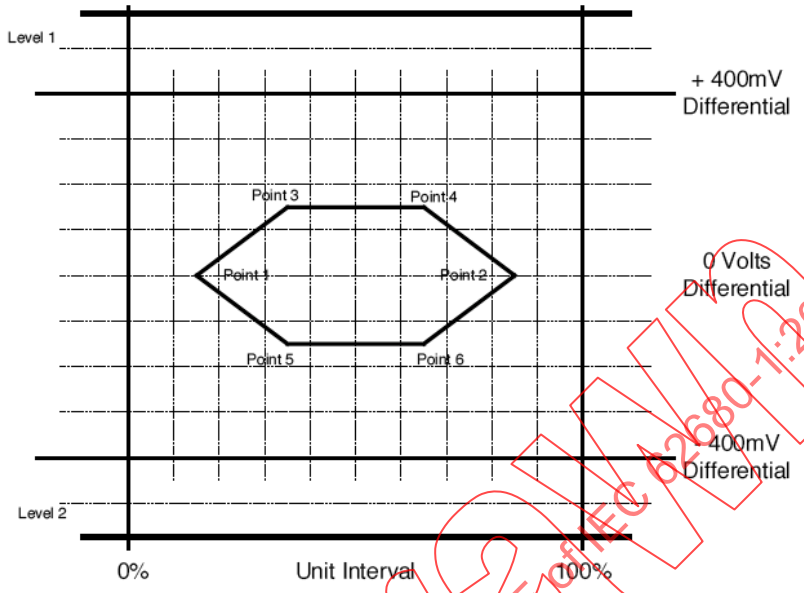
Figure 7-15. Template 3

Note: This eye is intended to specify differential data receiver sensitivity requirements. Levels 1 and 2 are outside the Disconnect Threshold values, but disconnection is detected at the source (after a minimum of 32 bit times without any transitions), not at the target receiver.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Template 4

Figure 7-16 shows receiver sensitivity requirements for a device (without a captive cable) when signal is applied at TP3, and for a hub when a signal is applied at TP2.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	575 mV	N/A
Level 2	-575 mV	N/A
Point 1	0 V	15% UI
Point 2	0 V	85% UI
Point 3	150 mV	35% UI
Point 4	150 mV	65% UI
Point 5	-150 mV	35% UI
Point 6	-150 mV	65% UI

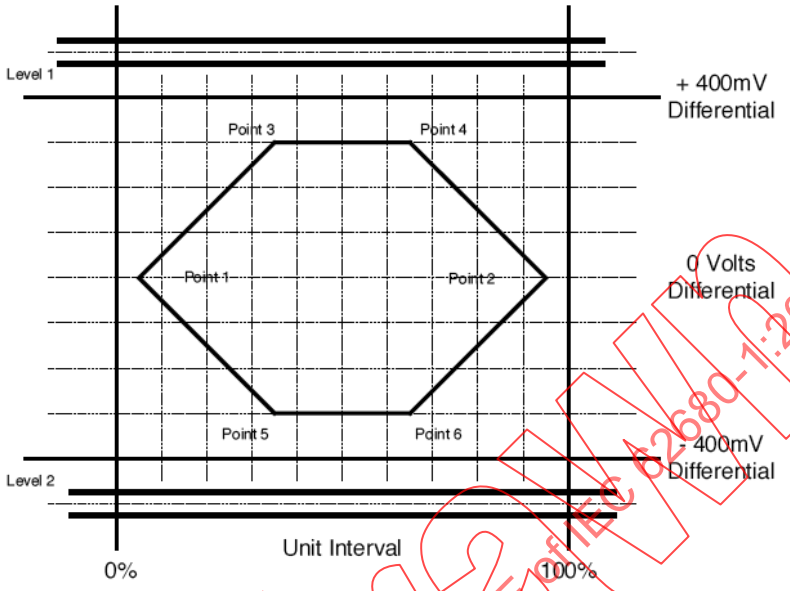
Figure 7-16. Template 4

Note: This eye is intended to specify differential data receiver sensitivity requirements. Levels 1 and 2 are outside the Disconnect Threshold values, but disconnection is detected at the source (after a minimum of 32 bit times without any transitions), not at the target receiver.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Template 5

Figure 7-17 shows transmit waveform requirements for a hub transceiver measured at TP1 and for a device transceiver measured at TP4.



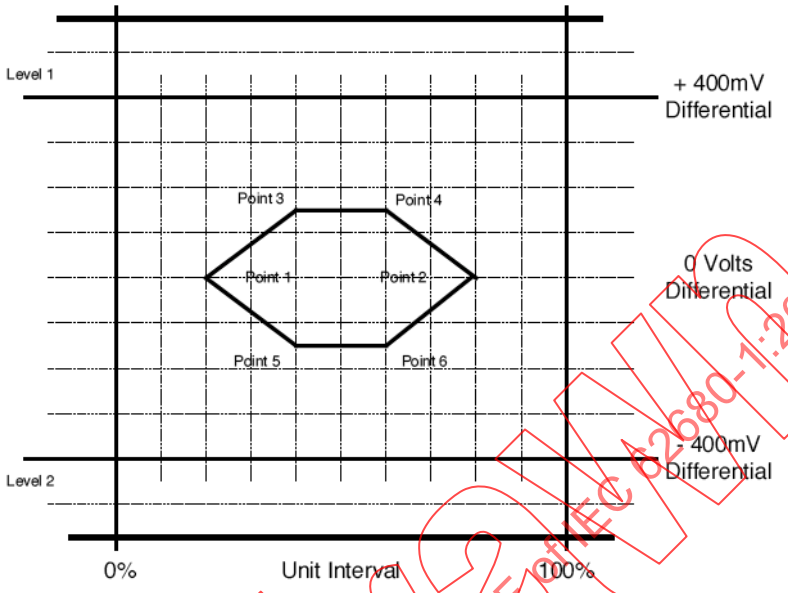
	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition, -475 in all others	N/A
Point 1	0 V	5% UI
Point 2	0 V	95% UI
Point 3	300 mV	35% UI
Point 4	300 mV	65% UI
Point 5	-300 mV	35% UI
Point 6	-300 mV	65% UI

Figure 7-17. Template 5

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Template 6

Figure 7-18 shows receiver sensitivity requirements for a device transceiver when a signal is applied at TP4 and for a hub transceiver when a signal is applied at TP1.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	575 mV	N/A
Level 2	-575 mV	N/A
Point 1	0 V	20% UI
Point 2	0 V	80% UI
Point 3	150 mV	40% UI
Point 4	150 mV	60% UI
Point 5	-150 mV	40% UI
Point 6	-150 mV	60% UI

Figure 7-18. Template 6

Note: This eye is intended to specify differential data receiver sensitivity requirements. Levels 1 and 2 are outside the Disconnect Threshold values, but disconnection is detected at the source (after a minimum of 32 bit times without any transitions), not at the target receiver.

High-speed Signaling Rise and Fall Times

The transition time of a high-speed driver must not be less than the specified minimum allowable differential rise and fall time (T_{HSR} and T_{HSF}). Transition times are measured when driving a reference load of 45 Ω to ground on D+ and D-. Figure 7-12 shows a recommended "Transmitter Test Fixture" for performing these measurements.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

For a hub, or for a device with detachable cable, the 10% to 90% high-speed differential rise and fall times must be 500 ps or longer when measured at the A or B receptacles (respectively).

For a device with a captive cable assembly, it is a recommended design guideline that the 10% to 90% high-speed differential rise and fall times must be 500 ps or longer when measured at the point where the cable is attached to the device circuit board.

It is required that high-speed data transitions be monotonic over the minimum vertical openings specified in the preceding eye pattern templates.

7.1.2.3 Driver Usage

The upstream facing ports of functions must use one and only one of the following three driver configurations:

1. Low-speed – Low-speed drivers only
2. Full-speed – Full-speed drivers only
3. Full-/high-speed – Combination full-speed and high-speed drivers

Upstream facing USB 2.0 hub ports must use full-/high-speed drivers. Such ports must be capable of transmitting data at low-speed and full-speed rates with full-speed signaling, and at the high-speed rate using high-speed signaling. Downstream facing ports (including the host) must support low-speed, full-speed, and high-speed signaling, and must be able to transmit data at each of the three associated data rates.

In this section, there is reference to a situation in which high-speed operation is “disallowed.” This topic is discussed in depth in Chapter 11 of this specification. In brief, a high-speed capable hub's downstream facing ports are “high-speed disallowed” if the hub is unable to establish a high-speed connection on its upstream facing port. For example, this would be the case for the downstream facing ports of a high-speed capable hub when the hub is connected to a USB 1.1 host controller.

When a full-/high-speed device is attached to a pre-USB 2.0 hub, or to a hub port which is high-speed disallowed, it is required to behave as a full-speed only device. When a full-/high-speed device is attached to a USB 2.0 hub which is not high-speed disallowed, it must operate with high-speed signaling and data rate.

7.1.3 Cable Skew

The maximum skew introduced by the cable between the differential signaling pair (i.e., D+ and D- (T_{SKEW})) must be less than 100 ps and is measured as described in Section 6.7.

7.1.4 Receiver Characteristics

This section discusses the receiver characteristics for low-speed, full-speed, and full-/high-speed transceivers.

7.1.4.1 Low-speed and Full-speed Receiver Characteristics

A differential input receiver must be used to accept the USB data signal. The receiver must feature an input sensitivity (V_{DI}) of at least 200 mV when both differential data inputs are in the differential common mode range (V_{CM}) of 0.8 V to 2.5 V, as shown in Figure 7-19

In addition to the differential receiver, there must be a single-ended receiver for each of the two data lines. The receivers must have a switching threshold between 0.8 V (V_{IL}) and 2.0 V (V_{IH}). It is recommended that the single-ended receivers incorporate hysteresis to reduce their sensitivity to noise.

Both D+ and D- may temporarily be less than V_{IH} (min) during differential signal transitions. This period can be up to 14 ns (T_{FST}) for full-speed transitions and up to 210 ns (T_{LST}) for

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

low-speed transitions. Logic in the receiver must ensure that that this is not interpreted as an SE0.

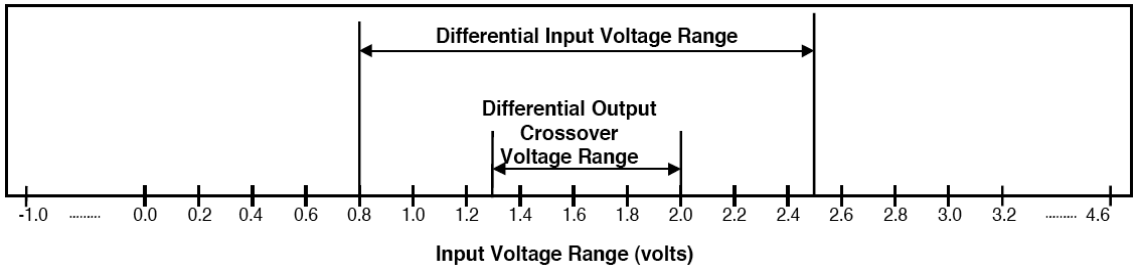


Figure 7-19. Differential Input Sensitivity Range for Low-/full-speed

7.1.4.2 High-speed Receiver Characteristics

A high-speed capable transceiver receiver must conform to the receiver characteristics specifications called out in Section 7.1.4.1 when receiving in low-speed or full-speed modes.

As shown in Figure 7-1, a high-speed capable transceiver which is operating in high-speed mode “listens” for an incoming serial data stream with the high-speed differential data receiver and the transmission envelope detector. Additionally, a downstream facing high-speed capable transceiver monitors the amplitude of the differential voltage on the lines with the disconnection envelope detector.

When receiving in high-speed mode, the differential receiver must be able to reliably receive signals that conform to the Receiver Eye Pattern templates shown in Section 7.1.2. Additionally, it is a strongly recommended guideline that a high-speed receiver should be able to reliably receive such signals in the presence of a common mode voltage component (V_{HSCM}) over the range of –50 mV to 500 mV (the nominal common mode component of high-speed signaling is 200 mV). Low frequency chirp J and K signaling, which occurs during the Reset handshake, should be reliably received with a common mode voltage range of –50 mV to 600 mV.

Reception of data is qualified by the output of the transmission envelope detector. The receiver must disable data recovery when the signal falls below the high-speed squelch level (V_{HSSQ}) defined in Table 7-3. (Detector must indicate squelch when the magnitude of the differential voltage envelope is ≤ 100 mV, and must not indicate squelch if the amplitude of differential voltage envelope is ≥ 150 mV.) Squelch detection must be done with a differential envelope detector, such as the one shown in Figure 7-1. The envelope detector used to detect the squelch state must incorporate a filtering mechanism that prevents indication of squelch during differential data crossovers.

The definition of a high-speed packet’s SYNC pattern, together with the requirements for high-speed hub repeaters, guarantee that a receiver will see at least 12 bits of SYNC (KJKJKJKJKJKK) followed by the data portion of the packet. This means that the combination of squelch response time, DLL lock time, and end of SYNC detection must occur within 12 bit times. This is required to assure that the first bit of the packet payload will be received correctly.

In the case of a downstream facing port, a high-speed capable transceiver must include a differential envelope detector that indicates when the signal on the data exceeds the high-speed Disconnect level (V_{HSDSC}) as defined in Table 7-3. (The detector must not indicate that the disconnection threshold has been exceeded if the differential signal amplitude is ≤525 mV, and must indicate that the threshold has been exceeded if the differential signal amplitude is ≥625 mV.)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

When sampled at the appropriate time, this detector provides indication that the device has been disconnected. The details of how the disconnection envelope detector is used are described in Section 7.1.7.3.

7.1.5 Device Speed Identification

The following sections specify the speed identification mechanisms for low-speed, full-speed, and high-speed.

7.1.5.1 Low-/Full-speed Device Speed Identification

The USB is terminated at the hub and function ends as shown in Figure 7-20 and Figure 7-21. Full-speed and low-speed devices are differentiated by the position of the pull-up resistor on the downstream end of the cable:

- Full-speed devices are terminated as shown in Figure 7-20 with the pull-up resistor on the D+ line.
- Low-speed devices are terminated as shown in Figure 7-21 with the pull-up resistor on the D- line.
- The pull-down terminators on downstream facing ports are resistors of 15 kΩ ±5% connected to ground.

The design of the pull-up resistor must ensure that the signal levels satisfy the requirements specified in Table 7-2. In order to facilitate bus state evaluation that may be performed at the end of a reset, the design must be able to pull-up D+ or D- from 0 V to VIH (min) within the minimum reset relaxation time of 2.5 µs. A device that has a detachable cable must use a 1.5 kΩ ±5% resistor tied to a voltage source between 3.0 V and 3.6 V (VTERM) to satisfy these requirements. Devices with captive cables may use alternative termination means. However, the Thevenin resistance of any termination must be no less than 900 Ω.

Note: Thevenin resistance of termination does not include the 15 kΩ ±5% resistor on host/hub.

The voltage source on the pull-up resistor must be derived from or controlled by the power supplied on the USB cable such that when VBUS is removed, the pull-up resistor does not supply current on the data line to which it is attached.

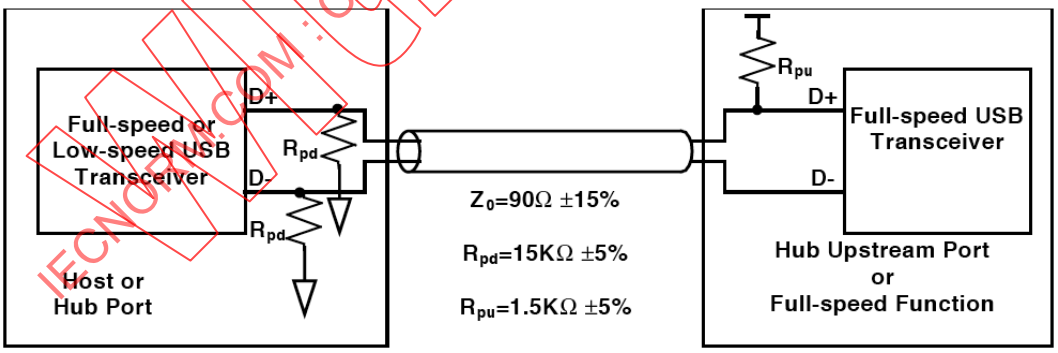


Figure 7-20. Full-speed Device Cable and Resistor Connections

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

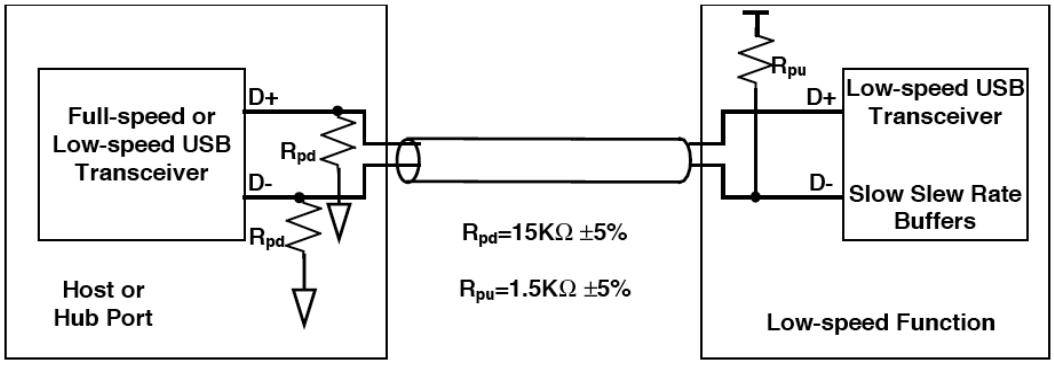


Figure 7-21. Low-speed Device Cable and Resistor Connections

7.1.5.2 High-speed Device Speed Identification

The high-speed Reset and Detection mechanisms follow the behavioral model for low-/full-speed. When reset is complete, the link must be operating in its appropriate signaling mode (low-speed, full-speed, or high-speed as governed by the preceding usage rules), and the speed indication bits in the port status register will correctly report this mode. Software need only initiate the assertion of reset and read the port status register upon notification of reset completion.

High-speed capable devices initially attach as full-speed devices. This means that for high-speed capable upstream facing ports, R_{PU} (1.5 kΩ ±5%) must be connected from D+ to the 3.3 V supply (as shown in Figure 7-1) through a switch which can be opened under SW control.

After the initial attachment, high-speed capable transceivers engage in a low level protocol during reset to establish a high-speed link and to indicate high-speed operation in the appropriate port status register. This protocol is described in Section 7.1.7.5.

7.1.6 Input Characteristics

The following sections describe the input characteristics for transceivers operating in low-speed, full-speed, and high-speed modes.

7.1.6.1 Low-speed and Full-speed Input Characteristics

The input impedance of D+ or D- without termination should be > 300 kΩ (Z_{INP}). The input capacitance of a port is measured at the connector pins. Upstream facing and downstream facing ports are allowed different values of capacitance. The maximum capacitance (differential or single-ended) (C_{IND}) allowed on a downstream facing port of a hub or host is 150 pF on D+ or D- when operating in low-speed or full-speed. This is comprised of up to 75 pF of lumped capacitance to ground on each line at the transceiver and in the connector, and an additional 75 pF capacitance on each conductor in the transmission line between the receptacle and the transceiver. The transmission line between the receptacle and R_S must be 90 Ω ±15%.

The maximum capacitance on an upstream facing port of a full-speed device with a detachable cable (C_{INUB}) is 100 pF on D+ or D-. This is comprised of up to 75 pF of lumped capacitance to ground on each line at the transceiver and in the connector and an additional 25 pF capacitance on each conductor in the transmission line between the receptacle and the transceiver. The difference in capacitance between D+ and D- must be less than 10%.

For full-speed devices with captive cables, the device itself may have up to 75 pF of lumped capacitance to ground on D+ and D-. The cable accounts for the remainder of the input capacitance.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A low-speed device is required to have a captive cable. The input capacitance of the low-speed device will include the cable. The maximum single-ended or differential input capacitance of a low-speed device is 450 pF (C_{LINUA}).

For devices with captive cables, the single-ended input capacitance must be consistent with the termination scheme used. The termination must be able to charge the D+ or D- line from 0 V to V_{IH} (min) within 2.5 µs. The capacitance on D+/D- includes the single-ended input-capacitance of the device (measured from the pins on the connector on the cable) and the 150 pF of input capacitance of the host/hub.

An implementation may use small capacitors at the transceiver for purposes of edge rate control. The sum of the capacitance of the added capacitor (CEDGE), the transceiver, and the trace connecting capacitor and transceiver to RS must not exceed 75 pF (either single-ended or differential) and the capacitance must be balanced to within 10%. The added capacitor, if present, must be placed between the transceiver pins and RS (see Figure 7-22).

Use of ferrite beads on the D+ or D- lines of full-speed devices is discouraged.

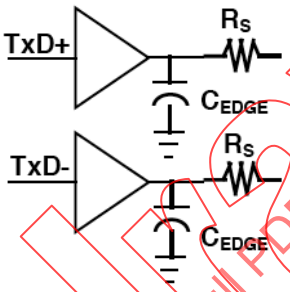


Figure 7-22. Placement of Optional Edge Rate Control Capacitors for Low-/full-speed

7.1.6.2 High-speed Input Characteristics

Figure 7-23 shows the simple equivalent loading circuit of a USB device operating in high-speed receive mode.

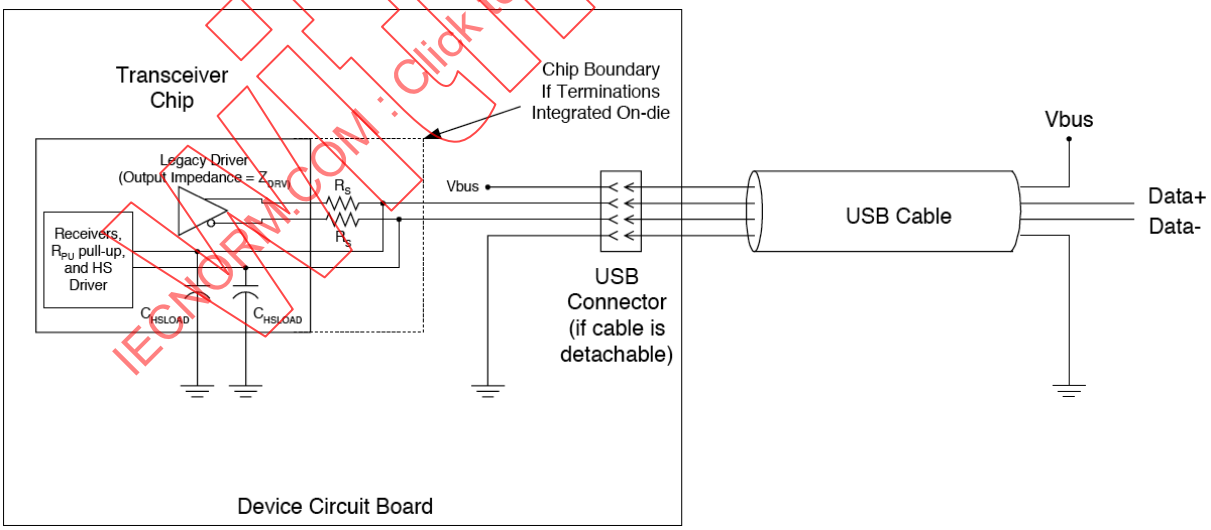


Figure 7-23. Diagram for High-speed Loading Equivalent Circuit

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

When operating in high-speed signaling mode, a transceiver must meet the following loading specifications:

1. DC output voltage and resistance specifications
2. TDR loading specification

Additionally, it is strongly recommended that a transceiver component operating in high-speed signaling mode should meet the following lumped capacitance guideline.

The use of ferrites on high-speed data lines is strongly discouraged.

DC output voltage and resistance specifications – A transceiver that is in high-speed mode must present a DC load on each of the data lines nominally equivalent to 45 Ω to ground. The actual resistance, Z_{HSDRV}, must be 40.5 Ω ≤ Z_{HSDRV} ≤ 49.5 Ω. The output voltage in the high-speed idle state (V_{HSTERM}) is specified in Table 7-3

TDR loading specification – The AC loading specifications of a transceiver in the high-speed idle state are specified in terms of differential TDR (Time Domain Reflectometer) measurements.

These measurements govern the maximum allowable transmission line discontinuities for the port connector, the interconnect leading from the connector to the transceiver, the transceiver package, and the transceiver IC itself. In the special case of a high-speed capable device with a captive cable, the transmission line discontinuities of the cable assembly are also governed.

The following specifications must be met with the incident rise time of the differential TDR set to 400 ps. It is important to note that all times are "as displayed" on the TDR and are hence "round trip times."

Termination Impedance (Z_{HSTERM}) is measured on the TDR trace at a specific measurement time following the connector reference time. The connector reference time is determined by disconnecting the TDR connection from the port connector and noting the time of the open circuit step. For an A connector, the measurement time is 8 ns after the connector reference location. For a B connector, the measurement time is 4 ns after the connector reference location. The differential termination impedance must be:

$$80\ \Omega \leq Z_{HSTERM} \leq 100\ \Omega$$

Through Impedance (Z_{HSTHRU}) is the impedance measured from 500 ps before the connector reference location until the time governed by the Termination impedance specification.

$$70\ \Omega \leq Z_{HSTHRU} \leq 110\ \Omega$$

In the Exception Window (a sliding 1.4 ns window inside the Through Impedance time window), the differential impedance may exceed the Through limits. No single excursion, however, may exceed the Through limits for more than twice the TDR rise time (400 ps).

In the special case of a high-speed capable device with a captive cable, the same specifications must be met, but the TDR measurements must be made through the captive cable assembly. Determination of the connector reference time can be more difficult in this case, since the cable may not be readily removable from the port being tested. It is left to the tester of a specific device to determine the connector reference location by whatever means are available.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Lumped capacitance guideline for the transceiver component

When characterizing a transceiver chip as an isolated component, the measurement can be performed effectively at the chip boundary shown in Figure 7-23 without USB connectors or cables. Parasitic capacitance of the test fixture can be corrected by measuring the capacitance of the fixture itself and subtracting this reading from the reading taken with the transceiver inserted. If the terminations are off-chip, discrete R_S resistors should be in place during the measurements, and measurements should be taken on the "connector side" of the resistors. The transceiver should be in Test_SE0_NAK mode during testing.

Capacitance measurements are taken from each of the data lines to ground while the other line is left open. The instrument used to perform this measurement must be able to determine the effective capacitance to ground in the presence of the parallel effective resistance to ground.

Capacitance to Ground on each line: $C_{HSLOAD} \leq 10$ pF
Matching of Capacitances to Ground: ≤ 1.0 pF

The guideline is to allow no more than 5.0 pF for the transceiver die itself and no more than an additional 5 pF for the package. The differential capacitance across the transceiver inputs should be no more than 5.0 pF

7.1.7 Signaling Levels

The following sections specify signaling levels for low-speed, full-speed, and high-speed operation.

7.1.7.1 Low-/Full-speed Signaling Levels

Table 7-2 summarizes the USB signaling levels. The source is required to drive the levels specified in the second column, and the target is required to identify the correct bus state when it sees the levels in the third column. (Target receivers can be more sensitive as long as they are within limits specified in the fourth column.)

Table 7-2. Low-/full-speed Signaling Levels

Bus State	Signaling Levels		
	At originating source connector (at end of bit time)	At final target connector	
		Required	Acceptable
Differential "1"	D+ > V_{OH} (min) and D- < V_{OL} (max)	(D+) - (D-) > 200 mV and D+ > V_{IH} (min)	(D+) - (D-) > 200 mV
Differential "0"	D- > V_{OH} (min) and D+ < V_{OL} (max)	(D-) - (D+) > 200 mV and D- > V_{IH} (min)	(D-) - (D+) > 200 mV
Single-ended 0 (SE0)	D+ and D- < V_{OL} (max)	D+ and D- < V_{IL} (max)	D+ and D- < V_{IH} (min)
Single-ended 1 (SE1)	D+ and D- > V_{OSE1}(min)	D+ and D- > V_{IL} (max)	
Data J state: Low-speed Full-speed	 Differential "0" Differential "1"	 Differential "0" Differential "1"	
Data K state: Low-speed Full-speed	 Differential "1" Differential "0"	 Differential "1" Differential "0"	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Bus State	Signaling Levels		
	At originating source connector (at end of bit time)	At final target connector	
		Required	Acceptable
Idle state: Low-speed Full-speed	NA	 D- > VIHZ (min) and D+ < VIL (max) D+ > VIHZ (min) and D- < VIL (max)	 D- > VIHZ (min) and D+ < VIH (min) D+ > VIHZ (min) and D- < VIH (min)
Resume state	Data K state	Data K state	
Start-of-Packet (SOP)	Data lines switch from Idle to K state		
End-of-Packet (EOP)[4]	SE0 for approximately 2 bit times[1] followed by a J for 1 bit time[3]	SE0 for ≥ 1 bit time[2] followed by a J state for 1 bit time	SE0 for ≥ 1 bit time[2] followed by a J state
Disconnect (at downstream port)	NA	SE0 for ≥2.5 μs	
Connect (at downstream port)	NA	Idle for ≥2 ms	Idle for ≥2.5 μs
Reset	D+ and D- < VOL (max) for ≥10ms	D+ and D- < VIL (max) for ≥10 ms	D+ and D- < VIL (max) for ≥2.5 μs

Note 1: The width of EOP is defined in bit times relative to the speed of transmission. (Specification EOP widths are given in Table 7-7 and Table 7-8.)

Note 2: The width of EOP is defined in bit times relative to the device type receiving the EOP. The bit time is approximate.

Note 3: The width of the J state following the EOP is defined in bit times relative to the buffer edge rate. The J state from a low-speed buffer must be a low-speed bit time wide and, from a full-speed buffer, a full-speed bit time wide.

Note 4: The keep-alive is a low-speed EOP.

The J and K data states are the two logical levels used to communicate differential data in the system. Differential signaling is measured from the point where the data line signals cross over. Differential data signaling is not concerned with the level at which the signals cross, as long as the crossover voltage meets the requirements in Section 7.1.2. Note that, at the receiver, the Idle and Resume states are logically equivalent to the J and K states respectively.

As shown in Table 7-2, the J and K states for full-speed signaling are inverted from those for low-speed signaling. The sense of data, idle, and resume signaling is set by the type of device that is being attached to a port. If a full-speed device is attached to a port, that segment of the USB uses full-speed signaling conventions (and fast rise and fall times), even if the data being sent across the data lines is at the low-speed data rate. The low-speed signaling conventions shown in Table 7-2 (plus slow rise and fall times) are used only between a low-speed device and the port to which it is attached.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

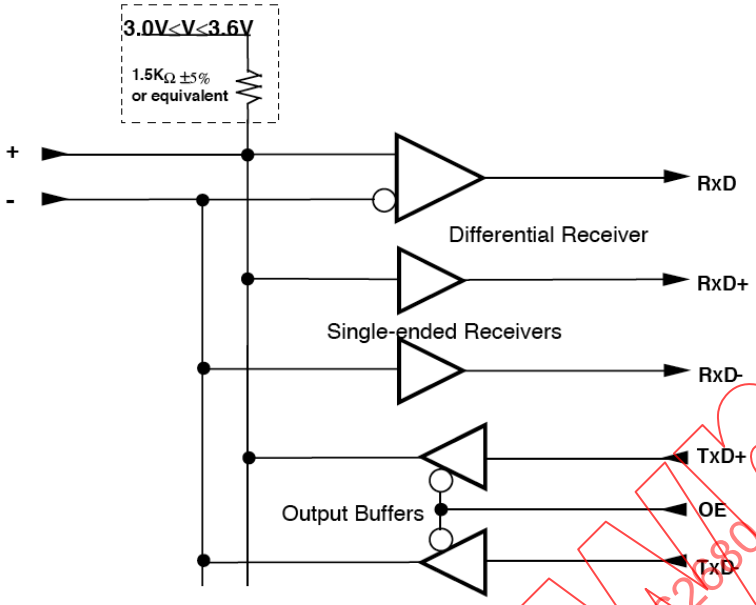


Figure 7-24. Upstream Facing Full-speed Port Transceiver

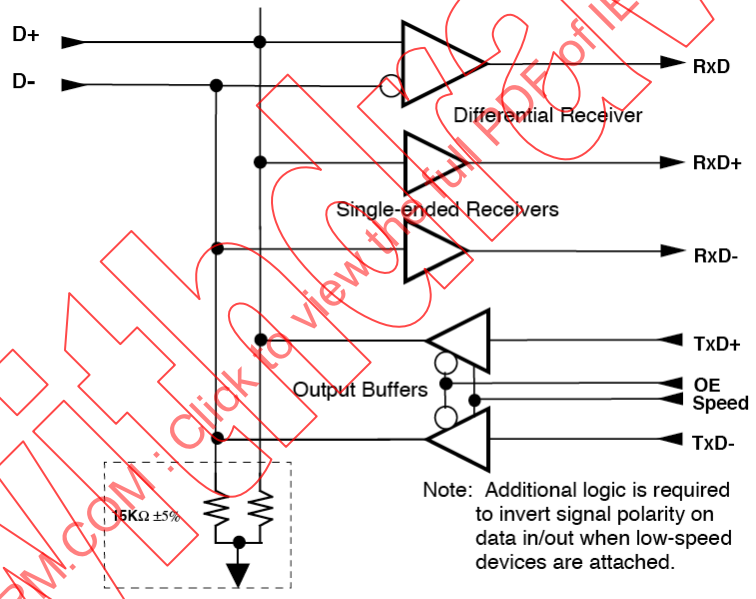


Figure 7-25. Downstream Facing Low-/full-speed Port Transceiver

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.1.7.2 Full-/High-speed Signaling Levels

The high-speed signaling voltage specifications in Table 7-3 must be met when measuring at the connector closest to the transceiver, using precision 45 Ω load resistors to the device ground as reference loads. All voltage measurements are taken with respect to the local device ground.

Table 7-3. High-speed Signaling Levels

Bus State	Required Signaling Level at Source Connector	Required Signaling Level at Target Connector
High-speed Differential “1”	**DC Levels:** VHSOH (min) ≤ D+ ≤ VHSOH (max) VHSOL (min) ≤ D- ≤ VHSOL (max) See Note 1. **AC Differential Levels:** A transmitter must conform to the eye pattern templates called out in Section 7.1.2. See Note 2.	**AC Differential Levels** The signal at the target connector must be recoverable, as defined by the eye pattern templates called out in Section 7.1.2. See Note 2.
High-speed Differential “0”	**DC Levels:** VHSOH (min) ≤ D- ≤ VHSOH (max) VHSOL (min) ≤ D+ ≤ VHSOL (max) See Note 1. **AC Differential Levels:** A transmitter must conform to the eye pattern templates called out in Section 7.1.2. See Note 2.	**AC Differential Levels:** The signal at the target connector must be recoverable, as defined by the eye pattern templates called out in Section 7.1.2. See Note 2.
High-speed J State	High-speed Differential “1”	High-speed Differential “1”
High-speed K State	High-speed Differential “0”	High-speed Differential “0”
Chirp J State (differential voltage; applies only during reset when both hub and device are high-speed capable)	**DC Levels:** VCHIRPJ (min) ≤ (D+ - D-) ≤ VCHIRPJ (max)	**AC Differential Levels** The differential signal at the target connector must be ≥ 300 mV
Chirp K State (differential voltage; applies only during reset when both hub and device are high-speed capable)	**DC Levels:** VCHIRPK (min) ≤ (D+ - D-) ≤ VCHIRPK (max)	**AC Differential Levels** The differential signal at the target connector must be ≤ -300 mV
High-speed Squelch State	NA	VHSSQ - Receiver must indicate squelch when magnitude of differential voltage is ≤100 mV; receiver must not indicate squelch if magnitude of differential voltage is ≥150 mV. See Note 3.
High-speed Idle State	NA	**DC Levels:** VHSOI min ≤ (D+, D-) ≤ VHSOI max See Note 1. **AC Differential Levels:** Magnitude of differential voltage is ≤ 100 mV See Note 3.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-3. High-speed Signaling Levels (Continued)

Bus State	Required Signaling Level at Source Connector	Required Signaling Level at Target Connector
Start of High-speed Packet (HSSOP)	Data lines switch from high-speed Idle to high-speed J or high-speed K state.	
End of High-speed Packet (HSEOP)	Data lines switch from high-speed J or K to high-speed Idle state.	
High-speed Disconnect State (at downstream facing port)	NA	VHSDSC - Downstream facing port must not indicate device disconnection if differential voltage is ≤ 525 mV, and must indicate device disconnection when magnitude of differential voltage is ≥ 625 mV, at the sample time discussed in Section 7.1.7.3.

Note 1: Note 1: Measured with a 45 Ω resistor to ground at each data line, using test modes Test_J and Test_K

Note 2: Note 2: Measured using test mode Test_Packet with fixture shown in Figure 7-12

Note 3: Note 3: Measured with fixture shown in Figure 7-12, using test mode SE0_NACK

Note 4: Note 4: A high-speed driver must never "intentionally" generate a signal in which both D+ and D- are driven to a level above 200 mV. The current-steering design of a high-speed driver should naturally preclude this possibility.

7.1.7.3 Connect and Disconnect Signaling

When no function is attached to the downstream facing port of a host or hub in low-/full-speed, the pull-down resistors present there will cause both D+ and D- to be pulled below the single-ended low threshold of the host or hub transceiver when that port is not being driven by the hub. This creates an SE0 state on the downstream facing port. A disconnect condition is indicated if the host or hub is not driving the data lines and an SE0 persists on a downstream facing port for more than TDDIS (see Figure 7-26). The specifications for TDDIS and TDCNN are defined in Table 7-13.

A connect condition will be detected when the hub detects that one of the data lines is pulled above its VIH threshold for more than TDCNN (see Figure 7-27 and Figure 7-28).

Hubs must determine the speed of the attached device by sampling the state of the bus immediately before driving SE0 to indicate a reset condition to the device.

All signaling levels given in Table 7-2 are set for this bus segment (and this segment alone) once the speed of the attached device is determined. The mechanics of speed detection are described in Section 11.8.2.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

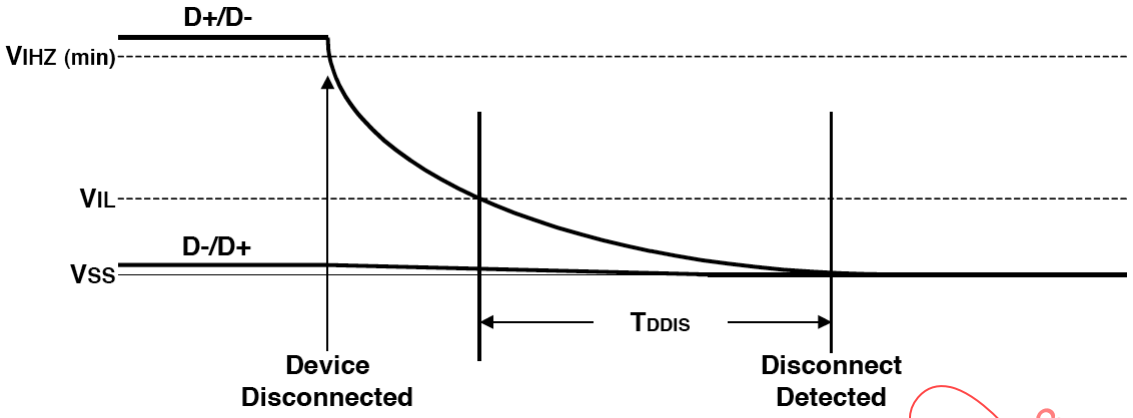


Figure 7-26. Low-/full-speed Disconnect Detection

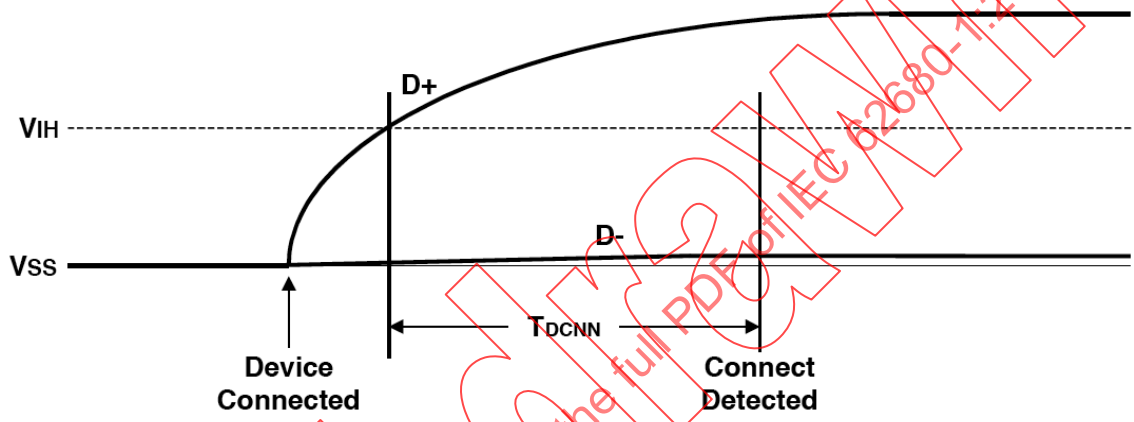


Figure 7-27. Full-/high-speed Device Connect Detection

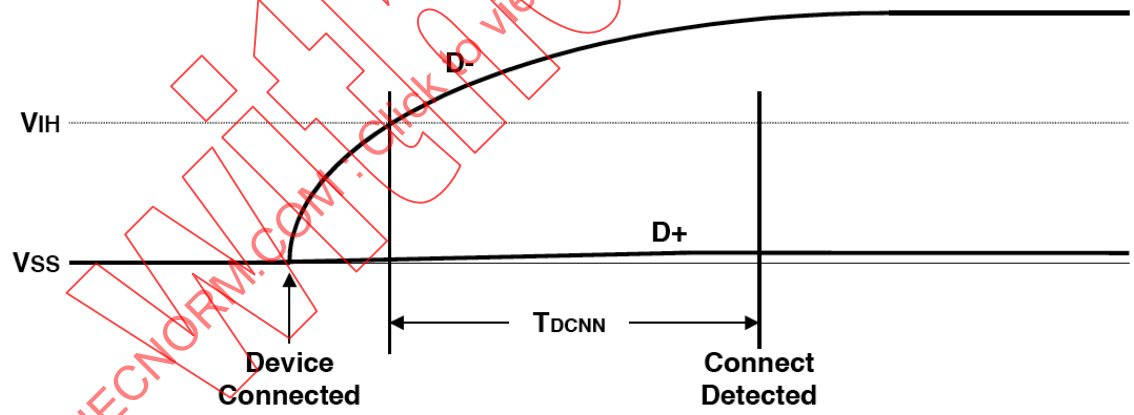


Figure 7-28. Low-speed Device Connect Detection

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Because USB components may be hot plugged, and hubs may implement power switching, it is necessary to comprehend the delays between power switching and/or device attach and when the device's internal power has stabilized. Figure 7-29 shows all the events associated with both turning on port power with a device connected and hot-plugging a device. There are six delays and a sequence of events that are defined by this specification.

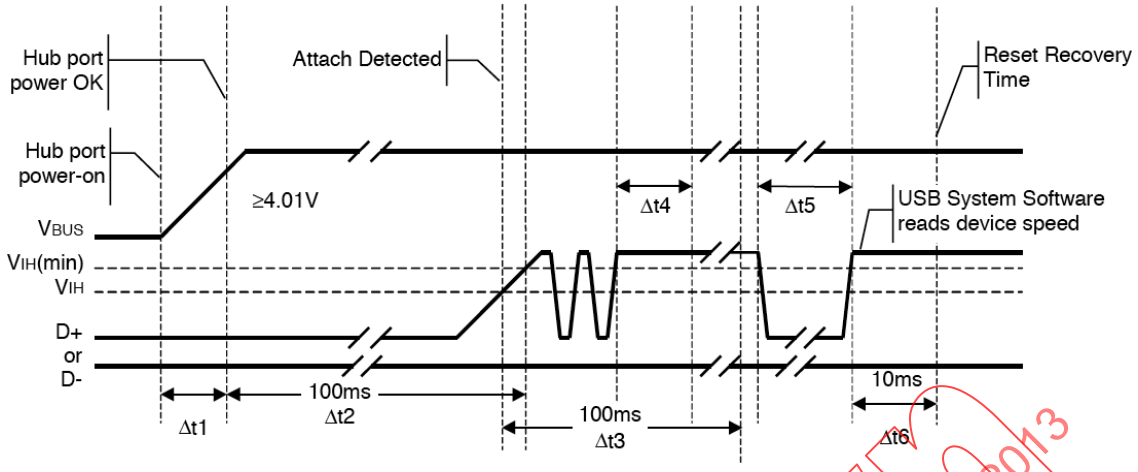


Figure 7-29. Power-on and Connection Events Timing

Δt1 This is the amount of time required for the hub port power switch to operate. This delay is a function of the type of hub port switch. Hubs report this time in the hub descriptor (see Section 11.15.2.1), which can be read via a request to the Hub Controller (see Section 11.16.2.4). If a device were plugged into a non-switched or already-switched on port, Δt1 is equal to zero.

Δt2 (TSIGATT) This is the maximum time from when VBUS is up to valid level (4.01 V) to when a device has to signal attach. Δt2 represents the time required for the device's internal power rail to stabilize and for D+ or D- to reach VIH (min) at the hub. Δt2 must be less than 100 ms for all hub and device implementations. (This requirement only applies if the device is drawing power from the bus.)

Δt3 (TATTDB) This is a debounce interval with a minimum duration of 100 ms that is provided by the USB System Software. It ensures that the electrical and mechanical connection is stable before software attempts to reset the attached device. The interval starts when the USB System Software is notified of a connection detection. The interval restarts if there is a disconnect. The debounce interval ensures that power is stable at the device for at least 100 ms before any requests will be sent to the device.

Δt4 (T2SUSP) Anytime a device observes no bus activity, it must obey the rules of going into suspend (see Section 7.1.7.6).

Δt5 (TDRST) This is the period of time hubs drive reset to a device. Refer to Section 7.1.7.5 and Section 11.5.1.5 for details.

Δt6 (TRSTRCY) The USB System Software guarantees a minimum of 10 ms for reset recovery. Device response to any bus transactions addressed to the default device address during the reset recovery time is undefined.

High-speed capable devices must initially attach as full-speed devices and must comply with all full-speed connection requirements. A high-speed capable downstream facing port must correctly detect the attachment of low-speed and full-speed devices and must also comply with all low-speed and full-speed connection behaviors.

Transition to high-speed signaling is accomplished by means of a low level electrical protocol which occurs during Reset. This protocol is specified in Section 7.1.7.5.

A downstream facing transceiver operating in high-speed mode detects disconnection of a high-speed device by sensing the doubling in differential signal amplitude across the D+ and D- lines that can occur when the device terminations are removed. The Disconnection Envelope Detector output goes high when the downstream facing transceiver transmits and positive reflections from the open line return with a phase which is additive with the transceiver driver signal. Signals with differential amplitudes ≥ 625 mV must reliably activate

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

the Disconnection Envelope Detector. Signals with differential amplitudes ≤ 525 mV must never activate the Disconnection Envelope Detector.

To assure that this additive effect occurs and is of sufficient duration to be detected, the EOP at the end of a high-speed SOF is lengthened to a continuous string of 40 bits without any transitions, as discussed in Section 7.1.13.2. This length is sufficient to guarantee that the voltage at the downstream facing port's connector will double, since the maximum allowable round trip signal delay is 30 bit times.

When a downstream facing port is transmitting in high-speed mode and detects that it has sent 32 bits without a transition, the disconnection envelope detector's output must be sampled once during transmission of the next 8 bits at the transceiver output. (In the absence of bus errors, the next 8 bits will not include a transition.) If the sample indicates that the disconnection detection threshold has been exceeded, the downstream facing port must indicate that the high-speed device has been disconnected. See Section 11.12.4.

7.1.7.4 Data Signaling

Data transmission within a packet is done with differential signals.

7.1.7.4.1 Low-/Full-Speed Signaling

The start of a packet (SOP) is signaled by the originating port by driving the D+ and D- lines from the Idle state to the opposite logic level (K state). This switch in levels represents the first bit of the SYNC field. Hubs must limit the change in the width of the first bit of SOP when it is retransmitted to less than ± 5 ns. Distortion can be minimized by matching the nominal data delay through the hub with the output enable delay of the hub.

The SE0 state is used to signal an end-of-packet (EOP). EOP will be signaled by driving D+ and D- to the SE0 state for two bit times followed by driving the lines to the J state for one bit time. The transition from the SE0 to the J state defines the end of the packet at the receiver. The J state is asserted for one bit time and then both the D+ and D- output drivers are placed in their high-impedance state. The bus termination resistors hold the bus in the Idle state. Figure 7-30 shows the signaling for start and end of a packet.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

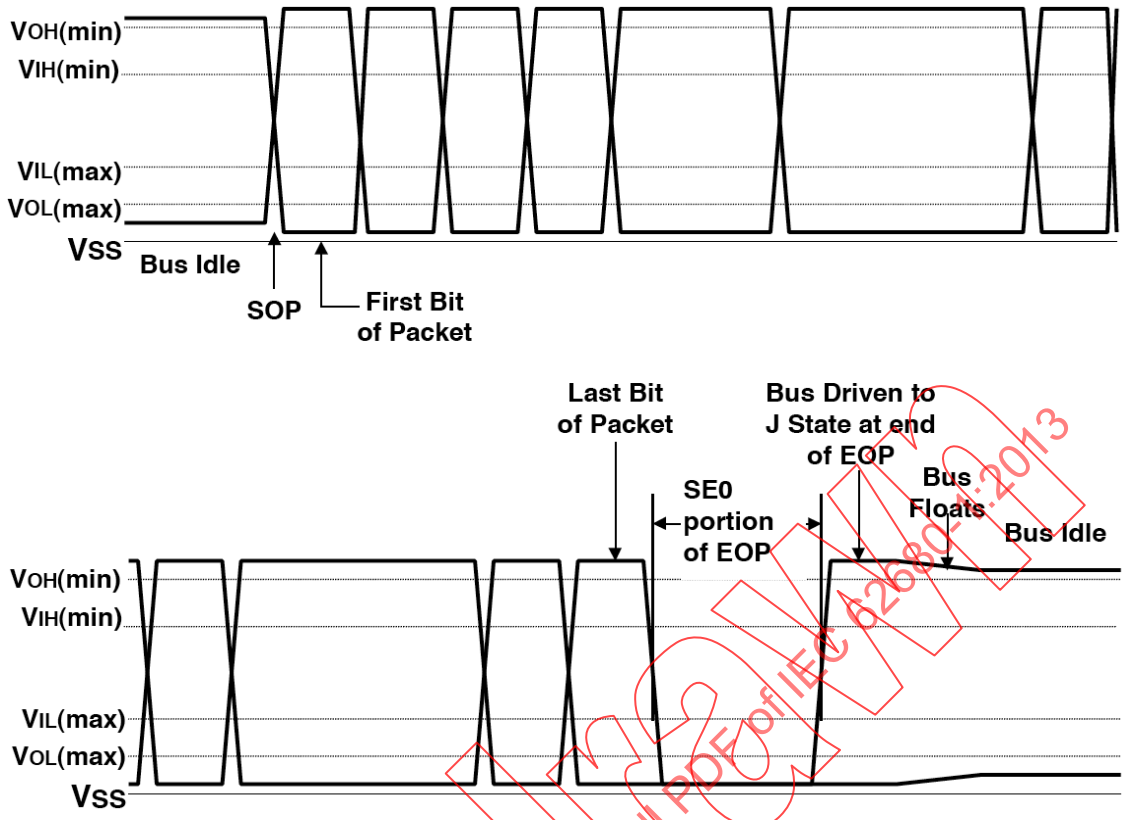


Figure 7-30. Low-/full-speed Packet Voltage Levels

7.1.7.4.2 High-speed Signaling

The high-speed Idle state is when both lines are nominally at GND.

The source of the packet signals the Start of Packet (SOP) in high-speed mode by driving the D+ and D- lines from the high-speed Idle state to the K state. This K is the first symbol of the SYNC pattern (NRZI sequence KJKJKJKJ KJKJKJKJ KJKJKJKJ KJKJKJKK) as described in Section 7.1.10.

The high-speed End of Packet (EOP) begins with a transition from the last symbol before the EOP to the opposite symbol. This opposite symbol is the first symbol in the EOP pattern (NRZ 01111111 with bit stuffing disabled) as described in Section 7.1.13.2. Upon completion of the EOP pattern, the driver ceases to inject current into the D+ or D- lines, and the lines return to the high-speed Idle state. The high-speed SOF EOP is a special case. This SOF EOP is 40 symbols without a transition (rather than 8 for a non-SOF packet).

The fact that the first symbol in the EOP pattern forces a transition simplifies the process of determining precisely which is the last bit in the packet prior to the EOP delimiter.

7.1.7.5 Reset Signaling

A hub signals reset to a downstream port by driving an extended SE0 at the port. After the reset is removed, the device will be in the Default state (refer to Section 9.1).

The reset signaling can be generated on any Hub or Host Controller port by request from the USB System Software. The reset signaling must be driven for a minimum of 10ms (T_{DRST}). After the reset, the hub port will transition to the Enabled state (refer to Section 11.5).

As an additional requirement, Host Controllers and the USB System Software must ensure that resets issued to the root ports drive reset long enough to overwhelm any concurrent resume attempts by downstream devices. It is required that resets from root ports have a

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

duration of at least 50 ms (TDRSTR). It is not required that this be 50 ms of continuous Reset signaling. However, if the reset is not continuous, the interval(s) between reset signaling must be less than 3 ms (TRHRSI), and the duration of each SE0 assertion must be at least 10 ms (TDRST).

A device operating in low-/full-speed mode that sees an SE0 on its upstream facing port for more than 2.5 µs (TDETRST) may treat that signal as a reset. The reset must have taken effect before the reset signaling ends.

Hubs will propagate traffic to a newly reset port after the port is in the Enabled state. The device attached to this port must recognize this bus activity and keep from going into the Suspend state.

Hubs must be able to accept all hub requests and devices must be able to accept a SetAddress() request (refer to Section 11.24.2 and Section 9.4 respectively) after the reset recovery time 10 ms (TRSTRCY) after the reset is removed. Failure to accept this request may cause the device not to be recognized by the USB system software. Hubs and devices must complete commands within the times specified in Chapter 9 and Chapter 11.

Reset must wake a device from the Suspend state.

It is required that a high-speed capable device can be reset while in the Powered, Default, Address, Configured, or Suspended states shown in Figure 9-1. The reset signaling is compatible with low-/full-speed reset. This means that a hub must successfully reset any device (even USB 1.X devices), and a device must be successfully reset by any hub (even USB1.X hubs).

If, and only if, a high-speed capable device is reset by a high-speed capable hub which is not high-speed disallowed, both hub and device must be operating in the default state in high-speed signaling mode at the end of reset. The hub port status register must indicate that the port is in high-speed signaling mode. This requirement is met by having such a device and such a hub engage in a low level protocol during the reset signaling time. The protocol is defined in such a way that USB 1.X devices will not be disrupted from their normal reset behaviors.

Note: Because the downstream facing port will not be in Transmit state during the Reset Protocol, high-speed Chirp signaling levels will not provoke disconnect detection. (Refer to Section 7.1.7.3 and Section 11.5.1.7.)

Reset Protocol for high-speed capable hubs and devices

1. The hub checks to make sure the attached device is not low-speed. (A low-speed device is not allowed to support high-speed operation. If the hub determines that it is attached to a low-speed device, it does not conduct the following high-speed detection protocol during reset.)
2. The hub drives SE0. In this description of the Reset Protocol and High-speed Detection Handshake, the start of SE0 is referred to as time T0.
3. The device detects assertion of SE0.
   a) If the device is being reset from suspend, then the device begins a high-speed detection handshake after the detection of SE0 for no less than 2.5 µs (TFILTSE0). Since a suspended device will generally have its clock oscillator disabled, the detection of SE0 will cause the oscillator to be restarted. The clock must be useable (although not necessarily settled to 500 ppm accuracy) in time to detect the high-speed hub chirp as described in Step 8.
   b) If the device is being reset from a non-suspended full-speed state, then the device begins a high-speed detection handshake after the detection of SE0 for no less than 2.5 µs and no more than 3.0 ms (TWTRSTFS).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

c) If the device is being reset from a non-suspended high-speed state, then the device must wait no less than 3.0 ms and no more than 3.125 ms (TWTREV) before reverting to full-speed. Reversion to full-speed is accomplished by removing the high-speed termination and reconnecting the D+ pull-up resistor. The device samples the bus state, and checks for SE0 (reset as opposed to suspend), no less than 100 µs and no more than 875 µs (TWTRSTHS) after starting reversion to full-speed. If SE0 (reset) is detected, then the device begins a high-speed detection handshake.

High-speed Detection Handshake (not performed if low-speed device detected by hub):

Note: In the following handshake, both the hub and device are required to detect Chirp J's and K's of specified minimum durations. It is strongly recommended that "gaps" in these Chirp signals as short as 16 high-speed bit times should restart the duration timers.

4. The high-speed device leaves the D+ pull-up resistor connected, leaves the high-speed terminations disabled, and drives the high-speed signaling current into the D- line. This creates a Chirp K on the bus. The device chirp must last no less than 1.0 ms (TUCH) and must end no more than 7.0 ms (TUCHEND) after high-speed Reset time T0.
5. The hub must detect the device chirp after it has seen assertion of the Chirp K for no less than 2.5 µs (TFILT). If the hub does not detect a device chirp, it must continue the assertion of SE0 until the end of reset.
6. No more than 100 µs (TWTDCH) after the bus leaves the Chirp K state, the hub must begin to send an alternating sequence of Chirp K's and Chirp J's. There must be no Idle states on the bus between the J's and K's. This sequence must continue until a time (TDCHSE0) no more than 500 µs before and no less than 100 µs before the end of Reset. (This will guarantee that the bus remains active, preventing the device from entering the high-speed Suspend state.) Each individual Chirp K and Chirp J must last no less than 40 µs and no more than 60 µs (TDCHBIT).
7. After completing the hub chirp sequence, the hub asserts SE0 until end of Reset. At the end of reset, the hub must transition to the high-speed Enabled state without causing any transitions on the data lines.
8. After the device completes its chirp, it looks for the high-speed hub chirp. At a minimum, the device is required to see the sequence Chirp K-J-K-J-K-J in order to detect a valid hub chirp. Each individual Chirp K and Chirp J must be detected for no less than 2.5 µs (TFILT).

   a) If the device detects the sequence Chirp K-J-K-J-K-J, then no more than 500 µs (TWTHS) after detection, the device is required to disconnect the D+ pull-up resistor, enable the high-speed terminations, and enter the high-speed Default state.

   b) If the device has not detected the sequence Chirp K-J-K-J-K-J by a time no less than 1.0 ms and no more than 2.5 ms (TWTFS) after completing its own chirp, then the device is required to revert to the full-speed Default state and wait for the end of Reset.

7.1.7.6 Suspending

All devices must support the Suspend state. Devices can go into the Suspend state from any powered state. They begin the transition to the Suspend state after they see a constant Idle state on their upstream facing bus lines for more than 3.0 ms. The device must actually be suspended, drawing only suspend current from the bus after no more than 10 ms of bus inactivity on all its ports. Any bus activity on the upstream facing port will keep a device out of the Suspend state. In the absence of any other bus traffic, the SOF token (refer to Section 8.4.3) will occur once per (micro)frame to keep full-/high-speed devices from suspending. In the absence of any low-speed traffic, low-speed devices will see at least one keep-alive (defined in Table 7-2) in every frame in which an SOF occurs, which keeps them from suspending. Hubs generate this keep-alive as described in Section 11.8.4.1.

While in the Suspend state, a device must continue to provide power to its D+ (full-/high-speed) or D- (low-speed) pull-up resistor to maintain an idle so that the upstream hub can maintain the correct connectivity status for the device.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Additional Requirements for High-speed Capable Devices

From the perspective of a device operating in high-speed mode, a Reset and a Suspend are initially indistinguishable, so the first part of the device response is the same as for a Reset. When a device operating in high-speed mode detects that the data lines have been in the high-speed Idle state for at least 3.0 ms, it must revert to the full-speed configuration no later than 3.125 ms (TWTREV) after the start of the idle state. Reversion to full-speed is accomplished by disconnecting its termination resistors and reconnecting its D+ pull-up resistor. No earlier than 100 µs and no later than 875 µs (TWTRSTHS) after reverting to full-speed, the device must sample the state of the line. If the state is a full-speed J, the device continues with the Suspend process. (SE0 would have indicated that the downstream facing port was driving reset, and the device would have gone into the "High-speed Detection Handshake" as described in Section 7.1.7.5.)

A device or downstream facing port which is suspended from high-speed operation actually transitions to full-speed signaling during the suspend process, but is required to remember that it was operating in high-speed mode when suspended. When the resume occurs, the device or downstream facing transceiver must revert to high-speed as discussed in Section 7.1.7.7 without the need for a reset.

7.1.7.6.1 Global Suspend

Global suspend is used when no communication is desired anywhere on the bus and the entire bus is placed in the Suspend state. The host signals the start of global suspend by ceasing all its transmissions (including the SOF token). As each device on the bus recognizes that the bus is in the Idle state for the appropriate length of time, it goes into the Suspend state.

After 3.0 ms of continuous idle state, a downstream facing transceiver operating in high-speed must revert to the full-speed idle configuration (high-speed terminations disabled), but it does not enable full-speed disconnect detection until 1.0 ms later. This is to make sure that the device has returned to the full-speed Idle state prior to the enabling of full-speed disconnect detection, thereby preventing an unintended disconnect detection. After re-enabling the full-speed disconnect detection mechanism, the hub continues with the suspend process.

7.1.7.6.2 Selective Suspend

Segments of the bus can be selectively suspended by sending the command SetPortFeature(PORT_SUSPEND) to the hub port to which that segment is attached. The suspended port will block activity to the suspended bus segment, and devices on that segment will go into the Suspend state after the appropriate delay as described above.

When a downstream facing port operating in high-speed mode receives the SetPortFeature(PORT_SUSPEND) command, the port immediately reverts to the full-speed Idle state and blocks any activity to the suspend segment. Full-speed disconnect detection is disabled until the port has been in full-speed idle for 4.0 ms. This prevents an unintended disconnect detection. After re-enabling the full-speed disconnect detection mechanism, the hub continues with the suspend process.

Section 11.5 describes the port Suspend state and its interaction with the port state machine. Suspend is further described in Section 11.9.

7.1.7.7 Resume

If a device is in the Suspend state, its operation is resumed when any non-idle signaling is received on its upstream facing port. Additionally, the device can signal the system to resume operation if its remote wakeup capability has been enabled by the USB System Software. Resume signaling is used by the host or a device to bring a suspended bus segment back to the active condition. Hubs play an important role in the propagation and generation of resume signaling. The following description is an outline of a general global resume sequence. A

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

complete description of the resume sequence, the special cases caused by selective suspend, and the role of the hub are given in Section 11.9.

The host may signal resume (TDRSMDN) at any time. It must send the resume signaling for at least 20 ms and then end the resume signaling in one of two ways, depending on the speed at which its port was operating when it was suspended. If the port was in low-/full-speed when suspended, the resume signaling must be ended with a standard, low-speed EOP (two low-speed bit times of SE0 followed by a J). If the port was operating in high-speed when it was suspended, the resume signaling must be ended with a transition to the high-speed idle state.

The 20 ms of resume signaling ensures that all devices in the network that are enabled to see the resume are awakened. The connectivity established by the resume signaling is torn down by the End of Resume, which prepares the hubs for normal operation. After resuming the bus, the host must begin sending bus traffic (at least the SOF token) within 3 ms of the start of the idle state to keep the system from going back into the Suspend state.

A device with remote wakeup capability may not generate resume signaling unless the bus has been continuously in the Idle state for 5 ms (TWTRSM). This allows the hubs to get into their Suspend state and prepare for propagating resume signaling. The remote wakeup device must hold the resume signaling for at least 1 ms but for no more than 15 ms (TDRSMUP). At the end of this period, the device stops driving the bus (puts its drivers into the high-impedance state and does not drive the bus to the J state).

If the hub upstream of a remote wakeup device is suspended, it will propagate the resume signaling to its upstream facing port and to all of its enabled downstream facing ports, including the port that originally signaled the resume. When a hub is propagating resume signaling from a downstream device, it may transition from the idle state to K with a risetime faster than is normally allowed. The hub must begin this rebroadcast (TURSM) of the resume signaling within 1 ms of receiving the original resume. The resume signal will propagate in this manner upstream until it reaches the host or a non-suspended hub (refer to Section 11.9), which will reflect the resume downstream and take control of resume timing. This hub is termed the controlling hub. Intermediate hubs (hubs between the resume initiator and the controlling hub) drive resume (TDRSMUP) on their upstream facing port for at least 1 ms during which time they also continue to drive resume on enabled downstream facing ports. An intermediate hub will stop driving resume on the upstream facing port and reverse the direction of connectivity from upstream to downstream within 15 ms after first asserting resume on its upstream facing port. When all intermediate hubs have reversed connectivity, resume is being driven from the controlling hub through all intermediate hubs and to all enabled ports. The controlling hub must rebroadcast the resume signaling within 1 ms (TURSM) and ensures that resume is signaled for at least 20 ms (TDRSMDN). The hub may then begin normal operation by terminating the resume process as described above.

The USB System Software must provide a 10 ms resume recovery time (TRSMRCY) during which it will not attempt to access any device connected to the affected (just-activated) bus segment.

Port connects and disconnects can also cause a hub to send a resume signal and awaken the system. These events will cause a hub to send a resume signal only if the hub has been enabled as a remote-wakeup source. Refer to Section 11.4.4 for more details.

Refer to Section 7.2.3 for a description of power control during suspend and resume.

If the hub port and device were operating in high-speed prior to suspend, they are required to "remember" that they were previously operating in high-speed, and they must transition back to high-speed operation, without arbitration, within two low-speed bit times of the K to SE0 transition. The inactivity timers must be started two low-speed bit times after the K to SE0 transition. Note that the transition from SE0 to J which would normally occur at the end of full-speed resume signaling is omitted if the link was operating in high-speed at the time when it was suspended.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

It is required that the host begin sending SOF's in time to prevent the high-speed tree from suspending.

7.1.8 Data Encoding/Decoding

The USB employs NRZI data encoding when transmitting packets. In NRZI encoding, a “1” is represented by no change in level and a “0” is represented by a change in level. Figure 7-31 shows a data stream and the NRZI equivalent. The high level represents the J state on the data lines in this and subsequent figures showing NRZI encoding. A string of zeros causes the NRZI data to toggle each bit time. A string of ones causes long periods with no transitions in the data.

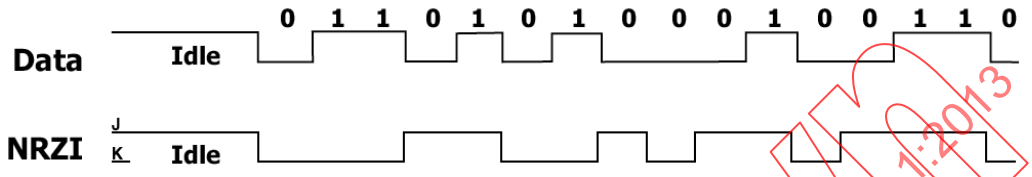


Figure 7-31. NRZI Data Encoding

7.1.9 Bit Stuffing

In order to ensure adequate signal transitions, bit stuffing is employed by the transmitting device when sending a packet on USB (see Figure 7-32 and Figure 7-34). A zero is inserted after every six consecutive ones in the data stream before the data is NRZI encoded, to force a transition in the NRZI data stream. This gives the receiver logic a data transition at least once every seven bit times to guarantee the data and clock lock. Bit stuffing is enabled beginning with the Sync Pattern. The data “one” that ends the Sync Pattern is counted as the first one in a sequence. Bit stuffing by the transmitter is always enforced, except during high-speed EOP. If required by the bit stuffing rules, a zero bit will be inserted even if it is the last bit before the end-of-packet (EOP) signal.

The receiver must decode the NRZI data, recognize the stuffed bits, and discard them.

7.1.9.1 Full-/low-speed

Full-/low-speed signaling uses bit stuffing throughout the packet without exception. If the receiver sees seven consecutive ones anywhere in the packet, then a bit stuffing error has occurred and the packet should be ignored. The time interval just before an EOP is a special case. The last data bit before the EOP can become stretched by hub switching skews. This is known as dribble and can lead to the case illustrated in Figure 7-33, which shows where dribble introduces a sixth bit that does not require a bit stuff. Therefore, the receiver must accept a packet for which there are up to six full bit times at the port with no transitions prior to the EOP.

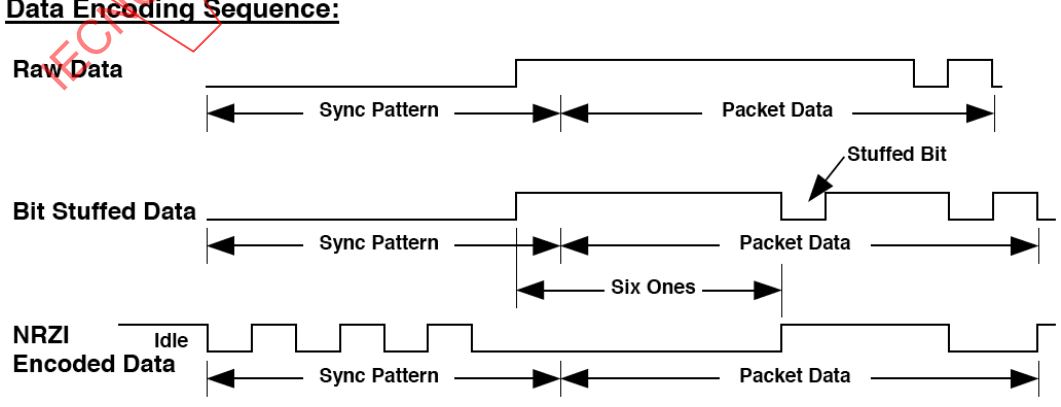


Figure 7-32. Bit Stuffing

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

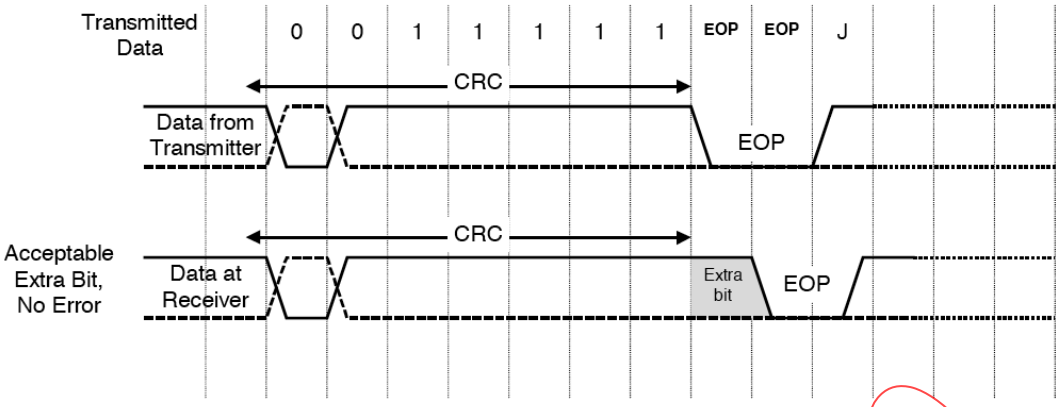


Figure 7-33. Illustration of Extra Bit Preceding EOP (Full-/low-speed)

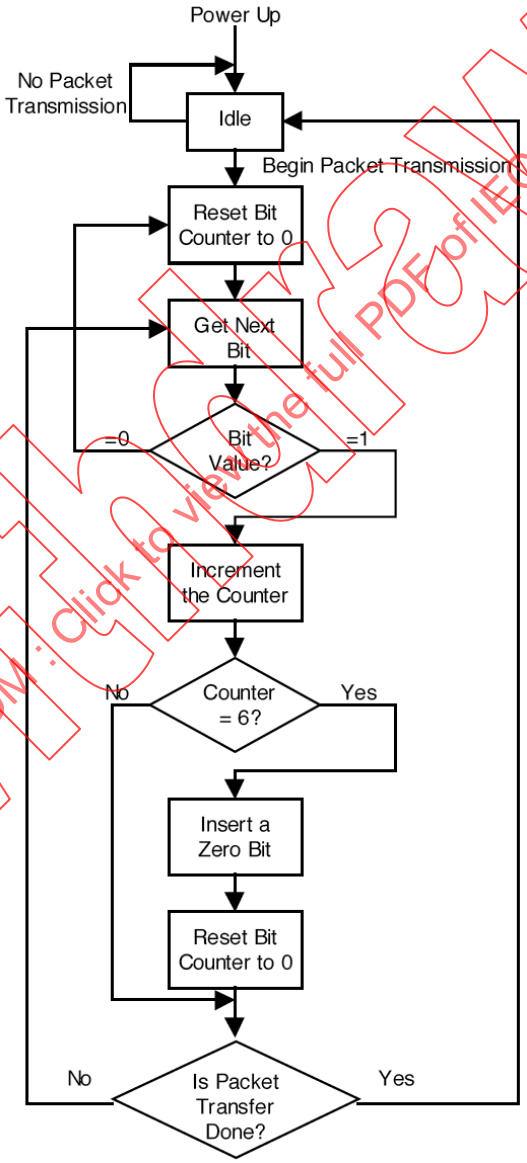


Figure 7-34. Flow Diagram for Bit Stuffing

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.1.9.2 High-Speed

High-speed signaling uses bit stuffing throughout the packet, with the exception of the intentional bit stuff errors used in the high-speed EOP as described in Section 7.1.13.2.

7.1.10 Sync Pattern

The SYNC pattern used for low-/full-speed transmission is required to be 3 KJ pairs followed by 2 K's for a total of eight symbols. Figure 7-35 shows the NRZI bit pattern, which is prefixed to each low-/full-speed packet.

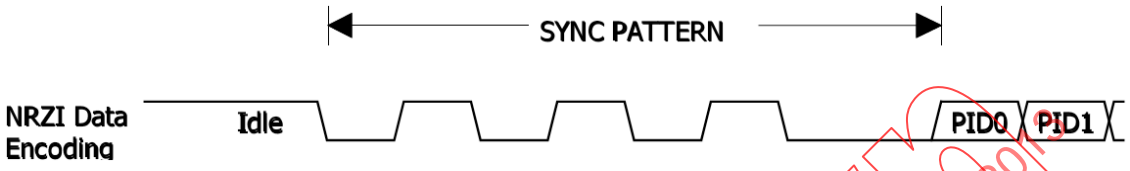


Figure 7-35. Sync Pattern (Low-/full-speed)

The SYNC pattern used for high-speed transmission is required to be 15 KJ pairs followed by 2 K's, for a total of 32 symbols. Hubs are allowed to drop up to 4 bits from the start of the SYNC pattern when repeating packets. Hubs must not corrupt any repeated bits of the SYNC field, however. Thus, after being repeated by 5 hubs, a packet's SYNC field may be as short as 12 bits.

7.1.11 Data Signaling Rate

The high-speed data rate (THSDRAT) is nominally 480.00 Mb/s, with a required bit rate accuracy of ±500 ppm. For hosts, hubs, and high-speed capable functions, the required data-rate accuracy when transmitting at any speed is ±0.05% (500 ppm). The full-speed rate for such hubs and functions is TFDRATHS. The low-speed rate for such hubs is TLDRATHS (a low-speed function must not support high-speed).

The full-speed data rate is nominally 12.000 Mb/s. For full-speed only functions, the required data-rate when transmitting (TFDRATE) is 12.000 Mb/s ±0.25% (2,500 ppm).

The low-speed data rate is nominally 1.50 Mb/s. For low-speed functions, the required data-rate when transmitting (TLDRATE) is 1.50 Mb/s ±1.5% (15,000 ppm). This allows the use of resonators in low cost, low-speed devices.

Hosts and hubs must be able to receive data from any compliant low-speed, full-speed, or high-speed source. High-speed capable functions must be able to receive data from any compliant full-speed or high-speed source. Full-speed only functions must be able to receive data from any compliant full-speed source. Low-speed only functions must be able to receive data from any compliant low-speed source.

The above accuracy numbers include contributions from all sources:

- Initial frequency accuracy
- Crystal capacitive loading
- Supply voltage on the oscillator
- Temperature
- Aging

7.1.12 Frame Interval

The USB defines a frame interval (TFRAME) to be 1.000 ms ±500 ns long. The USB defines a microframe interval (THSFRAM) to be 125.0 µs ±62.5 ns long. The (micro)frame interval is measured from any point in an SOF token in one (micro)frame to the same point in the SOF token of the next (micro)frame.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Since the Host Controller and hubs must meet clock accuracy specification of ±0.05%, they will automatically meet the frame interval requirements without the need for adjustment.

The frame interval repeatability, TRFI (difference in frame interval between two successive frames), must be less than 0.5 full-speed bit times. The microframe interval repeatability, THSRFI (difference in the microframe interval between two successive microframes, measured at the host), must be less than 4 high-speed bit times. Each hub may introduce at most 4 additional high-speed bits of microframe jitter.

Hubs and certain full-/high-speed functions need to track the (micro)frame interval. They also are required to have sufficient frame timing adjustment to compensate for their own frequency inaccuracy.

7.1.13 Data Source Signaling

This section covers the timing characteristics of data produced and sent from a port (the data source). Section 7.1.14 covers the timing characteristics of data that is transmitted through the Hub Repeater section of a hub. In this section, TPERIOD is defined as the actual period of the data rate that can have a range as defined in Section 7.1.11.

7.1.13.1 Data Source Jitter

This section describes the maximum allowable data source jitter for low-speed, full-speed, and high-speed signaling.

7.1.13.1.1 Low-/full-speed Data Source Jitter

The source of data can have some variation (jitter) in the timing of edges of the data transmitted. The time between any set of data transitions is (N * TPERIOD) ± jitter time, where 'N' is the number of bits between the transitions. The data jitter is measured with the same load used for maximum rise and fall times and is measured at the crossover points of the data lines, as shown in Figure 7-36.

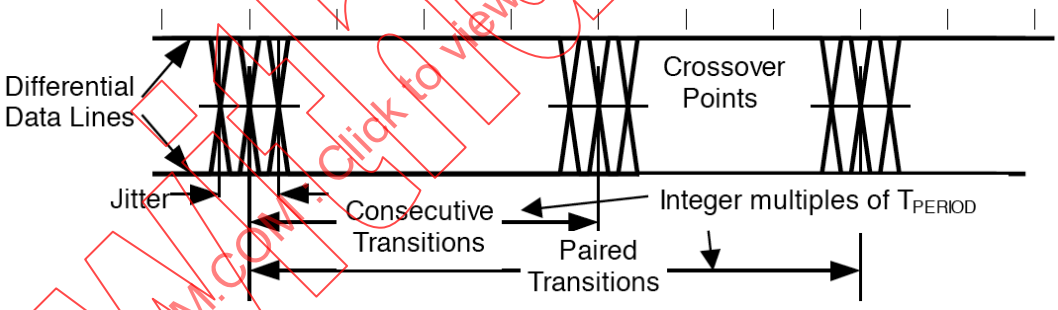


Figure 7-36. Data Jitter Taxonomy

- For full-speed transmissions, the jitter time for any consecutive differential data transitions must be within ±2.0 ns and within ±1.0 ns for any set of paired (JK-to-next JK transition or KJ-to-next KJ transition) differential data transitions.
- For low-speed transmissions, the jitter time for any consecutive differential data transitions must be within ±25 ns and within ±10 ns for any set of paired differential data transitions.

These jitter numbers include timing variations due to differential buffer delay and rise and fall time mismatches, internal clock source jitter, and noise and other random effects.

7.1.13.1.2 High-speed Data Source Jitter

High-speed data within a single packet must be transmitted with no more jitter than is allowed by the eye patterns defined in Section 7.1.2 when measured over a sliding window of 480 high-speed bit times.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.1.13.2 EOP Width

This section describes low-speed, full-speed, and high-speed EOP width.

7.1.13.2.1 Low-/full-speed EOP

The width of the SE0 in the EOP is approximately 2 * TPERIOD. The SE0 width is measured with the same load used for maximum rise and fall times and is measured at the same level as the differential signal crossover points of the data lines (see Figure 7-37).

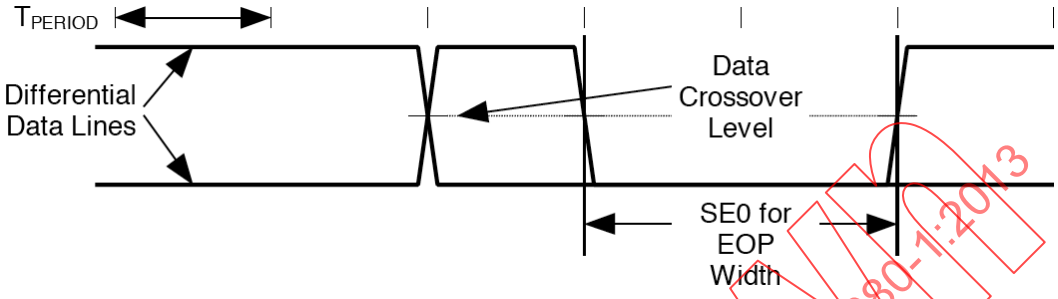


Figure 7-37. SE0 for EOP Width Timing

- For full-speed transmissions, the SE0 for EOP width from the transmitter must be between 160 ns and 175 ns.
- For low-speed transmissions, the transmitter's SE0 for EOP width must be between 1.25 μs and 1.50 μs.

These ranges include timing variations due to differential buffer delay and rise and fall time mismatches and to noise and other random effects.

A receiver must accept any valid EOP. Receiver design should note that the single-ended input threshold voltage can be different from the differential crossover voltage and the SE0 transitions will in general be asynchronous to the clock encoded in the NRZI stream.

- A full-speed EOP may have the SE0 interval reduced to as little as 82 ns (TFEOPR) and a low-speed SE0 interval may be as short as 670 ns (TLEOPR).

A hub may tear down connectivity if it sees an SE0 of at least TFST or TLST followed by a transition to the J state. A hub must tear down connectivity on any valid EOP.

7.1.13.2.2 High-speed EOP

In high-speed signaling, a bit stuff error is intentionally generated to indicate EOP. A receiver is required to interpret any bit stuff error as an EOP.

For high-speed packets other than SOF's, the transmitted EOP delimiter is required to be an NRZ byte of 01111111 without bit stuffing. For example, if the last symbol prior to the EOP field is a J, this would lead to an EOP of KKKKKKKK.

For high-speed SOF's, the transmitted EOP delimiter is required to be 5 NRZ bytes without bit stuffing, consisting of 01111111 11111111 11111111 11111111 11111111. Thus if the last bit prior to the EOP field is a J, this would lead to 40 K's on the wire, at the end of which the lines must return to the high-speed Idle state. This extra EOP length is of no significance to a receiver; it is used for disconnect detection as discussed in Section 7.1.7.3.

A hub may add at most 4 random bits to the end of the EOP field when repeating a packet. Thus after 5 repeaters, a packet can have up to 20 random bits following the EOP field. A hub, however, must not corrupt any of the 8 (or 40 in the case of a SOF) required bits of the EOP field.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.1.14 Hub Signaling Timings

This section describes low-speed, full-speed, and high-speed hub signaling timings.

7.1.14.1 Low-/full-speed Hub Signaling Timings

The propagation of a full-speed, differential data signal through a hub is shown in Figure 7-38. The downstream signaling is measured without a cable connected to the port and with the load used for measuring rise and fall times. The total delay through the upstream cable and hub electronics must be a maximum of 70 ns (T_{HDD1}). If the hub has a detachable USB cable, then the delay (T_{HDD2}) through hub electronics and the associated transmission line must be a maximum of 44 ns to allow for a maximum cable delay of 26 ns (T_{FSCBL}). The delay through this hub is measured in both the upstream and downstream directions, as shown in Figure 7-38B, from data line crossover at the input port to data line crossover at the output port.

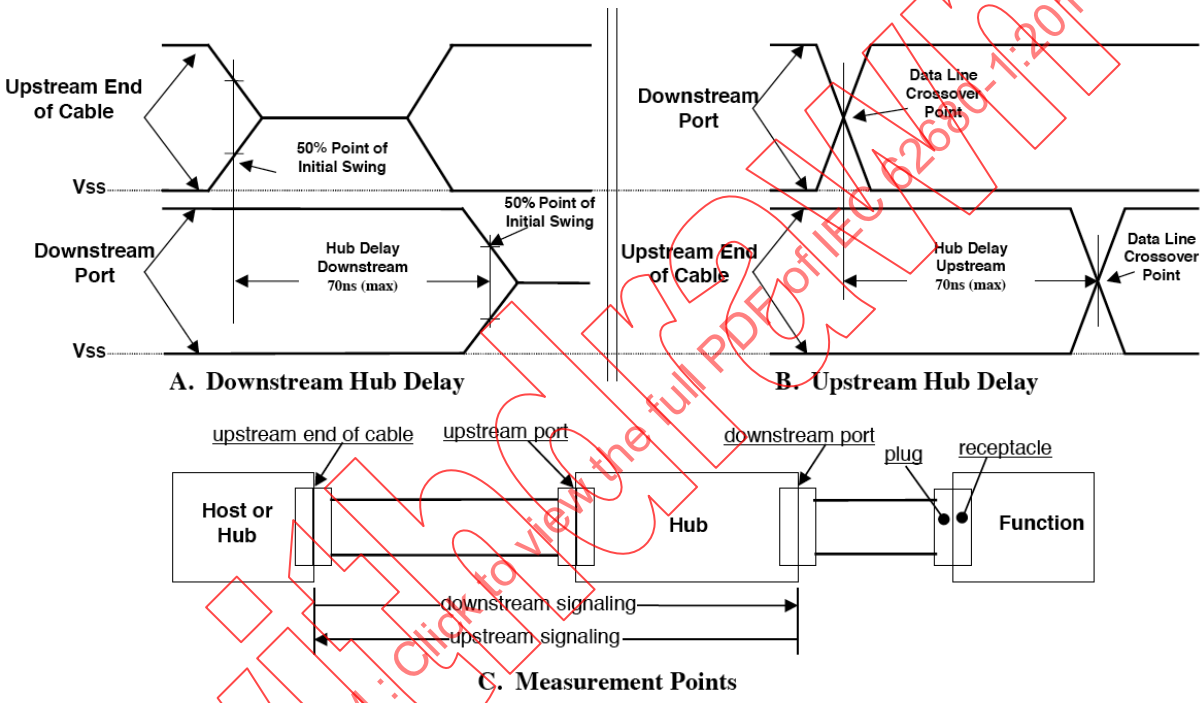


Figure 7-38. Hub Propagation Delay of Full-speed Differential Signals

Low-speed propagation delay for differential signals is measured in the same fashion as for full-speed signaling. The maximum low-speed hub delay is 300 ns (T_{LHDD}). This allows for the slower low-speed buffer propagation delay and rise and fall times. It also provides time for the hub to re-clock the low-speed data in the upstream direction.

When the hub acts as a repeater, it must reproduce the received, full-speed signal accurately on its outputs. This means that for differential signals, the propagation delays of a J-to-K state transition must match closely to the delays of a K-to-J state transition. For full-speed propagation, the maximum difference allowed between these two delays (T_{HDJ1}) (see Figure 7-38 and Figure 7-52) for a hub plus cable is ±3.0 ns. Similarly, the difference in delay between any two J-to-K or K-to-J transitions through a hub (T_{HDJ2}) must be less than ±1.0 ns. For low-speed propagation in the downstream direction, the corresponding allowable jitter (T_{LDHJ1}) is ±45 ns and (T_{LDHJ2}) ±15 ns, respectively. For low-speed propagation in the upstream direction, the allowable jitter is ±45 ns in both cases (T_{LUHJ1} and T_{LUHJ2}).

An exception to this case is the skew that can be introduced in the Idle-to-K state transition at SOP (T_{FSOP} and T_{LSOP}) (refer to Section 7.1.7.4). In this case, the delay to the opposite port includes the time to enable the output buffer. However, the delays should be closely matched

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

to the normal hub delay and the maximum additional delay difference over a normal J-to-K transition is ±5.0 ns. This limits the maximum distortion of the first bit in the packet.

Note: Because of this distortion of the SOP transition relative to the next K-to-J state transition, the first SYNC field bit should not be used to synchronize the receiver to the data stream.

The EOP must be propagated through a hub in the same way as the differential signaling. The propagation delay for sensing an SE0 must be no less than the greater of the J-to-K or K-to-J differential data delay (to avoid truncating the last data bit in a packet), but not more than 15 ns greater than the larger of these differential delays at full-speed and 200 ns at low-speed (to prevent creating a bit stuff error at the end of the packet). EOP delays are shown in Figure 7-53.

Because the sense levels for the SE0 state are not at the midpoint of the signal swing, the width of SE0 state will be changed as it passes through each hub. A hub may not change the width of the SE0 state in a full-speed EOP by more than ±15 ns (TFHESK), as measured by the difference of the leading edge and trailing edge delays of the SE0 state (see Figure 7-53). An SE0 from a low-speed device has long rise and fall times and is subject to greater skew, but these conditions exist only on the cable from the low-speed device to the port to which it is connected. Thereafter, the signaling uses full-speed buffers and their faster rise and fall times. The SE0 from the low-speed device cannot be changed by more than ±300 ns (TLHESK) as it passes through the hub to which the device is connected. This time allows for some signal conditioning in the low-speed transceiver to reduce its sensitivity to noise.

7.1.14.2 High-speed Hub Signaling Timings

When a hub acts as a repeater for high-speed data, the delay of the hub (THSHDD) must not exceed 36 high-speed bit times plus 4 ns (the trace delays allowed for the hub circuit board). This delay is measured from the last bit of the SYNC field at the input connector to the last bit of the SYNC field at the output connector.

A high-speed hub repeater must digitally resynchronize the buffered data, so there is no allowance for cumulative jitter (within a single packet) as a high-speed packet passes through multiple repeater stages. Within a single packet, the jitter must not exceed the eye pattern templates defined in Section 7.1.2 over a sliding window of 480 high-speed bit times.

Due to the data synchronization process, the propagation delay of a hub repeater is allowed to vary at most 5 high-speed bit times (THSHDV). The delay including this allowed variation must not exceed 36 high-speed bit times plus 4 ns. (This allows for some uncertainty as to when an incoming packet arrives at the hub with respect to the phase of the synchronization clock.)

7.1.15 Receiver Data Jitter

This section describes low-speed, full-speed, and high-speed receiver data jitter.

7.1.15.1 Low-/full-speed Receiver Data Jitter

The data receivers for all types of devices must be able to properly decode the differential data in the presence of jitter. The more of the bit cell that any data edge can occupy and still be decoded, the more reliable the data transfer will be. Data receivers are required to decode differential data transitions that occur in a window plus and minus a nominal quarter bit cell from the nominal (centered) data edge position. (A simple 4X over-sampling state machine DPLL can be built that satisfies these requirements.) This requirement is derived in Table 7-4 and Table 7-5. The tables assume a worst-case topology of five hubs between the host and device and the worst-case number of seven bits between transitions. The derived numbers are rounded up for ease of specification.

Jitter will be caused by the delay mismatches discussed above and by mismatches in the source and destination data rates (frequencies). The receive data jitter budgets for full- and

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

low-speed are given in Table 7-4 and Table 7-5. These tables give the value and totals for each source of jitter for both consecutive (next) and paired transitions. Note that the jitter component related to the source or destination frequency tolerance has been allocated to the appropriate device (i.e., the source jitter includes bit shifts due to source frequency inaccuracy over the worst-case data transition interval). The output driver jitter can be traded off against the device clock accuracy in a particular implementation as long as the jitter specification is met.

The low-speed jitter budget table has an additional line in it because the jitter introduced by the hub to which the low-speed device is attached is different from all the other devices in the data path. The remaining devices operate with full-speed signaling conventions (though at low-speed data rate).

Table 7-4. Full-speed Jitter Budget

Jitter Source	Full-speed			
	Next Transition		Paired Transition	
	Each (ns)	Total (ns)	Each (ns)	Total (ns)
Source Driver Jitter	2.0	2.0	1.0	1.0
Source Frequency Tolerance (worst-case)	0.21/bit	1.5	0.21/bit	3.0
Source Jitter Total		**3.5**		**4.0**
Hub Jitter	3.0	15.0	1.0	5.0
Jitter Specification		**18.5**		**9.0**
Destination Frequency Tolerance	0.21/bit	1.5	0.21/bit	3.0
Receiver Jitter Budget		**20.0**		**12.0**

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 7-5. Low-speed Jitter Budget

Jitter Source	Low-speed Upstream			
	Next Transition		Paired Transition	
	Each (ns)	Total (ns)	Each (ns)	Total (ns)
Function Driver Jitter	25.0	25.0	10.0	10.0
Function Frequency Tolerance (worst-case)	10.0/bit	70.0	10.0/bit	140.0
Source (Function) Jitter Total		**95.0**		**150.0**
Hub with Low-speed Device Jitter	45.0	45.0	45.0	45.0
Remaining (full-speed) Hubs' Jitter	3.0	12.0	1.0	4.0
Jitter Specification		**152.0**		**199.0**
Host Frequency Tolerance	1.7/bit	12.0	1.7/bit	24.0
Host Receiver Jitter Budget		**164.0**		**223.0**
	Low-speed Downstream			
	Next Transition		**Paired Transition**	
	Each (ns)	**Total (ns)**	**Each (ns)**	**Total (ns)**
Host Driver Jitter	2.0	2.0	1.0	1.0
Host Frequency Tolerance (worst-case)	1.7/bit	12.0	1.7/bit	24.0
Source (Host) Jitter Total		**14.0**		**25.0**
Hub with Low-speed Device Jitter	45.0	45.0	15.0	15.0
Remaining (full-speed) Hubs' Jitter	3.0	12.0	1.0	4.0
Jitter Spec		**71.0**		**44.0**
Function Frequency Tolerance	10.0/bit	70.0	10.0/bit	140.0
Function Receiver Jitter Budget		**141.0**		**184.0**

Note: This table describes the host transmitting at low-speed data rate using full-speed signaling to a low-speed device through the maximum number of hubs. When the host is directly connected to the low-speed device, it uses low-speed data rate and low-speed signaling, and the host has to meet the source jitter listed in the “Jitter Specification” row.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.1.15.2 High-speed Receiver Data Jitter

A high-speed capable receiver must reliably recover high-speed data when the waveforms at its inputs conform to the receiver sensitivity eye pattern templates. The templates, which are called out in Section 7.1.2.2, specify the horizontal and vertical eye pattern opening over a 480 bit time sliding window over the duration of a packet. Thus, for example, a high-speed receiver within a function must reliably recover data with a peak to peak jitter of 30%, measured at its B receptacle (as described by Template 4).

Such conformance is tested using Test Mode Test_Packet, as defined in Section 7.1.20.

It is a recommended design guideline that a receiver’s BER should be <= 10^{-12} when the receiver sensitivity requirement is met.

7.1.16 Cable Delay

The maximum total one-way signal propagation delay allowed is 30 ns. The allocation for cable delay is 26 ns. A maximum delay of 3 ns is allowed from a Host or Hub Controller downstream facing transceiver to its exterior downstream facing connector, while a maximum delay of 1 ns is allowed from the upstream facing connector to the upstream facing transceiver of any device. For a standard USB detachable cable, the cable delay is measured from the Series A connector pins to the Series B connector pins and is no more than 26 ns.

For other cables, the delay is measured from the series A connector to the point where the cable is connected to the device. The cable delay must also be less than 5.2 ns per meter.

The maximum one-way data delay on a full-speed cable is measured as shown in Figure 7-39.

One-way cable delay for low-speed cables must be less than 18 ns. It is measured as shown in Figure 7-40.

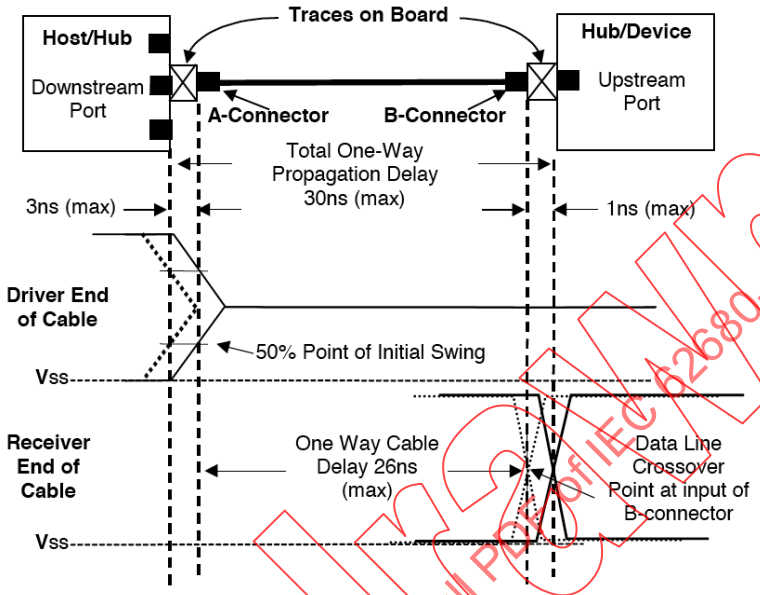


Figure 7-39. Full-speed Cable Delay

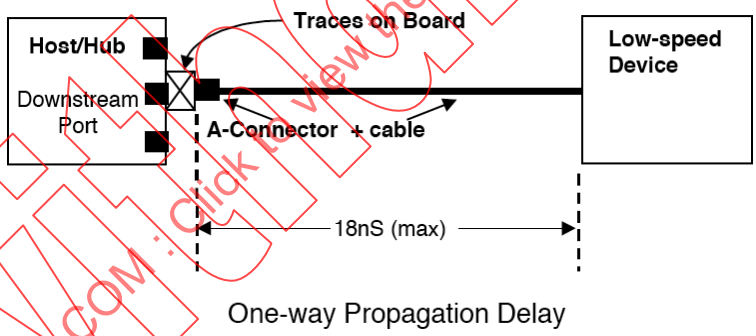


Figure 7-40. Low-speed Cable Delay

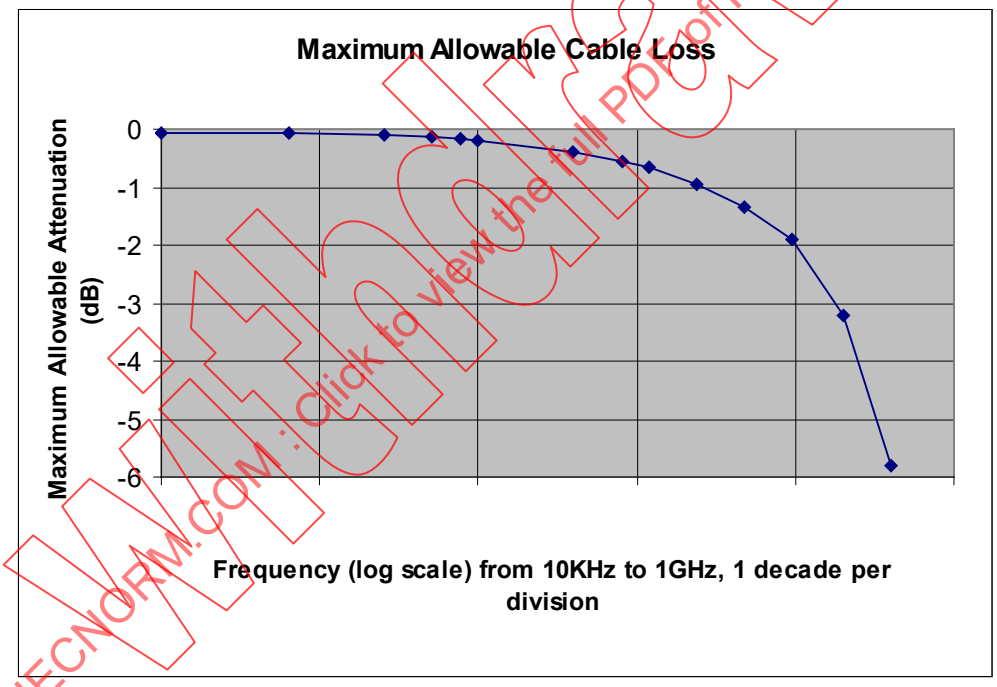
7.1.17 Cable Attenuation

USB cables must not exceed the loss figures shown in Table 7-6. Between the frequencies called out in the table, the cable loss should be no more than is shown in the accompanying graph.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-6. Maximum Allowable Cable Loss

Frequency (MHz)	Attenuation (maximum) dB/cable
0.064	0.08
0.256	0.11
0.512	0.13
0.772	0.15
1.000	0.20
4.000	0.39
8.000	0.57
12.000	0.67
24.000	0.95
48.000	1.35
96.000	1.9
200.00	3.2
400.00	5.8



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.1.18 Bus Turn-around Time and Inter-packet Delay

This section describes low-speed, full-speed, and high-speed bus turn-around time and inter-packet delay.

7.1.18.1 Low-/Full-Speed Bus Turn-around Time and Inter-packet Delay

Inter-packet delays are measured from the SE0-to-J transition at the end of the EOP to the J-to-K transition that starts the next packet.

A device must provide at least two bit times of inter-packet delay. The delay is measured at the responding device with a bit time defined in terms of the response. This provides adequate time for the device sending the EOP to drive J for one bit time and then turn off its output buffers.

The host must provide at least two bit times of J after the SE0 of an EOP and the start of a new packet (TIPD). If a function is expected to provide a response to a host transmission, the maximum inter-packet delay for a function or hub with a detachable (TRSPIPD1) cable is 6.5 bit times measured at the Series B receptacle. If the device has a captive cable, the inter-packet delay (TRSPIPD2) must be less than 7.5 bit times as measured at the Series A plug. These timings apply to both full-speed and low-speed devices and the bit times are referenced to the data rate of the packet.

The maximum inter-packet delay for a host response is 7.5 bit times measured at the host's port pins. There is no maximum inter-packet delay between packets in unrelated transactions.

7.1.18.2 High-Speed Bus Turn-around Time and Inter-packet Delay

High-speed inter-packet delays are measured from time when the line returns to a high-speed Idle State at the end of one packet to when the line leaves the high-speed Idle State at the start of the next packet.

When transmitting after receiving a packet, hosts and devices must provide an inter-packet delay of at least 8 bit times (THSIPDOD) measured at their A or B connectors (receptacles or plugs).

Additionally, if a host is transmitting two packets in a row, the inter-packet delay must be a minimum of 88 bit times (THSIPDSD), measured at the host's A receptacle. This will guarantee an inter-packet delay of at least 32 bit times at all devices (when receiving back to back packets). The maximum inter-packet delay provided by a host is 192 bit times within a transaction (THSRSPIPD1) measured at the A receptacle. When a host responds to a packet from a device, it will provide an inter-packet delay of at most 192 bit times measured at the A receptacle. There is no maximum inter-packet delay between packets in unrelated transactions.

When a device with a detachable cable responds to a packet from a host, it will provide an inter-packet delay of at most 192 bit times measured at the B receptacle. If the device has a captive cable, it will provide an inter-packet delay of at most 192 bit times plus 52 ns (2 times the max cable length) measured at the cable's A plug (THSRSPIPD2).

7.1.19 Maximum End-to-end Signal Delay

This section describes low-speed, full-speed, and high-speed end-to-end delay.

7.1.19.1 Low-/full-speed End-to-end Signal Delay

A device expecting a response to a transmission will invalidate the transaction if it does not see the start-of-packet (SOP) transition within the timeout period after the end of the transmission (after the SE0-to-J state transition in the EOP). This can occur between an IN token and the following data packet or between a data packet and the handshake packet (refer to Chapter 8). The device expecting the response will not time out before 16 bit times but will timeout before 18 bit times (measured at the data pins of the device from the SE0-to-J transition at the end of the EOP). The host will wait at least 18 bit times for a response to start before it will start a new transaction.

Figure 7-41 depicts the configuration of six signal hops (cables) that results in allowable worst-case signal delay. The maximum propagation delay from the upstream end of a hub's cable to any downstream facing connector on that hub is 70 ns.

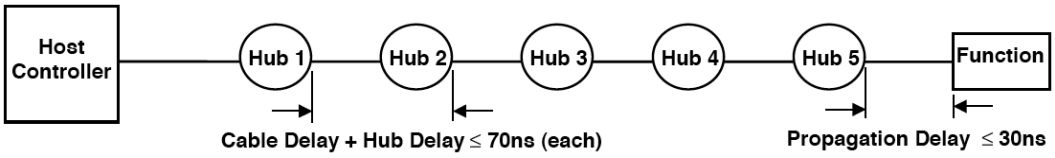


Figure 7-41. Worst-case End-to-end Signal Delay Model for Low-/full-speed

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.1.19.2 High-Speed End-to-end Delay

A high-speed host or device expecting a response to a transmission must not timeout the transaction if the inter-packet delay is less than 736 bit times, and it must timeout the transaction if no signaling is seen within 816 bit times.

These timeout limits allow a response to be seen even for the worst-case round trip signal delay. In high-speed mode, the worst-case round trip signal delay model is the sum of the following components:

12 max length cable delays (6 cables)	= 312 ns	
10 max delay hubs (5 hubs)	= 40 ns +	360 bit times
1 max device response time	=	192 bit times
Worst-case round trip delay	= 352 ns +	552 bit times = 721 bit times

7.1.20 Test Mode Support

To facilitate compliance testing, host controllers, hubs, and high-speed capable functions must support the following test modes:

- Test mode Test_SE0_NAK: Upon command, a port's transceiver must enter the high-speed receive mode and remain in that mode until the exit action is taken. This enables the testing of output impedance, low level output voltage, and loading characteristics. In addition, while in this mode, upstream facing ports (and only upstream facing ports) must respond to any IN token packet with a NAK handshake (only if the packet CRC is determined to be correct) within the normal allowed device response time. This enables testing of the device squelch level circuitry and, additionally, provides a general purpose stimulus/response test for basic functional testing.
- Test mode Test_J: Upon command, a port's transceiver must enter the high-speed J state and remain in that state until the exit action is taken. This enables the testing of the high output drive level on the D+ line.
- Test mode Test_K: Upon command, a port's transceiver must enter the high-speed K state and remain in that state until the exit action is taken. This enables the testing of the high output drive level on the D- line.
- Test mode Test_Packet: Upon command, a port must repetitively transmit the following test packet until the exit action is taken. This enables the testing of rise and fall times, eye patterns, jitter, and any other dynamic waveform specifications.

  The test packet is made up by concatenating the following strings. (Note: For J/K NRZI data, and for NRZ data, the bit on the left is the first one transmitted. "S" indicates that a bit stuff occurs, which inserts an "extra" NRZI data bit. "* N" is used to indicate N occurrences of a string of bits or symbols.)

NRZI Symbols (Fields)	NRZ Bit Strings	Number of NRZ Bits
{KJ * 15}, KK (SYNC)	{00000000 * 3}, 00000001	32
KKJKJKKK (DATA0 PID)	11000011	8
JKJKJKJK * 9	00000000 * 9	72
JJKKJJKK * 8	01010101 * 8	64
JJJJKKKK * 8	01110111 * 8	64
JJJJJJJKKKKKKK * 8	0, {111111S *15}, 111111	97
JJJJJJJK * 8	S, 111111S, {0111111S * 7}	55

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

NRZI Symbols (Fields)	NRZ Bit Strings	Number of NRZ Bits
{JKKKKKKK * 10}, JK	00111111, {S0111111 * 9}, S0	72
JJJKKKJJKKKKJKKK (CRC16)	0110110101110011	16
JJJJJJJJ (EOP)	01111111	8

A port in Test_Packet mode must send this packet repetitively. The inter-packet timing must be no less than the minimum allowable inter-packet gap as defined in Section 7.1.18 and no greater than 125 µs.

- Test mode Test_Force_Enable: Upon command, downstream facing hub ports (and only downstream facing hub ports) must be enabled in high-speed mode, even if there is no device attached. Packets arriving at the hub's upstream facing port must be repeated on the port which is in this test mode. This enables testing of the hub's disconnect detection; the disconnect detect bit can be polled while varying the loading on the port, allowing the disconnect detection threshold voltage to be measured.

Test Mode Entry and Exit

Test mode of a port is entered by using a device specific standard request (for an upstream facing port) or a port specific hub class request (for a downstream facing port). The device standard request SetFeature(TEST_MODE) is defined in Section 9.4.9. The hub class request SetPortFeature(PORT_TEST) is defined in Section 11.24.2.13. All high-speed capable devices/hubs must support these requests. These requests are not supported for non-high-speed devices.

The transition to test mode must be complete no later than 3 ms after the completion of the status stage of the request.

For an upstream facing port, the exit action is to power cycle the device. For a downstream facing port, the exit action is to reset the hub, as defined in Section 11.24.2.13.

7.2 Power Distribution

This section describes the USB power distribution specification.

7.2.1 Classes of Devices

The power source and sink requirements of different device classes can be simplified with the introduction of the concept of a unit load. A unit load is defined to be 100 mA. The number of unit loads a device can draw is an absolute maximum, not an average over time. A device may be either low-power at one unit load or high-power, consuming up to five unit loads. All devices default to low-power. The transition to high-power is under software control. It is the responsibility of software to ensure adequate power is available before allowing devices to consume high-power.

The USB supports a range of power sourcing and power consuming agents; these include the following:

- **Root port hubs**: Are directly attached to the USB Host Controller. Hub power is derived from the same source as the Host Controller. Systems that obtain operating power externally, either AC or DC, must supply at least five unit loads to each port. Such ports are called high-power ports. Battery-powered systems may supply either one or five unit loads. Ports that can supply only one unit load are termed low-power ports.
- **Bus-powered hubs**: Draw all of their power for any internal functions and downstream facing ports from VBUS on the hub's upstream facing port. Bus-powered hubs may only draw up to one unit load upon power-up and five unit loads after configuration. The configuration power is split between allocations to the hub, any non-removable functions

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

and the external ports. External ports in a bus-powered hub can supply only one unit load per port regardless of the current draw on the other ports of that hub. The hub must be able to supply this port current when the hub is in the Active or Suspend state.

- **Self-powered hubs**: Power for the internal functions and downstream facing ports does not come from VBUS. However, the USB interface of the hub may draw up to one unit load from VBUS on its upstream facing port to allow the interface to function when the remainder of the hub is powered down. Hubs that obtain operating power externally (from the USB) must supply five unit loads to each port. Battery-powered hubs may supply either one or five unit loads per port.
- **Low-power bus-powered functions**: All power to these devices comes from VBUS. They may draw no more than one unit load at any time.
- **High-power bus-powered functions**: All power to these devices comes from VBUS. They must draw no more than one unit load upon power-up and may draw up to five unit loads after being configured.
- **Self-powered functions**: May draw up to one unit load from VBUS to allow the USB interface to function when the remainder of the function is powered down. All other power comes from an external (to the USB) source.

No device shall supply (source) current on VBUS at its upstream facing port at any time. From VBUS on its upstream facing port, a device may only draw (sink) current. They may not provide power to the pull-up resistor on D+/D- unless VBUS is present (see Section 7.1.5). When VBUS is removed, the device must remove power from the D+/D- pull-up resistor within 10 seconds. On power-up, a device needs to ensure that its upstream facing port is not driving the bus, so that the device is able to receive the reset signaling. Devices must also ensure that the maximum operating current drawn by a device is one unit load, until configured. Any device that draws power from the bus must be able to detect lack of activity on the bus, enter the Suspend state, and reduce its current consumption from VBUS (refer to Section 7.2.3 and Section 9.2.5.1).

7.2.1.1 Bus-powered Hubs

Bus-powered hub power requirements can be met with a power control circuit such as the one shown in Figure 7-42. Bus-powered hubs often contain at least one non-removable function. Power is always available to the hub's controller, which permits host access to power management and other configuration registers during the enumeration process. A non-removable function(s) may require that its power be switched, so that upon power-up, the entire device (hub and non-removable functions) draws no more than one unit load. Power switching on any non-removable function may be implemented either by removing its power or by shutting off the clock. Switching on the non-removable function is not required if the aggregate power drawn by it and the Hub Controller is less than one unit load. However, as long as the hub port associated with the function is in the Power-off state, the function must be logically reset and the device must appear to be not connected. The total current drawn by a bus-powered device is the sum of the current to the Hub Controller, any non-removable function(s), and the downstream facing ports.

Figure 7-42 shows the partitioning of power based upon the maximum current draw (from upstream) of five unit loads: one unit load for the Hub Controller and the non-removable function and one unit load for each of the external downstream facing ports. If more than four external ports are required, then the hub will need to be self-powered. If the non-removable function(s) and Hub Controller draw more than one unit load, then the number of external ports must be appropriately reduced. Power control to a bus-powered hub may require a regulator. If present, the regulator is always enabled to supply the Hub Controller. The regulator can also power the non-removable functions(s). Inrush current limiting must also be incorporated into the regulator subsystem.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
WITHDRAWN

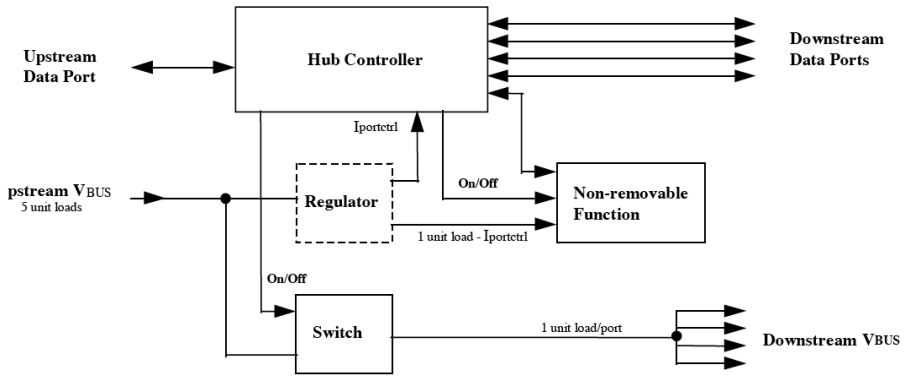


Figure 7-42. Compound Bus-powered Hub

Power to external downstream facing ports of a bus-powered hub must be switched. The Hub Controller supplies a software controlled on/off signal from the host, which is in the "off" state when the device is powered up or after reset signaling. When switched to the "on" state, the switch implements a soft turn-on function that prevents excessive transient current from being drawn from upstream. The voltage drop across the upstream cable, connectors, and switch in a bus-powered hub must not exceed 350 mV at maximum rated current.

7.2.1.2 Self-powered Hubs

Self-powered hubs have a local power supply that furnishes power to any non-removable functions and to all downstream facing ports, as shown in Figure 7-43. Power for the Hub Controller, however, may be supplied from the upstream VBUS (a "hybrid" powered hub) or the local power supply. The advantage of supplying the Hub Controller from the upstream supply is that communication from the host is possible even if the device's power supply remains off. This makes it possible to differentiate between a disconnected and an unpowered device. If the hub draws power for its upstream facing port from VBUS, it may not draw more than one unit load.

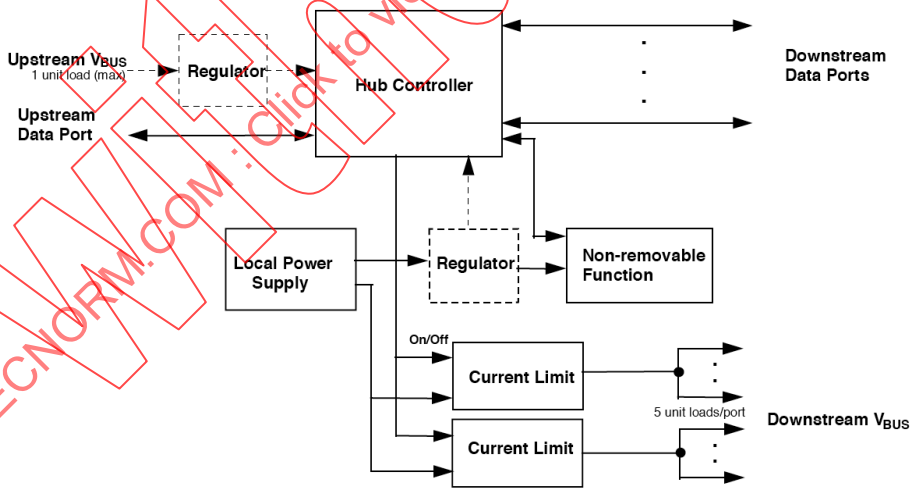


Figure 7-43. Compound Self-powered Hub

The number of ports that can be supported is limited only by the address capability of the hub and the local supply.

Self-powered hubs may experience loss of power. This may be the result of disconnecting the power cord or exhausting the battery. Under these conditions, the hub may force a re-enumeration of itself as a bus-powered hub. This requires the hub to implement port power switching on all external ports. When power is lost, the hub must ensure that upstream current does not exceed low-power. All the rules of a bus-powered hub then apply.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.2.1.2.1 Over-current Protection

The host and all self-powered hubs must implement over-current protection for safety reasons, and the hub must have a way to detect the over-current condition and report it to the USB software. Should the aggregate current drawn by a gang of downstream facing ports exceed a preset value, the over-current protection circuit removes or reduces power from all affected downstream facing ports. The over-current condition is reported through the hub to Host Controller, as described in Section 11.12.5. The preset value cannot exceed 5.0 A and must be sufficiently above the maximum allowable port current such that transient currents (e.g., during power up or dynamic attach or reconfiguration) do not trip the over-current protector. If an over-current condition occurs on any port, subsequent operation of the USB is not guaranteed, and once the condition is removed, it may be necessary to reinitialize the bus as would be done upon power-up. The over-current limiting mechanism must be resettable without user mechanical intervention. Polymeric PTCs and solid-state switches are examples of methods, which can be used for over-current limiting.

7.2.1.3 Low-power Bus-powered Functions

A low-power function is one that draws up to one unit load from the USB cable when operational. Figure 7-44 shows a typical bus-powered, low-power function, such as a mouse. Low-power regulation can be integrated into the function silicon. Low-power functions must be capable of operating with input VBUS voltages as low as 4.40 V, measured at the plug end of the cable.

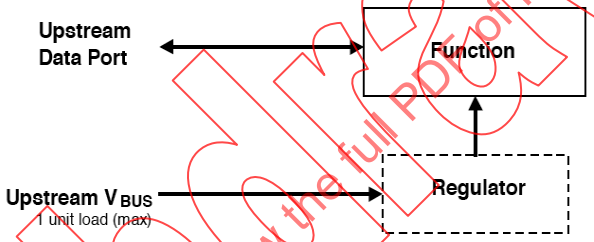


Figure 7-44. Low-power Bus-powered Function

7.2.1.4 High-power Bus-powered Functions

A function is defined as being high-power if, when fully powered, it draws over one but no more than five unit loads from the USB cable. A high-power function requires staged switching of power. It must first come up in a reduced power state of less than one unit load. At bus enumeration time, its total power requirements are obtained and compared against the available power budget. If sufficient power exists, the remainder of the function may be powered on. A typical high-power function is shown in Figure 7-45. The function's electronics have been partitioned into two sections. The function controller contains the minimum amount of circuitry necessary to permit enumeration and power budgeting. The remainder of the function resides in the function block. High-power functions must be capable of operating in their low-power (one unit load) mode with an input voltage as low as 4.40 V, so that it may be detected and enumerated even when plugged into a bus-powered hub. They must also be capable of operating at full power (up to five unit loads) with a VBUS voltage of 4.75 V, measured at the upstream plug end of the cable.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

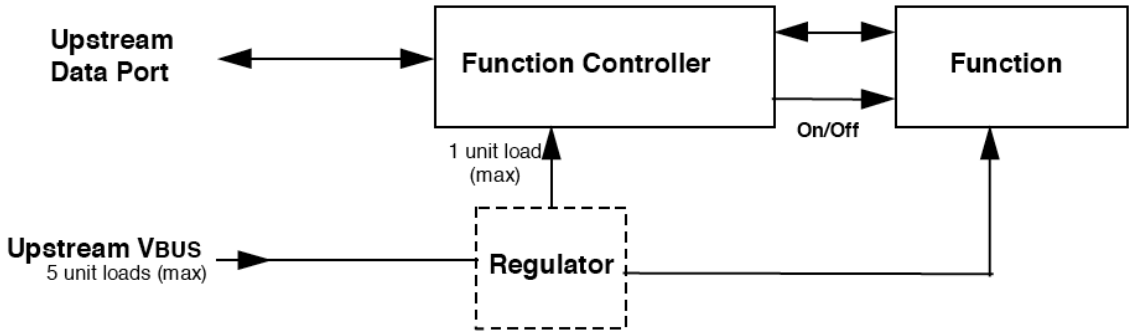


Figure 7-45. High-power Bus-powered Function

7.2.1.5 Self-powered Functions

Figure 7-46 shows a typical self-powered function. The function controller is powered either from the upstream bus via a low-power regulator or from the local power supply. The advantage of the former scheme is that it permits detection and enumeration of a self-powered function whose local power supply is turned off. The maximum upstream power that the function controller can draw is one unit load, and the regulator block must implement inrush current limiting. The amount of power that the function block may draw is limited only by the local power supply. Because the local power supply is not required to power any downstream bus ports, it does not need to implement current limiting, soft start, or power switching.

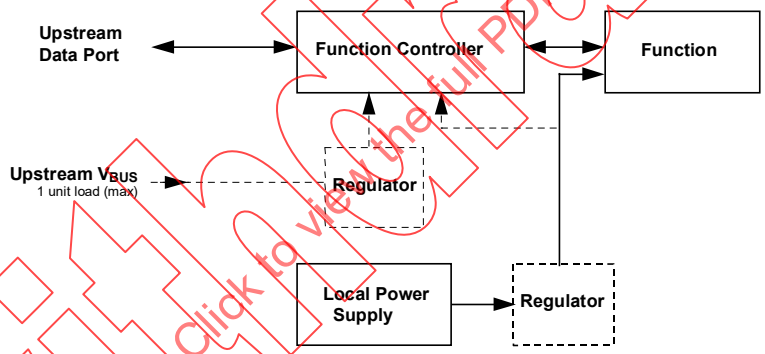


Figure 7-46. Self-powered Function

7.2.2 Voltage Drop Budget

The voltage drop budget is determined from the following:

- The voltage supplied by high-powered hub ports is 4.75 V to 5.25 V.
- The voltage supplied by low-powered hub ports is 4.4 V to 5.25 V.
- Bus-powered hubs can have a maximum drop of 350 mV from their cable plug (where they attach to a source of power) to their output port connectors (where they supply power).
- The maximum voltage drop (for detachable cables) between the A-series plug and B-series plug on VBUS is 125 mV (VBUSD).
- The maximum voltage drop for all cables between upstream and downstream on GND is 125 mV (VGNDD).
- All hubs and functions must be able to provide configuration information with as little as 4.40 V at the connector end of their upstream cables. Only low-power functions need to be operational with this minimum voltage.
- Functions drawing more than one unit load must operate with a 4.75 V minimum input voltage at the connector end of their upstream cables.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 7-47 shows the minimum allowable voltages in a worst-case topology consisting of a bus-powered hub driving a bus-powered function.

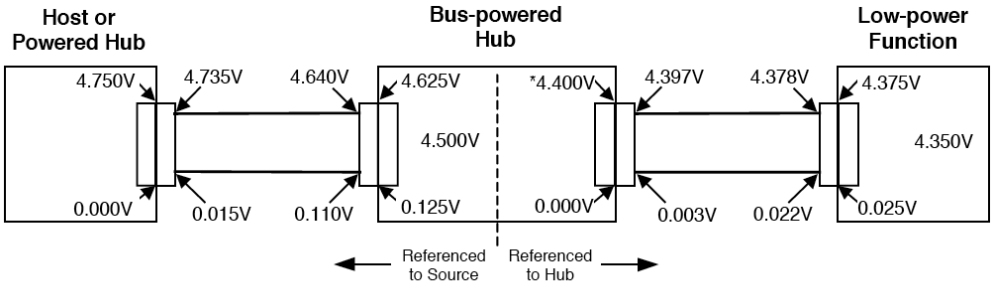


Figure 7-47. Worst-case Voltage Drop Topology (Steady State)

7.2.3 Power Control During Suspend/Resume

Suspend current is a function of unit load allocation. All USB devices initially default to low-power. Low-power devices or high-power devices operating at low-power are limited to 500 μA of suspend current. If the device is configured for high-power and enabled as a remote wakeup source, it may draw up to 2.5 mA during suspend. When computing suspend current, the current from VBUS through the bus pull-up and pull-down resistors must be included. Configured bus-powered hubs may also consume a maximum of 2.5 mA, with 500 μA allocated to each available external port and the remainder available to the hub and its internal functions. If a hub is not configured, it is operating as a low-power device and must limit its suspend current to 500 μA.

While in the Suspend state, a device may briefly draw more than the average current. The amplitude of the current spike cannot exceed the device power allocation 100 mA (or 500 mA). A maximum of 1.0 second is allowed for an averaging interval. The average current cannot exceed the average suspend current limit (ICCSH or ICCSL, see Table 7-7) during any 1.0-second interval (TSUSAVG1). The profile of the current spike is restricted so the transient response of the power supply (which may be an efficient, low-capacity, trickle power supply) is not overwhelmed. The rising edge of the current spike must be no more than 100 mA/μs. Downstream facing ports must be able to absorb the 500 mA peak current spike and meet the voltage droop requirements defined for inrush current during dynamic attach (see Section 7.2.4.1). Figure 7-48 illustrates a typical example profile for an averaging interval. If the supply to the pull-up resistor on D+/D- is derived from VBUS, then the suspend current will never go to zero because the pull-up and pull-down resistors will always draw power.

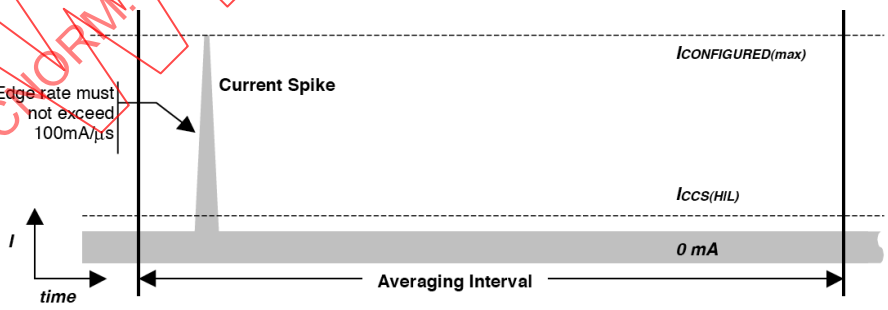


Figure 7-48. Typical Suspend Current Averaging Profile

Devices are responsible for handling the bus voltage reduction due to the inductive and resistive effects of the cable. When a hub is in the Suspend state, it must still be able to provide the maximum current per port (one unit load of current per port for bus-powered hubs and five unit loads per port for self-powered hubs). This is necessary to support remote wakeup-capable devices that will power-up while the remainder of the system is still suspended. Such devices, when enabled to do remote wakeup, must drive resume signaling

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

upstream within 10 ms of starting to draw the higher, non-suspend current. Devices not capable of remote wakeup must draw the higher current only when not suspended.

When devices wakeup, either by themselves (remote wakeup) or by seeing resume signaling, they must limit the inrush current on VBUS. The target maximum droop in the hub VBUS is 330 mV. The device must have sufficient on-board bypass capacitance or a controlled power-on sequence such that the current drawn from the hub does not exceed the maximum current capability of the port at any time while the device is waking up.

7.2.4 Dynamic Attach and Detach

The act of plugging or unplugging a hub or function must not affect the functionality of another device on other segments of the network. Unplugging a function will stop the transaction between that function and the host. However, the hub to which this function was attached will recover from this condition and will alert the host that the port has been disconnected.

7.2.4.1 Inrush Current Limiting

When a function or hub is plugged into the network, it has a certain amount of on-board capacitance between VBUS and ground. In addition, the regulator on the device may supply current to its output bypass capacitance and to the function as soon as power is applied. Consequently, if no measures are taken to prevent it, there could be a surge of current into the device which might pull the VBUS on the hub below its minimum operating level. Inrush currents can also occur when a high-power function is switched into its high-power mode. This problem must be solved by limiting the inrush current and by providing sufficient capacitance in each hub to prevent the power supplied to the other ports from going out of tolerance. An additional motivation for limiting inrush current is to minimize contact arcing, thereby prolonging connector contact life.

The maximum droop in the hub VBUS is 330 mV, or about 10% of the nominal signal swing from the function. In order to meet this requirement, the following conditions must be met:

- The maximum load (CRPB) that can be placed at the downstream end of a cable is 10 μF in parallel with 44 Ω. The 10 μF capacitance represents any bypass capacitor directly connected across the VBUS lines in the function plus any capacitive effects visible through the regulator in the device. The 44 Ω resistance represents one unit load of current drawn by the device during connect.
- If more bypass capacitance is required in the device, then the device must incorporate some form of VBUS surge current limiting, such that it matches the characteristics of the above load.
- The hub downstream facing port VBUS power lines must be bypassed (CHPB) with no less than 120 μF of low-ESR capacitance per hub. Standard bypass methods should be used to minimize inductance and resistance between the bypass capacitors and the connectors to reduce droop. The bypass capacitors themselves should have a low dissipation factor to allow decoupling at higher frequencies.

The upstream facing port of a hub is also required to meet the above requirements. Furthermore, a bus-powered hub must provide additional surge limiting in the form of a soft-start circuit when it enables power to its downstream facing ports.

A high-power bus-powered device that is switching from a lower power configuration to a higher power configuration must not cause droop > 330 mV on the VBUS at its upstream hub. The device can meet this by ensuring that changes in the capacitive load it presents do not exceed 10 μF.

Signal pins are protected from excessive currents during dynamic attach by being recessed in the connector such that the power pins make contact first. This guarantees that the power rails to the downstream device are referenced before the signal pins make contact. In

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

addition, the signal lines are in a high-impedance state during connect, so that no current flows for standard signal levels.

7.2.4.2 Dynamic Detach

When a device is detached from the network with power flowing in the cable, the inductance of the cable will cause a large flyback voltage to occur on the open end of the device cable. This flyback voltage is not destructive. Proper bypass measures on the hub ports will suppress any coupled noise. The frequency range of this noise is inversely dependent on the length of the cable, to a maximum of 60 MHz for a one-meter cable. This will require some low capacitance, very low inductance bypass capacitors on each hub port connector. The flyback voltage and the noise it creates is also moderated by the bypass capacitance on the device end of the cable. Also, there must be some minimum capacitance on the device end of the cable to ensure that the inductive flyback on the open end of the cable does not cause the voltage on the device end to reverse polarity. A minimum of 1.0 µF is recommended for bypass across VBUS.

7.3 Physical Layer

The physical layer specifications are described in the following subsections.

7.3.1 Regulatory Requirements

All USB devices should be designed to meet the applicable regulatory requirements.

7.3.2 Bus Timing/Electrical Characteristics

Table 7-7. DC Electrical Characteristics

Parameter	Symbol	Conditions	Min.	Max.	Units
Supply Voltage:					
High-power Port	VBUS	Note 2, Section 7.2.1	4.75	5.25	V
Low-power Port	VBUS	Note 2, Section 7.2.1	4.40	5.25	V
Supply Current:					
High-power Hub Port (out)	ICCPRT	Section 7.2.1	500		mA
Low-power Hub Port (out)	ICCUPT	Section 7.2.1	100		mA
High-power Function (in)	ICCHPF	Section 7.2.1		500	mA
Low-power Function (in)	ICCLPF	Section 7.2.1		100	mA
Unconfigured Function/Hub (in)	ICCINIT	Section 7.2.1.4		100	mA
Suspended High-power Device	ICCSH	Section 7.2.3; Note 15		2.5	mA
Suspended Low-power Device	ICCSL	Section 7.2.3		500	µA
Input Levels for Low-/full-speed:					
High (driven)	Vih	Note 4, Section 7.1.4	2.0		V
High (floating)	VIHZ	Note 4, Section 7.1.4	2.7	3.6	V
Low	Vil	Note 4, Section 7.1.4		0.8	V
Differential Input Sensitivity	VDI	|(D+)-(D-)|; Figure 7-19; Note 4	0.2		V
Differential Common Mode Range	VCM	Includes VDI range; Figure 7-19; Note 4	0.8	2.5	V

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-7. DC Electrical Characteristics (Continued)

Parameter	Symbol	Conditions	Min.	Max.	Units
Input Levels for High-speed:					
High-speed squelch detection threshold (differential signal amplitude)	VHSSQ	Section 7.1.7.2 (specification refers to differential signal amplitude)	100	150	mV
High speed disconnect detection threshold (differential signal amplitude)	VHSDSC	Section 7.1.7.2 (specification refers to differential signal amplitude)	525	625	mV
High-speed differential input signaling levels		Section 7.1.7.2 Specified by eye pattern templates			
High-speed data signalling common mode voltage range (guideline for receiver)	VHSCM	Section 7.1.4.2	-50	500	mV
Output Levels for Low-/full-speed:					
Low	VOL	Note 4, 5, Section 7.1.1	0.0	0.3	V
High (Driven)	VOH	Note 4, 6, Section 7.1.1	2.8	3.6	V
SE1	VOSE1	Section 7.1.1	0.8		V
Output Signal Crossover Voltage	VCRS	Measured as in Figure 7-8; Note 10	1.3	2.0	V
Output Levels for High-speed:					
High-speed idle level	VHSOI	Section 7.1.7.2	-10.0	10.0	mV
High-speed data signaling high	VHSOH	Section 7.1.7.2	360	440	mV
High-speed data signaling low	VHSOL	Section 7.1.7.2	-10.0	10.0	mV
Chirp J level (differential voltage)	VCHIRPJ	Section 7.1.7.2	700	1100	mV
Chirp K level (differential voltage)	VCHIRPK	Section 7.1.7.2	-900	-500	mV
Decoupling Capacitance:					
Downstream Facing Port Bypass Capacitance (per hub)	CHPB	VBUS to GND, Section 7.2.4.1	120		µF
Upstream Facing Port Bypass Capacitance	CRPB	VBUS to GND; Note 9, Section 7.2.4.1	1.0	10.0	µF
Input Capacitance for Low-/full-speed:					
Downstream Facing Port	CIND	Note 2; Section 7.1.6.1		150	pF
Upstream Facing Port (w/o cable)	CINUB	Note 3; Section 7.1.6.1		100	pF
Transceiver edge rate control capacitance	CEDGE	Section 7.1.6.1		75	pF
Input Impedance for High-speed:					
TDR spec for high-speed termination		Section 7.1.6.2			

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-7. DC Electrical Characteristics (Continued)

Parameter	Symbol	Conditions	Min.	Max.	Units
Terminations:					
Bus Pull-up Resistor on Upstream Facing Port	RPU	1.5 kΩ ±5% Section 7.1.5	1.425	1.575	kΩ
Bus Pull-down Resistor on Downstream Facing Port	RPD	15 kΩ ±5% Section 7.1.5	14.25	15.75	kΩ
Input impedance exclusive of pullup/pulldown (for low-/full-speed)	ZINP	Section 7.1.6	300		kΩ
Termination voltage for upstream facing port pullup (RPU)	VTERM	Section 7.1.5	3.0	3.6	V
Terminations in High-speed:					
Termination voltage in high-speed	VHSTERM	Section 7.1.6.2	-10	10	mV

Table 7-8. High-speed Source Electrical Characteristics

Parameter	Symbol	Conditions	Min.	Max.	Units
Driver Characteristics:					
Rise Time (10% - 90%)	THSR	Section 7.1.2	500		ps
Fall Time (10% - 90%)	THSF	Section 7.1.2	500		ps
Driver waveform requirements		Specified by eye pattern templates in Section 7.1.2			
Driver Output Resistance (which also serves as high-speed termination)	ZHSDRV	Section 7.1.1.1	40.5	49.5	Ω
Clock Timings:					
High-speed Data Rate	THSDRAT	Section 7.1.11	479.760	480.240	Mb/s
Microframe Interval	THSFRAM	Section 7.1.12	124.9375	125.0625	µs
Consecutive Microframe Interval Difference	THSRFI	Section 7.1.12		4 high-speed bit times	
High-speed Data Timings:					
Data source jitter		Source and receiver jitter specified by the eye pattern templates in Section 7.1.2.2			
Receiver jitter tolerance					

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 7-9. Full-speed Source Electrical Characteristics

Parameter	Symbol	Conditions	Min.	Max.	Units
Driver Characteristics:					
Rise Time	TFR	Figure 7-8; Figure 7-9	4	20	ns
Fall Time	TFF	Figure 7-8; Figure 7-9	4	20	ns
Differential Rise and Fall Time Matching	TFRFM	(TFR/TFF) Note 10, Section 7.1.2	90	111.11	%
Driver Output Resistance for driver which is not high-speed capable	ZDRV	Section 7.1.1.1	28	44	Ω
Clock Timings:					
Full-speed Data Rate for hubs and devices which are high-speed capable	TFDRATHS	Average bit rate, Section 7.1.11	11.9940	12.0060	Mb/s
Full-speed Data Rate for devices which are not high-speed capable	TFDRATE	Average bit rate, Section 7.1.11	11.9700	12.0300	Mb/s
Frame Interval	TFRAME	Section 7.1.12	0.9995	1.0005	ms
Consecutive Frame Interval Jitter	TRFI	No clock adjustment Section 7.1.12		42	ns
Full-speed Data Timings:					
Source Jitter Total (including frequency tolerance): To Next Transition For Paired Transitions	 TDJ1 TDJ2	Note 7, 8, 12, 10; Measured as in Figure 7-49;	 -3.5 -4	 3.5 4	 ns ns
Source Jitter for Differential Transition to SE0 Transition	TFDEOP	Note 8; Figure 7-50; Note 11	-2	5	ns
Receiver Jitter: To Next Transition For Paired Transitions	 TJR1 TJR2	Note 8; Figure 7-51	 -18.5 -9	 18.5 9	 ns ns
Source SE0 interval of EOP	TFEOPT	Figure 7-50	160	175	ns
Receiver SE0 interval of EOP	TFEOPR	Note 13; Section 7.1.13.2; Figure 7-50	82		ns
Width of SE0 interval during differential transition	TFST	Section 7.1.4		14	ns

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-10. Low-speed Source Electrical Characteristics

Parameter	Symbol	Conditions	Min.	Max.	Units
Driver Characteristics:					
Transition Time: Rise Time Fall Time	 TLR TLF	 Measured as in Figure 7-8	 75 75	 300 300	 ns ns
Rise and Fall Time Matching	TLRFM	(TLR/TLF) Note 10	80	125	%
Upstream Facing Port (w/cable, low-speed only)	CLINUA	Note 1; Section 7.1.6	200	450	pF
Clock Timings:					
Low-speed Data Rate for hubs which are high-speed capable	TLDRATHS	Section 7.1.11	1.49925	1.50075	Mb/s
Low-speed Data Rate for devices which are not high-speed capable	TLDRATE	Section 7.1.11	1.4775	1.5225	Mb/s
Low-speed Data Timings:					
Upstream facing port source Jitter Total (including frequency tolerance): To Next Transition For Paired Transitions	 TUDJ1 TUDJ2	Note 7, 8; Figure 7-49	 -95 -150	 95 150	 ns ns
Upstream facing port source Jitter for Differential Transition to SE0 Transition	TLDEOP	Note 8; Figure 7-50; Note 11	-40	100	ns
Upstream facing port differential Receiver Jitter: To Next Transition For Paired Transitions	 TDJR1 TDJR2	Note 8; Figure 7-51	 -75 -45	 75 45	 ns ns
Downstream facing port source Jitter Total (including frequency tolerance): To Next Transition For Paired Transitions	 TDDJ1 TDDJ2	Note 7, 8; Figure 7-49	 -25 -14	 25 14	 ns ns
Downstream facing port source Jitter for Differential Transition to SE0 Transition		Note 8; Figure 7-50; Note 11			ns
Downstream facing port Differential Receiver Jitter: To Next Transition For Paired Transitions	 TUJR1 TUJR2	Note 8; Figure 7-50	 -152 -200	 152 200	 ns ns
Source SE0 interval of EOP	TLEOPT	Figure 7-50	1.25	1.50	µs
Receiver SE0 interval of EOP	TLEOPR	Note 13; Section 7.1.13.2; Figure 7-50	670		ns
Width of SE0 interval during differential transition	TLST	Section 7.1.4		210	ns

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 7-11. Hub/Repeater Electrical Characteristics

Parameter	Symbol	Conditions	Min.	Max.	Units
Full-speed Hub Characteristics (as measured at connectors):					
Driver Characteristics: (Refer to Table 7-9)		Upstream facing port and downstream facing ports configured as full-speed			
Hub Differential Data Delay:		Note 7, 8			
(with cable)	THDD1	Figure 7-52A		70	ns
(without cable)	THDD2	Figure 7-52B		44	ns
Hub Differential Driver Jitter: (including cable)		Note 7, 8; Figure 7-52, Section 7.1.14			
To Next Transition	THDJ1		-3	3	ns
For Paired Transitions	THDJ2		-1	1	ns
Data Bit Width Distortion after SOP	TFSOP	Note 8; Figure 7-52	-5	5	ns
Hub EOP Delay Relative to THDD	TFEOPD	Note 8; Figure 7-53	0	15	ns
Hub EOP Output Width Skew	TFHESK	Note 8; Figure 7-53	-15	15	ns
Low-speed Hub Characteristics (as measured at connectors):					
Driver Characteristics: (Refer to Table 7-10)		Downstream facing ports configured as low-speed			
Hub Differential Data Delay	TLHDD	Note 7, 8; Figure 7-52		300	ns
Hub Differential Driver Jitter (including cable):		Note 7, 8; Figure 7-52			
Downstream facing port :					
To Next Transition	TLDHJ1		-45	45	ns
For Paired Transitions	TLDHJ2		-15	15	ns
Upstream facing port:					
To Next Transition	TLUHJ1		-45	45	ns
For Paired Transitions	TLUHJ2		-45	45	ns
Data Bit Width Distortion after SOP	TLSOP	Note 8; Figure 7-52	-60	60	ns
Hub EOP Delay Relative to THDD	TLEOPD	Note 8; Figure 7-53	0	200	ns
Hub EOP Output Width Skew	TLHESK	Note 8; Figure 7-53	-300	+300	ns
High-speed Hub Characteristics (as measured at connectors):					
Driver Characteristics: (Refer to Table 7-8)		Upstream facing port and downstream facing ports configured as high-speed			
Hub Data Delay (without cable):	THSHDD	Section 7.1.14.2		36 high-speed bit times + 4 ns	
Hub Data Jitter:		Specified by eye patterns in Section 7.1.2.2			
Hub Delay Variation Range:	THSHDV	Section 7.1.14.2		5 high-speed bit times	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-12. Cable Characteristics (Note 14)

Parameter	Symbol	Conditions	Min	Max	Units
VBUS Voltage drop for detachable cables	VBUSD	Section 7.2.2		125	mV
GND Voltage drop (for all cables)	VGNDD	Section 7.2.2		125	mV
Differential Cable Impedance (full-/high-speed)	ZO	(90 Ω ±15%);	76.5	103.5	Ω
Common mode cable impedance (full-/high-speed)	ZCM	(30 Ω ±30%);	21.0	39.0	Ω
Cable Delay (one way) Full-/high-speed Low-speed	 TFSCBL TLSCBL	Section 7.1.16		 26 18	 ns ns
Cable Skew	TSKEW	Section 7.1.3		100	ps
Unmated Contact Capacitance	CUC	Section 6.7		2	pF
Cable loss		Specified by table and graph in Section 7.1.17			

Note 1: Measured at A plug.

Note 2: Measured at A receptacle.

Note 3: Measured at B receptacle.

Note 4: Measured at A or B connector.

Note 5: Measured with RL of 1.425 kΩ to 3.6 V.

Note 6: Measured with RL of 14.25 kΩ to GND.

Note 7: Timing difference between the differential data signals.

Note 8: Measured at crossover point of differential data signals.

Note 9: The maximum load specification is the maximum effective capacitive load allowed that meets the target VBUS drop of 330 mV.

Note 10: Excluding the first transition from the Idle state.

Note 11: The two transitions should be a (nominal) bit time apart.

Note 12: For both transitions of differential signaling.

Note 13: Must accept as valid EOP.

Note 14: Single-ended capacitance of D+ or D- is the capacitance of D+/D- to all other conductors and, if present, shield in the cable. That is, to measure the single-ended capacitance of D+, short D-, VBUS, GND, and the shield line together and measure the capacitance of D+ to the other conductors.

Note 15: For high power devices (non-hubs) when enabled for remote wakeup.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-13. Hub Event Timings

Event Description	Symbol	Conditions	Min	Max	Unit
Time to detect a downstream facing port connect event Awake Hub Suspended Hub	TDCNN	Section 11.5 and Section 7.1.7.3	 2.5 2.5	 2000 12000	 µs µs
Time to detect a disconnect event at a hub's downstream facing port	TDDIS	Section 7.1.7.3	2	2.5	µs
Duration of driving resume to a downstream port; only from a controlling hub	TDRSMDN	Nominal; Section 7.1.7.7 and Section 11.5	20		ms
Time from detecting downstream resume to rebroadcast	TURSM	Section 7.1.7.7		1.0	ms
Duration of driving reset to a downstream facing port	TDRST	Only for a SetPortFeature (PORT_RESET) request; Section 7.1.7.5 and Section 11.5	10	20	ms
Overall duration of driving reset to downstream facing port, root hub	TDRSTR	Only for root hubs; Section 7.1.7.5	50		ms
Maximum interval between reset segments used to create TDRSTR	TRHRSI	Only for root hubs; each reset pulse must be of length TDRST; Section 7.1.7.5		3	ms
Time to detect a long K from upstream	TURLK	Section 11.6	2.5	100	µs
Time to detect a long SE0 from upstream	TURLSE0	Section 11.6	2.5	10000	µs
Duration of repeating SE0 upstream (for low-/full-speed repeater)	TURPSE0	Section 11.6		23	FS bit times
Duration of sending SE0 upstream after EOF1 (for low-/full-speed repeater)	TUDEOP	*Optional* Section 11.6		2	FS bit times
Inter-packet Delay (for high-speed) for packets traveling in same direction	THSIPDSD	Section 7.1.18.2	88		bit times
Inter-packet Delay (for high-speed) for packets traveling in opposite direction	THSIPDOD	Section 7.1.18.2	8		bit times
Inter-packet delay for device/root hub response w/detachable cable for high-speed	THSRSPIPD1	Section 7.1.18.2		192	bit times

ECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-13. Hub Event Timings (Continued)

Event Description	Symbol	Conditions	Min	Max	Unit
Reset Handshake Protocol:					
Time for which a Chirp J or Chirp K must be continuously detected (filtered) by hub or device during Reset handshake	TFILT	Section 7.1.7.5	2.5		µs
Time after end of device Chirp K by which hub must start driving first Chirp K in the hub's chirp sequence	TWTDCH	Section 7.1.7.5		100	µs
Time for which each individual Chirp J or Chirp K in the chirp sequence is driven downstream by hub during reset	TDCHBIT	Section 7.1.7.5	40	60	µs
Time before end of reset by which a hub must end its downstream chirp sequence	TDCHSE0	Section 7.1.7.5	100	500	µs

Table 7-14. Device Event Timings

Parameter	Symbol	Conditions	Min	Max	Units
Time from internal power good to device pulling D+/D- beyond VIHZ (min) (signaling attach)	TSIGATT	Figure 7-29		100	ms
Debounce interval provided by USB system software after attach	TATTDB	Figure 7-29		100	ms
Maximum time a device can draw power >suspend power when bus is continuously in idle state	T2SUSP	Section 7.1.7.6		10	ms
Maximum duration of suspend averaging interval	TSUSAVGI	Section 7.2.3		1	s
Period of idle bus before device can initiate resume	TWTRSM	Device must be remote-wakeup enabled Section 7.1.7.5	5		ms
Duration of driving resume upstream	TDRSMUP	Section 7.1.7.7	1	15	ms
Resume Recovery Time	TRSMRCY	Provided by USB System Software; Section 7.1.7.7	10		ms
Time to detect a reset from upstream for non high-speed capable devices	TDETRST	Section 7.1.7.5	2.5	10000	µs
Reset Recovery Time	TRSTRCY	Section 7.1.7.5		10	ms
Inter-packet Delay (for low-/full-speed)	TIPD	Section 7.1.18	2		bit times
Inter-packet delay for device response w/detachable cable for low-/full-speed	TRSPIPD1	Section 7.1.18		6.5	bit times
Inter-packet delay for device response w/captive cable for low-/full-speed	TRSPIPD2	Section 7.1.18		7.5	bit times

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-14. Device Event Timings (Continued)

Parameter	Symbol	Conditions	Min	Max	Units
SetAddress() Completion Time	TDSETADDR	Section 9.2.6.3		50	ms
Time to complete standard request with no data	TDRQCMPLTND	Section 9.2.6.4		50	ms
Time to deliver first and subsequent (except last) data for standard request	TDRETDATA1	Section 9.2.6.4		500	ms
Time to deliver last data for standard request	TDRETDATAN	Section 9.2.6.4		50	ms
Inter-packet delay for device response w/captive cable (high-speed)	THSRSPIPD2	Section 7.1.18.2		192 bit times + 52 ns	
SetAddress() Completion Time	TDSETADDR	Section 9.2.6.3		50	ms
Time to complete standard request with no data	TDRQCMPLTND	Section 9.2.6.4		50	ms
Reset Handshake Protocol:					
Time for which a suspended high-speed capable device must see a continuous SE0 before beginning the high-speed detection handshake	TFILTSE0	Section 7.1.7.5	2.5		µs
Time a high-speed capable device operating in non-suspended full-speed must wait after start of SE0 before beginning the high-speed detection handshake	TWTRSTFS	Section 7.1.7.5	2.5	3000	µs
Time a high-speed capable device operating in high-speed must wait after start of SE0 before reverting to full-speed	TWTREV	Section 7.1.7.5	3.0	3.125	ms
Time a device must wait after reverting to full-speed before sampling the bus state for SE0 and beginning the high-speed detection handshake	TWTRSTHS	Section 7.1.7.5	100	875	µs
Minimum duration of a Chirp K from a high-speed capable device within the reset protocol	TUCH	Section 7.1.7.5	1.0		ms
Time after start of SE0 by which a high-speed capable device is required to have completed its Chirp K within the reset protocol	TUCHEND	Section 7.1.7.5		7.0	ms
Time between detection of downstream chirp and entering high-speed state	TWTHS	Section 7.1.7.5		500	µs
Time after end of upstream chirp at which device reverts to full-speed default state if no downstream chirp is detected	TWTFS	Section 7.1.7.5	1.0	2.5	ms

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.3.3 Timing Waveforms

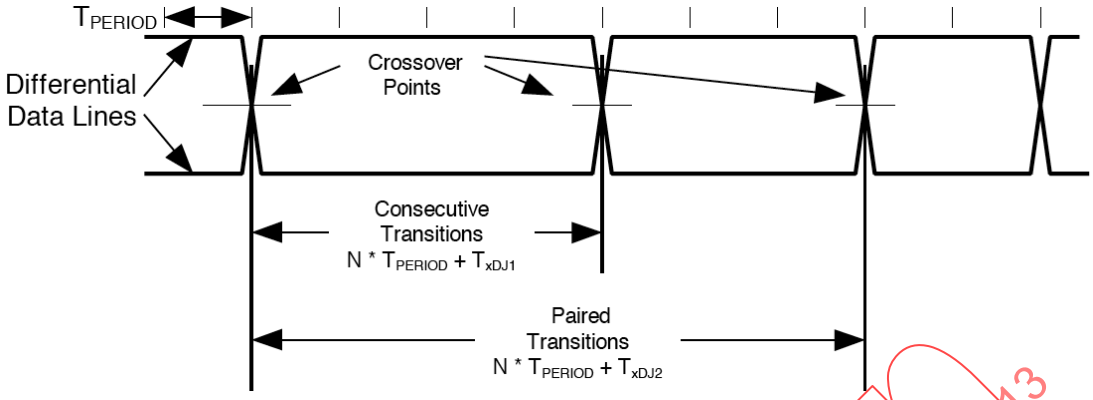


Figure 7-49. Differential Data Jitter for Low-/full-speed

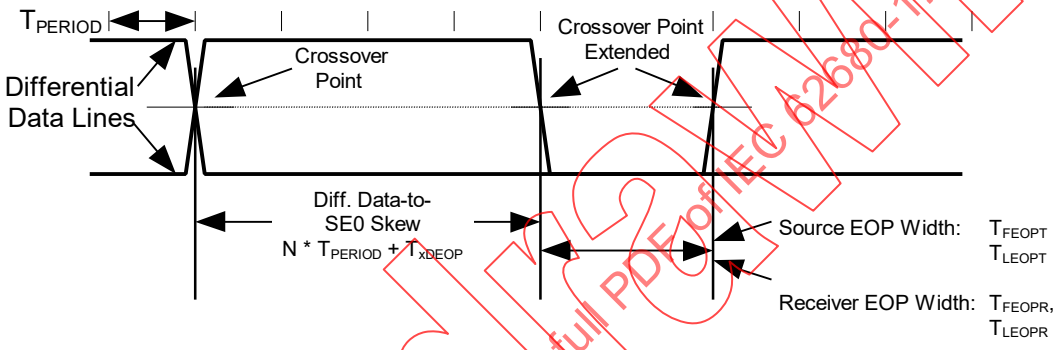


Figure 7-50. Differential-to-EOP Transition Skew and EOP Width for Low-/full-speed

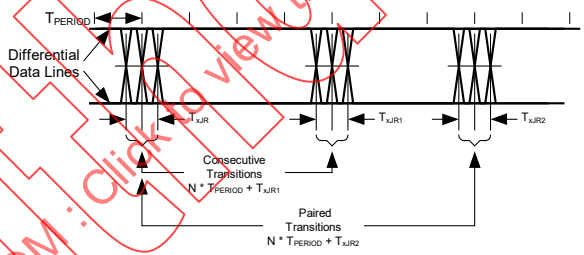
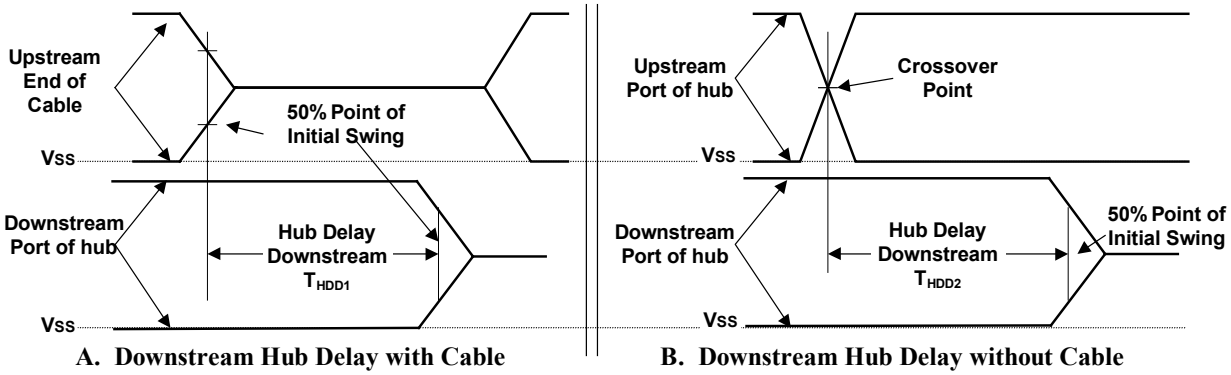


Figure 7-51. Receiver Jitter Tolerance for Low-/full-speed

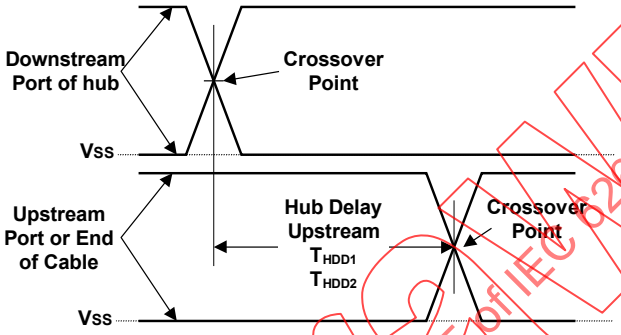
TPERIOD is the data rate of the receiver that can have the range as defined in Section 7.1.11.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



A. Downstream Hub Delay with Cable

B. Downstream Hub Delay without Cable



C. Upstream Hub Delay with or without Cable

Hub Differential Jitter:

$T_{HDJ1} = T_{HDDx}(J) - T_{HDDx}(K)$ or $T_{HDDx}(K) - T_{HDDx}(J)$ Consecutive Transitions

$T_{HDJ2} = T_{HDDx}(J) - T_{HDDx}(J)$ or $T_{HDDx}(K) - T_{HDDx}(K)$ Paired Transitions

Bit after SOP Width Distortion (same as data jitter for SOP and next J transition):

$T_{FSOP} = T_{HDDx}(\text{next J}) - T_{HDDx}(\text{SOP})$

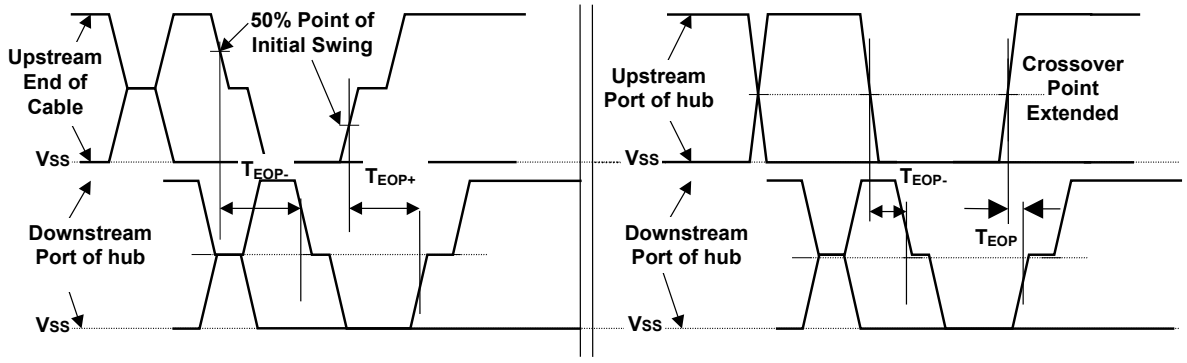
Low-speed timings are determined in the same way for:

T_{LHDD}, T_{LDHJ1}, T_{LDJH2}, T_{LUHJ1}, T_{LUJH2}, and T_{LSOP}

Figure 7-52. Hub Differential Delay, Differential Jitter, and SOP Distortion for Low-/full-speed

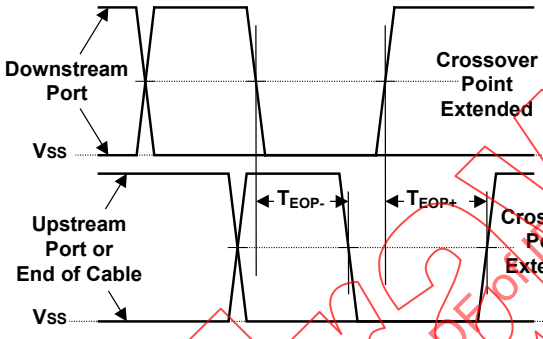
Measurement locations referenced in Figure 7-52 and Figure 7-53 are specified in Figure 7-38.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



A. Downstream EOP Delay with Cable

B. Downstream EOP Delay without Cable



C. Upstream EOP Delay with or Without Cable

EOP Delay:

$T_{FEOPD} = T_{EOPy} - T_{HDDx}$
(T_{EOPy} means that this equation applies to T_{EOP-} and T_{EOP+})

EOP Skew:

$T_{FHESK} = T_{EOP+} - T_{EOP-}$

Low-speed timings are determined in the same way for:

T_{LEOPD} and T_{LHESK}

Figure 7-53. Hub EOP Delay and EOP Skew for Low-/full-speed

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8 Chapter 8 Protocol Layer

This chapter presents a bottom-up view of the USB protocol, starting with field and packet definitions. This is followed by a description of packet transaction formats for different transaction types. Link layer flow control and transaction level fault recovery are then covered. The chapter finishes with a discussion of retry synchronization, babble, loss of bus activity recovery, and high-speed PING protocol.

8.1 Byte/Bit Ordering

Bits are sent out onto the bus least-significant bit (LSb) first, followed by the next LSb, through to the most-significant bit (MSb) last. In the following diagrams, packets are displayed such that both individual bits and fields are represented (in a left to right reading order) as they would move across the bus.

Multiple byte fields in standard descriptors, requests, and responses are interpreted as and moved over the bus in little-endian order, i.e., LSB to MSB.

8.2 SYNC Field

All packets begin with a synchronization (SYNC) field, which is a coded sequence that generates a maximum edge transition density. It is used by the input circuitry to align incoming data with the local clock. A SYNC from an initial transmitter is defined to be eight bits in length for full/low-speed and 32 bits for high-speed. Received SYNC fields may be shorter as described in Chapter 7. SYNC serves only as a synchronization mechanism and is not shown in the following packet diagrams (refer to Section 7.1.10). The last two bits in the SYNC field are a marker that is used to identify the end of the SYNC field and, by inference, the start of the PID.

8.3 Packet Field Formats

Field formats for the token, data, and handshake packets are described in the following section. Packet bit definitions are displayed in unencoded data format. The effects of NRZI coding and bit stuffing have been removed for the sake of clarity. All packets have distinct Start- and End-of-Packet delimiters. The Start-of-Packet (SOP) delimiter is part of the SYNC field, and the End-of-Packet (EOP) delimiter is described in Chapter 7.

8.3.1 Packet Identifier Field

A packet identifier (PID) immediately follows the SYNC field of every USB packet. A PID consists of a four-bit packet type field followed by a four-bit check field as shown in Figure 8-1. The PID indicates the type of packet and, by inference, the format of the packet and the type of error detection applied to the packet. The four-bit check field of the PID ensures reliable decoding of the PID so that the remainder of the packet is interpreted correctly. The PID check field is generated by performing a one's complement of the packet type field. A PID error exists if the four PID check bits are not complements of their respective packet identifier bits.

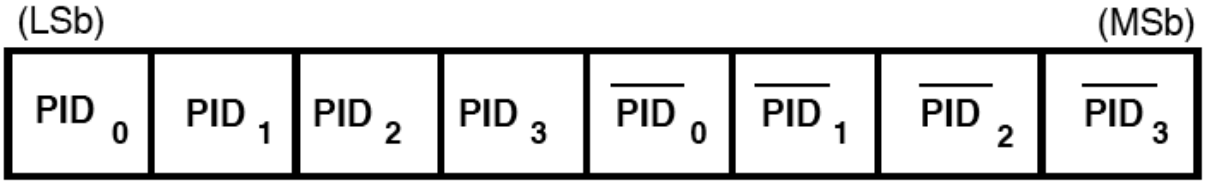


Figure 8-1. PID Format

The host and all functions must perform a complete decoding of all received PID fields. Any PID received with a failed check field or which decodes to a non-defined value is assumed to be corrupted and it, as well as the remainder of the packet, is ignored by the packet receiver.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

If a function receives an otherwise valid PID for a transaction type or direction that it does not support, the function must not respond. For example, an IN-only endpoint must ignore an OUT token. PID types, codings, and descriptions are listed in Table 8-1.

Table 8-1. PID Types

PID Type	PID Name	PID<3:0>*	Description
Token	OUT	0001B	Address + endpoint number in host-to-function transaction
	IN	1001B	Address + endpoint number in function-to-host transaction
	SOF	0101B	Start-of-Frame marker and frame number
	SETUP	1101B	Address + endpoint number in host-to-function transaction for SETUP to a control pipe
Data	DATA0	0011B	Data packet PID even
	DATA1	1011B	Data packet PID odd
	DATA2	0111B	Data packet PID high-speed, high bandwidth isochronous transaction in a microframe (see Section 5.9.2 for more information)
	MDATA	1111B	Data packet PID high-speed for split and high bandwidth isochronous transactions (see Sections 5.9.2, 11.20, and 11.21 for more information)
Handshake	ACK	0010B	Receiver accepts error-free data packet
	NAK	1010B	Receiving device cannot accept data or transmitting device cannot send data
	STALL	1110B	Endpoint is halted or a control pipe request is not supported
	NYET	0110B	No response yet from receiver (see Sections 8.5.1 and 11.17-11.21)
Special	PRE	1100B	(Token) Host-issued preamble. Enables downstream bus traffic to low-speed devices.
	ERR	1100B	(Handshake) Split Transaction Error Handshake (reuses PRE value)
	SPLIT	1000B	(Token) High-speed Split Transaction Token (see Section 8.4.2)
	PING	0100B	(Token) High-speed flow control probe for a bulk/control endpoint (see Section 8.5.1)
	Reserved	0000B	Reserved PID

*Note: PID bits are shown in MSb order. When sent on the USB, the rightmost bit (bit 0) will be sent first.

PIDs are divided into four coding groups: token, data, handshake, and special, with the first two transmitted PID bits (PID<0:1>) indicating which group. This accounts for the distribution of PID codes.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

8.3.2 Address Fields

Function endpoints are addressed using two fields: the function address field and the endpoint field. A function needs to fully decode both address and endpoint fields. Address or endpoint aliasing is not permitted, and a mismatch on either field must cause the token to be ignored. Accesses to non-initialized endpoints will also cause the token to be ignored.

8.3.2.1 Address Field

The function address (ADDR) field specifies the function, via its address, that is either the source or destination of a data packet, depending on the value of the token PID. As shown in Figure 8-2, a total of 128 addresses are specified as ADDR<6:0>. The ADDR field is

specified for IN, SETUP, and OUT tokens and the PING and SPLIT special token. By definition, each ADDR value defines a single function. Upon reset and power-up, a function's address defaults to a value of zero and must be programmed by the host during the enumeration process. Function address zero is reserved as the default address and may not be assigned to any other use.

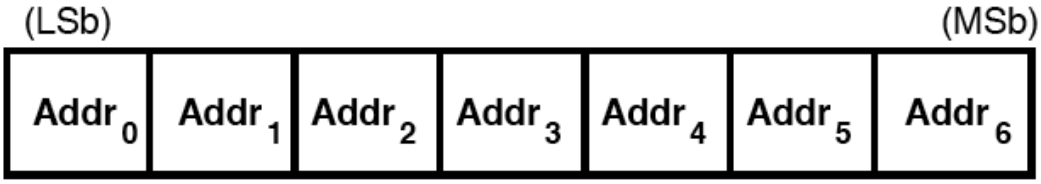


Figure 8-2. ADDR Field

8.3.2.2 Endpoint Field

An additional four-bit endpoint (ENDP) field, shown in Figure 8-3 permits more flexible addressing of functions in which more than one endpoint is required. Except for endpoint address zero, endpoint numbers are function-specific. The endpoint field is defined for IN, SETUP, and OUT tokens and the PING special token. All functions must support a control pipe at endpoint number zero (the Default Control Pipe). Low-speed devices support a maximum of three pipes per function: a control pipe at endpoint number zero plus two additional pipes (either two control pipes, a control pipe and a interrupt endpoint, or two interrupt endpoints). Full-speed and high-speed functions may support up to a maximum of 16 IN and OUT endpoints.

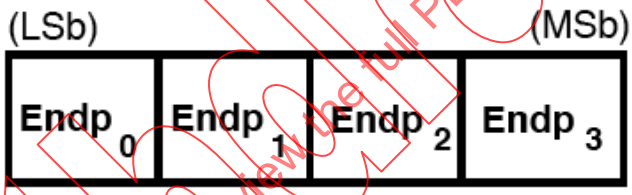


Figure 8-3. Endpoint Field

8.3.3 Frame Number Field

The frame number field is an 11-bit field that is incremented by the host on a per-frame basis. The frame number field rolls over upon reaching its maximum value of 7FFH and is sent only in SOF tokens at the start of each (micro)frame.

8.3.4 Data Field

The data field may range from zero to 1,024 bytes and must be an integral number of bytes. Figure 8-4 shows the format for multiple bytes. Data bits within each byte are shifted out LSb first.

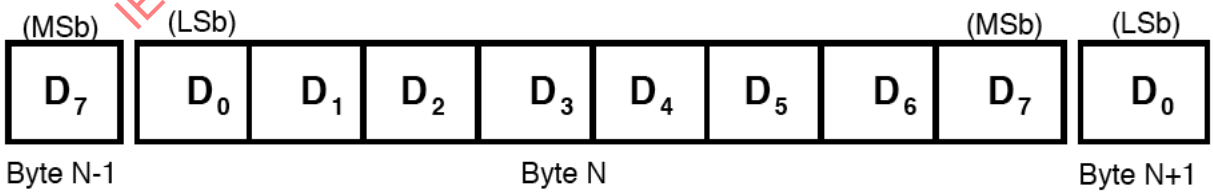


Figure 8-4. Data Field Format

Data packet size varies with the transfer type, as described in Chapter 5.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

8.3.5 Cyclic Redundancy Checks

Cyclic redundancy checks (CRCs) are used to protect all non-PID fields in token and data packets. In this context, these fields are considered to be protected fields. The PID is not included in the CRC check of a packet containing a CRC. All CRCs are generated over their respective fields in the transmitter before bit stuffing is performed. Similarly, CRCs are decoded in the receiver after stuffed bits have been removed. Token and data packet CRCs provide 100% coverage for all single- and double-bit errors. A failed CRC is considered to indicate that one or more of the protected fields is corrupted and causes the receiver to ignore those fields and, in most cases, the entire packet.

For CRC generation and checking, the shift registers in the generator and checker are seeded with an all-ones pattern. For each data bit sent or received, the high order bit of the current remainder is XORed with the data bit and then the remainder is shifted left one bit and the low-order bit set to zero. If the result of that XOR is one, then the remainder is XORed with the generator polynomial.

When the last bit of the checked field is sent, the CRC in the generator is inverted and sent to the checker MSb first. When the last bit of the CRC is received by the checker and no errors have occurred, the remainder will be equal to the polynomial residual.

A CRC error exists if the computed checksum remainder at the end of a packet reception does not match the residual.

Bit stuffing requirements must be met for the CRC, and this includes the need to insert a zero at the end of a CRC if the preceding six bits were all ones.

8.3.5.1 Token CRCs

A five-bit CRC field is provided for tokens and covers the ADDR and ENDP fields of IN, SETUP, and OUT tokens or the time stamp field of an SOF token. The PING and SPLIT special tokens also include a five-bit CRC field. The generator polynomial is:

$$G(X) = X^5 + X^2 + 1$$

The binary bit pattern that represents this polynomial is 00101B. If all token bits are received without error, the five-bit residual at the receiver will be 01100B.

8.3.5.2 Data CRCs

The data CRC is a 16-bit polynomial applied over the data field of a data packet. The generating polynomial is:

$$G(X) = X^{16} + X^{15} + X^2 + 1$$

The binary bit pattern that represents this polynomial is 1000000000000101B. If all data and CRC bits are received without error, the 16-bit residual will be 1000000000001101B.

8.4 Packet Formats

This section shows packet formats for token, data, and handshake packets. Fields within a packet are displayed in these figures in the order in which bits are shifted out onto the bus.

8.4.1 Token Packets

Figure 8-5 shows the field formats for a token packet. A token consists of a PID, specifying either IN, OUT, or SETUP packet type and ADDR and ENDP fields. The PING special token packet also has the same fields as a token packet. For OUT and SETUP transactions, the address and endpoint fields uniquely identify the endpoint that will receive the subsequent Data packet. For IN transactions, these fields uniquely identify which endpoint should transmit a Data packet. For PING transactions, these fields uniquely identify which endpoint will respond with a handshake packet. Only the host can issue token packets. An IN PID

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

defines a Data transaction from a function to the host. OUT and SETUP PIDs define Data transactions from the host to a function. A PING PID defines a handshake transaction from the function to the host.

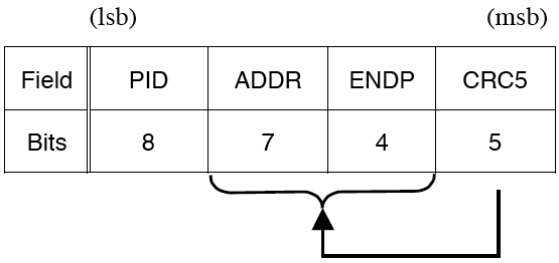


Figure 8-5. Token Format

Token packets have a five-bit CRC that covers the address and endpoint fields as shown above. The CRC does not cover the PID, which has its own check field. Token and SOF packets are delimited by an EOP after three bytes of packet field data. If a packet decodes as an otherwise valid token or SOF but does not terminate with an EOP after three bytes, it must be considered invalid and ignored by the receiver.

8.4.2 Split Transaction Special Token Packets

USB defines a special token for split transactions: SPLIT. This is a 4 byte token packet compared to other normal 3 byte token packets. The split transaction token packet provides additional transaction types with additional transaction specific information. The split transaction token is used to support split transactions between the host controller communicating with a hub operating at high speed with full-/low-speed devices to some of its downstream facing ports. There are two split transactions defined that use the SPLIT special token: a start-split transaction (SSPLIT) and a complete-split transaction (CSPLIT). A field in the SPLIT special token, described in the following sections, indicates the specific split transaction.

8.4.2.1 Split Transactions

A high-speed split transaction is used only between the host controller and a hub when the hub has full-/low-speed devices attached to it. This high-speed split transaction is used to initiate a full-/low-speed transaction via the hub and some full-/low-speed device endpoint. The high-speed split transaction also allows the completion status of the full-/low-speed transaction to be retrieved from the hub. This approach allows the host controller to start a full-/low-speed transaction via a high-speed transaction and then continue with other high-speed transactions without having to wait for the full-/low-speed transaction to proceed/complete at the slower speed. See Chapter 11 for more details about the state machines and transaction definitions of split transactions.

A high-speed split transaction has two parts: a start-split and a complete-split. Split transactions are only defined to be used between the host controller and a hub. No other high-speed or full-/low-speed devices ever use split transactions.

Figure 8-6 shows the packets composing a generic start-split transaction. There are two packets in the token phase: the SPLIT special token and a full-/low-speed token. Depending on the direction of data transfer and whether a handshake is defined for the transaction type, the token phase is optionally followed by a data packet and a handshake packet. Start split transactions can consist of 2, 3, or 4 packets as determined by the specific transfer type and data direction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

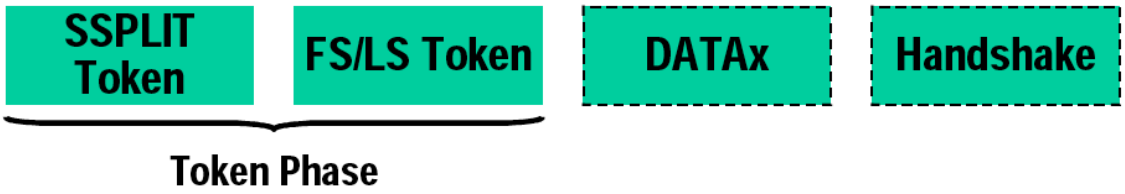


Figure 8-6. Packets in a Start-split Transaction

Figure 8-7 shows the packets composing a generic complete-split transaction. There are two packets in the token phase: the SPLIT special token and a full-/low-speed token. A data or handshake packet follows the token phase packets in the complete-split depending on the data transfer direction and specific transaction type. Complete split transactions can consist of 2 or 3 packets as determined by the specific transfer type and data direction.

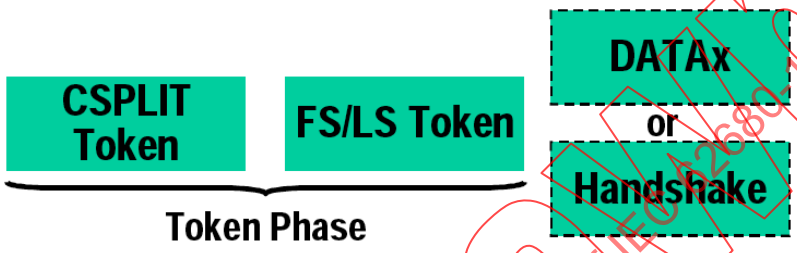


Figure 8-7. Packets in a Complete-split Transaction

The results of a split transaction are returned by a complete-split transaction. Figure 8-8 shows this conceptual “conversion” for an example interrupt IN transfer type. The host issues a start-split (indicated with 1) to the hub and then can proceed with other high-speed transactions. The start-split causes the hub to issue a full-/low-speed IN token sometime later (indicated by 2). The device responds to the IN token (in this example) with a data packet and the hub responds with a handshake to the device. Finally, the host sometime later issues a complete-split (indicated by 3) to retrieve the data provided by the device. Note that in the example, the hub provided the full-/low-speed handshake (ACK in this example) to the device endpoint before the complete-split, and the complete-split did not provide a high-speed handshake to the hub.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

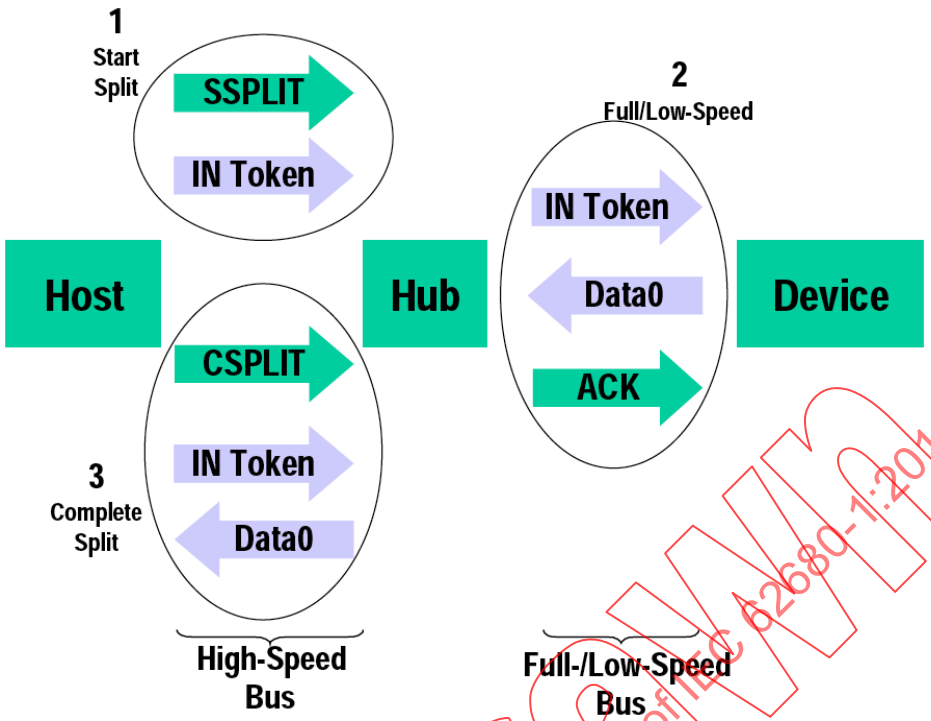


Figure 8-8. Relationship of Interrupt IN Transaction to High-speed Split Transaction

A normal full-/low-speed OUT transaction is similarly conceptually “converted” into start-split and complete-split transactions. Figure 8-9 shows this “conversion” for an example interrupt OUT transfer type. The host issues a start-split transaction consisting of a SSPLIT special token, an OUT token, and a DATA packet. The hub sometime later issues the OUT token and DATA packet on the full-/low-speed bus. The device responds with a handshake. Sometime later, the host issues the complete-split transaction and the hub responds with the results (either full-/low-speed data or handshake) provided by the device.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

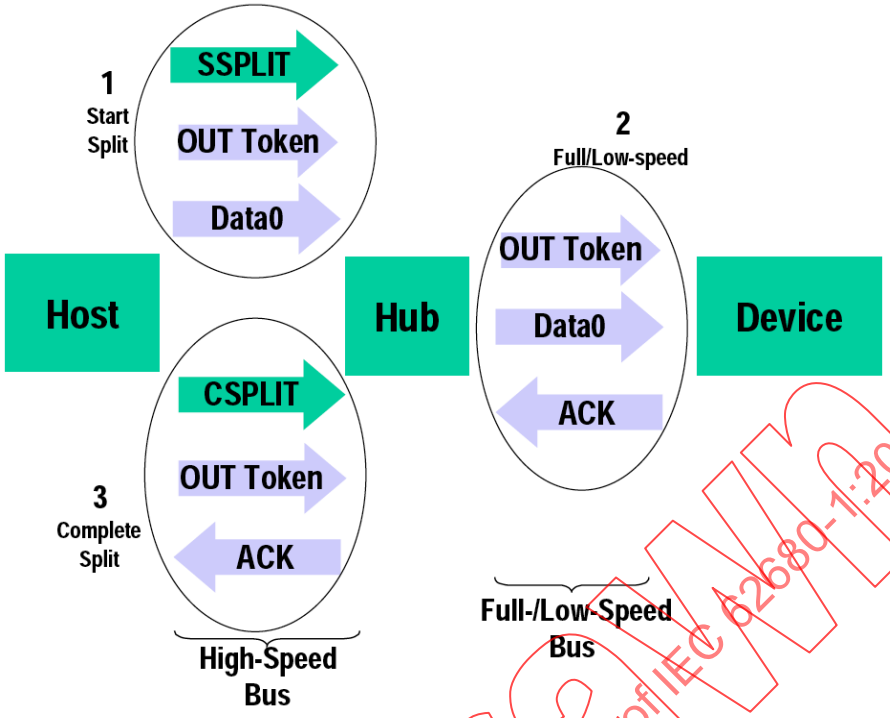


Figure 8-9. Relationship of Interrupt OUT Transaction to High-speed Split OUT Transaction

The next two sections describe the fields composing the detailed start- and complete-split token packets. Figure 8-10 and Figure 8-12 show the fields in the split-transaction token packet. The SPLIT special token follows the general token format and starts with a PID field (after a SYNC) and ends with a CRC5 field (and EOP). Start-split and complete-split token packets are both 4 bytes long. SPLIT transactions must only originate from the host. The start-split token is defined in Section 8.4.2.2 and the complete-split token is defined in Section 8.4.2.3.

8.4.2.2 Start-Split Transaction Token

(lsb) (msb)

Field	SPLIT PID	Hub Addr	SC	Port	S	E	ET	CRC5
Bits	8	7	1	7	1	1	2	5

Figure 8-10. Start-split (SSPLIT) Token

The Hub addr field contains the USB device address of the hub supporting the specified full-/low-speed device for this full-/low-speed transaction. This field has the same definition as the ADDR field definition in Section 8.3.2.1.

A SPLIT special token packet with the SC (Start/Complete) field set to zero indicates that this is a start-split transaction (SSPLIT).

The Port field contains the port number of the target hub for which this full-/low-speed transaction is destined. As shown in Figure 8-11, a total of 128 ports are specified as PORT<6:0>. The host must correctly set the port field for single and multiple TT hub implementations. A single TT hub implementation may ignore the port field.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

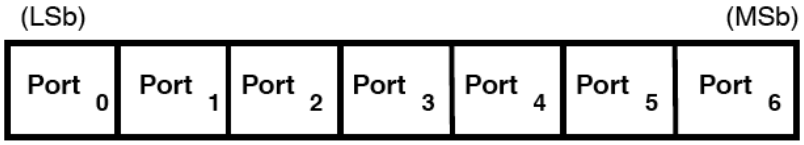


Figure 8-11. Port Field

The S (Speed) field specifies the speed for this interrupt or control transaction as follows:

- 0 – Full speed
- 1 – Low speed

For bulk IN/OUT and isochronous IN start-splits, the S field must be set to zero. For bulk/control IN/OUT, interrupt IN/OUT, and isochronous IN start-splits, the E field must be set to zero.

For full-speed isochronous OUT start-splits, the S[1] (Start) and E (End) fields specify how the high-speed data payload corresponds to data for a full-speed data packet as shown in Table 8-2.

Table 8-2. Isochronous OUT Payload Continuation Encoding

S	E	High-speed to Full-speed Data Relation
0	0	High-speed data is the middle of the full-speed data payload
0	1	High-speed data is the end of the full-speed data payload
1	0	High-speed data is the beginning of the full-speed data payload
1	1	High-speed data is all of the full-speed data payload.

Isochronous OUT start-split transactions use these encodings to allow the hub to detect various error cases due to lack of receiving start-split transactions for an endpoint with a data payload that requires multiple start-splits. For example, a large full-speed data payload may require three start-split transactions: a start-split/beginning, a start-split/middle and a start-split/end. If any of these transactions is not received by the hub, it will either ignore the full-speed transaction (if the start-split/beginning is not received), or it will force an error for the corresponding full-speed transaction (if one of the other two transactions are not received). Other error conditions can be detected by not receiving a start-split during a microframe.

The ET (Endpoint Type) field specifies the endpoint type of the full-/low-speed transaction as shown in Table 8-3.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

[1] The S bit can be reused for these encodings since isochronous transactions must not be low speed.

Table 8-3. Endpoint Type Values in Split Special Token

ET value (msb:lsb)	Endpoint Type
00	Control
01	Isochronous
10	Bulk
11	Interrupt

This field tells the hub which split transaction state machine to use for this full-/low-speed transaction.

The full-/low-speed device address and endpoint number information is contained in the normal token packet that follows the SPLIT special token packet.

8.4.2.3 Complete-Split Transaction Token

(lsb) (msb)

Field	SPLIT PID	Hub Addr	SC	Port	S	U	ET	CRC5
Bits	8	7	1	7	1	1	2	5

Figure 8-12. Complete-split (CSPLIT) Transaction Token

A SPLIT special token packet with the SC field set to one indicates that this is a complete-split transaction (CSPLIT).

The U bit is reserved/unused and must be reset to zero(0B).

The other fields of the complete-split token packet have the same definitions as for the start-split token packet.

8.4.3 Start-of-Frame Packets

Start-of-Frame (SOF) packets are issued by the host at a nominal rate of once every 1.00 ms ±0.0005 ms for a full-speed bus and 125 µs ±0.0625 µs for a high-speed bus. SOF packets consist of a PID indicating packet type followed by an 11-bit frame number field as illustrated in Figure 8-13.

(lsb) (msb)

Field	PID	FrameNumber	CRC5
Bits	8	11	5

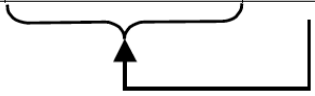


Figure 8-13. SOF Packet

The SOF token comprises the token-only transaction that distributes an SOF marker and accompanying frame number at precisely timed intervals corresponding to the start of each frame. All high-speed and full-speed functions, including hubs, receive the SOF packet. The SOF token does not cause any receiving function to generate a return packet; therefore, SOF delivery to any given function cannot be guaranteed. The SOF packet delivers two pieces of timing information. A function is informed that an SOF has occurred when it detects the SOF

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

PID. Frame timing sensitive functions, that do not need to keep track of frame number (e.g., a full-speed operating hub), need only decode the SOF PID; they can ignore the frame number and its CRC. If a function needs to track frame number, it must comprehend both the PID and the time stamp. Full-speed devices that have no particular need for bus timing information may ignore the SOF packet.

8.4.3.1 USB Frames and Microframes

USB defines a full-speed 1 ms frame time indicated by a Start Of Frame (SOF) packet each and every 1ms period with defined jitter tolerances. USB also defines a high-speed microframe with a 125 μs frame time with related jitter tolerances (See Chapter 7). SOF packets are generated (by the host controller or hub transaction translator) every 1ms for full-speed links. SOF packets are also generated after the next seven 125 μs periods for high-speed links.

Figure 8-14 shows the relationship between microframes and frames.

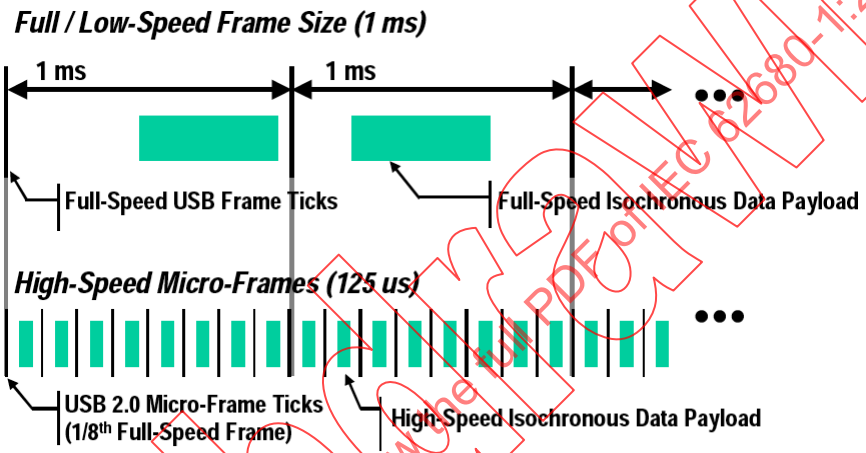


Figure 8-14. Relationship between Frames and Microframes

High-speed devices see an SOF packet with the same frame number eight times (every 125 μs) during each 1 ms period. If desired, a high-speed device can locally determine a particular microframe "number" by detecting the SOF that had a different frame number than the previous SOF and treating that as the zeroth microframe. The next seven SOFs with the same frame number can be treated as microframes 1 through 7.

8.4.4 Data Packets

A data packet consists of a PID, a data field containing zero or more bytes of data, and a CRC as shown in Figure 8-15. There are four types of data packets, identified by differing PIDs: DATA0, DATA1, DATA2 and MDATA. Two data packet PIDs (DATA0 and DATA1) are defined to support data toggle synchronization (refer to Section 8.6). All four data PIDs are used in data PID sequencing for high bandwidth high-speed isochronous endpoints (refer to Section 5.9). Three data PIDs (MDATA, DATA0, DATA1) are used in split transactions (refer to Sections 11.17-11.21).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

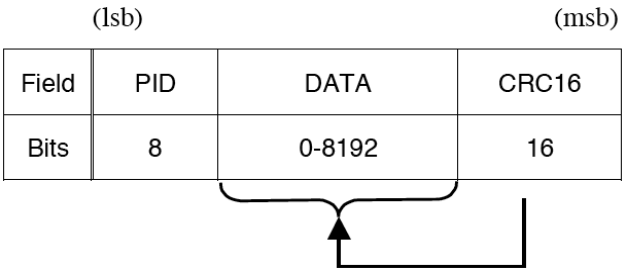


Figure 8-15. Data Packet Format

Data must always be sent in integral numbers of bytes. The data CRC is computed over only the data field in the packet and does not include the PID, which has its own check field.

The maximum data payload size allowed for low-speed devices is 8 bytes. The maximum data payload size for full-speed devices is 1023. The maximum data payload size for high-speed devices is 1024 bytes.

8.4.5 Handshake Packets

Handshake packets, as shown in Figure 8-16, consist of only a PID. Handshake packets are used to report the status of a data transaction and can return values indicating successful reception of data, command acceptance or rejection, flow control, and halt conditions. Only transaction types that support flow control can return handshakes. Handshakes are always returned in the handshake phase of a transaction and may be returned, instead of data, in the data phase. Handshake packets are delimited by an EOP after one byte of packet field. If a packet decodes as an otherwise valid handshake but does not terminate with an EOP after one byte, it must be considered invalid and ignored by the receiver.

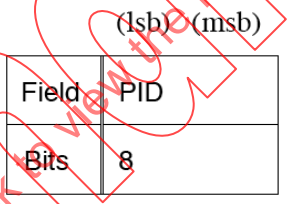


Figure 8-16. Handshake Packet

There are four types of handshake packets and one special handshake packet:

- **ACK** indicates that the data packet was received without bit stuff or CRC errors over the data field and that the data PID was received correctly. ACK may be issued either when sequence bits match and the receiver can accept data or when sequence bits mismatch and the sender and receiver must resynchronize to each other (refer to Section 8.6 for details). An ACK handshake is applicable only in transactions in which data has been transmitted and where a handshake is expected. ACK can be returned by the host for IN transactions and by a function for OUT, SETUP, or PING transactions.
- **NAK** indicates that a function was unable to accept data from the host (OUT) or that a function has no data to transmit to the host (IN). NAK can only be returned by functions in the data phase of IN transactions or the handshake phase of OUT or PING transactions. The host can never issue NAK. NAK is used for flow control purposes to indicate that a function is temporarily unable to transmit or receive data, but will eventually be able to do so without need of host intervention.
- **STALL** is returned by a function in response to an IN token or after the data phase of an OUT or in response to a PING transaction (see Figure 8-30 and Figure 8-38). STALL indicates that a function is unable to transmit or receive data, or that a control pipe request is not supported. The state of a function after returning a STALL (for any endpoint except the default endpoint) is undefined. The host is not permitted to return a STALL

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

under any condition.

The STALL handshake is used by a device in one of two distinct occasions. The first case, known as “functional stall,” is when the *Halt* feature associated with the endpoint is set. (The *Halt* feature is specified in Chapter 9 of this document.) A special case of the functional stall is the “commanded stall.” Commanded stall occurs when the host explicitly sets the endpoint’s *Halt* feature, as detailed in Chapter 9. Once a function’s endpoint is halted, the function must continue returning STALL until the condition causing the halt has been cleared through host intervention.

The second case, known as “protocol stall,” is detailed in Section 8.5.3. Protocol stall is unique to control pipes. Protocol stall differs from functional stall in meaning and duration. A protocol STALL is returned during the Data or Status stage of a control transfer, and the STALL condition terminates at the beginning of the next control transfer (Setup). The remainder of this section refers to the general case of a functional stall.

- **NYET** is a high-speed only handshake that is returned in two circumstances. It is returned by a high-speed endpoint as part of the PING protocol described later in this chapter. NYET may also be returned by a hub in response to a split-transaction when the full-/low-speed transaction has not yet been completed or the hub is otherwise not able to handle the split-transaction. See Chapter 11 for more details.
- **ERR** is a high-speed only handshake that is returned to allow a high-speed hub to report an error on a full-/low-speed bus. It is only returned by a high-speed hub as part of the split transaction protocol. See Chapter 11 for more details.

8.4.6 Handshake Responses

Transmitting and receiving functions must return handshakes based upon an order of precedence detailed in Table 8-4 through Table 8-6. Not all handshakes are allowed, depending on the transaction type and whether the handshake is being issued by a function or the host. Note that if an error occurs during the transmission of the token to the function, the function will not respond with any packets until the next token is received and successfully decoded.

8.4.6.1 Function Response to IN Transactions

Table 8-4 shows the possible responses a function may make in response to an IN token. If the function is unable to send data, due to a halt or a flow control condition, it issues a STALL or NAK handshake, respectively. If the function is able to issue data, it does so. If the received token is corrupted, the function returns no response.

Table 8-4. Function Responses to IN Transactions

Token Received Corrupted	Function Tx Endpoint Halt Feature	Function Can Transmit Data	Action Taken
Yes	Don’t care	Don’t care	Return no response
No	Set	Don’t care	Issue STALL handshake
No	Not set	No	Issue NAK handshake
No	Not set	Yes	Issue data packet

8.4.6.2 Host Response to IN Transactions

Table 8-5 shows the host response to an IN transaction. The host is able to return only one type of handshake: ACK. If the host receives a corrupted data packet, it discards the data and issues no response. If the host cannot accept data from a function, (due to problems such as internal buffer overrun) this condition is considered to be an error and the host returns no response. If the host is able to accept data and the data packet is received error-free, the host accepts the data and issues an ACK handshake.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 8-5. Host Responses to IN Transactions

Data Packet Corrupted	Host Can Accept Data	Handshake Returned by Host
Yes	N/A	Discard data, return no response
No	No	Discard data, return no response
No	Yes	Accept data, issue ACK

8.4.6.3 Function Response to an OUT Transaction

Handshake responses for an OUT transaction are shown in Table 8-6. Assuming successful token decode, a function, upon receiving a data packet, may return any one of the three handshake types. If the data packet was corrupted, the function returns no handshake. If the data packet was received error-free and the function's receiving endpoint is halted, the function returns STALL. If the transaction is maintaining sequence bit synchronization and a mismatch is detected (refer to Section 8.6 for details), then the function returns ACK and discards the data. If the function can accept the data and has received the data error-free, it returns ACK. If the function cannot accept the data packet due to flow control reasons, it returns NAK.

Table 8-6. Function Responses to OUT Transactions in Order of Precedence

Data Packet Corrupted	Receiver Halt Feature	Sequence Bits Match	Function Can Accept Data	Handshake Returned by Function
Yes	N/A	N/A	N/A	None
No	Set	N/A	N/A	STALL
No	Not set	No	N/A	ACK
No	Not set	Yes	Yes	ACK
No	Not set	Yes	No	NAK

8.4.6.4 Function Response to a SETUP Transaction

SETUP defines a special type of host-to-function data transaction that permits the host to initialize an endpoint's synchronization bits to those of the host. Upon receiving a SETUP token, a function must accept the data. A function may not respond to a SETUP token with either STALL or NAK, and the receiving function must accept the data packet that follows the SETUP token. If a non-control endpoint receives a SETUP token, it must ignore the transaction and return no response.

8.5 Transaction Packet Sequences

The packets that comprise a transaction varies depending on the endpoint type. There are four endpoint types: bulk, control, interrupt, and isochronous.

A host controller and device each require different state machines to correctly sequence each type of transaction. Figures in the following sections show state machines that define the correct sequencing of packets within a transaction of each type. The diagrams should not be taken as a required implementation, but to specify the required behavior.

Figure 8-17 shows the legend for the state machine diagrams. A circle with a three-line border indicates a reference to another (hierarchical) state machine. A circle with a two-line border indicates an initial state. A circle with a single-line border represents a simple state.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

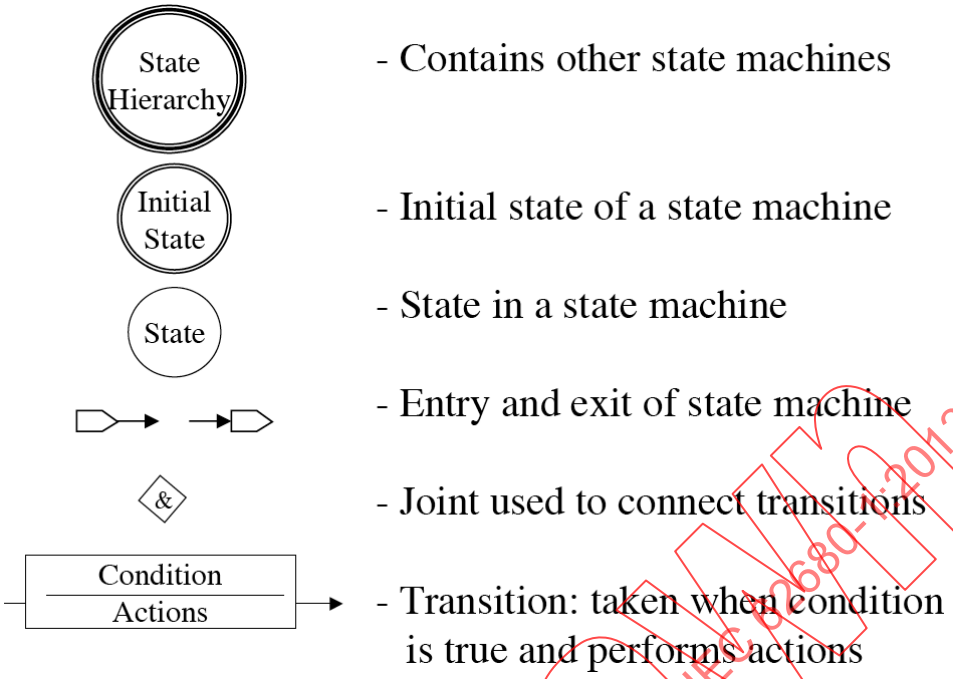


Figure 8-17. Legend for State Machines

The “tab” shapes with arrows are the entry or exit (respectively in the legend) to/from the state machine. The entry/exit relates to another state in a state machine at a higher level in the state machine hierarchy.

A diamond (joint) is used to join several transitions to a common point. A joint allows a single input transition with multiple output transitions or multiple input transitions and a single output transition. All conditions on the transitions of a path involving a joint must be true for the path to be taken. A path is simply a sequence of transitions involving one or more joints.

A transition is labeled with a block with a line in the middle separating the (upper) condition and the (lower) actions. The condition is required to be true to take the transition. The syntax for actions and conditions is VHDL. The actions are performed if the transition is taken. A circle includes a name in bold and optionally one or more actions that are performed upon entry to the state.

The host controller and device state machines are in a context as shown in Figure 8-18. The host controller determines the next transaction to run for an endpoint and issues a command (HC_cmd) to the host controller state machines. This causes the host controller state machines to issue one or more packets to move over the downstream bus (HSD1).

The device receives these packets from the bus (HSD2), reacts to the received packet, and interacts with its function(s) via the state of the corresponding endpoint (in the EP_array). Then the device may respond with a packet on the upstream bus (HSU1). The host controller state machines can receive a packet from the bus (HSU2) and provide a result of the transaction back to the host controller (HC_resp). The details of what packets are sent on the bus is determined by the transfer type for the endpoint and what bus activity the state machines observe.

The state machines are presented in a hierarchical form. Figure 8-19 shows the top level state machines for the host controller. The non-split transactions are presented in the remainder of this chapter. The split transaction state machines (HC_Do_start and HC_Do_complete) are described and shown in Chapter 11.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

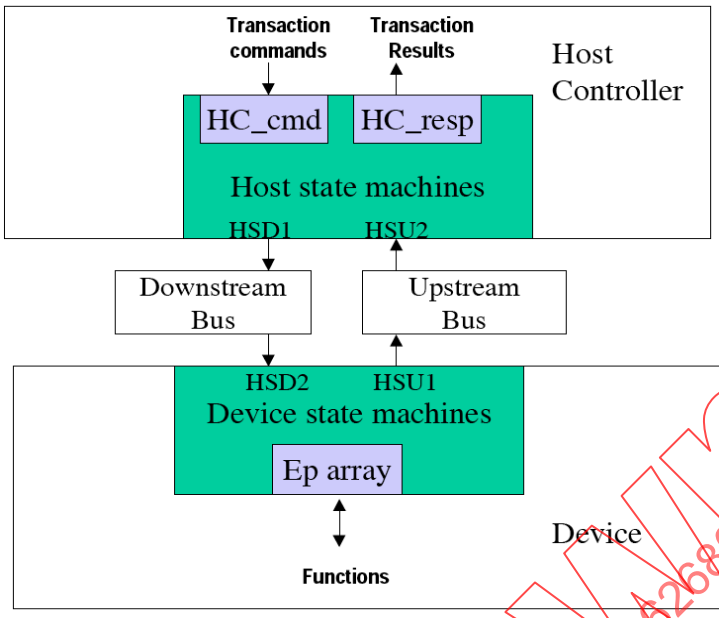


Figure 8-18. State Machine Context Overview

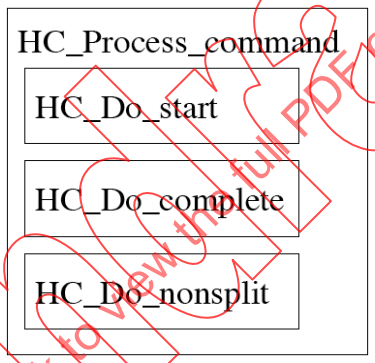


Figure 8-19. Host Controller Top Level Transaction State Machine Hierarchy Overview

The host controller state machines are located in the host controller. The host controller causes packets to be issued downstream (labeled as HSD1) and it receives upstream packets (labeled as HSU2).

The device state machines are located in the device. The device causes packets to be issued upstream (labeled as HSU1) and it receives downstream packets (labeled as HSD2).

The host controller has commands that tell it what transaction to issue next for an endpoint. The host controller tracks transactions for several endpoints. The host controller state machines sequence to determine what the host controller needs to do next for the current endpoint. The device has a state for each of its endpoints. The device state machines sequence to determine what reaction the device has to a transaction.

The appendix includes some declarations that were used in constructing the state machines and may be useful in understanding additional details of the state machines. There are several pseudo-code procedures and functions for conditions and actions. Simple descriptions of them are also included in the appendix.

Figure 8-20 shows an overview of the overall state machine hierarchy for the host controller for the non-split transaction types. Figure 8-21 shows the hierarchy of the device state machines. The state machines common to endpoint types are presented first. The lowest

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

level endpoint type specific state machines are presented in each following endpoint type section.

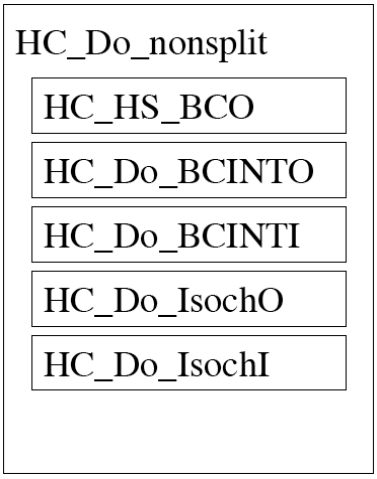


Figure 8-20. Host Controller Non-split Transaction State Machine Hierarchy Overview

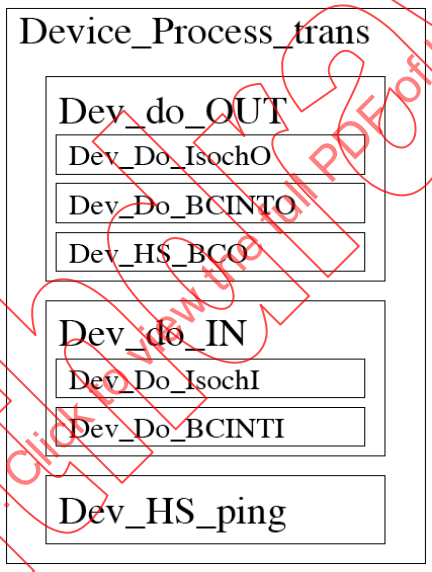


Figure 8-21. Device Transaction State Machine Hierarchy Overview

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Global Actions **Concurrent Statements** **Architecture Declarations** **Signals Status** **State Register Statements**

Package List

ieee std_logic_1164
ieee numeric_std
usb2statemachines behav_package

```
SIGNAL  SCOPE DEFAULT
hsul    OUT   (BULK,NAK,0,0,ok,in_dir,TRUE,ALLDATA,FALSE,FA
device  INT   '0'
token   INT   '0'
```

Process Declarations

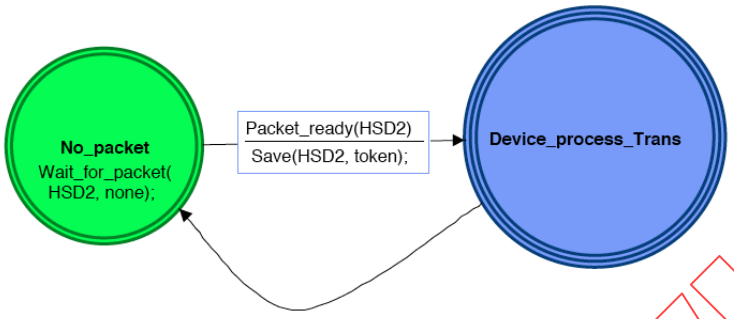
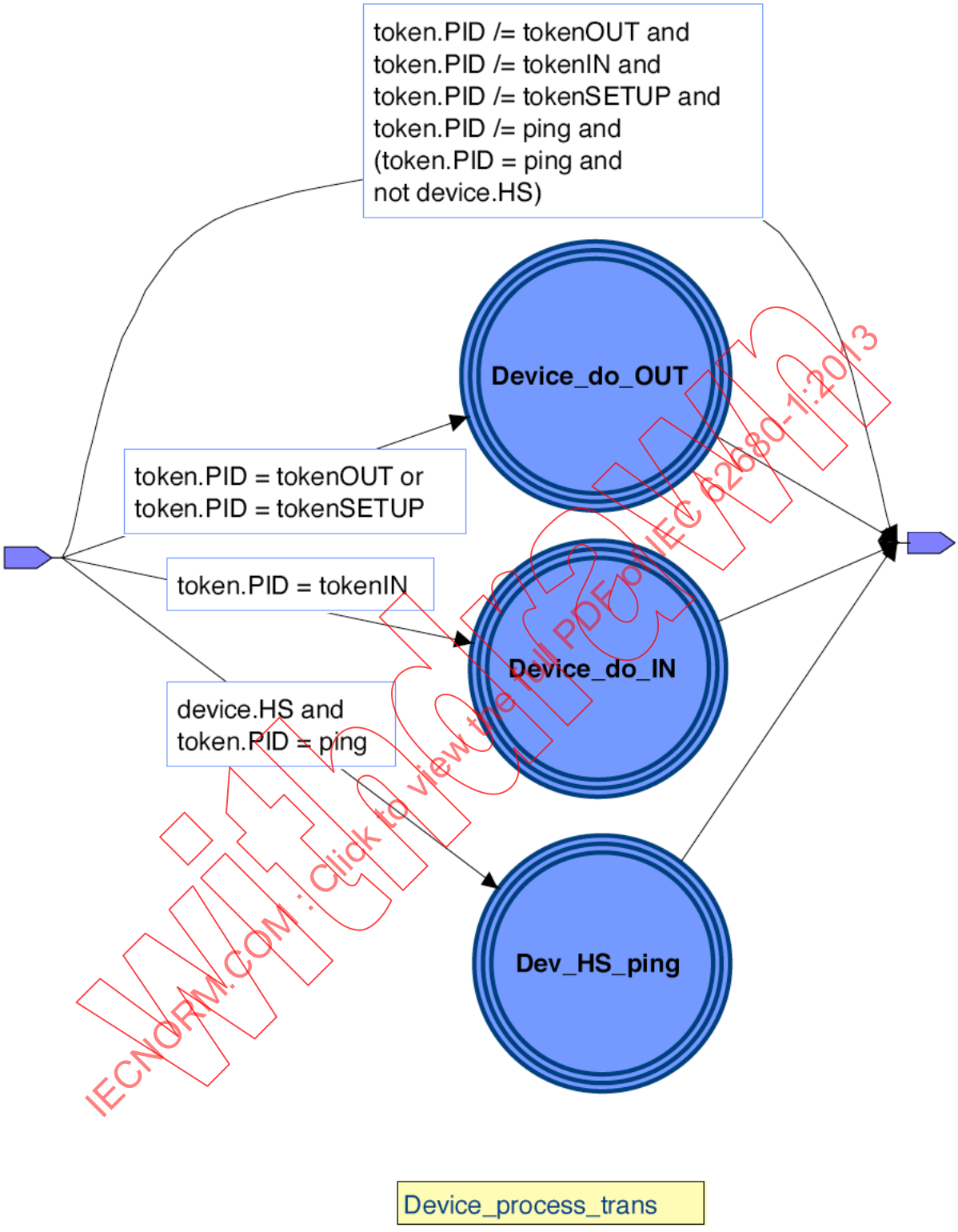


Figure 8-22. Device Top Level State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 8-23. Device_process_Trans State Machine

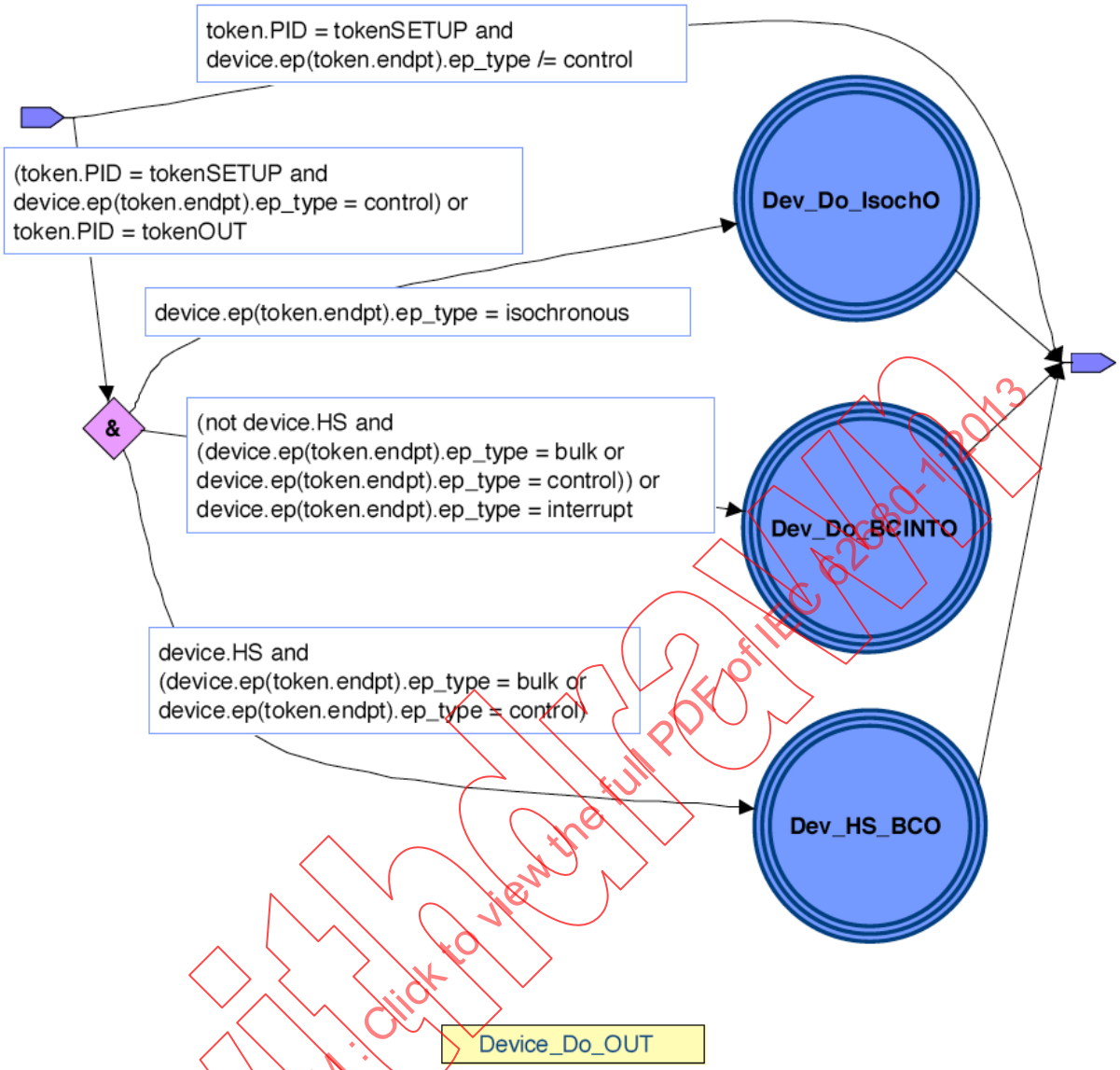


Figure 8-24. Dev_do_OUT State Machine

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

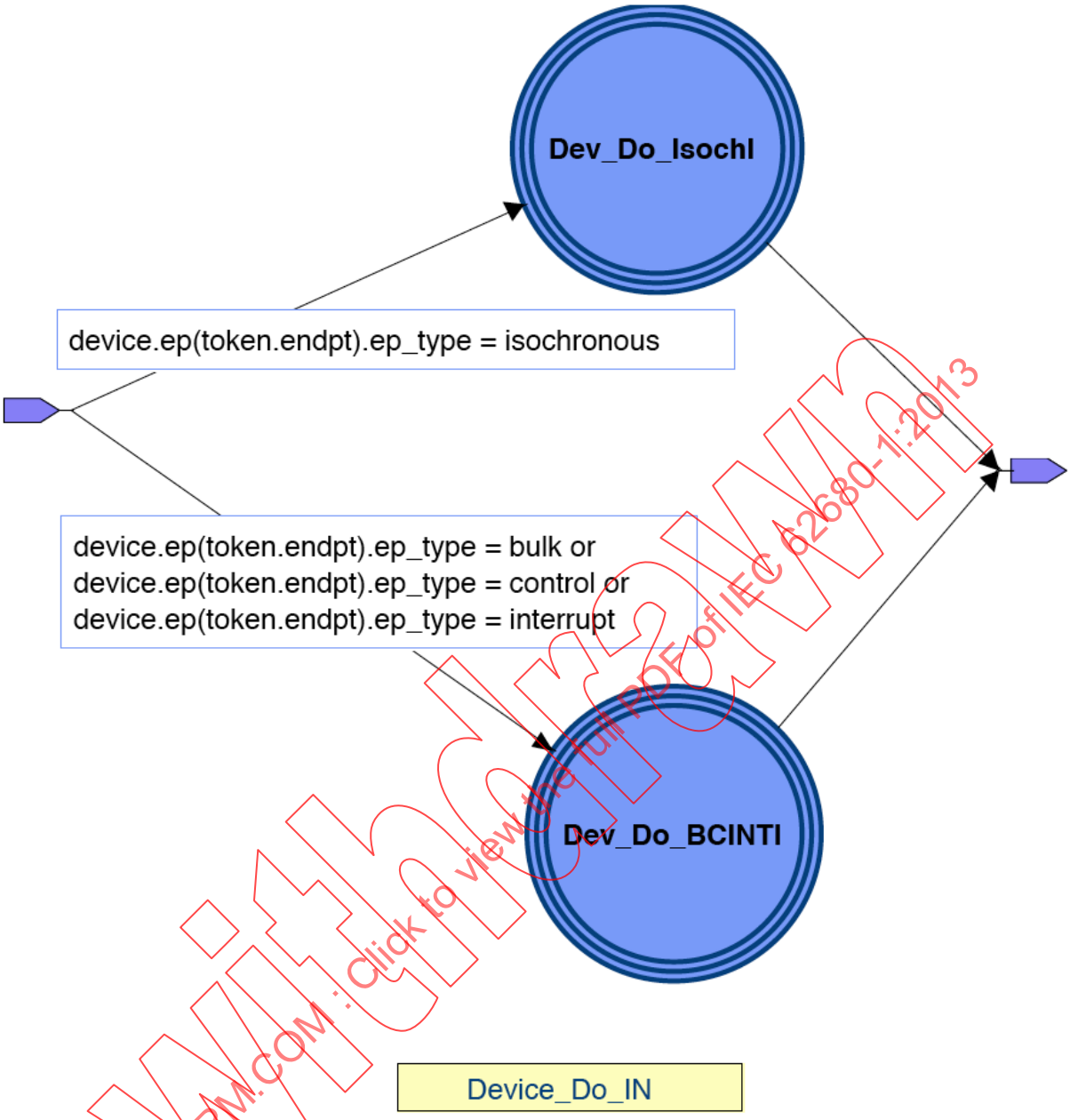


Figure 8-25. Dev_do_IN State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

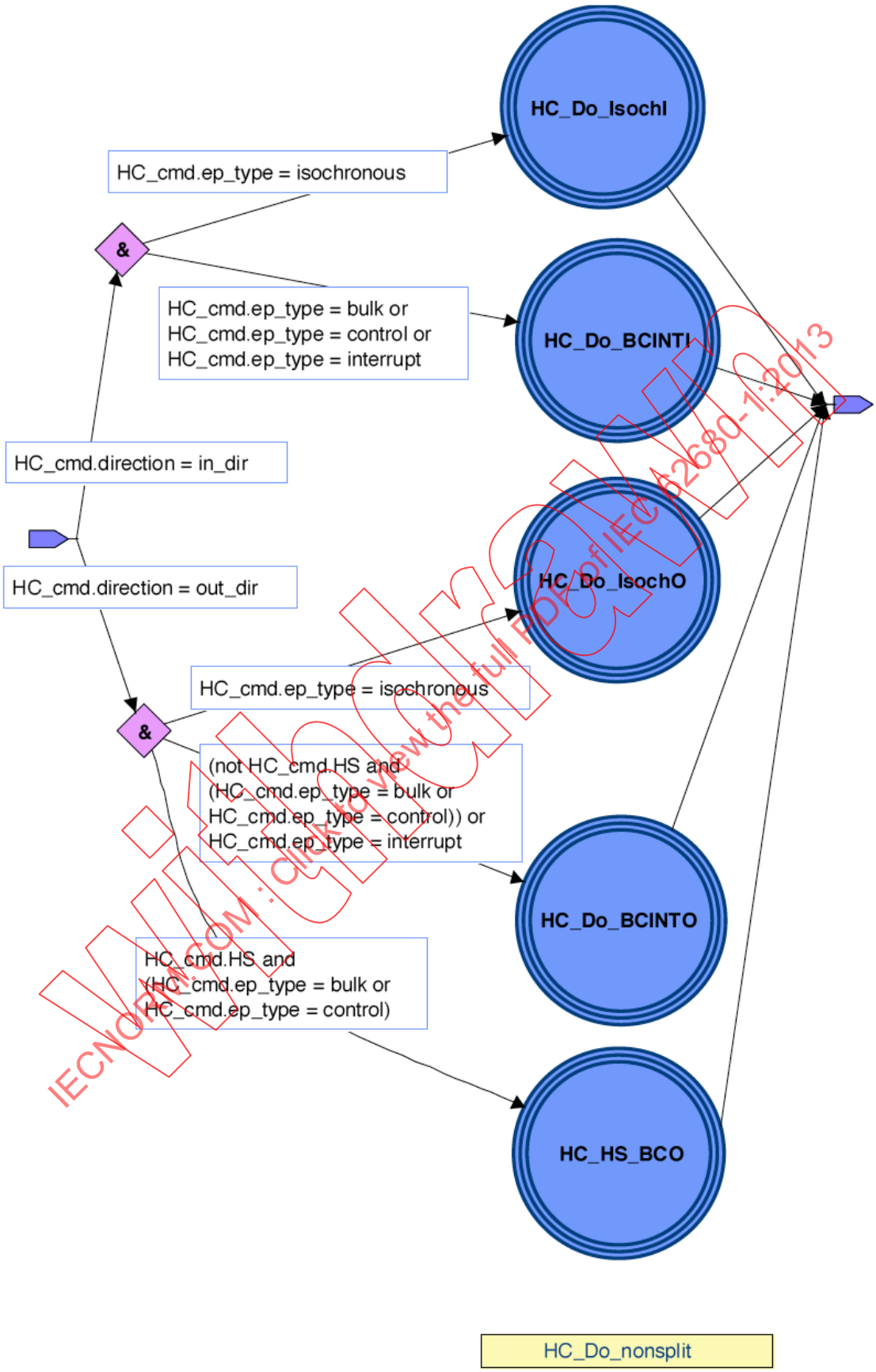


Figure 8-26. HC_Do_nonsplit State Machine

8.5.1 NAK Limiting via Ping Flow Control

Full-/low-speed devices can have bulk/control endpoints that take time to process their data and, therefore, respond to OUT transactions with a NAK handshake. This handshake response indicates that the endpoint did not accept the data because it did not have space for the data. The host controller is expected to retry the transaction at some future time when the endpoint has space available. Unfortunately, by the time the endpoint NAKs, most of the full-/low-speed bus time for the transaction had been used. This means that the full-/low-speed bus has poor utilization when there is a high frequency of NAK'd OUT transactions.

High-speed devices must support an improved NAK mechanism for Bulk OUT and Control endpoints and transactions. Control endpoints must support this protocol for an OUT transaction in the data and status stages. The control Setup stage must not support the PING protocol.

This mechanism allows the device to tell the host controller whether it has sufficient endpoint space for the next OUT transaction. If the device endpoint does not have space, the host controller can choose to delay a transaction attempt for this endpoint and instead try some other transaction. This can lead to improved bus utilization. The mechanism avoids using bus time to send data until the host controller knows that the endpoint has space for the data.

The host controller queries the high-speed device endpoint with a PING special token. The PING special token packet is a normal token packet as shown in Figure 8-5. The endpoint either responds to the PING with a NAK or an ACK handshake.

A NAK handshake indicates that the endpoint does not have space for a *wMaxPacketSize* data payload. The host controller will retry the PING at some future time to query the endpoint again. A device can respond to a PING with a NAK for long periods of time. A NAK response is not a reason for the host controller to retire a transfer request. If a device responds with a NAK in a (micro)frame, the host controller may choose to issue the next transaction in the next *bInterval* specified for the endpoint. However, the device must be prepared to receive PINGs as sequential transactions, e.g., one immediately after the other.

An ACK handshake indicates the endpoint has space for a *wMaxPacketSize* data payload. The host controller must generate an OUT transaction with a DATA phase as the next transaction to the endpoint. The host controller may generate other transactions to other devices or endpoints before the OUT/DATA transaction for this endpoint.

If the endpoint responds to the OUT/DATA transaction with an ACK handshake, this means the endpoint accepted the data successfully and has room for another *wMaxPacketSize* data payload. The host controller continues with OUT/DATA transactions (which are not required to be the next transactions on the bus) as long as it has transactions to generate.

If the endpoint instead responds to the OUT/DATA transaction with a NYET handshake, this means that the endpoint accepted the data but does not have room for another *wMaxPacketSize* data payload. The host controller must return to using a PING token until the endpoint indicates it has space.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

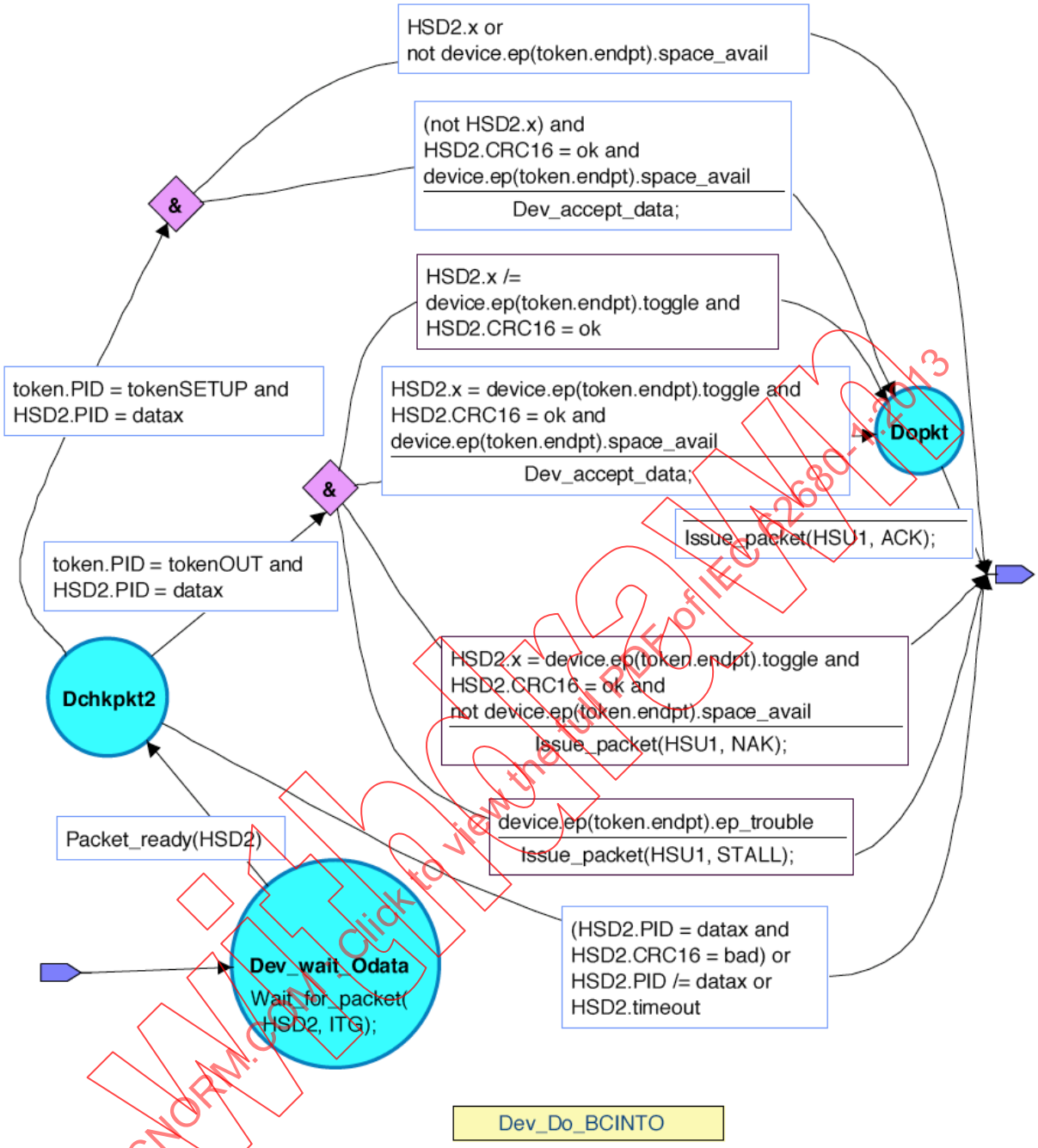


Figure 8-27. Host High-speed Bulk OUT/Control Ping State Machine

8.5.1.1 NAK Responses to OUT/DATA During PING Protocol

The endpoint may also respond to the OUT/DATA transaction with a NAK handshake. This means that the endpoint did not accept the data and does not have space for a *wMaxPacketSize* data payload at this time. The host controller must return to using a PING token until the endpoint indicates it has space.

A NAK response is expected to be an unusual occurrence. A high-speed bulk/control endpoint must specify its maximum NAK rate in its endpoint descriptor. The endpoint is allowed to NAK at most one time each *bInterval* period. A NAK suggests that the endpoint responded to a previous OUT or PING with an inappropriate handshake, or that the endpoint transitioned into a state where it (temporarily) could not accept data. An endpoint can use a

bInterval of zero to indicate that it never NAKs. An endpoint must always be able to accept a PING from the host, even if it never NAKs.

If a timeout occurs after the data phase, the host must return to using a PING token. Note that a transition back to the PING state does not affect the data toggle state of the transaction data phase.

Figure 8-27 shows the host controller state machine for the interactions and transitions between PING and OUT/DATA tokens and the allowed ACK, NAK, and NYET handshakes for the PING mechanism.

Figure 8-29 shows the device endpoint state machine for PING based on the buffer space the endpoint has available.

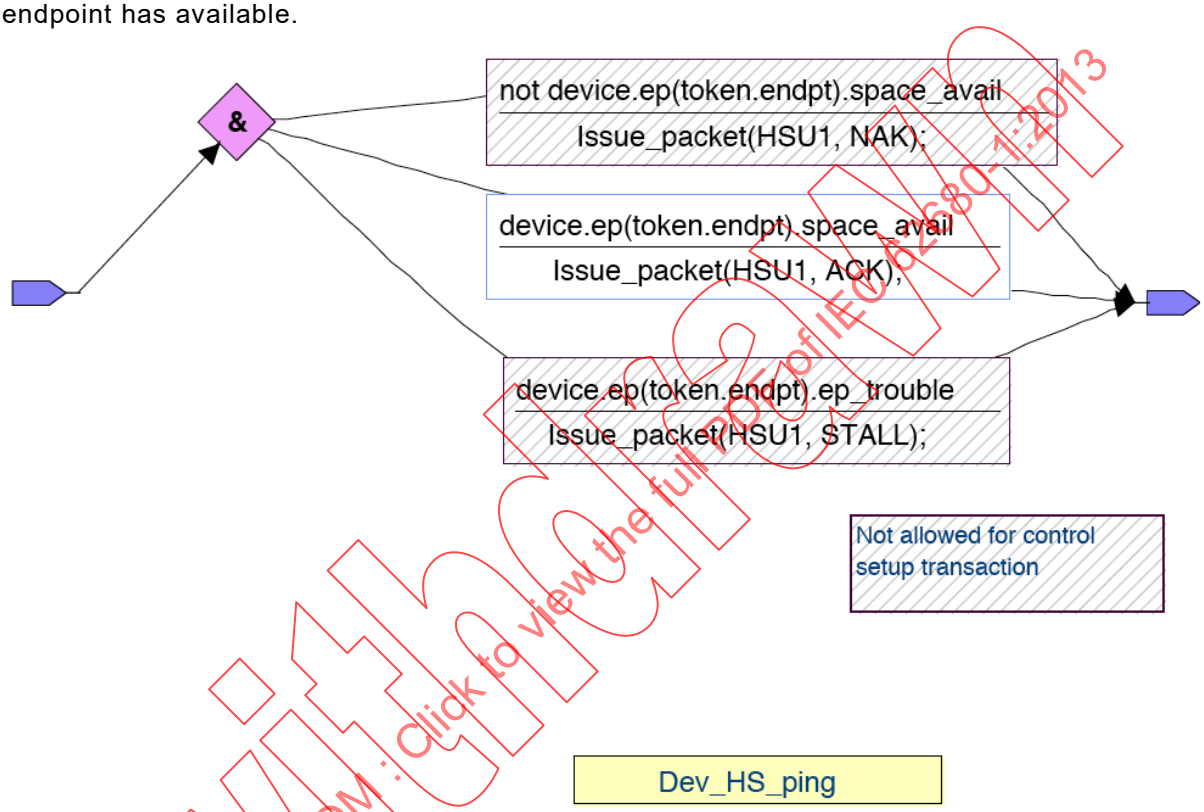


Figure 8-28. Dev_HS_ping State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

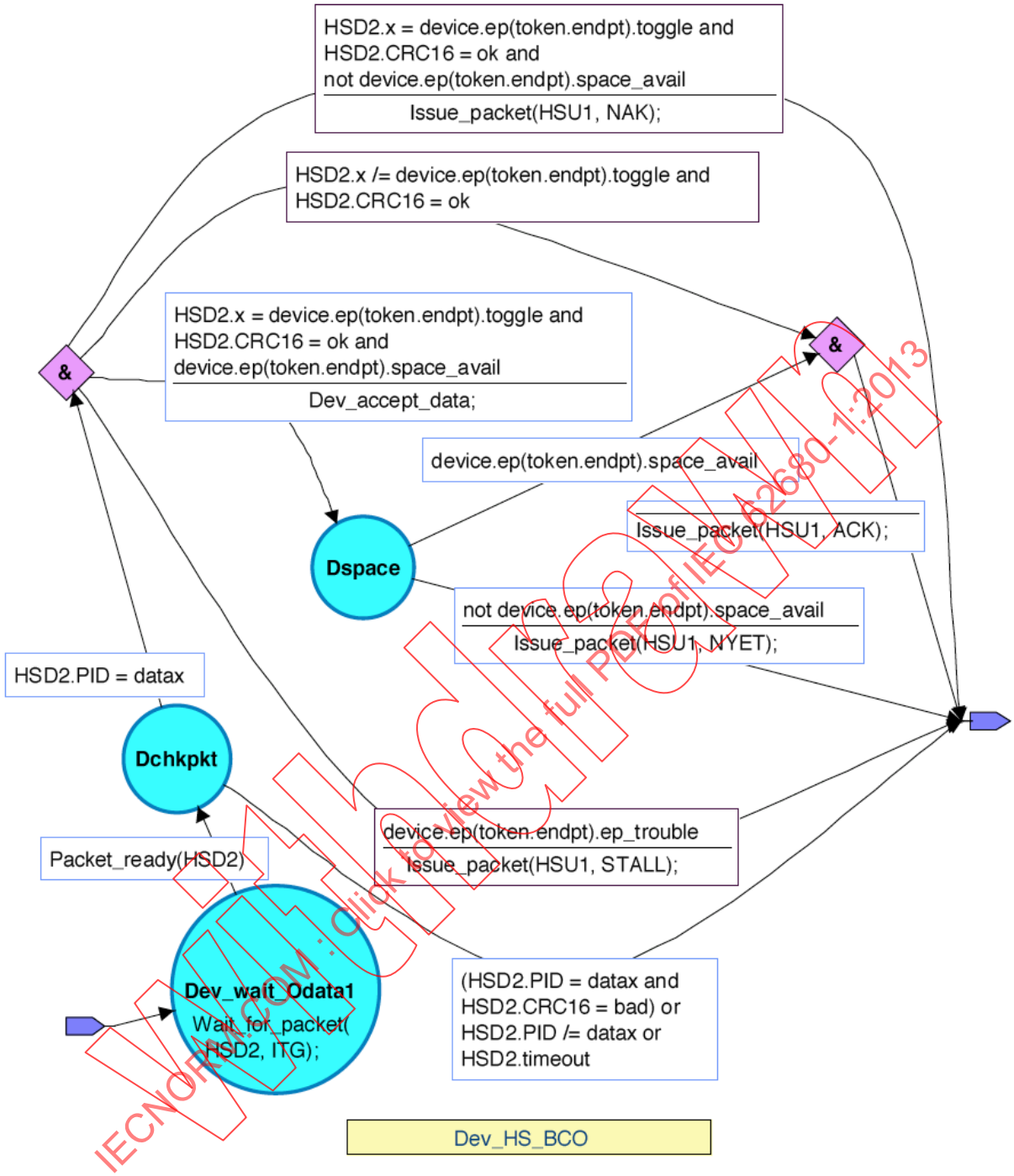


Figure 8-29. Device High-speed Bulk OUT /Control State Machine

Full-/low-speed devices/endpoints must not support the PING protocol. Host controllers must not support the PING protocol for full-/low-speed devices.

Note: The PING protocol is also not included as part of the split-transaction protocol definition. Some split-transactions have equivalent flow control without using PING. Other split-transactions will not benefit from PING as defined. In any case, split-transactions that can return a NAK handshake have small data payloads which should have minor high-speed bus impact. Hubs must support PING on their control endpoint, but PING is not defined for the split-transactions that are used to communicate with full-/low-speed devices supported by a hub.

8.5.2 Bulk Transactions

Bulk transaction types are characterized by the ability to guarantee error-free delivery of data between the host and a function by means of error detection and retry. Bulk transactions use a three-phase transaction consisting of token, data, and handshake packets as shown in Figure 8-30. Under certain flow control and halt conditions, the data phase may be replaced with a handshake resulting in a two-phase transaction in which no data is transmitted. The PING and NYET packets must only be used with devices operating at high-speed.

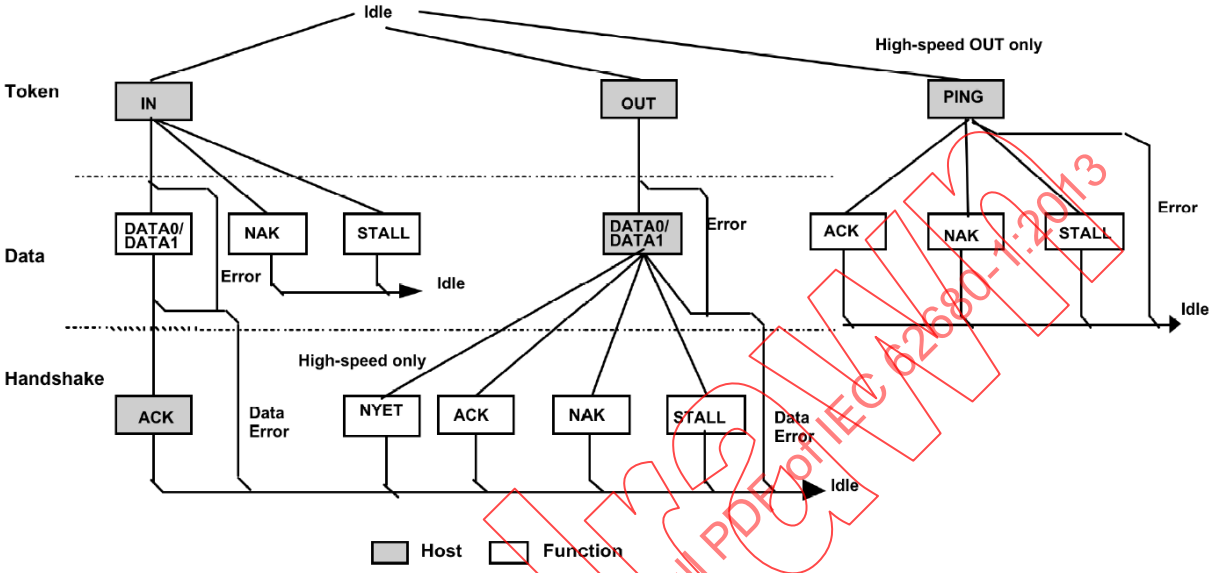


Figure 8-30. Bulk Transaction Format

When the host is ready to receive bulk data, it issues an IN token. The function endpoint responds by returning either a data packet or, should it be unable to return data, a NAK or STALL handshake. NAK indicates that the function is temporarily unable to return data, while STALL indicates that the endpoint is permanently halted and requires USB System Software intervention. If the host receives a valid data packet, it responds with an ACK handshake. If the host detects an error while receiving data, it returns no handshake packet to the function.

When the host is ready to transmit bulk data, it first issues an OUT token packet followed by a data packet (or PING special token packet, see Section 8.5.1). If the data is received without error by the function, it will return one of three (or four including NYET, for a device operating at high-speed) handshakes:

- ACK indicates that the data packet was received without errors and informs the host that it may send the next packet in the sequence.
- NAK indicates that the data was received without error but that the host should resend the data because the function was in a temporary condition preventing it from accepting the data (e.g., buffer full).
- If the endpoint was halted, STALL is returned to indicate that the host should not retry the transmission because there is an error condition on the function.

If the data packet was received with a CRC or bit stuff error, no handshake is returned.

Figure 8-31 and Figure 8-32 show the host and device state machines respectively for bulk, control, and interrupt OUT full/low-speed transactions. Figure 8-27, Figure 8-28, and Figure 8-29 show the state machines for high-speed transactions. Figure 8-33 and Figure 8-34 show the host and device state machines respectively for bulk, control, and interrupt IN transactions.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

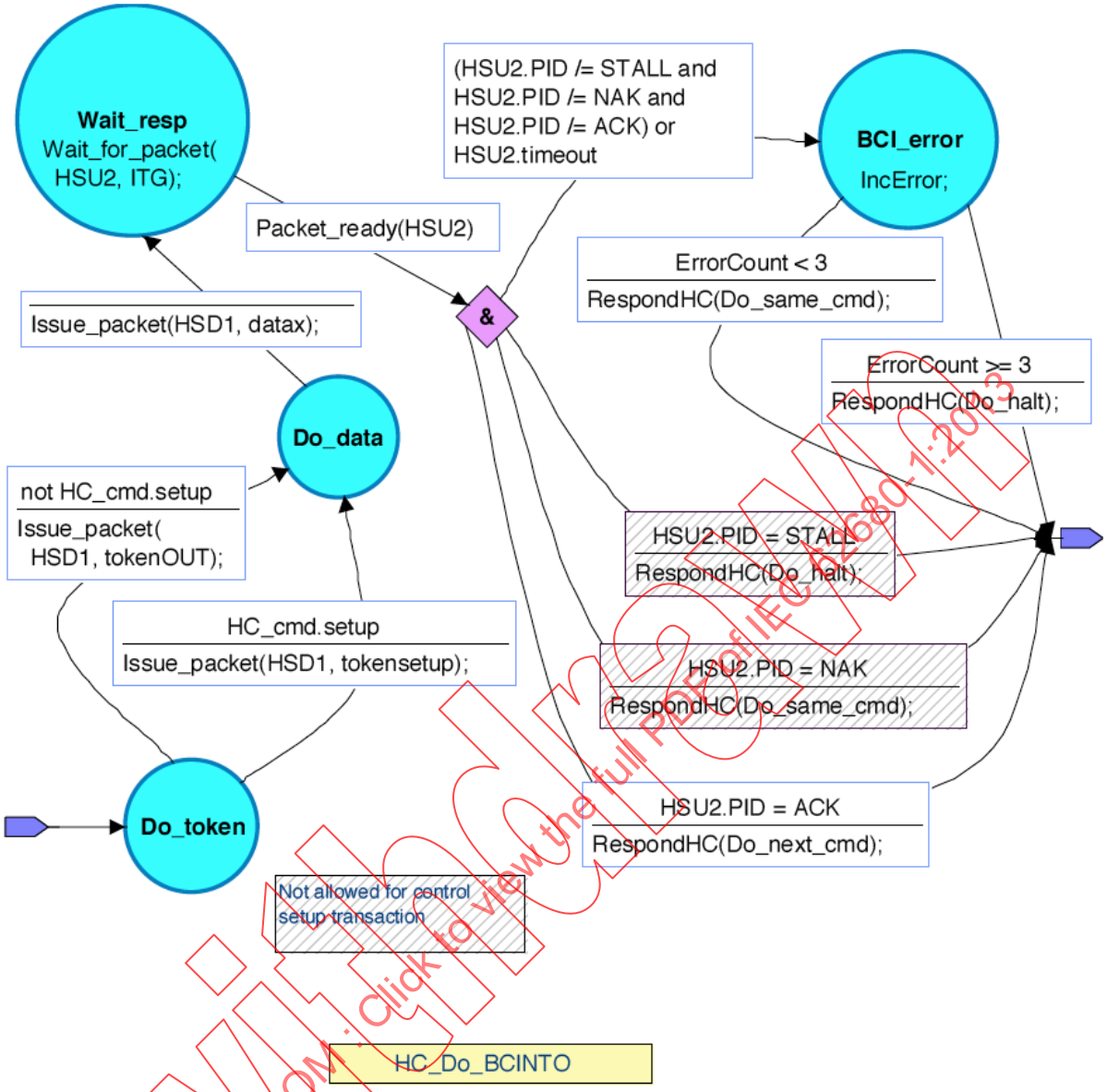


Figure 8-31. Bulk/Control/Interrupt OUT Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

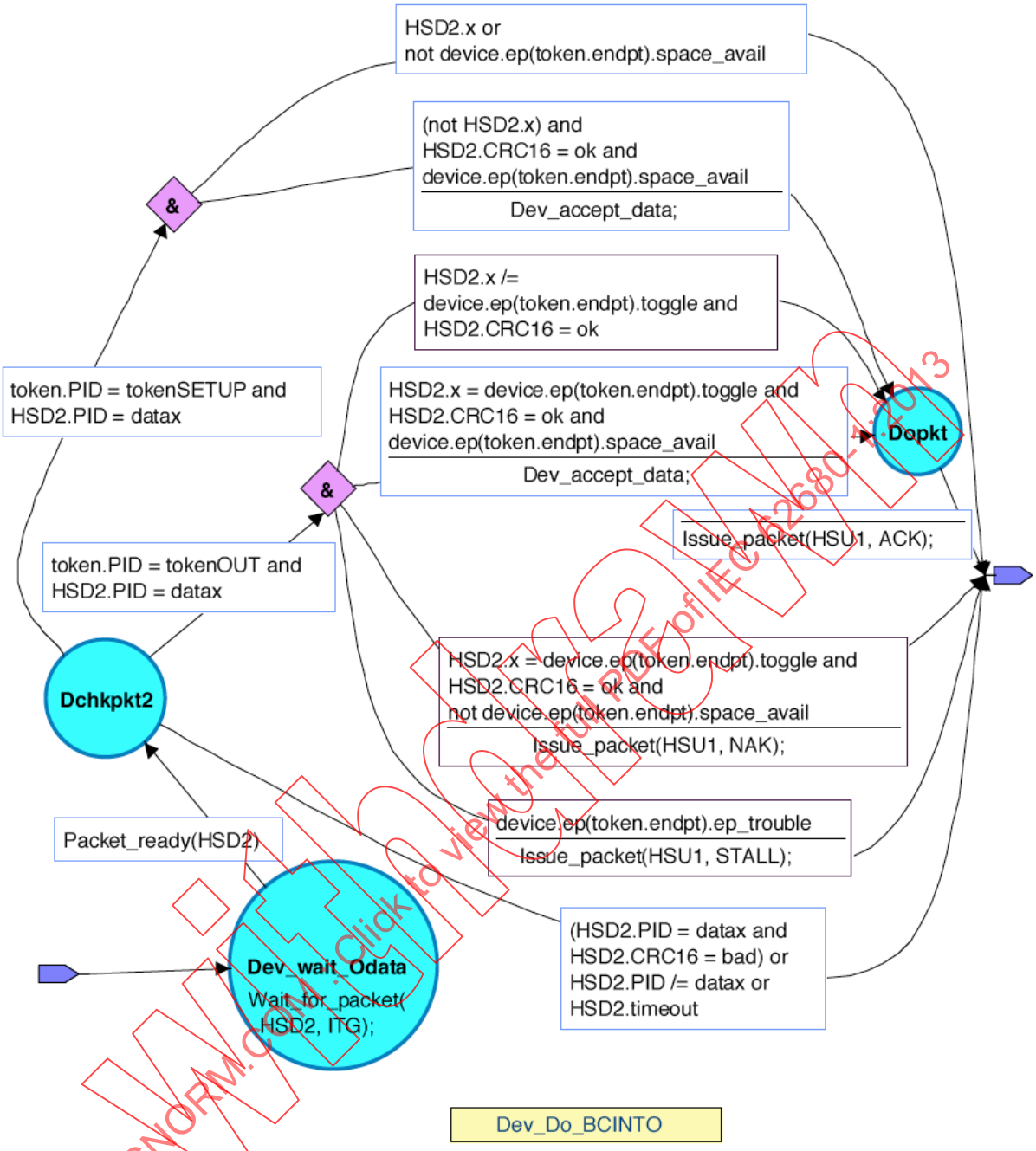


Figure 8-32. Bulk/Control/Interrupt OUT Transaction Device State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

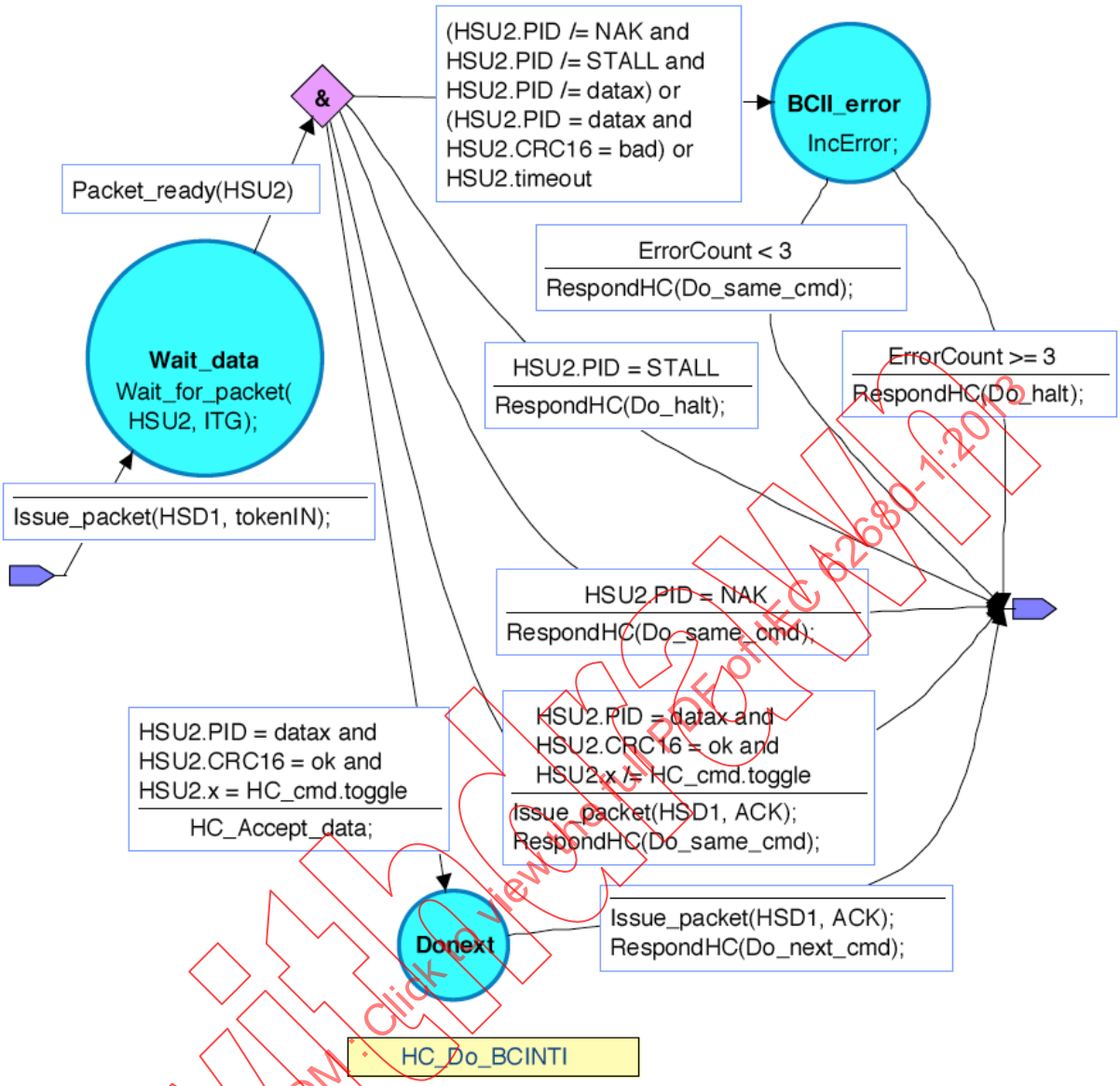


Figure 8-33. Bulk/Control/Interrupt IN Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

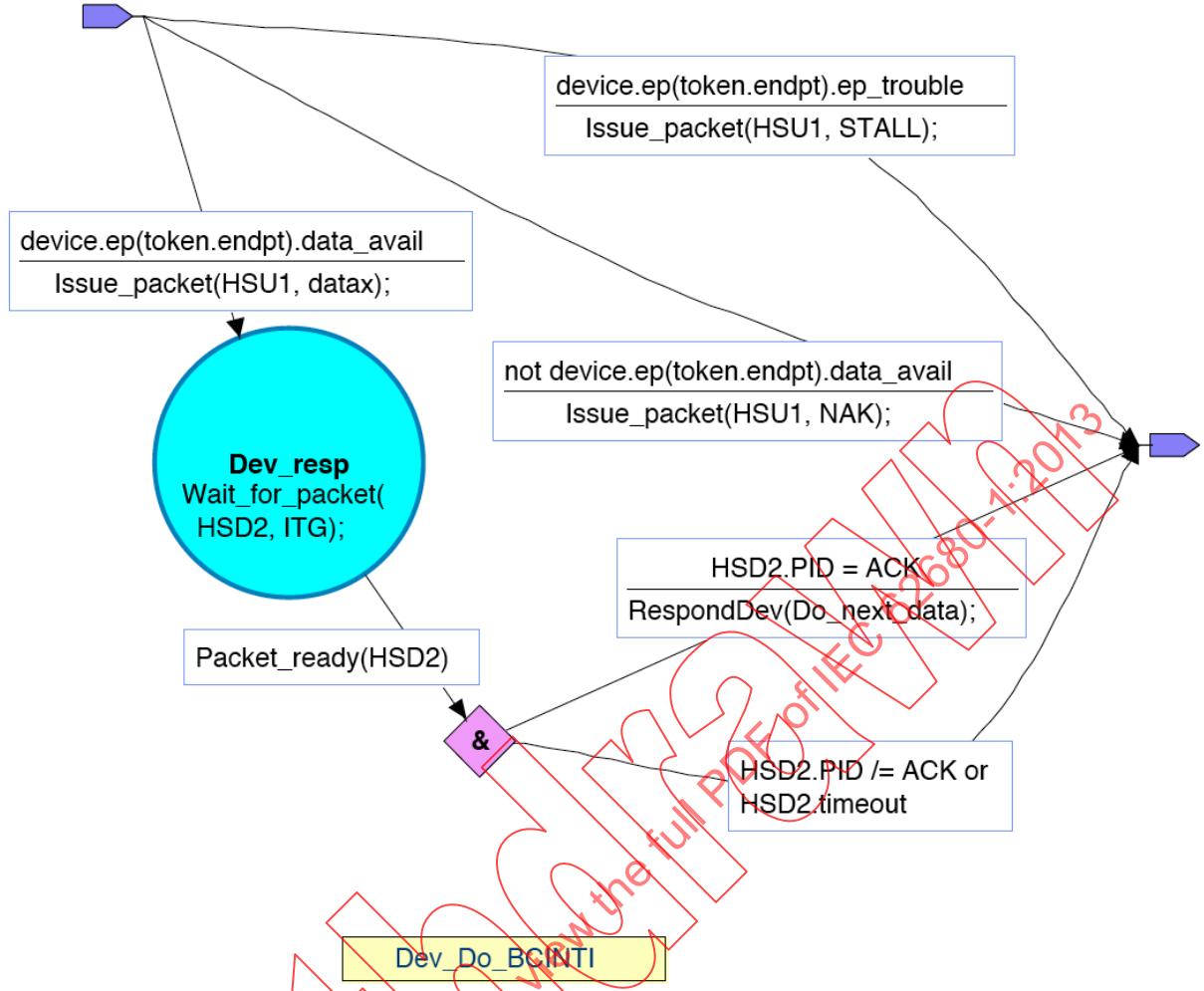


Figure 8-34. Bulk/Control/Interrupt IN Transaction Device State Machine

Figure 8-35 shows the sequence bit and data PID usage for bulk reads and writes. Data packet synchronization is achieved via use of the data sequence toggle bits and the DATA0/DATA1 PIDs. A bulk endpoint's toggle sequence is initialized to DATA0 when the endpoint experiences any configuration event (configuration events are explained in Sections 9.1.1.5 and 9.4.5). Data toggle on an endpoint is NOT initialized as the direct result of a short packet transfer or the retirement of an IRP.

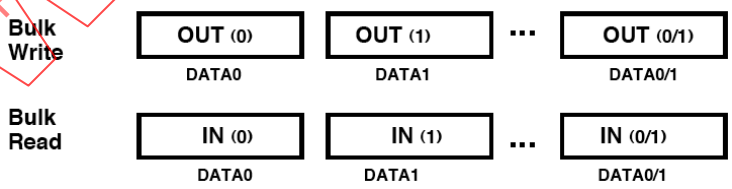


Figure 8-35. Bulk Reads and Writes

The host always initializes the first transaction of a bus transfer to the DATA0 PID with a configuration event. The second transaction uses a DATA1 PID, and successive data transfers alternate for the remainder of the bulk transfer. The data packet transmitter toggles upon receipt of ACK, and the receiver toggles upon receipt and acceptance of a valid data packet (refer to Section 8.6).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8.5.3 Control Transfers

Control transfers minimally have two transaction stages: Setup and Status. A control transfer may optionally contain a Data stage between the Setup and Status stages. During the Setup stage, a SETUP transaction is used to transmit information to the control endpoint of a function. SETUP transactions are similar in format to an OUT but use a SETUP rather than an OUT PID. Figure 8-36 shows the SETUP transaction format. A SETUP always uses a DATA0 PID for the data field of the SETUP transaction. The function receiving a SETUP must accept the SETUP data and respond with ACK; if the data is corrupted, discard the data and return no handshake.

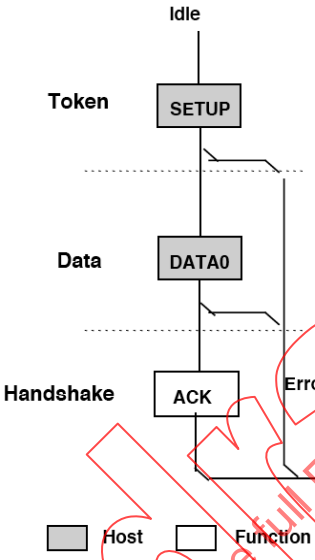


Figure 8-36. Control SETUP Transaction

The Data stage, if present, of a control transfer consists of one or more IN or OUT transactions and follows the same protocol rules as bulk transfers. All the transactions in the Data stage must be in the same direction (i.e., all INs or all OUTs). The amount of data to be sent during the data stage and its direction are specified during the Setup stage. If the amount of data exceeds the prenegotiated data packet size, the data is sent in multiple transactions (INs or OUTs) that carry the maximum packet size. Any remaining data is sent as a residual in the last transaction.

The Status stage of a control transfer is the last transaction in the sequence. The status stage transactions follow the same protocol sequence as bulk transactions. Status stage for devices operating at high-speed also includes the PING protocol. A Status stage is delineated by a change in direction of data flow from the previous stage and always uses a DATA1 PID. If, for example, the Data stage consists of OUTs, the status is a single IN transaction. If the control sequence has no Data stage, then it consists of a Setup stage followed by a Status stage consisting of an IN transaction.

Figure 8-37 shows the transaction order, the data sequence bit value, and the data PID types for control read and write sequences. The sequence bits are displayed in parentheses.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

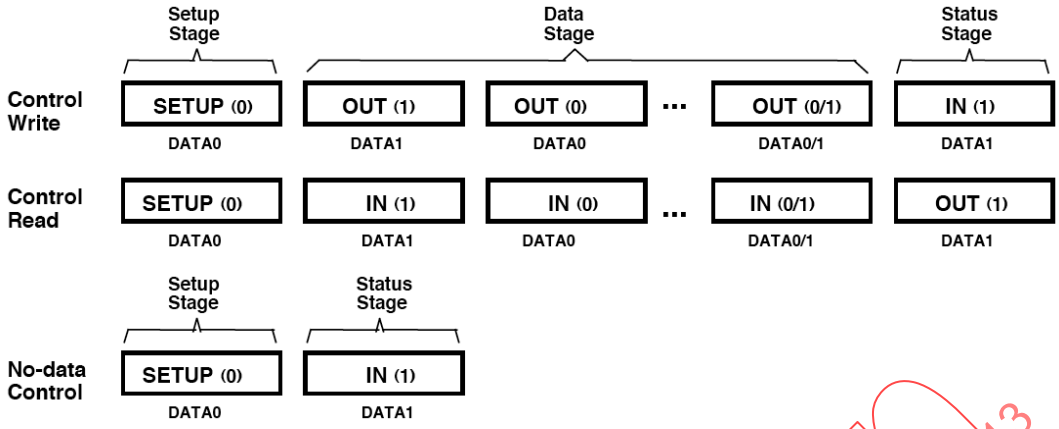


Figure 8-37. Control Read and Write Sequences

When a STALL handshake is sent by a control endpoint in either the Data or Status stages of a control transfer, a STALL handshake must be returned on all succeeding accesses to that endpoint until a SETUP PID is received. The endpoint is not required to return a STALL handshake after it receives a subsequent SETUP PID. For the default endpoint, if an ACK handshake is returned for the SETUP transaction, the host expects that the endpoint has automatically recovered from the condition that caused the STALL and the endpoint must operate normally.

8.5.3.1 Reporting Status Results

The Status stage reports to the host the outcome of the previous Setup and Data stages of the transfer. Three possible results may be returned:

- The command sequence completed successfully.
- The command sequence failed to complete.
- The function is still busy completing the command.

Status reporting is always in the function-to-host direction. Table 8-7 summarizes the type of responses required for each. Control write transfers return status information in the data phase of the Status stage transaction. Control read transfers return status information in the handshake phase of a Status stage transaction, after the host has issued a zero-length data packet during the previous data phase.

Table 8-7. Status Stage Responses

Status Response	Control Write Transfer (sent during data phase)	Control Read Transfer (sent during handshake phase)
Function completes	Zero-length data packet	ACK handshake
Function has an error	STALL handshake	STALL handshake
Function is busy	NAK handshake	NAK handshake

For control reads, the host must send either an OUT token or PING special token (for a device operating at high-speed) to the control pipe to initiate the Status stage. The host may only send a zero-length data packet in this phase but the function may accept any length packet as a valid status inquiry. The pipe's handshake response to this data packet indicates the current status. NAK indicates that the function is still processing the command and that the host should continue the Status stage. ACK indicates that the function has completed the command and is ready to accept a new command. STALL indicates that the function has an error that prevents it from completing the command.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

For control writes, the host sends an IN token to the control pipe to initiate the Status stage. The function responds with either a handshake or a zero-length data packet to indicate its current status. NAK indicates that the function is still processing the command and that the host should continue the Status stage; return of a zero-length packet indicates normal completion of the command; and STALL indicates that the function cannot complete the command. The function expects the host to respond to the data packet in the Status stage with ACK. If the function does not receive ACK, it remains in the Status stage of the command and will continue to return the zero-length data packet for as long as the host continues to send IN tokens.

If during a Data stage a command pipe is sent more data or is requested to return more data than was indicated in the Setup stage (see Section 8.5.3.2), it should return STALL. If a control pipe returns STALL during the Data stage, there will be no Status stage for that control transfer.

8.5.3.2 Variable-length Data Stage

A control pipe may have a variable-length data phase in which the host requests more data than is contained in the specified data structure. When all of the data structure is returned to the host, the function should indicate that the Data stage is ended by returning a packet that is shorter than the *MaxPacketSize* for the pipe. If the data structure is an exact multiple of *wMaxPacketSize* for the pipe, the function will return a zero-length packet to indicate the end of the Data stage.

8.5.3.3 Error Handling on the Last Data Transaction

If the ACK handshake on an IN transaction is corrupted, the function and the host will temporarily disagree on whether the transaction was successful. If the transaction is followed by another IN, the toggle retry mechanism will detect the mismatch and recover from the error. If the ACK was on the last IN of a Data stage, the toggle retry mechanism cannot be used and an alternative scheme must be used.

The host that successfully received the data of the last IN will send ACK. Later, the host will issue an OUT token to start the Status stage of the transfer. If the function did not receive the ACK that ended the Data stage, the function will interpret the start of the Status stage as verification that the host successfully received the data. Control writes do not have this ambiguity. If an ACK handshake on an OUT gets corrupted, the host does not advance to the Status stage and retries the last data instead. A detailed analysis of retry policy is presented in Section 8.6.4.

8.5.3.4 STALL Handshakes Returned by Control Pipes

Control pipes have the unique ability to return a STALL handshake due to function problems in control transfers. If the device is unable to complete a command, it returns a STALL in the Data and/or Status stages of the control transfer. Unlike the case of a functional stall, protocol stall does not indicate an error with the device. The protocol STALL condition lasts until the receipt of the next SETUP transaction, and the function will return STALL in response to any IN or OUT transaction on the pipe until the SETUP transaction is received. In general, protocol stall indicates that the request or its parameters are not understood by the device and thus provides a mechanism for extending USB requests.

A control pipe may also support functional stall as well, but this is not recommended. This is a degenerative case, because a functional stall on a control pipe indicates that it has lost the ability to communicate with the host. If the control pipe does support functional stall, then it must possess a *Halt* feature, which can be set or cleared by the host. Chapter 9 details how to treat the special case of a *Halt* feature on a control pipe. A well-designed device will associate all of its functions and *Halt* features with non-control endpoints. The control pipes should be reserved for servicing USB requests.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8.5.4 Interrupt Transactions

Interrupt transactions may consist of IN or OUT transfers. Upon receipt of an IN token, a function may return data, NAK, or STALL. If the endpoint has no new interrupt information to return (i.e., no interrupt is pending), the function returns a NAK handshake during the data phase. If the *Halt* feature is set for the interrupt endpoint, the function will return a STALL handshake. If an interrupt is pending, the function returns the interrupt information as a data packet. The host, in response to receipt of the data packet, issues either an ACK handshake if data was received error-free or returns no handshake if the data packet was received corrupted. Figure 8-38 shows the interrupt transaction format.

Section 5.9.1 contains additional information about high-speed, high-bandwidth interrupt endpoints. Such endpoints use multiple transactions in a microframe as defined in that section. Each transaction for a high-bandwidth endpoint follows the transaction format shown in Figure 8-38.

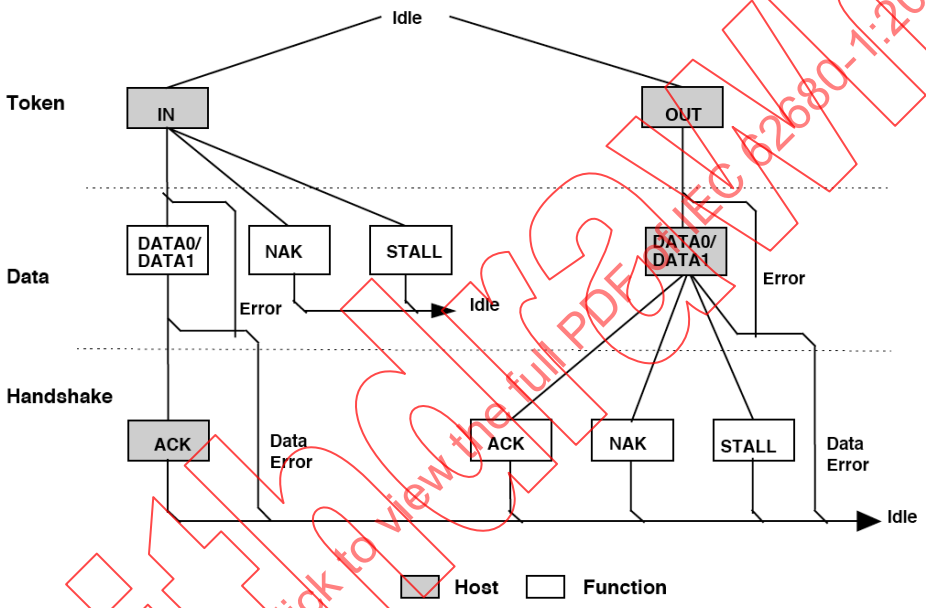


Figure 8-38. Interrupt Transaction Format

When an endpoint is using the interrupt transfer mechanism for actual interrupt data, the data toggle protocol must be followed. This allows the function to know that the data has been received by the host and the event condition may be cleared. This "guaranteed" delivery of events allows the function to only send the interrupt information until it has been received by the host rather than having to send the interrupt data every time the function is polled and until the USB System Software clears the interrupt condition. When used in the toggle mode, an interrupt endpoint is initialized to the DATA0 PID by any configuration event on the endpoint and behaves the same as the bulk transactions shown in Figure 8-35.

8.5.5 Isochronous Transactions

Isochronous transactions have a token and data phase, but no handshake phase, as shown in Figure 8-39. The host issues either an IN or an OUT token followed by the data phase in which the endpoint (for INs) or the host (for OUTs) transmits data. Isochronous transactions do not support a handshake phase or retry capability.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

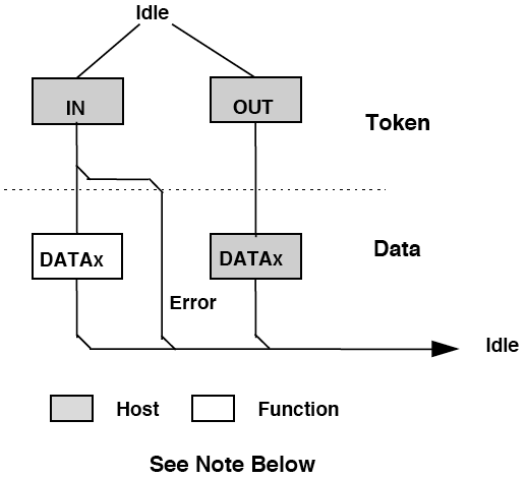


Figure 8-39. Isochronous Transaction Format

Note: A full-speed device or Host Controller should be able to accept either DATA0 or DATA1 PIDs in data packets. A full-speed device or Host Controller should only send DATA0 PIDs in data packets. A high-speed Host Controller must be able to accept and send DATA0, DATA1, DATA2, or MDATA PIDs in data packets. A high-speed device with at most 1 transaction per microframe must only send DATA0 PIDs in data packets. A high-speed device with high-bandwith endpoints (e.g., one that has more than 1 transaction per microframe) must be able to accept and/or send DATA0, DATA1, DATA2, or MDATA PIDs in data packets.

Full-speed isochronous transactions do not support toggle sequencing. High-speed isochronous transactions with a single transaction per microframe do not support toggle sequencing. High bandwidth, high-speed isochronous transactions support data PID sequencing (see Section 5.9.1 for more details).

Figure 8-40 and Figure 8-41 show the host and device state machines respectively for isochronous OUT transactions. Figure 8-42 and Figure 8-43 show the host and device state machines respectively for isochronous IN transactions.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

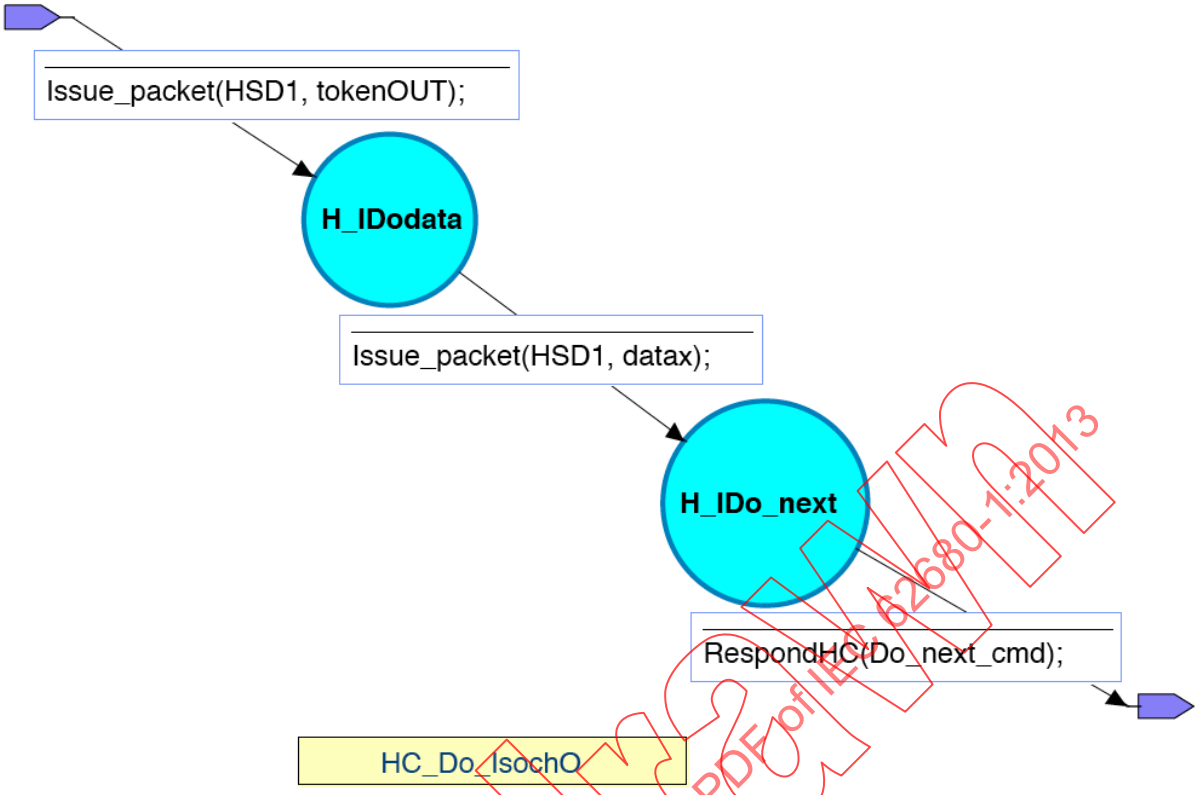


Figure 8-40. Isochronous OUT Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

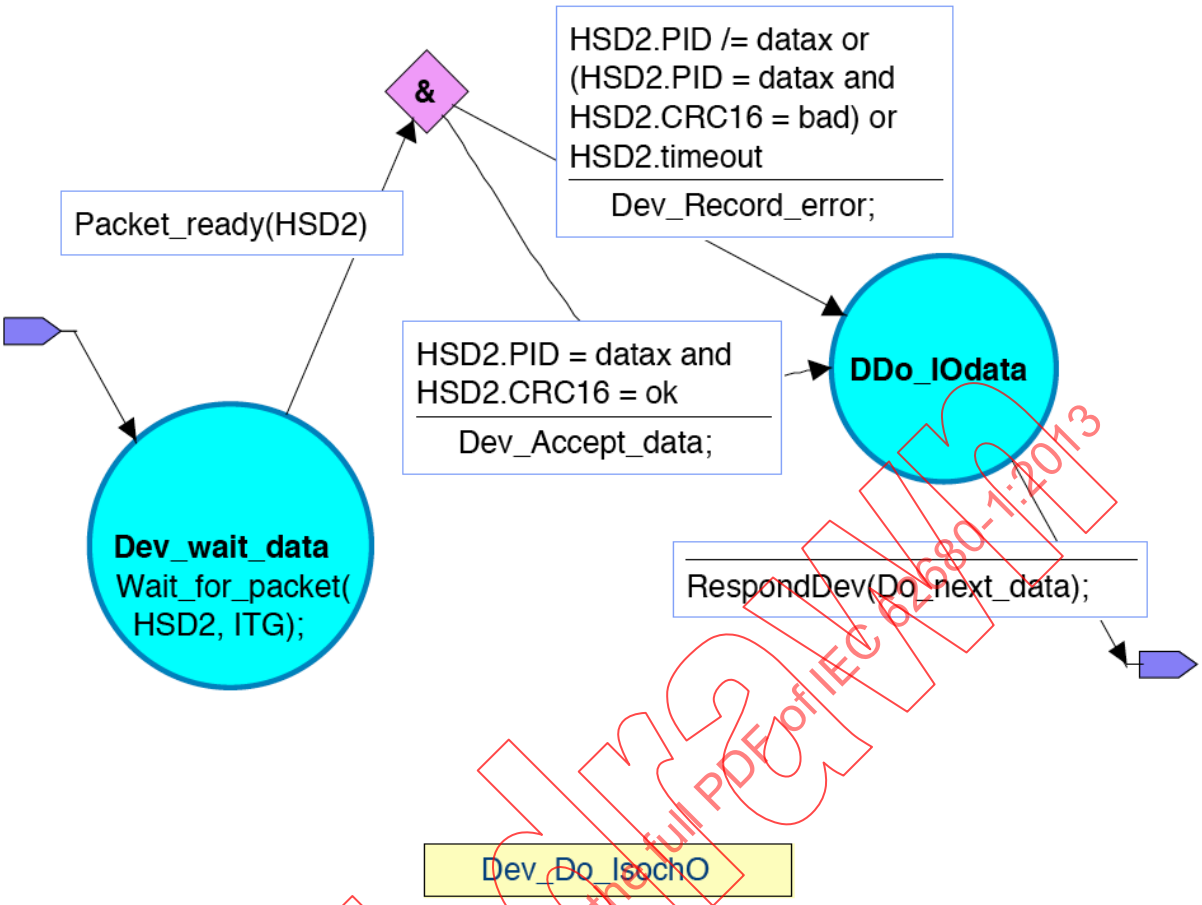


Figure 8-41. Isochronous OUT Transaction Device State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

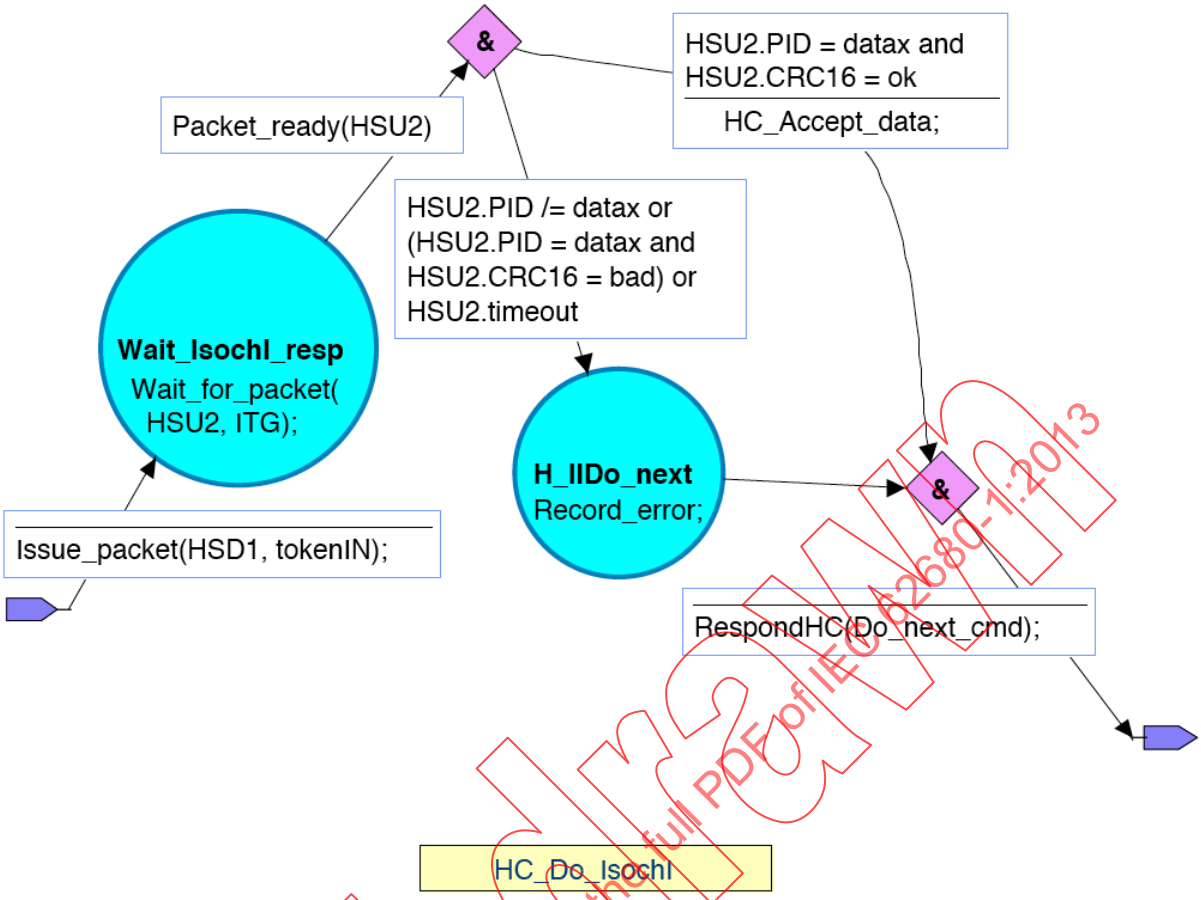


Figure 8-42. Isochronous IN Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

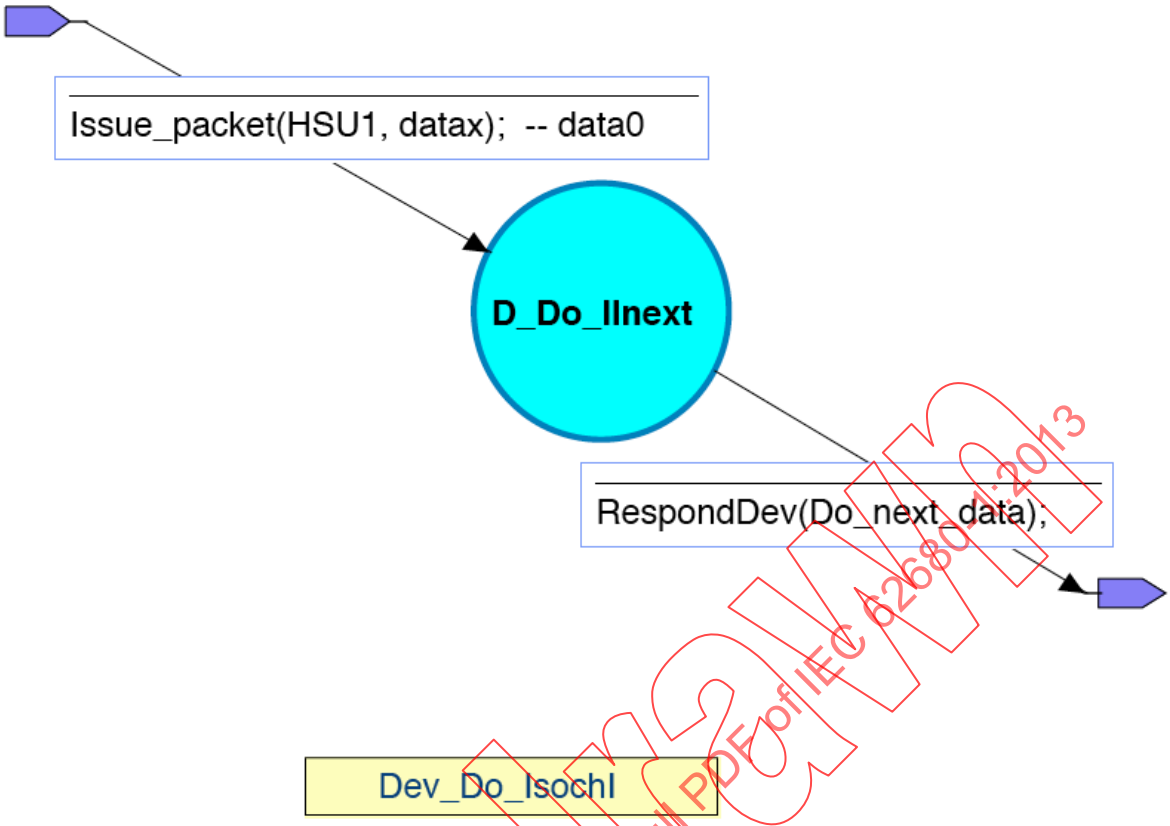


Figure 8-43. Isochronous IN Transaction Device State Machine

8.6 Data Toggle Synchronization and Retry

The USB provides a mechanism to guarantee data sequence synchronization between data transmitter and receiver across multiple transactions. This mechanism provides a means of guaranteeing that the handshake phase of a transaction was interpreted correctly by both the transmitter and receiver. Synchronization is achieved via use of the DATA0 and DATA1 PIDs and separate data toggle sequence bits for the data transmitter and receiver. Receiver sequence bits toggle only when the receiver is able to accept data and receives an error-free data packet with the correct data PID. Transmitter sequence bits toggle only when the data transmitter receives a valid ACK handshake. The data transmitter and receiver must have their sequence bits synchronized at the start of a transaction. The synchronization mechanism used varies with the transaction type. Data toggle synchronization is not supported for isochronous transfers.

The state machines contained in this chapter and in Chapter 11 describe data toggle synchronization in a more compact form. Instead of explicitly identifying DATA0 and DATA1, it uses a value “DATAx” to represent either/both DATA0/DATA1 PIDs. In some cases where the specific data PID is important, another variable labeled “x” is used that has the value 0 for DATA0 and 1 for DATA1.

High-speed, high-bandwidth isochronous and interrupt endpoints support a similar but different data synchronization technique called data PID sequencing. That technique is used instead of data toggle synchronization. Section 5.9.1 defines data PID sequencing.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8.6.1 Initialization via SETUP Token

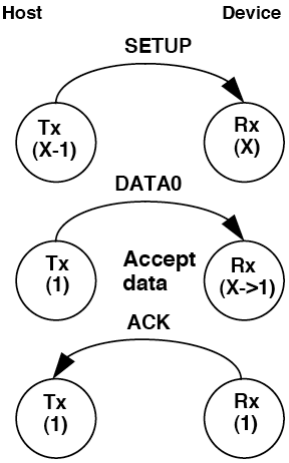


Figure 8-44. SETUP Initialization

Control transfers use the SETUP token for initializing host and function sequence bits. Figure 8-44 shows the host issuing a SETUP packet to a function followed by an OUT transaction. The numbers in the circles represent the transmitter and receiver sequence bits. The function must accept the data and return ACK. When the function accepts the transaction, it must set its sequence bit so that both the host's and function's sequence bits are equal to one at the end of the SETUP transaction.

8.6.2 Successful Data Transactions

Figure 8-45 shows the case where two successful transactions have occurred. For the data transmitter, this means that it toggles its sequence bit upon receipt of ACK. The receiver toggles its sequence bit only if it receives a valid data packet and the packet's data PID matches the current value of its sequence bit. The transmitter only toggles its sequence bit after it receives an ACK to a data packet.

During each transaction, the receiver compares the transmitter sequence bit (encoded in the data packet PID as either DATA0 or DATA1) with its receiver sequence bit. If data cannot be accepted, the receiver must issue NAK and the sequence bits of both the transmitter and receiver remain unchanged. If data can be accepted and the receiver's sequence bit matches the PID sequence bit, then data is accepted and the sequence bit is toggled. Two-phase transactions in which there is no data packet leave the transmitter and receiver sequence bits unchanged.

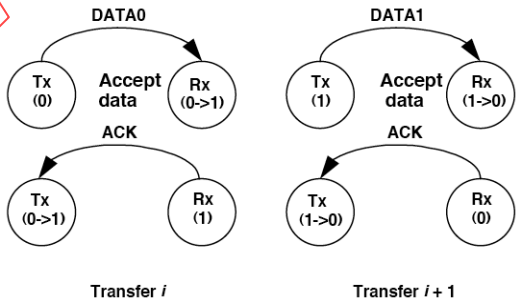


Figure 8-45. Consecutive Transactions

8.6.3 Data Corrupted or Not Accepted

If data cannot be accepted or the received data packet is corrupted, the receiver will issue a NAK or STALL handshake, or timeout, depending on the circumstances, and the receiver will not toggle its sequence bit. Figure 8-46 shows the case where a transaction is NAKed and

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

then retried. Any non-ACK handshake or timeout will generate similar retry behavior. The transmitter, having not received an ACK handshake, will not toggle its sequence bit. As a result, a failed data packet transaction leaves the transmitter's and receiver's sequence bits synchronized and untoggled. The transaction will then be retried and, if successful, will cause both transmitter and receiver sequence bits to toggle.

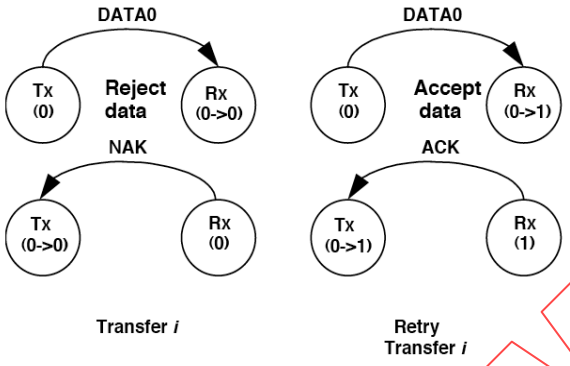


Figure 8-46. NAKed Transaction with Retry

8.6.4 Corrupted ACK Handshake

The transmitter is the last and only agent to know for sure whether a transaction has been successful, due to its receiving an ACK handshake. A lost or corrupted ACK handshake can lead to a temporary loss of synchronization between transmitter and receiver as shown in Figure 8-47. Here the transmitter issues a valid data packet, which is successfully acquired by the receiver; however, the ACK handshake is corrupted.

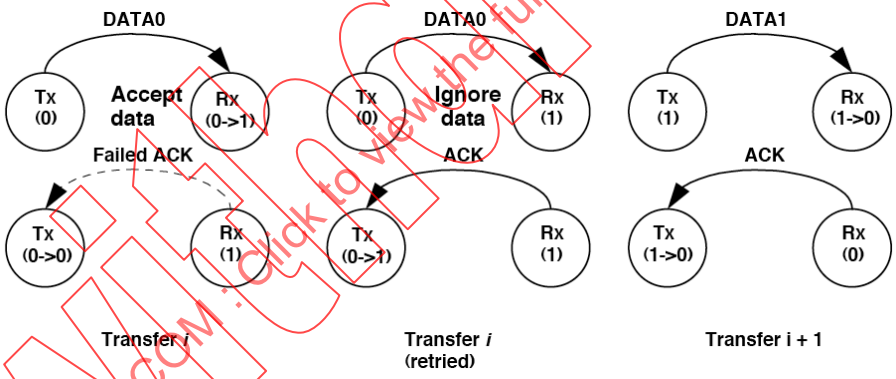


Figure 8-47. Corrupted ACK Handshake with Retry

At the end of transaction *i*, there is a temporary loss of coherency between transmitter and receiver, as evidenced by the mismatch between their respective sequence bits. The receiver has received good data, but the transmitter does not know whether it has successfully sent data. On the next transaction, the transmitter will resend the previous data using the previous DATA0 PID. The receiver's sequence bit and the data PID will not match, so the receiver knows that it has previously accepted this data. Consequently, it discards the incoming data packet and does not toggle its sequence bit. The receiver then issues ACK, which causes the transmitter to regard the retried transaction as successful. Receipt of ACK causes the transmitter to toggle its sequence bit. At the beginning of transaction *i*+1, the sequence bits have toggled and are again synchronized.

The data transmitter must guarantee that any retried data packet is identical (same length and content) as that sent in the original transaction. If the data transmitter is unable, because of problems such as a buffer underrun condition, to transmit the identical amount of data as was in the original data packet, it must abort the transaction by generating a bit stuffing violation for full-/low-speed. An error for high-speed must be forced by taking the currently calculated

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

CRC and complementing it before transmitting it. This causes a detectable error at the receiver and guarantees that a partial packet will not be interpreted as a good packet. The transmitter should not try to force an error at the receiver by sending a constant known bad CRC. A combination of a bad packet with a "bad" CRC may be interpreted by the receiver as a good packet.

8.6.5 Low-speed Transactions

The USB supports signaling at three speeds: high-speed signaling at 480 Mb/s, full-speed signaling at 12.0 Mb/s, and low-speed signaling at 1.5 Mb/s. Hubs isolate high-speed signaling from full-/low-speed signaling environments.

Within a full-/low-speed signaling environment, hubs disable downstream bus traffic to all ports to which low-speed devices are attached during full-speed downstream signaling. This is required both for EMI reasons and to prevent any possibility that a low-speed device might misinterpret downstream a full-speed packet as being addressed to it.

Figure 8-48 shows an IN low-speed transaction in which the host (or TT) issues a token and handshake and receives a data packet.

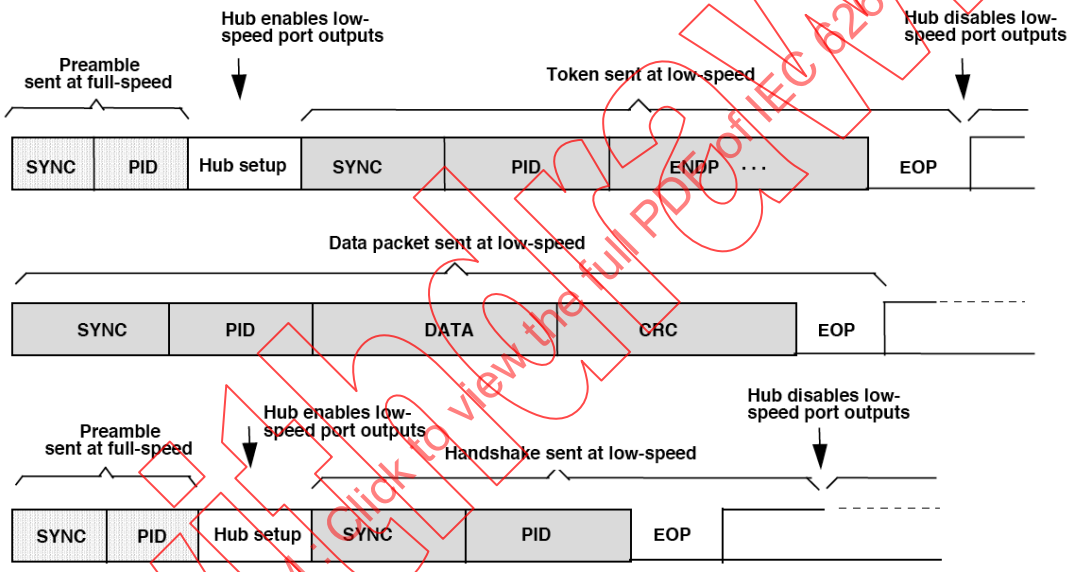


Figure 8-48. Low-speed Transaction

All downstream packets transmitted to low-speed devices within a full-/low-speed signaling environment require a preamble. Preambles are never used in a high-speed signaling environment. The preamble consists of a SYNC followed by a PRE PID, both sent at full-speed. Hubs must comprehend the PRE PID; all other USB devices may ignore it and treat it as undefined. At the end of the preamble PID, the host (or TT) drives the bus to the Idle state for at least one full-speed bit time. This Idle period on the bus is termed the hub setup interval and lasts for at least four full-speed bit times. During this hub setup interval, hubs must drive their full-speed and low-speed ports to their respective Idle states. Hubs must be ready to repeat low-speed signaling on low-speed ports before the end of the hub setup interval. Low-speed connectivity rules are summarized below:

1. Low-speed devices are identified during the connection process, and the hub ports to which they are connected are identified as low-speed.
2. All downstream low-speed packets must be prefaced with a preamble (sent at full-speed), which turns on the output buffers on low-speed hub ports.
3. Low-speed hub port output buffers are turned off upon receipt of EOP and are not turned on again until a preamble PID is detected.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

4. Upstream connectivity is not affected by whether a hub port is full- or low-speed.

Low-speed signaling begins with the host (or TT) issuing SYNC at low-speed, followed by the remainder of the packet. The end of the packet is identified by an End-of-Packet (EOP), at which time all hubs tear down connectivity and disable any ports to which low-speed devices are connected. Hubs do not switch ports for upstream signaling; low-speed ports remain enabled in the upstream direction for both low-speed and full-speed signaling.

Low-speed and full-speed transactions maintain a high degree of protocol commonality. However, low-speed signaling does have certain limitations which include:

- Data payload is limited to eight bytes, maximum.
- Only interrupt and control types of transfers are supported.
- The SOF packet is not received by low-speed devices.

8.7 Error Detection and Recovery

The USB permits reliable end-to-end communication in the presence of errors on the physical signaling layer. This includes the ability to reliably detect the vast majority of possible errors and to recover from errors on a transaction-type basis. Control transactions, for example, require a high degree of data reliability; they support end-to-end data integrity using error detection and retry. Isochronous transactions, by virtue of their bandwidth and latency requirements, do not permit retries and must tolerate a higher incidence of uncorrected errors.

8.7.1 Packet Error Categories

The USB employs three error detection mechanisms: bit stuff violations, PID check bits, and CRCs. Bit stuff violations are defined in Section 7.1.9. PID errors are defined in Section 8.3.1. CRC errors are defined in Section 8.3.5.

With the exception of the SOF token, any packet that is received corrupted causes the receiver to ignore it and discard any data or other field information that came with the packet. Table 8-8 lists error detection mechanisms, the types of packets to which they apply, and the appropriate packet receiver response.

Table 8-8. Packet Error Types

Field	Error	Action
PID	PID Check, Bit Stuff	Ignore packet
Address	Bit Stuff, Address CRC	Ignore token
Frame Number	Bit Stuff, Frame Number CRC	Ignore Frame Number field
Data	Bit Stuff, Data CRC	Discard data

8.7.2 Bus Turn-around Timing

Neither the device nor the host will send an indication that a received packet had an error. This absence of positive acknowledgement is considered to be the indication that there was an error. As a consequence of this method of error reporting, the host and USB function need to keep track of how much time has elapsed from when the transmitter completes sending a packet until it begins to receive a response packet. This time is referred to as the bus turn-around time. Devices and hosts require turn-around timers to measure this time.

For full-/low-speed transactions, the timer starts counting on the SE0-to-'J' transition of the EOP strobe and stops counting when the Idle-to-'K' SOP transition is detected. For high-speed transactions, the timer starts counting when the data lines return to the squelch level and stops counting when the data lines leave the squelch level.

The device bus turn-around time is defined by the worst case round trip delay plus the maximum device response delay (refer to Sections 7.1.18 and 7.1.19 for specific bus turn-

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

around times). If a response is not received within this worst case timeout, then the transmitter considers that the packet transmission has failed.

Timeout is used and interpreted as a transaction error condition for many transfer types. If the host wishes to indicate an error condition for a transaction via a timeout, it must wait the full bus turn-around time before issuing the next token to ensure that all downstream devices have timed out.

As shown in Figure 8-49, the device uses its bus turn-around timer between token and data or data and handshake phases. The host uses its timer between data and handshake or token and data phases.

If the host receives a corrupted data packet, it may require additional wait time before sending out the next token. This additional wait interval guarantees that the host properly handles false EOPs.

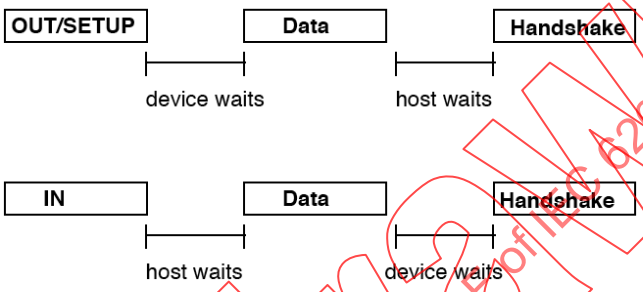


Figure 8-49. Bus Turn-around Timer Usage

8.7.3 False EOPs

False EOPs must be handled in a manner which guarantees that the packet currently in progress completes before the host or any other device attempts to transmit a new packet. If such an event were to occur, it would constitute a bus collision and have the ability to corrupt up to two consecutive transactions. Detection of false EOP relies upon the fact that a packet into which a false EOP has been inserted will appear as a truncated packet with a CRC failure. (The last 16 bits of the data packet will have a very low probability of appearing to be a correct CRC.)

The host and devices handle false EOP situations differently. When a device receives a corrupted data packet, it issues no response and waits for the host to send the next token. This scheme guarantees that the device will not attempt to return a handshake while the host may still be transmitting a data packet. If a false EOP has occurred, the host data packet will eventually end, and the device will be able to detect the next token. If a device issues a data packet that gets corrupted with a false EOP, the host will ignore the packet and not issue the handshake. The device, expecting to see a handshake from the host, will timeout the transaction.

If the host receives a corrupted full-/low-speed data packet, it assumes that a false EOP may have occurred and waits for 16 bit times to see if there is any subsequent upstream traffic. If no bus transitions are detected within the 16 bit interval and the bus remains in the Idle state, the host may issue the next token.

Otherwise, the host waits for the device to finish sending the remainder of its full-/low-speed packet. Waiting 16 bit times guarantees two conditions:

- The first condition is to make sure that the device has finished sending its packet. This is guaranteed by a timeout interval (with no bus transitions) greater than the worst case six-bit time bit stuff interval.
- The second condition is that the transmitting device's bus turn-around timer must be guaranteed to expire.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Note that the timeout interval is transaction speed sensitive. For full-speed transactions, the host must wait full-speed bit times; for low-speed transactions, it must wait low-speed bit times.

If the host receives a corrupted high-speed data packet, it ignores any data until the data lines return to the squelch level before issuing the next token. For high-speed transactions, the host does not need to wait additional time (beyond the normal inter-transaction gap time) after the data lines return to the squelch level.

If the host receives a data packet with a valid CRC, it assumes that the packet is complete and requires no additional delay (beyond normal inter-transaction gap time) in issuing the next token.

8.7.4 Babble and Loss of Activity Recovery

The USB must be able to detect and recover from conditions which leave it waiting indefinitely for a full-/low-speed EOP or which leave the bus in something other than the Idle state at the end of a (micro)frame.

- Full-/low-speed loss of activity (LOA) is characterized by an SOP followed by lack of bus activity (bus remains driven to a ‘J’ or ‘K’) and no EOP at the end of a frame.
- Full-/low-speed babble is characterized by an SOP followed by the presence of bus activity past the end of a frame.
- High-speed babble/LOA is characterized by the data lines being at an unsquelched level at the end of a microframe.

LOA and babble have the potential to either deadlock the bus or delay the beginning of the next (micro)frame. Neither condition is acceptable, and both must be prevented from occurring. As the USB component responsible for controlling connectivity, hubs are responsible for babble/LOA detection and recovery. All USB devices that fail to complete their transmission at the end of a (micro)frame are prevented from transmitting past a (micro)frame’s end by having the nearest hub disable the port to which the offending device is attached. Details of the hub babble/LOA recovery mechanism appear in Section 11.2.5.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9 Chapter 9
USB Device Framework

A USB device may be divided into three layers:

- The bottom layer is a bus interface that transmits and receives packets.
- The middle layer handles routing data between the bus interface and various endpoints on the device. An endpoint is the ultimate consumer or provider of data. It may be thought of as a source or sink for data.
- The top layer is the functionality provided by the serial bus device, for instance, a mouse or ISDN interface.

This chapter describes the common attributes and operations of the middle layer of a USB device. These attributes and operations are used by the function-specific portions of the device to communicate through the bus interface and ultimately with the host.

9.1 USB Device States

A USB device has several possible states. Some of these states are visible to the USB and the host, while others are internal to the USB device. This section describes those states.

9.1.1 Visible Device States

This section describes USB device states that are externally visible (see Figure 9-1). Table 9-1 summarizes the visible device states.

Note: USB devices perform a reset operation in response to reset signaling on the upstream facing port. When reset signaling has completed, the USB device is reset.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

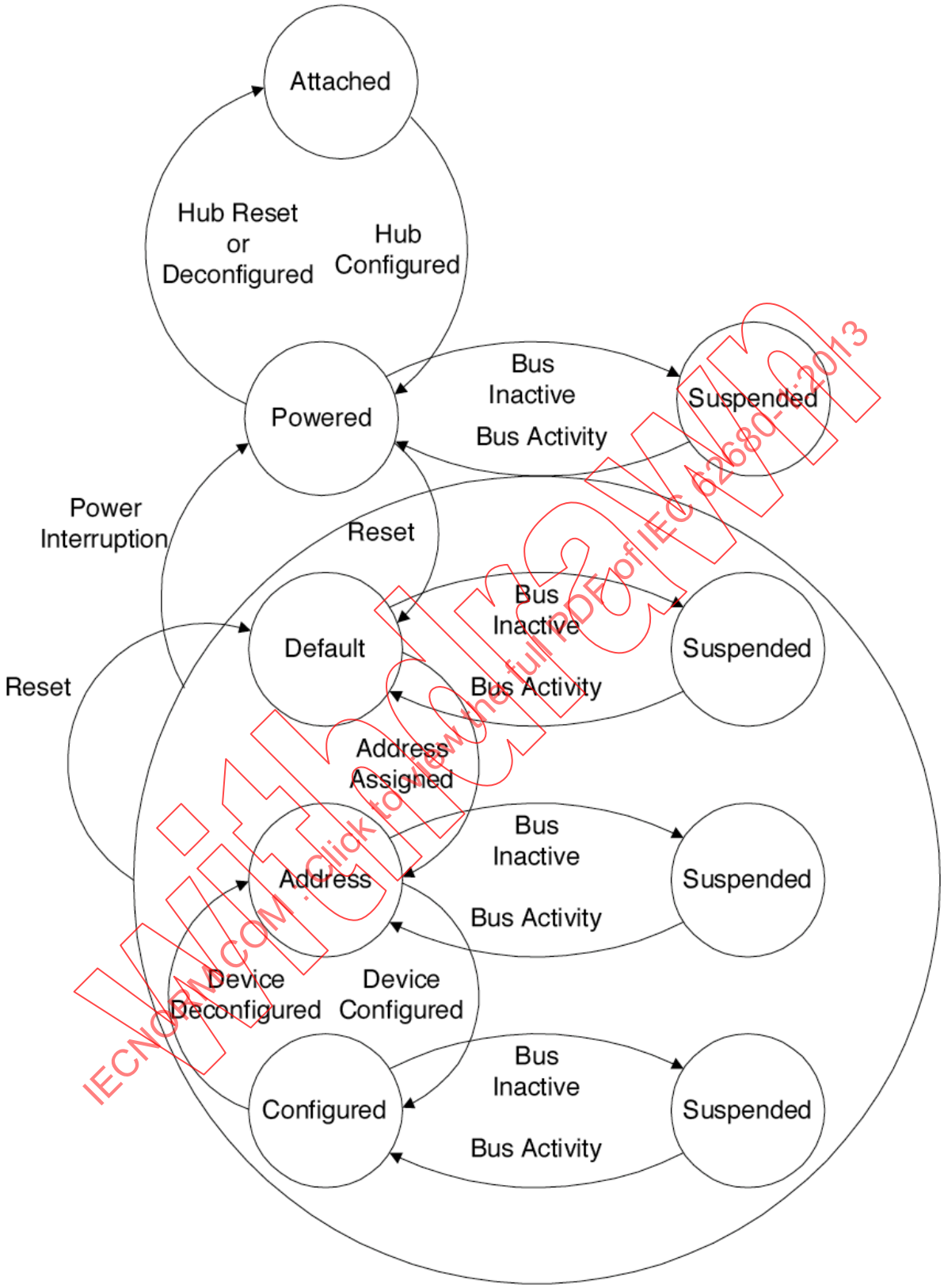


Figure 9-1. Device State Diagram

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 9-1. Visible Device States

Attached	Powered	Default	Address	Configured	Suspended	State
No	--	--	--	--	--	Device is not attached to the USB. Other attributes are not significant.
Yes	No	--	--	--	--	Device is attached to the USB, but is not powered. Other attributes are not significant.
Yes	Yes	No	--	--	--	Device is attached to the USB and powered, but has not been reset.
Yes	Yes	Yes	No	--	--	Device is attached to the USB and powered and has been reset, but has not been assigned a unique address. Device responds at the default address.
Yes	Yes	Yes	Yes	No	--	Device is attached to the USB, powered, has been reset, and a unique device address has been assigned. Device is not configured.
Yes	Yes	Yes	Yes	Yes	No	Device is attached to the USB, powered, has been reset, has a unique address, is configured, and is not suspended. The host may now use the function provided by the device.
Yes	Yes	--	--	--	Yes	Device is, at minimum, attached to the USB and is powered and has not seen bus activity for 3 ms. It may also have a unique address and be configured for use. However, because the device is suspended, the host may not use the device's function.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

9.1.1.1 Attached

A USB device may be attached or detached from the USB. The state of a USB device when it is detached from the USB is not defined by this specification. This specification only addresses required operations and attributes once the device is attached.

9.1.1.2 Powered

USB devices may obtain power from an external source and/or from the USB through the hub to which they are attached. Externally powered USB devices are termed self-powered. Although self-powered devices may already be powered before they are attached to the USB, they are not considered to be in the Powered state until they are attached to the USB and VBUS is applied to the device.

A device may support both self-powered and bus-powered configurations. Some device configurations support either power source. Other device configurations may be available only if the device is self-powered. Devices report their power source capability through the configuration descriptor. The current power source is reported as part of a device's status. Devices may change their power source at any time, e.g., from self- to bus-powered. If a

configuration is capable of supporting both power modes, the power maximum reported for that configuration is the maximum the device will draw from VBUS in either mode. The device must observe this maximum, regardless of its mode. If a configuration supports only one power mode and the power source of the device changes, the device will lose its current configuration and address and return to the Powered state. If a device is self-powered and its current configuration requires more than 100 mA, then if the device switches to being bus-powered, it must return to the Address state. Self-powered hubs that use VBUS to power the Hub Controller are allowed to remain in the Configured state if local power is lost. Refer to Section 11.13 for details.

A hub port must be powered in order to detect port status changes, including attach and detach. Bus-powered hubs do not provide any downstream power until they are configured, at which point they will provide power as allowed by their configuration and power source. A USB device must be able to be addressed within a specified time period from when power is initially applied (refer to Chapter 7). After an attachment to a port has been detected, the host may enable the port, which will also reset the device attached to the port.

9.1.1.3 Default

After the device has been powered, it must not respond to any bus transactions until it has received a reset from the bus. After receiving a reset, the device is then addressable at the default address.

When the reset process is complete, the USB device is operating at the correct speed (i.e., low-/full-/high-speed). The speed selection for low- and full-speed is determined by the device termination resistors. A device that is capable of high-speed operation determines whether it will operate at high-speed as a part of the reset process (see Chapter 7 for more details).

A device capable of high-speed operation must reset successfully at full-speed when in an electrical environment that is operating at full-speed. After the device is successfully reset, the device must also respond successfully to device and configuration descriptor requests and return appropriate information. The device may or may not be able to support its intended functionality when operating at full-speed.

9.1.1.4 Address

All USB devices use the default address when initially powered or after the device has been reset. Each USB device is assigned a unique address by the host after attachment or after reset. A USB device maintains its assigned address while suspended.

A USB device responds to requests on its default pipe whether the device is currently assigned a unique address or is using the default address.

9.1.1.5 Configured

Before a USB device's function may be used, the device must be configured. From the device's perspective, configuration involves correctly processing a SetConfiguration() request with a non-zero configuration value. Configuring a device or changing an alternate setting causes all of the status and configuration values associated with endpoints in the affected interfaces to be set to their default values. This includes setting the data toggle of any endpoint using data toggles to the value data0.

9.1.1.6 Suspended

In order to conserve power, USB devices automatically enter the Suspended state when the device has observed no bus traffic for a specified period (refer to Chapter 7). When suspended, the USB device maintains any internal status, including its address and configuration.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

All devices must suspend if bus activity has not been observed for the length of time specified in Chapter 7. Attached devices must be prepared to suspend at any time they are powered, whether they have been assigned a non-default address or are configured. Bus activity may cease due to the host entering a suspend mode of its own. In addition, a USB device shall also enter the Suspended state when the hub port it is attached to is disabled. This is referred to as selective suspend.

A USB device exits suspend mode when there is bus activity. A USB device may also request the host to exit suspend mode or selective suspend by using electrical signaling to indicate remote wakeup. The ability of a device to signal remote wakeup is optional. If a USB device is capable of remote wakeup signaling, the device must support the ability of the host to enable and disable this capability. When the device is reset, remote wakeup signaling must be disabled.

9.1.2 Bus Enumeration

When a USB device is attached to or removed from the USB, the host uses a process known as bus enumeration to identify and manage the device state changes necessary. When a USB device is attached to a powered port, the following actions are taken:

1. The hub to which the USB device is now attached informs the host of the event via a reply on its status change pipe (refer to Section 11.12.3 for more information). At this point, the USB device is in the Powered state and the port to which it is attached is disabled.
2. The host determines the exact nature of the change by querying the hub.
3. Now that the host knows the port to which the new device has been attached, the host then waits for at least 100 ms to allow completion of an insertion process and for power at the device to become stable. The host then issues a port enable and reset command to that port. Refer to Section 7.1.7.5 for sequence of events and timings of connection through device reset.
4. The hub performs the required reset processing for that port (see Section 11.5.1.5). When the reset signal is released, the port has been enabled. The USB device is now in the Default state and can draw no more than 100 mA from VBUS. All of its registers and state have been reset and it answers to the default address.
5. The host assigns a unique address to the USB device, moving the device to the Address state.
6. Before the USB device receives a unique address, its Default Control Pipe is still accessible via the default address. The host reads the device descriptor to determine what actual maximum data payload size this USB device's default pipe can use.
7. The host reads the configuration information from the device by reading each configuration zero to *n*-1, where *n* is the number of configurations. This process may take several milliseconds to complete.
8. Based on the configuration information and how the USB device will be used, the host assigns a configuration value to the device. The device is now in the Configured state and all of the endpoints in this configuration have taken on their described characteristics. The USB device may now draw the amount of VBUS power described in its descriptor for the selected configuration. From the device's point of view, it is now ready for use.

When the USB device is removed, the hub again sends a notification to the host. Detaching a device disables the port to which it had been attached. Upon receiving the detach notification, the host will update its local topological information.

9.2 Generic USB Device Operations

All USB devices support a common set of operations. This section describes those operations.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9.2.1 Dynamic Attachment and Removal

USB devices may be attached and removed at any time. The hub that provides the attachment point or port is responsible for reporting any change in the state of the port.

The host enables the hub port where the device is attached upon detection of an attachment, which also has the effect of resetting the device. A reset USB device has the following characteristics:

- Responds to the default USB address
- Is not configured
- Is not initially suspended

When a device is removed from a hub port, the hub disables the port where the device was attached and notifies the host of the removal.

9.2.2 Address Assignment

When a USB device is attached, the host is responsible for assigning a unique address to the device. This is done after the device has been reset by the host, and the hub port where the device is attached has been enabled.

9.2.3 Configuration

A USB device must be configured before its function(s) may be used. The host is responsible for configuring a USB device. The host typically requests configuration information from the USB device to determine the device's capabilities.

As part of the configuration process, the host sets the device configuration and, where necessary, selects the appropriate alternate settings for the interfaces.

Within a single configuration, a device may support multiple interfaces. An interface is a related set of endpoints that present a single feature or function of the device to the host. The protocol used to communicate with this related set of endpoints and the purpose of each endpoint within the interface may be specified as part of a device class or vendor-specific definition.

In addition, an interface within a configuration may have alternate settings that redefine the number or characteristics of the associated endpoints. If this is the case, the device must support the GetInterface() request to report the current alternate setting for the specified interface and SetInterface() request to select the alternate setting for the specified interface.

Within each configuration, each interface descriptor contains fields that identify the interface number and the alternate setting. Interfaces are numbered from zero to one less than the number of concurrent interfaces supported by the configuration. Alternate settings range from zero to one less than the number of alternate settings for a specific interface. The default setting when a device is initially configured is alternate setting zero.

In support of adaptive device drivers that are capable of managing a related group of USB devices, the device and interface descriptors contain *Class*, *SubClass*, and *Protocol* fields. These fields are used to identify the function(s) provided by a USB device and the protocols used to communicate with the function(s) on the device. A class code is assigned to a group of related devices that has been characterized as a part of a USB Class Specification. A class of devices may be further subdivided into subclasses, and, within a class or subclass, a protocol code may define how the Host Software communicates with the device.

Note: The assignment of class, subclass, and protocol codes must be coordinated but is beyond the scope of this specification.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9.2.4 Data Transfer

Data may be transferred between a USB device endpoint and the host in one of four ways. Refer to Chapter 5 for the definition of the four types of transfers. An endpoint number may be used for different types of data transfers in different alternate settings. However, once an alternate setting is selected (including the default setting of an interface), a USB device endpoint uses only one data transfer method until a different alternate setting is selected.

9.2.5 Power Management

Power management on USB devices involves the issues described in the following sections.

9.2.5.1 Power Budgeting

USB bus power is a limited resource. During device enumeration, a host evaluates a device's power requirements. If the power requirements of a particular configuration exceed the power available to the device, Host Software shall not select that configuration.

USB devices shall limit the power they consume from VBUS to one unit load or less until configured. Suspended devices, whether configured or not, shall limit their bus power consumption as defined in Chapter 7. Depending on the power capabilities of the port to which the device is attached, a USB device may be able to draw up to five unit loads from VBUS after configuration.

9.2.5.2 Remote Wakeup

Remote wakeup allows a suspended USB device to signal a host that may also be suspended. This notifies the host that it should resume from its suspended mode, if necessary, and service the external event that triggered the suspended USB device to signal the host. A USB device reports its ability to support remote wakeup in a configuration descriptor. If a device supports remote wakeup, it must also allow the capability to be enabled and disabled using the standard USB requests.

Remote wakeup is accomplished using electrical signaling described in Section 7.1.7.7.

9.2.6 Request Processing

With the exception of SetAddress() requests (see Section 9.4.6), a device may begin processing of a request as soon as the device returns the ACK following the Setup. The device is expected to "complete" processing of the request before it allows the Status stage to complete successfully. Some requests initiate operations that take many milliseconds to complete. For requests such as this, the device class is required to define a method other than Status stage completion to indicate that the operation has completed. For example, a reset on a hub port takes at least 10 ms to complete. The SetPortFeature(PORT_RESET) (see Chapter 11) request "completes" when the reset on the port is initiated. Completion of the reset operation is signaled when the port's status change is set to indicate that the port is now enabled. This technique prevents the host from having to constantly poll for a completion when it is known that the request will take a relatively long period of time.

9.2.6.1 Request Processing Timing

All devices are expected to handle requests in a timely manner. USB sets an upper limit of 5 seconds as the upper limit for any command to be processed. This limit is not applicable in all instances. The limitations are described in the following sections. It should be noted that the limitations given below are intended to encompass a wide range of implementations. If all devices in a USB system used the maximum allotted time for request processing, the user experience would suffer. For this reason, implementations should strive to complete requests in times that are as short as possible.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9.2.6.2 Reset/Resume Recovery Time

After a port is reset or resumed, the USB System Software is expected to provide a “recovery” interval of 10 ms before the device attached to the port is expected to respond to data transfers. The device may ignore any data transfers during the recovery interval.

After the end of the recovery interval (measured from the end of the reset or the end of the EOP at the end of the resume signaling), the device must accept data transfers at any time.

9.2.6.3 Set Address Processing

After the reset/resume recovery interval, if a device receives a SetAddress() request, the device must be able to complete processing of the request and be able to successfully complete the Status stage of the request within 50 ms. In the case of the SetAddress() request, the Status stage successfully completes when the device sends the zero-length Status packet or when the device sees the ACK in response to the Status stage data packet.

After successful completion of the Status stage, the device is allowed a SetAddress() recovery interval of 2 ms. At the end of this interval, the device must be able to accept Setup packets addressed to the new address. Also, at the end of the recovery interval, the device must not respond to tokens sent to the old address (unless, of course, the old and new address is the same).

9.2.6.4 Standard Device Requests

For standard device requests that require no Data stage, a device must be able to complete the request and be able to successfully complete the Status stage of the request within 50 ms of receipt of the request. This limitation applies to requests to the device, interface, or endpoint.

For standard device requests that require data stage transfer to the host, the device must be able to return the first data packet to the host within 500 ms of receipt of the request. For subsequent data packets, if any, the device must be able to return them within 500 ms of successful completion of the transmission of the previous packet. The device must then be able to successfully complete the status stage within 50 ms after returning the last data packet.

For standard device requests that require a data stage transfer to the device, the 5-second limit applies. This means that the device must be capable of accepting all data packets from the host and successfully completing the Status stage if the host provides the data at the maximum rate at which the device can accept it. Delays between packets introduced by the host add to the time allowed for the device to complete the request.

9.2.6.5 Class-specific Requests

Unless specifically exempted in the class document, all class-specific requests must meet the timing limitations for standard device requests. If a class document provides an exemption, the exemption may only be specified on a request-by-request basis.

A class document may require that a device respond more quickly than is specified in this section. Faster response may be required for standard and class-specific requests.

9.2.6.6 Speed Dependent Descriptors

A device capable of operation at high-speed can operate in either full- or high-speed. The device always knows its operational speed due to having to manage its transceivers correctly as part of reset processing (See Chapter 7 for more details on reset). A device also operates at a single speed after completing the reset sequence. In particular, there is no speed switch during normal operation. However, a high-speed capable device may have configurations that are speed dependent. That is, it may have some configurations that are only possible when operating at high-speed or some that are only possible when operating at full-speed. High-speed capable devices must support reporting their speed dependent configurations.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A high-speed capable device responds with descriptor information that is valid for the current operating speed. For example, when a device is asked for configuration descriptors, it only returns those for the current operating speed (e.g., full speed). However, there must be a way to determine the capabilities for both high- and full-speed operation.

Two descriptors allow a high-speed capable device to report configuration information about the other operating speed. The two descriptors are: the (other_speed) device_qualifier descriptor and the other_speed_configuration descriptor. These two descriptors are retrieved by the host by using the GetDescriptor request with the corresponding descriptor type values.

Note: These descriptors are not retrieved unless the host explicitly issues the corresponding GetDescriptor requests. If these two requests are not issued, the device would simply appear to be a single speed device.

Devices that are high-speed capable must set the version number in the *bcdUSB* field of their descriptors to 0200H. This indicates that such devices support the other_speed requests defined by USB 2.0. A device with descriptor version numbers less than 0200H should cause a Request Error response (see next section) if it receives these other_speed requests. A USB 1.x device (i.e., one with a device descriptor version less than 0200H) should not be issued the other_speed requests.

9.2.7 Request Error

When a request is received by a device that is not defined for the device, is inappropriate for the current setting of the device, or has values that are not compatible with the request, then a Request Error exists. The device deals with the Request Error by returning a STALL PID in response to the next Data stage transaction or in the Status stage of the message. It is preferred that the STALL PID be returned at the next Data stage transaction, as this avoids unnecessary bus activity.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9.3 USB Device Requests

All USB devices respond to requests from the host on the device's Default Control Pipe. These requests are made using control transfers. The request and the request's parameters are sent to the device in the Setup packet. The host is responsible for establishing the values passed in the fields listed in Table 9-2. Every Setup packet has eight bytes.

Table 9-2. Format of Setup Data

Offset	Field	Size	Value	Description
0	*bmRequestType*	1	Bitmap	Characteristics of request: D7: Data transfer direction 0 = Host-to-device 1 = Device-to-host D6...5: Type 0 = Standard 1 = Class 2 = Vendor 3 = Reserved D4...0: Recipient 0 = Device 1 = Interface 2 = Endpoint 3 = Other 4...31 = Reserved
1	*bRequest*	1	Value	Specific request (refer to Table 9-3)
2	*wValue*	2	Value	Word-sized field that varies according to request
4	*wIndex*	2	Index or Offset	Word-sized field that varies according to request; typically used to pass an index or offset
6	*wLength*	2	Count	Number of bytes to transfer if there is a Data stage

9.3.1 bmRequestType

This bitmapped field identifies the characteristics of the specific request. In particular, this field identifies the direction of data transfer in the second phase of the control transfer. The state of the *Direction* bit is ignored if the *wLength* field is zero, signifying there is no Data stage.

The USB Specification defines a series of standard requests that all devices must support. These are enumerated in Table 9-3. In addition, a device class may define additional requests. A device vendor may also define requests supported by the device.

Requests may be directed to the device, an interface on the device, or a specific endpoint on a device. This field also specifies the intended recipient of the request. When an interface or endpoint is specified, the *wIndex* field identifies the interface or endpoint.

9.3.2 bRequest

This field specifies the particular request. The *Type* bits in the *bmRequestType* field modify the meaning of this field. This specification defines values for the *bRequest* field only when the bits are reset to zero, indicating a standard request (refer to Table 9-3).

9.3.3 wValue

The contents of this field vary according to the request. It is used to pass a parameter to the device, specific to the request.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9.3.4 wIndex

The contents of this field vary according to the request. It is used to pass a parameter to the device, specific to the request.

The *wIndex* field is often used in requests to specify an endpoint or an interface. Figure 9-2 shows the format of *wIndex* when it is used to specify an endpoint.

D7	D6	D5	D4	D3	D2	D1	D0
Direction	Reserved (Reset to zero)			Endpoint Number			
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Reserved (Reset to zero)							

Figure 9-2. wIndex Format when Specifying an Endpoint

The *Direction* bit is set to zero to indicate the OUT endpoint with the specified *Endpoint Number* and to one to indicate the IN endpoint. In the case of a control pipe, the request should have the *Direction* bit set to zero but the device may accept either value of the *Direction* bit.

Figure 9-3 shows the format of *wIndex* when it is used to specify an interface.

D7	D6	D5	D4	D3	D2	D1	D0
Interface Number							
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Reserved (Reset to zero)							

Figure 9-3. wIndex Format when Specifying an Interface

9.3.5 wLength

This field specifies the length of the data transferred during the second phase of the control transfer. The direction of data transfer (host-to-device or device-to-host) is indicated by the *Direction* bit of the *bmRequestType* field. If this field is zero, there is no data transfer phase.

On an input request, a device must never return more data than is indicated by the *wLength* value; it may return less. On an output request, *wLength* will always indicate the exact amount of data to be sent by the host. Device behavior is undefined if the host should send more data than is specified in *wLength.*

9.4 Standard Device Requests

This section describes the standard device requests defined for all USB devices. Table 9-3 outlines the standard device requests, while Table 9-4 and Table 9-5 give the standard request codes and descriptor types, respectively.

USB devices must respond to standard device requests, even if the device has not yet been assigned an address or has not been configured.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 9-3. Standard Device Requests

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B 00000001B 00000010B	CLEAR_FEATURE	Feature Selector	Zero Interface Endpoint	Zero	None
10000000B	GET_CONFIGURATION	Zero	Zero	One	Configuration Value
10000000B	GET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor
10000001B	GET_INTERFACE	Zero	Interface	One	Alternate Interface
10000000B 10000001B 10000010B	GET_STATUS	Zero	Zero Interface Endpoint	Two	Device, Interface, or Endpoint Status
00000000B	SET_ADDRESS	Device Address	Zero	Zero	None
00000000B	SET_CONFIGURATION	Configuration Value	Zero	Zero	None
00000000B	SET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor
00000000B 00000001B 00000010B	SET_FEATURE	Feature Selector	Zero Interface Endpoint	Zero	None
00000001B	SET_INTERFACE	Alternate Setting	Interface	Zero	None
10000010B	SYNCH_FRAME	Zero	Endpoint	Two	Frame Number

Table 9-4. Standard Request Codes

bRequest	Value
GET_STATUS	0
CLEAR_FEATURE	1
Reserved for future use	2
SET_FEATURE	3
Reserved for future use	4
SET_ADDRESS	5
GET_DESCRIPTOR	6
SET_DESCRIPTOR	7
GET_CONFIGURATION	8
SET_CONFIGURATION	9
GET_INTERFACE	10
SET_INTERFACE	11
SYNCH_FRAME	12

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 9-5. Descriptor Types

Descriptor Types	Value
DEVICE	1
CONFIGURATION	2
STRING	3
INTERFACE	4
ENDPOINT	5
DEVICE_QUALIFIER	6
OTHER_SPEED_CONFIGURATION	7
INTERFACE_POWER[1]	8

[1] The INTERFACE_POWER descriptor is defined in the current revision of the *USB Interface Power Management Specification.*

Feature selectors are used when enabling or setting features, such as remote wakeup, specific to a device, interface, or endpoint. The values for the feature selectors are given in Table 9-6.

Table 9-6. Standard Feature Selectors

Feature Selector	Recipient	Value
device_remote_wakeup	Device	1
endpoint_HALT	Endpoint	0
TEST_MODE	Device	2

If an unsupported or invalid request is made to a USB device, the device responds by returning STALL in the Data or Status stage of the request. If the device detects the error in the Setup stage, it is preferred that the device returns STALL at the earlier of the Data or Status stage. Receipt of an unsupported or invalid request does NOT cause the optional *Halt* feature on the control pipe to be set. If for any reason, the device becomes unable to communicate via its Default Control Pipe due to an error condition, the device must be reset to clear the condition and restart the Default Control Pipe.

9.4.1 Clear Feature

This request is used to clear or disable a specific feature.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B 00000001B 00000010B	CLEAR_FEATURE	Feature Selector	Zero Interface Endpoint	Zero	None

Feature selector values in wValue must be appropriate to the recipient. Only device feature selector values may be used when the recipient is a device, only interface feature selector values may be used when the recipient is an interface, and only endpoint feature selector values may be used when the recipient is an endpoint.

Refer to Table 9-6 for a definition of which feature selector values are defined for which recipients.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A ClearFeature() request that references a feature that cannot be cleared, that does not exist, or that references an interface or endpoint that does not exist, will cause the device to respond with a Request Error.

If *wLength* is non-zero, then the device behavior is not specified.

Default state: Device behavior when this request is received while the device is in the Default state is not specified.

Address state: This request is valid when the device is in the Address state; references to interfaces or to endpoints other than endpoint zero shall cause the device to respond with a Request Error.

Configured state: This request is valid when the device is in the Configured state.

Note: The Test_Mode feature cannot be cleared by the ClearFeature() request.

9.4.2 Get Configuration

This request returns the current device configuration value.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10000000B	GET_CONFIGURATION	Zero	Zero	One	Configuration Value

If the returned value is zero, the device is not configured.

If *wValue*, *wIndex*, or *wLength* are not as specified above, then the device behavior is not specified.

Default state: Device behavior when this request is received while the device is in the Default state is not specified.

Address state: The value zero must be returned.

Configured state: The non-zero *bConfigurationValue* of the current configuration must be returned.

9.4.3 Get Descriptor

This request returns the specified descriptor if the descriptor exists.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10000000B	GET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID (refer to Section 9.6.7)	Descriptor Length	Descriptor

The wValue field specifies the descriptor type in the high byte (refer to Table 9-5) and the descriptor index in the low byte. The descriptor index is used to select a specific descriptor (only for configuration and string descriptors) when several descriptors of the same type are implemented in a device. For example, a device can implement several configuration descriptors. For other standard descriptors that can be retrieved via a GetDescriptor() request, a descriptor index of zero must be used. The range of values used for a descriptor index is from 0 to one less than the number of descriptors of that type implemented by the device.

The wIndex field specifies the Language ID for string descriptors or is reset to zero for other descriptors. The wLength field specifies the number of bytes to return. If the descriptor is longer than the wLength field, only the initial bytes of the descriptor are returned. If the descriptor is shorter than the wLength field, the device indicates the end of the control transfer by sending a short packet when further data is requested. A short packet is defined as a packet shorter than the maximum payload size or a zero length data packet (refer to Chapter 5).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The standard request to a device supports three types of descriptors: device (also device_qualifier), configuration (also other_speed_configuration), and string. A high-speed capable device supports the device_qualifier descriptor to return information about the device for the speed at which it is not operating (including *wMaxPacketSize* for the default endpoint and the number of configurations for the other speed). The other_speed_configuration returns information in the same structure as a configuration descriptor, but for a configuration if the device were operating at the other speed. A request for a configuration descriptor returns the configuration descriptor, all interface descriptors, and endpoint descriptors for all of the interfaces in a single request. The first interface descriptor follows the configuration descriptor. The endpoint descriptors for the first interface follow the first interface descriptor. If there are additional interfaces, their interface descriptor and endpoint descriptors follow the first interface's endpoint descriptors. Class-specific and/or vendor-specific descriptors follow the standard descriptors they extend or modify.

All devices must provide a device descriptor and at least one configuration descriptor. If a device does not support a requested descriptor, it responds with a Request Error.

Default state: This is a valid request when the device is in the Default state.

Address state: This is a valid request when the device is in the Address state.

Configured state: This is a valid request when the device is in the Configured state.

9.4.4 Get Interface

This request returns the selected alternate setting for the specified interface.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10000001B	GET_INTERFACE	Zero	Interface	One	Alternate Setting

Some USB devices have configurations with interfaces that have mutually exclusive settings. This request allows the host to determine the currently selected alternate setting.

If *wValue* or *wLength* are not as specified above, then the device behavior is not specified.

If the interface specified does not exist, then the device responds with a Request Error.

Default state: Device behavior when this request is received while the device is in the Default state is not specified.

Address state: A Request Error response is given by the device.

Configured state: This is a valid request when the device is in the Configured state.

9.4.5 Get Status

This request returns status for the specified recipient.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10000000B 10000001B 10000010B	GET_STATUS	Zero	Zero Interface Endpoint	Two	Device, Interface, or Endpoint Status

The *Recipient* bits of the bmRequestType field specify the desired recipient. The data returned is the current status of the specified recipient.

If *wValue* or *wLength* are not as specified above, or if *wIndex* is non-zero for a device status request, then the behavior of the device is not specified.

If an interface or an endpoint is specified that does not exist, then the device responds with a Request Error.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Default state: Device behavior when this request is received while the device is in the Default state is not specified.

Address state: If an interface or an endpoint other than endpoint zero is specified, then the device responds with a Request Error.

Configured state: If an interface or endpoint that does not exist is specified, then the device responds with a Request Error.

A GetStatus() request to a device returns the information shown in Figure 9-4.

D7	D6	D5	D4	D3	D2	D1	D0
Reserved (Reset to zero)						Remote Wakeup	Self Powered
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Reserved (Reset to zero)							

Figure 9-4. Information Returned by a GetStatus() Request to a Device

The Self Powered field indicates whether the device is currently self-powered. If D0 is reset to zero, the device is bus-powered. If D0 is set to one, the device is self-powered. The Self Powered field may not be changed by the SetFeature() or ClearFeature() requests.

The Remote Wakeup field indicates whether the device is currently enabled to request remote wakeup. The default mode for devices that support remote wakeup is disabled. If D1 is reset to zero, the ability of the device to signal remote wakeup is disabled. If D1 is set to one, the ability of the device to signal remote wakeup is enabled. The Remote Wakeup field can be modified by the SetFeature() and ClearFeature() requests using the device_remote_wakeup feature selector. This field is reset to zero when the device is reset.

A GetStatus() request to an interface returns the information shown in Figure 9-5.

D7	D6	D5	D4	D3	D2	D1	D0
Reserved (Reset to zero)							
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Reserved (Reset to zero)							

Figure 9-5. Information Returned by a GetStatus() Request to an Interface

A GetStatus() request to an endpoint returns the information shown in Figure 9-6.

D7	D6	D5	D4	D3	D2	D1	D0
Reserved (Reset to zero)							Halt
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Reserved (Reset to zero)							

Figure 9-6. Information Returned by a GetStatus() Request to an Endpoint

The *Halt* feature is required to be implemented for all interrupt and bulk endpoint types. If the endpoint is currently halted, then the Halt feature is set to one. Otherwise, the Halt feature is reset to zero. The Halt feature may optionally be set with the SetFeature(ENDPOINT_HALT) request. When set by the SetFeature() request, the endpoint exhibits the same stall behavior as if the field had been set by a hardware condition. If the condition causing a halt has been removed, clearing the *Halt* feature via a ClearFeature(ENDPOINT_HALT) request results in the endpoint no longer returning a STALL. For endpoints using data toggle, regardless of whether an endpoint has the *Halt* feature set, a ClearFeature(ENDPOINT_HALT) request always results in the data toggle being reinitialized to data0. The *Halt* feature is reset to zero

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

after either a SetConfiguration() or SetInterface() request even if the requested configuration or interface is the same as the current configuration or interface.

It is neither required nor recommended that the *Halt* feature be implemented for the Default Control Pipe. However, devices may set the *Halt* feature of the Default Control Pipe in order to reflect a functional error condition. If the feature is set to one, the device will return STALL in the Data and Status stages of each standard request to the pipe except GetStatus(), SetFeature(), and ClearFeature() requests. The device need not return STALL for class-specific and vendor-specific requests.

9.4.6 Set Address

This request sets the device address for all future device accesses.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B	SET_ADDRESS	Device Address	Zero	Zero	None

The wValue field specifies the device address to use for all subsequent accesses.

As noted elsewhere, requests actually may result in up to three stages. In the first stage, the Setup packet is sent to the device. In the optional second stage, data is transferred between the host and the device. In the final stage, status is transferred between the host and the device. The direction of data and status transfer depends on whether the host is sending data to the device or the device is sending data to the host. The Status stage transfer is always in the opposite direction of the Data stage. If there is no Data stage, the Status stage is from the device to the host.

Stages after the initial Setup packet assume the same device address as the Setup packet. The USB device does not change its device address until after the Status stage of this request is completed successfully. Note that this is a difference between this request and all other requests. For all other requests, the operation indicated must be completed before the Status stage.

If the specified device address is greater than 127, or if *wIndex* or *wLength* are non-zero, then the behavior of the device is not specified.

Device response to SetAddress() with a value of 0 is undefined.

Default state: If the address specified is non-zero, then the device shall enter the Address state; otherwise, the device remains in the Default state (this is not an error condition).

Address state: If the address specified is zero, then the device shall enter the Default state; otherwise, the device remains in the Address state but uses the newly-specified address.

Configured state: Device behavior when this request is received while the device is in the Configured state is not specified.

9.4.7 Set Configuration

This request sets the device configuration.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B	SET_CONFIGURATION	Configuration Value	Zero	Zero	None

The lower byte of the wValue field specifies the desired configuration. This configuration value must be zero or match a configuration value from a configuration descriptor. If the configuration value is zero, the device is placed in its Address state. The upper byte of the *wValue* field is reserved.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

If *wIndex*, *wLength*, or the upper byte of *wValue* is non-zero, then the behavior of this request is not specified.

Default state: Device behavior when this request is received while the device is in the Default state is not specified.

Address state: If the specified configuration value is zero, then the device remains in the Address state. If the specified configuration value matches the configuration value from a configuration descriptor, then that configuration is selected and the device enters the Configured state. Otherwise, the device responds with a Request Error.

Configured state: If the specified configuration value is zero, then the device enters the Address state. If the specified configuration value matches the configuration value from a configuration descriptor, then that configuration is selected and the device remains in the Configured state. Otherwise, the device responds with a Request Error.

9.4.8 Set Descriptor

This request is optional and may be used to update existing descriptors or new descriptors may be added.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B	SET_DESCRIPTOR	Descriptor Type and Descriptor Index	Language ID (refer to Section 9.6.7) or zero	Descriptor Length	Descriptor

The wValue field specifies the descriptor type in the high byte (refer to Table 9-5) and the descriptor index in the low byte. The descriptor index is used to select a specific descriptor (only for configuration and string descriptors) when several descriptors of the same type are implemented in a device. For example, a device can implement several configuration descriptors. For other standard descriptors that can be set via a SetDescriptor() request, a descriptor index of zero must be used. The range of values used for a descriptor index is from 0 to one less than the number of descriptors of that type implemented by the device.

The wIndex field specifies the Language ID for string descriptors or is reset to zero for other descriptors. The wLength field specifies the number of bytes to transfer from the host to the device.

The only allowed values for descriptor type are device, configuration, and string descriptor types.

If this request is not supported, the device will respond with a Request Error.

Default state: Device behavior when this request is received while the device is in the Default state is not specified.

Address state: If supported, this is a valid request when the device is in the Address state.

Configured state: If supported, this is a valid request when the device is in the Configured state.

9.4.9 Set Feature

This request is used to set or enable a specific feature.

bmRequestType	bRequest	wValue	wIndex		wLength	Data
00000000B 00000001B 00000010B	SET_FEATURE	Feature Selector	Test Selector	Zero Interface Endpoint	Zero	None

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Feature selector values in wValue must be appropriate to the recipient. Only device feature selector values may be used when the recipient is a device; only interface feature selector values may be used when the recipient is an interface, and only endpoint feature selector values may be used when the recipient is an endpoint.

Refer to Table 9-6 for a definition of which feature selector values are defined for which recipients.

The TEST_MODE feature is only defined for a device recipient (i.e., bmRequestType = 0) and the lower byte of wIndex must be zero. Setting the TEST_MODE feature puts the device upstream facing port into test mode. The device will respond with a request error if the request contains an invalid test selector. The transition to test mode must be complete no later than 3 ms after the completion of the status stage of the request. The transition to test mode of an upstream facing port must not happen until after the status stage of the request. The power to the device must be cycled to exit test mode of an upstream facing port of a device. See Section 7.1.20 for definitions of each test mode. A device must support the TEST_MODE feature when in the Default, Address or Configured high-speed device states.

A SetFeature() request that references a feature that cannot be set or that does not exist causes a STALL to be returned in the Status stage of the request.

Table 9-7. Test Mode Selectors

Value	Description
00H	Reserved
01H	Test_J
02H	Test_K
03H	Test_SE0_NAK
04H	Test_Packet
05H	Test_Force_Enable
06H-3FH	Reserved for standard test selectors
3FH-BFH	Reserved
C0H-FFH	Reserved for vendor-specific test modes.

If the feature selector is *TEST_MODE*, then the most significant byte of *wIndex* is used to specify the specific test mode. The recipient of a SetFeature(TEST_MODE...) must be the device; i.e., the lower byte of *wIndex* must be zero and the *bmRequestType* must be set to zero. The device must have its power cycled to exit test mode. The valid test mode selectors are listed in Table 9-7. See Section 7.1.20 for more information about the specific test modes.

If *wLength* is non-zero, then the behavior of the device is not specified.

If an endpoint or interface is specified that does not exist, then the device responds with a Request Error.

Default state: A device must be able to accept a SetFeature(TEST_MODE, TEST_SELECTOR) request when in the Default State. Device behavior for other SetFeature requests while the device is in the Default state is not specified.

Address state: If an interface or an endpoint other than endpoint zero is specified, then the device responds with a Request Error.

Configured state: This is a valid request when the device is in the Configured state.

9.4.10 Set Interface

This request allows the host to select an alternate setting for the specified interface.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000001B	SET_INTERFACE	Alternate Setting	Interface	Zero	None

Some USB devices have configurations with interfaces that have mutually exclusive settings. This request allows the host to select the desired alternate setting. If a device only supports a default setting for the specified interface, then a STALL may be returned in the Status stage of the request. This request cannot be used to change the set of configured interfaces (the SetConfiguration() request must be used instead).

If the interface or the alternate setting does not exist, then the device responds with a Request Error. If *wLength* is non-zero, then the behavior of the device is not specified.

Default state: Device behavior when this request is received while the device is in the Default state is not specified.

Address state: The device must respond with a Request Error.

Configured state: This is a valid request when the device is in the Configured state.

9.4.11 Synch Frame

This request is used to set and then report an endpoint's synchronization frame.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10000010B	SYNCH_FRAME	Zero	Endpoint	Two	Frame Number

When an endpoint supports isochronous transfers, the endpoint may also require per-frame transfers to vary in size according to a specific pattern. The host and the endpoint must agree on which frame the repeating pattern begins. The number of the frame in which the pattern began is returned to the host.

If a high-speed device supports the Synch Frame request, it must internally synchronize itself to the zeroth microframe and have a time notion of classic frame. Only the frame number is used to synchronize and reported by the device endpoint (i.e., no microframe number). The endpoint must synchronize to the zeroth microframe.

This value is only used for isochronous data transfers using implicit pattern synchronization. If *wValue* is non-zero or *wLength* is not two, then the behavior of the device is not specified.

If the specified endpoint does not support this request, then the device will respond with a Request Error.

Default state: Device behavior when this request is received while the device is in the Default state is not specified.

Address state: The device shall respond with a Request Error.

Configured state: This is a valid request when the device is in the Configured state.

9.5 Descriptors

USB devices report their attributes using descriptors. A descriptor is a data structure with a defined format. Each descriptor begins with a byte-wide field that contains the total number of bytes in the descriptor followed by a byte-wide field that identifies the descriptor type.

Using descriptors allows concise storage of the attributes of individual configurations because each configuration may reuse descriptors or portions of descriptors from other configurations that have the same characteristics. In this manner, the descriptors resemble individual data records in a relational database.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Where appropriate, descriptors contain references to string descriptors that provide displayable information describing a descriptor in human-readable form. The inclusion of string descriptors is optional. However, the reference fields within descriptors are mandatory. If a device does not support string descriptors, string reference fields must be reset to zero to indicate no string descriptor is available.

If a descriptor returns with a value in its length field that is less than defined by this specification, the descriptor is invalid and should be rejected by the host. If the descriptor returns with a value in its length field that is greater than defined by this specification, the extra bytes are ignored by the host, but the next descriptor is located using the length returned rather than the length expected.

A device may return class- or vendor-specific descriptors in two ways:

1. If the class or vendor specific descriptors use the same format as standard descriptors (e.g., start with a length byte and followed by a type byte), they must be returned interleaved with standard descriptors in the configuration information returned by a GetDescriptor(Configuration) request. In this case, the class or vendor-specific descriptors must follow a related standard descriptor they modify or extend.
2. If the class or vendor specific descriptors are independent of configuration information or use a non-standard format, a GetDescriptor() request specifying the class or vendor specific descriptor type and index may be used to retrieve the descriptor from the device. A class or vendor specification will define the appropriate way to retrieve these descriptors.

9.6 Standard USB Descriptor Definitions

The standard descriptors defined in this specification may only be modified or extended by revision of the Universal Serial Bus Specification.

Note: An extension to the USB 1.0 standard endpoint descriptor has been published in Device Class Specification for Audio Devices Revision 1.0. This is the only extension defined outside USB Specification that is allowed. Future revisions of the USB Specification that extend the standard endpoint descriptor will do so as to not conflict with the extension defined in the Audio Device Class Specification Revision 1.0.

9.6.1 Device

A device descriptor describes general information about a USB device. It includes information that applies globally to the device and all of the device's configurations. A USB device has only one device descriptor.

A high-speed capable device that has different device information for full-speed and high-speed must also have a device_qualifier descriptor (see Section 9.6.2).

The DEVICE descriptor of a high-speed capable device has a version number of 2.0 (0200H). If the device is full-speed only or low-speed only, this version number indicates that it will respond correctly to a request for the device_qualifier desciptor (i.e., it will respond with a request error).

The *bcdUSB* field contains a BCD version number. The value of the *bcdUSB* field is 0xJJMN for version JJ.M.N (JJ – major version number, M – minor version number, N – sub-minor version number), e.g., version 2.1.3 is represented with value 0x0213 and version 2.0 is represented with a value of 0x0200.

The *bNumConfigurations* field indicates the number of configurations at the current operating speed. Configurations for the other operating speed are not included in the count. If there are specific configurations of the device for specific speeds, the *bNumConfigurations* field only reflects the number of configurations for a single speed, not the total number of configurations for both speeds.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

If the device is operating at high-speed, the *bMaxPacketSize0* field must be 64 indicating a 64 byte maximum packet. High-speed operation does not allow other maximum packet sizes for the control endpoint (endpoint 0).

All USB devices have a Default Control Pipe. The maximum packet size of a device's Default Control Pipe is described in the device descriptor. Endpoints specific to a configuration and its interface(s) are described in the configuration descriptor. A configuration and its interface(s) do not include an endpoint descriptor for the Default Control Pipe. Other than the maximum packet size, the characteristics of the Default Control Pipe are defined by this specification and are the same for all USB devices.

The bNumConfigurations field identifies the number of configurations the device supports. Table 9-8 shows the standard device descriptor.

Table 9-8. Standard Device Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	DEVICE Descriptor Type
2	*bcdUSB*	2	BCD	USB Specification Release Number in Binary-Coded Decimal (i.e., 2.10 is 210H). This field identifies the release of the USB Specification with which the device and its descriptors are compliant.
4	*bDeviceClass*	1	Class	Class code (assigned by the USB-IF). If this field is reset to zero, each interface within a configuration specifies its own class information and the various interfaces operate independently. If this field is set to a value between 1 and FEH, the device supports different class specifications on different interfaces and the interfaces may not operate independently. This value identifies the class definition used for the aggregate interfaces. If this field is set to FFH, the device class is vendor-specific.
5	*bDeviceSubClass*	1	SubClass	Subclass code (assigned by the USB-IF). These codes are qualified by the value of the bDeviceClass field. If the bDeviceClass field is reset to zero, this field must also be reset to zero. If the bDeviceClass field is not set to FFH, all values are reserved for assignment by the USB-IF.
6	*bDeviceProtocol*	1	Protocol	Protocol code (assigned by the USB-IF). These codes are qualified by the value of the bDeviceClass and the bDeviceSubClass fields. If a device supports class-specific protocols on a device basis as opposed to an interface basis, this code identifies the protocols that the device uses as defined by the specification of the device class. If this field is reset to zero, the device does not use class-specific protocols on a device basis. However, it may use class-specific protocols on an interface basis. If this field is set to FFH, the device uses a vendor-specific protocol on a device basis.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 9-8 Standard Device Descriptor

7	*bMaxPacketSize0*	1	Number	Maximum packet size for endpoint zero (only 8, 16, 32, or 64 are valid)
8	*idVendor*	2	ID	Vendor ID (assigned by the USB-IF)
10	*idProduct*	2	ID	Product ID (assigned by the manufacturer)
12	*bcdDevice*	2	BCD	Device release number in binary-coded decimal
14	*iManufacturer*	1	Index	Index of string descriptor describing manufacturer
15	*iProduct*	1	Index	Index of string descriptor describing product
16	*iSerialNumber*	1	Index	Index of string descriptor describing the device's serial number
17	*bNumConfigurations*	1	Number	Number of possible configurations

9.6.2 Device_Qualifier

The device_qualifier descriptor describes information about a high-speed capable device that would change if the device were operating at the other speed. For example, if the device is currently operating at full-speed, the device_qualifier returns information about how it would operate at high-speed and vice-versa. Table 9-9 shows the fields of the device_qualifier descriptor.

Table 9-9. Device_Qualifier Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of descriptor
1	*bDescriptorType*	1	Constant	Device Qualifier Type
2	*bcdUSB*	2	BCD	USB specification version number (e.g., 0200H for V2.00)
4	*bDeviceClass*	1	Class	Class Code
5	*bDeviceSubClass*	1	SubClass	SubClass Code
6	*bDeviceProtocol*	1	Protocol	Protocol Code
7	*bMaxPacketSize0*	1	Number	Maximum packet size for other speed
8	*bNumConfigurations*	1	Number	Number of Other-speed Configurations
9	*bReserved*	1	Zero	Reserved for future use, must be zero

The vendor, product, device, manufacturer, product, and serialnumber fields of the standard device descriptor are not included in this descriptor since that information is constant for a device for all supported speeds. The version number for this descriptor must be at least 2.0 (0200H).

The host accesses this descriptor using the GetDescriptor() request. The descriptor type in the GetDescriptor() request is set to device_qualifier (see Table 9-5).

If a full-speed only device (with a device descriptor version number equal to 0200H) receives a GetDescriptor() request for a device_qualifier, it must respond with a request error. The host must not make a request for an other_speed_configuration descriptor unless it first successfully retrieves the device_qualifier descriptor.

9.6.3 Configuration

The configuration descriptor describes information about a specific device configuration. The descriptor contains a bConfigurationValue field with a value that, when used as a parameter to the SetConfiguration() request, causes the device to assume the described configuration.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

The descriptor describes the number of interfaces provided by the configuration. Each interface may operate independently. For example, an ISDN device might be configured with two interfaces, each providing 64 Kb/s bi-directional channels that have separate data sources or sinks on the host. Another configuration might present the ISDN device as a single interface, bonding the two channels into one 128 Kb/s bi-directional channel.

When the host requests the configuration descriptor, all related interface and endpoint descriptors are returned (refer to Section 9.4.3).

A USB device has one or more configuration descriptors. Each configuration has one or more interfaces and each interface has zero or more endpoints. An endpoint is not shared among interfaces within a single configuration unless the endpoint is used by alternate settings of the same interface. Endpoints may be shared among interfaces that are part of different configurations without this restriction.

Once configured, devices may support limited adjustments to the configuration. If a particular interface has alternate settings, an alternate may be selected after configuration. Table 9-10 shows the standard configuration descriptor.

Table 9-10. Standard Configuration Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	CONFIGURATION Descriptor Type
2	*wTotalLength*	2	Number	Total length of data returned for this configuration. Includes the combined length of all descriptors (configuration, interface, endpoint, and class- or vendor-specific) returned for this configuration.
4	*bNumInterfaces*	1	Number	Number of interfaces supported by this configuration
5	*bConfigurationValue*	1	Number	Value to use as an argument to the SetConfiguration() request to select this configuration
6	*iConfiguration*	1	Index	Index of string descriptor describing this configuration
7	*bmAttributes*	1	Bitmap	Configuration characteristics D7: Reserved (set to one) D6: Self-powered D5: Remote Wakeup D4...0: Reserved (reset to zero) D7 is reserved and must be set to one for historical reasons. A device configuration that uses power from the bus and a local source reports a non-zero value in *bMaxPower* to indicate the amount of bus power required and sets D6. The actual power source at runtime may be determined using the GetStatus(DEVICE) request (see Section 9.4.5). If a device configuration supports remote wakeup, D5 is set to one.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 9-10. Standard Configuration Descriptor (Continued)

Offset	Field	Size	Value	Description
8	*bMaxPower*	1	mA	Maximum power consumption of the USB device from the bus in this specific configuration when the device is fully operational. Expressed in 2 mA units (i.e., 50 = 100 mA). Note: A device configuration reports whether the configuration is bus-powered or self-powered. Device status reports whether the device is currently self-powered. If a device is disconnected from its external power source, it updates device status to indicate that it is no longer self-powered. A device may not increase its power draw from the bus, when it loses its external power source, beyond the amount reported by its configuration. If a device can continue to operate when disconnected from its external power source, it continues to do so. If the device cannot continue to operate, it fails operations it can no longer support. The USB System Software may determine the cause of the failure by checking the status and noting the loss of the device's power source.

9.6.4 Other_Speed_Configuration

The other_speed_configuration descriptor shown in Table 9-11 describes a configuration of a high-speed capable device if it were operating at its other possible speed. The structure of the other_speed_configuration is identical to a configuration descriptor.

Table 9-11. Other_Speed_Configuration Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of descriptor
1	*bDescriptorType*	1	Constant	Other_speed_Configuration Type
2	*wTotalLength*	2	Number	Total length of data returned
4	*bNumInterfaces*	1	Number	Number of interfaces supported by this speed configuration
5	*bConfigurationValue*	1	Number	Value to use to select configuration
6	*iConfiguration*	1	Index	Index of string descriptor
7	*bmAttributes*	1	Bitmap	Same as Configuration descriptor
8	*bMaxPower*	1	mA	Same as Configuration descriptor

The host accesses this descriptor using the GetDescriptor() request. The descriptor type in the GetDescriptor() request is set to other_speed_configuration (see Table 9-5).

9.6.5 Interface

The interface descriptor describes a specific interface within a configuration. A configuration provides one or more interfaces, each with zero or more endpoint descriptors describing a unique set of endpoints within the configuration. When a configuration supports more than one interface, the endpoint descriptors for a particular interface follow the interface descriptor in the data returned by the GetConfiguration() request. An interface descriptor is always

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

returned as part of a configuration descriptor. Interface descriptors cannot be directly accessed with a GetDescriptor() or SetDescriptor() request.

An interface may include alternate settings that allow the endpoints and/or their characteristics to be varied after the device has been configured. The default setting for an interface is always alternate setting zero. The SetInterface() request is used to select an alternate setting or to return to the default setting. The GetInterface() request returns the selected alternate setting.

Alternate settings allow a portion of the device configuration to be varied while other interfaces remain in operation. If a configuration has alternate settings for one or more of its interfaces, a separate interface descriptor and its associated endpoints are included for each setting.

If a device configuration supported a single interface with two alternate settings, the configuration descriptor would be followed by an interface descriptor with the *bInterfaceNumber* and *bAlternateSetting* fields set to zero and then the endpoint descriptors for that setting, followed by another interface descriptor and its associated endpoint descriptors. The second interface descriptor's *bInterfaceNumber* field would also be set to zero, but the *bAlternateSetting* field of the second interface descriptor would be set to one.

If an interface uses only endpoint zero, no endpoint descriptors follow the interface descriptor. In this case, the *bNumEndpoints* field must be set to zero.

An interface descriptor never includes endpoint zero in the number of endpoints. Table 9-12 shows the standard interface descriptor.

Table 9-12. Standard Interface Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	INTERFACE Descriptor Type
2	*bInterfaceNumber*	1	Number	Number of this interface. Zero-based value identifying the index in the array of concurrent interfaces supported by this configuration.
3	*bAlternateSetting*	1	Number	Value used to select this alternate setting for the interface identified in the prior field
4	*bNumEndpoints*	1	Number	Number of endpoints used by this interface (excluding endpoint zero). If this value is zero, this interface only uses the Default Control Pipe.
5	*bInterfaceClass*	1	Class	Class code (assigned by the USB-IF). A value of zero is reserved for future standardization. If this field is set to FFH, the interface class is vendor-specific. All other values are reserved for assignment by the USB-IF.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 9-12. Standard Interface Descriptor (Continued)

Offset	Field	Size	Value	Description
6	*bInterfaceSubClass*	1	SubClass	Subclass code (assigned by the USB-IF). These codes are qualified by the value of the bInterfaceClass field. If the bInterfaceClass field is reset to zero, this field must also be reset to zero. If the bInterfaceClass field is not set to FFH, all values are reserved for assignment by the USB-IF.
7	*bInterfaceProtocol*	1	Protocol	Protocol code (assigned by the USB). These codes are qualified by the value of the bInterfaceClass and the bInterfaceSubClass fields. If an interface supports class-specific requests, this code identifies the protocols that the device uses as defined by the specification of the device class. If this field is reset to zero, the device does not use a class-specific protocol on this interface. If this field is set to FFH, the device uses a vendor-specific protocol for this interface.
8	*iInterface*	1	Index	Index of string descriptor describing this interface

9.6.6 Endpoint

Each endpoint used for an interface has its own descriptor. This descriptor contains the information required by the host to determine the bandwidth requirements of each endpoint. An endpoint descriptor is always returned as part of the configuration information returned by a GetDescriptor(Configuration) request. An endpoint descriptor cannot be directly accessed with a GetDescriptor() or SetDescriptor() request. There is never an endpoint descriptor for endpoint zero. Table 9-13 shows the standard endpoint descriptor.

Table 9-13. Standard Endpoint Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	ENDPOINT Descriptor Type
2	*bEndpointAddress*	1	Endpoint	The address of the endpoint on the USB device described by this descriptor. The address is encoded as follows: Bit 3...0: The endpoint number Bit 6...4: Reserved, reset to zero Bit 7: Direction, ignored for control endpoints 0 = OUT endpoint 1 = IN endpoint

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 9-13. Standard Endpoint Descriptor (Continued)

Offset	Field	Size	Value	Description
3	*bmAttributes*	1	Bitmap	This field describes the endpoint's attributes when it is configured using the *bConfigurationValue*. Bits 1..0: Transfer Type 00 = Control 01 = Isochronous 10 = Bulk 11 = Interrupt If not an isochronous endpoint, bits 5..2 are reserved and must be set to zero. If isochronous, they are defined as follows: Bits 3..2: Synchronization Type 00 = No Synchronization 01 = Asynchronous 10 = Adaptive 11 = Synchronous Bits 5..4: Usage Type 00 = Data endpoint 01 = Feedback endpoint 10 = Implicit feedback Data endpoint 11 = Reserved Refer to Chapter 5 for more information. All other bits are reserved and must be reset to zero. Reserved bits must be ignored by the host.
4	*wMaxPacketSize*	2	Number	Maximum packet size this endpoint is capable of sending or receiving when this configuration is selected. For isochronous endpoints, this value is used to reserve the bus time in the schedule, required for the per-(micro)frame data payloads. The pipe may, on an ongoing basis, actually use less bandwidth than that reserved. The device reports, if necessary, the actual bandwidth used via its normal, non-USB defined mechanisms. For all endpoints, bits 10..0 specify the maximum packet size (in bytes). For high-speed isochronous and interrupt endpoints: Bits 12..11 specify the number of additional transaction opportunities per microframe: 00 = None (1 transaction per microframe) 01 = 1 additional (2 per microframe) 10 = 2 additional (3 per microframe) 11 = Reserved Bits 15..13 are reserved and must be set to zero. Refer to Chapter 5 for more information.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 9-13. Standard Endpoint Descriptor (Continued)

Offset	Field	Size	Value	Description
6	*bInterval*	1	Number	Interval for polling endpoint for data transfers. Expressed in frames or microframes depending on the device operating speed (i.e., either 1 millisecond or 125 µs units). For full-/high-speed isochronous endpoints, this value must be in the range from 1 to 16. The *bInterval* value is used as the exponent for a $2^{(bInterval-1)}$ value; e.g., a *bInterval* of 4 means a period of 8 ($2^{(4-1)}$). For full-/low-speed interrupt endpoints, the value of this field may be from 1 to 255. For high-speed interrupt endpoints, the *bInterval* value is used as the exponent for a $2^{(bInterval-1)}$ value; e.g., a *bInterval* of 4 means a period of 8 ($2^{(4-1)}$). This value must be from 1 to 16. For high-speed bulk/control OUT endpoints, the *bInterval* must specify the maximum NAK rate of the endpoint. A value of 0 indicates the endpoint never NAKs. Other values indicate at most 1 NAK each *bInterval* number of microframes. This value must be in the range from 0 to 255. See Chapter 5 description of periods for more detail.

The *bmAttributes* field provides information about the endpoint's Transfer Type (bits 1..0) and Synchronization Type (bits 3..2). In addition, the Usage Type bit (bits 5..4) indicate whether this is an endpoint used for normal data transfers (bits 5..4=00B), whether it is used to convey explicit feedback information for one or more data endpoints (bits 5..4=01B) or whether it is a data endpoint that also serves as an implicit feedback endpoint for one or more data endpoints (bits 5..4=10B). Bits 5..2 are only meaningful for isochronous endpoints and must be reset to zero for all other transfer types.

If the endpoint is used as an explicit feedback endpoint (bits 5..4=01B), then the Transfer Type must be set to isochronous (bits1..0 = 01B) and the Synchronization Type must be set to No Synchronization (bits 3..2=00B).

A feedback endpoint (explicit or implicit) needs to be associated with one (or more) isochronous data endpoints to which it provides feedback service. The association is based on endpoint number matching. A feedback endpoint always has the opposite direction from the data endpoint(s) it services. If multiple data endpoints are to be serviced by the same feedback endpoint, the data endpoints must have ascending ordered–but not necessarily consecutive–endpoint numbers. The first data endpoint and the feedback endpoint must have the same endpoint number (and opposite direction). This ensures that a data endpoint can uniquely identify its feedback endpoint by searching for the first feedback endpoint that has an endpoint number equal or less than its own endpoint number.

Example: Consider the extreme case where there is a need for five groups of OUT asynchronous isochronous endpoints and at the same time four groups of IN adaptive isochronous endpoints. Each group needs a separate feedback endpoint and the groups are composed as shown in Figure 9-7.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

OUT Group	Nr of OUT Endpoints	IN Group	Nr of IN Endpoints
1	1	6	1
2	2	7	2
3	2	8	3
4	3	9	4
5	3		

Figure 9-7. Example of Feedback Endpoint Numbers

The endpoint numbers can be intertwined as illustrated in Figure 9-8.

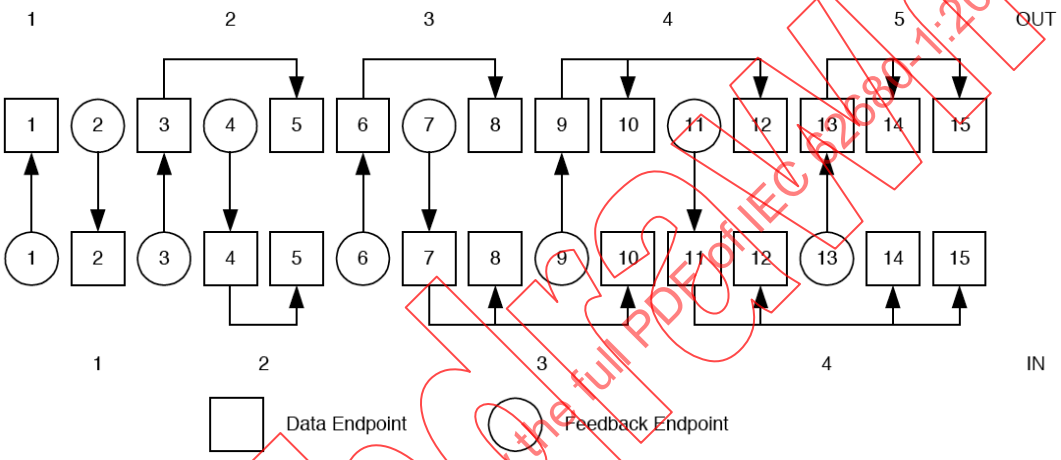


Figure 9-8. Example of Feedback Endpoint Relationships

High-speed isochronous and interrupt endpoints use bits 12..11 of *wMaxPacketSize* to specify multiple transactions for each microframe specified by *bInterval.* If bits 12..11 of *wMaxPacketSize* are zero, the maximum packet size for the endpoint can be any allowed value (as defined in Chapter 5). If bits 12..11 of *wMaxPacketSize* are not zero (0), the allowed values for *wMaxPacketSize* bits 10..0 are limited as shown in Table 9-14.

Table 9-14. Allowed wMaxPacketSize Values for Different Numbers of Transactions per Microframe

***wMaxPacketSize* bits 12..11**	***wMaxPacketSize* bits 10..0 Values Allowed**
00	1 – 1024
01	513 – 1024
10	683 – 1024
11	N/A; reserved

For high-speed bulk and control OUT endpoints, the *bInterval* field is only used for compliance purposes; the host controller is not required to change its behavior based on the value in this field.

9.6.7 String

String descriptors are optional. As noted previously, if a device does not support string descriptors, all references to string descriptors within device, configuration, and interface descriptors must be reset to zero.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

String descriptors use UNICODE encodings as defined by *The Unicode Standard, Worldwide Character Encoding, Version 3.0*, The Unicode Consortium, Addison-Wesley Publishing Company, Reading, Massachusetts (URL: http://www.unicode.com). The strings in a USB device may support multiple languages. When requesting a string descriptor, the requester specifies the desired language using a sixteen-bit language ID (LANGID) defined by the USB-IF. The list of currently defined USB LANGIDs can be found at http://www.usb.org/developers/docs.html. String index zero for all languages returns a string descriptor that contains an array of two-byte LANGID codes supported by the device. Table 9-15 shows the LANGID code array. A USB device may omit all string descriptors. USB devices that omit all string descriptors must not return an array of LANGID codes.

The array of LANGID codes is not NULL-terminated. The size of the array (in bytes) is computed by subtracting two from the value of the first byte of the descriptor.

Table 9-15. String Descriptor Zero, Specifying Languages Supported by the Device

Offset	Field	Size	Value	Description
0	*bLength*	1	N+2	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	STRING Descriptor Type
2	*wLANGID[0]*	2	Number	LANGID code zero
...	...	...	...	...
N	*wLANGID[x]*	2	Number	LANGID code *x*

The UNICODE string descriptor (shown in Table 9-16) is not NULL-terminated. The string length is computed by subtracting two from the value of the first byte of the descriptor.

Table 9-16. UNICODE String Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	STRING Descriptor Type
2	*bString*	N	Number	UNICODE encoded string

9.7 Device Class Definitions

All devices must support the requests and descriptor definitions described in this chapter. Most devices provide additional requests and, possibly, descriptors for device-specific extensions. In addition, devices may provide extended services that are common to a group of devices. In order to define a class of devices, the following information must be provided to completely define the appearance and behavior of the device class.

9.7.1 Descriptors

If the class requires any specific definition of the standard descriptors, the class definition must include those requirements as part of the class definition. In addition, if the class defines a standard extended set of descriptors, they must also be fully defined in the class definition. Any extended descriptor definitions must follow the approach used for standard descriptors; for example, all descriptors must begin with a length field.

9.7.2 Interface(s) and Endpoint Usage

When a class of devices is standardized, the interfaces used by the devices, including how endpoints are used, must be included in the device class definition. Devices may further extend a class definition with proprietary features as long as they meet the base definition of the class.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9.7.3 Requests

All of the requests specific to the class must be defined.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

10 Chapter 10
USB Host: Hardware and Software

The USB interconnect supports data traffic between a host and a USB device. This chapter describes the host interfaces necessary to facilitate USB communication between a software client, resident on the host, and a function implemented on a device. The implementation described in this chapter is not required. This implementation is provided as an example to illustrate the host system behavior expected by a USB device. A host system may provide a different host software implementation as long as a USB device experiences the same host behavior.

10.1 Overview of the USB Host

10.1.1 Overview

The basic flow and interrelationships of the USB communications model are shown in Figure 10-1.

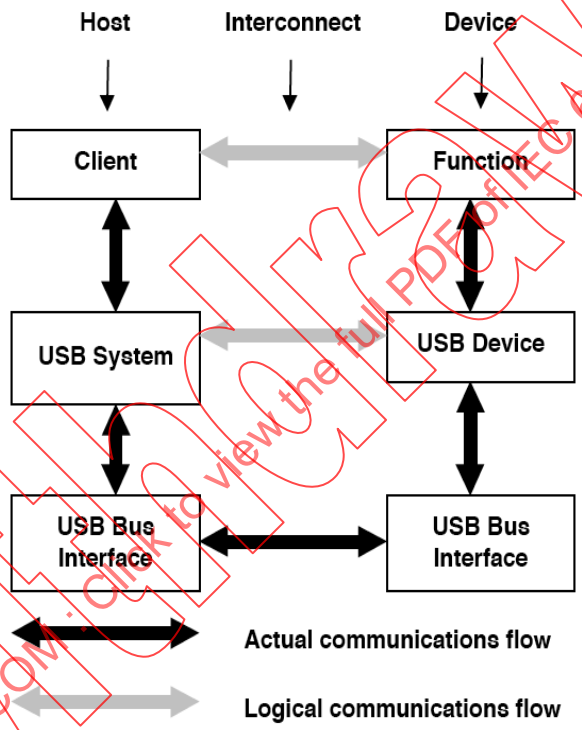


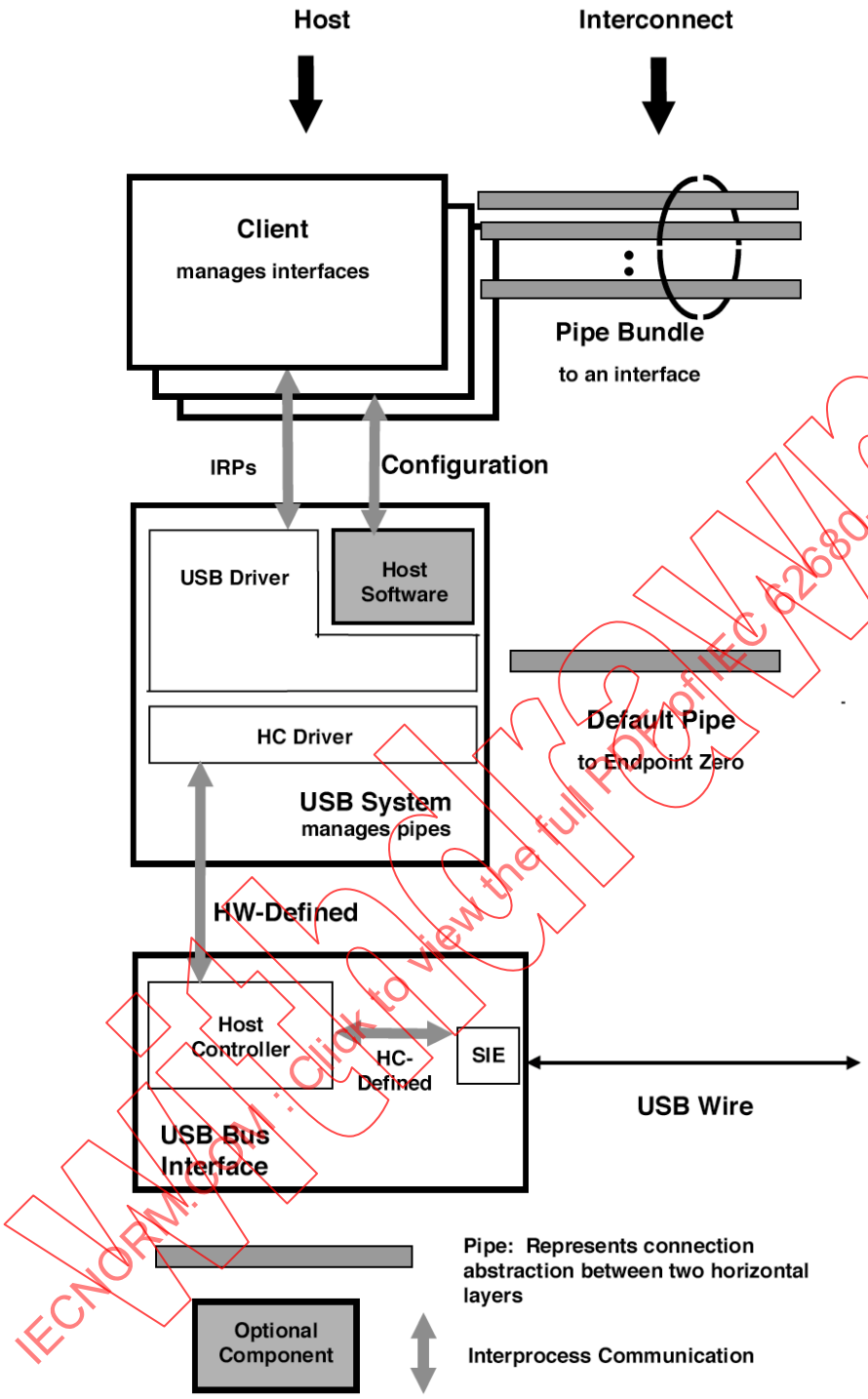
Figure 10-1. Interlayer Communications Model

The host and the device are divided into the distinct layers depicted in Figure 10-1. Vertical arrows indicate the actual communication on the host. The corresponding interfaces on the device are implementation-specific. All communications between the host and device ultimately occur on the physical USB wire. However, there are logical host-device interfaces between each horizontal layer. These communications, between client software resident on the host and the function provided by the device, are typified by a contract based on the needs of the application currently using the device and the capabilities provided by the device.

This client-function interaction creates the requirements for all of the underlying layers and their interfaces.

This chapter describes this model from the point of view of the host and its layers. Figure 10-2 illustrates, based on the overall view introduced in Chapter 5, the host's view of its communication with the device.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 10-2. Host Communications

There is only one host for each USB. The major layers of a host consist of the following:

- USB bus interface
- USB System
- Client

The USB bus interface handles interactions for the electrical and protocol layers (refer to Chapter 7 and Chapter 8). From the interconnect point of view, a similar USB bus interface is

provided by both the USB device and the host, as exemplified by the Serial Interface Engine (SIE). On the host, however, the USB bus interface has additional responsibilities due to the unique role of the host on the USB and is implemented as the Host Controller. The Host Controller has an integrated root hub providing attachment points to the USB wire.

The USB System uses the Host Controller to manage data transfers between the host and USB devices. The interface between the USB System and the Host Controller is dependent on the hardware definition of the Host Controller. The USB System, in concert with the Host Controller, performs the translation between the client's view of data transfers and the USB transactions appearing on the interconnect. This includes the addition of any USB feature support such as protocol wrappers. The USB System is also responsible for managing USB resources, such as bandwidth and bus power, so that client access to the USB is possible.

The USB System has three basic components:

- Host Controller Driver
- USB Driver
- Host Software

The Host Controller Driver (HCD) exists to more easily map the various Host Controller implementations into the USB System, such that a client can interact with its device without knowing to which Host Controller the device is connected. The USB Driver (USBD) provides the basic host interface (USBDI) for clients to USB devices. The interface between the HCD and the USBD is known as the Host Controller Driver Interface (HCDI). This interface is never available directly to clients and thus is not defined by the USB Specification. A particular HCDI is, however, defined by each operating system that supports various Host Controller implementations.

The USBD provides data transfer mechanisms in the form of I/O Request Packets (IRPs), which consist of a request to transport data across a specific pipe. In addition to providing data transfer mechanisms, the USBD is responsible for presenting to its clients an abstraction of a USB device that can be manipulated for configuration and state management. As part of this abstraction, the USBD owns the default pipe (see Chapter 5 and Chapter 9) through which all USB devices are accessed for the purposes of standard USB control. This default pipe represents a logical communication between the USBD and the abstraction of a USB device as shown in Figure 10-2.

In some operating systems, additional non-USB System Software is available that provides configuration and loading mechanisms to device drivers. In such operating systems, the device driver shall use the provided interfaces instead of directly accessing the USBDI mechanisms.

The client layer describes all the software entities that are responsible for directly interacting with USB devices. When each device is attached to the system, these clients might interact directly with the peripheral hardware. The shared characteristics of the USB place USB System Software between the client and its device; that is, a client cannot directly access the device's hardware.

Overall, the host layers provide the following capabilities:

- Detecting the attachment and removal of USB devices
- Managing USB standard control flow between the host and USB devices
- Managing data flow between the host and USB devices
- Collecting status and activity statistics
- Controlling the electrical interface between the Host Controller and USB devices, including the provision of a limited amount of power

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The following sections describe these responsibilities and the requirements placed on the USBDI in greater detail. The actual interfaces used for a specific combination of host platform and operating system are described in the appropriate operating system environment guide.

All hubs (see Chapter 11) report internal status changes and their port change status via the status change pipe. This includes a notification of when a USB device is attached to or removed from one of their ports. A USBD client generically known as the hub driver receives these notifications as owner of the hub's Status Change pipe. For device attachments, the hub driver then initiates the device configuration process. In some systems, this hub driver is a part of the host software provided by the operating system for managing devices.

10.1.2 Control Mechanisms

Control information may be passed between the host and a USB device using in-band or out-of-band signaling. In-band signaling mixes control information with data in a pipe outside the awareness of the host. Out-of-band signaling places control information in a separate pipe.

There is a message pipe called the default pipe for each attached USB device. This logical association between a host and a USB device is used for USB standard control flow such as device enumeration and configuration. The default pipe provides a standard interface to all USB devices. The default pipe may also be used for device-specific communications, as mediated by the USBD, which owns the default pipes of all of the USB devices.

A particular USB device may allow the use of additional message pipes to transfer device-specific control information. These pipes use the same communications protocol as the default pipe, but the information transferred is specific to the USB device and is not standardized by the USB Specification.

The USBD supports the sharing of the default pipe, which it owns and uses, with its clients. It also provides access to any other control pipes associated with the device.

10.1.3 Data Flow

The Host Controller is responsible for transferring streams of data between the host and USB devices. These data transfers are treated as a continuous stream of bytes. The USB supports four basic types of data transfers:

- Control transfers
- Isochronous transfers
- Interrupt transfers
- Bulk transfers

For additional information on transfer types, refer to Chapter 5.

Each device presents one or more interfaces that a client may use to communicate with the device. Each interface is composed of zero or more pipes that individually transfer data between the client and a particular endpoint on the device. The USBD establishes interfaces and pipes at the explicit request of the Host Software. The Host Controller provides service based on parameters provided by the Host Software when the configuration request is made.

A pipe has several characteristics based on the delivery requirements of the data to be transferred. Examples of these characteristics include the following:

- The rate at which data needs to be transferred
- Whether data is provided at a steady rate or sporadically
- How long data may be delayed before delivery
- Whether the loss of data being transferred is catastrophic

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A USB device endpoint describes the characteristics required for a specific pipe. Endpoints are described as part of a USB device's characterization information. For additional details, refer to Chapter 9.

10.1.4 Collecting Status and Activity Statistics

As a common communicant for all control and data transfers between the host and USB devices, the USB System and the Host Controller are well-positioned to track status and activity information. Such information is provided upon request to the Host Software, allowing that software to manage status and activity information. This specification does not identify any specific information that should be tracked or require any particular format for reporting activity and status information.

10.1.5 Electrical Interface Considerations

The host provides power to USB devices attached to the root hub. The amount of power provided by a port is specified in Chapter 7.

10.2 Host Controller Requirements

In all implementations, Host Controllers perform the same basic duties with regard to the USB and its attached devices. These basic duties are described below.

The Host Controller has requirements from both the host and the USB. The following is a brief overview of the functionality provided. Each capability is discussed in detail in subsequent sections.

State Handling	As a component of the host, the Host Controller reports and manages its states.
Serializer/Deserializer	For data transmitted from the host, the Host Controller converts protocol and data information from its native format to a bit stream transmitted on the USB. For data being received into the host, the reverse operation is performed.
(micro)frame Generation	The Host Controller produces SOF tokens at a period of 1 ms when operating with full-speed devices, and at a period of 125 µs when operating with high-speed devices.
Data Processing	The Host Controller processes requests for data transmission to and from the host.
Protocol Engine	The Host Controller supports the protocol specified by the USB.
Transmission Error Handling	All Host Controllers exhibit the same behavior when detecting and reacting to the defined error categories.
Remote Wakeup	All Host Controllers must have the ability to place the bus into the Suspended state and to respond to bus wakeup events.
Root Hub	The root hub provides standard hub function to link the Host Controller to one or more USB ports.
Host System Interface	Provides a high-speed data path between the Host Controller and host system.

The following sections present a more detailed discussion of the required capabilities of the Host Controller.

10.2.1 State Handling

The Host Controller has a series of states that the USB System manages. Additionally, the Host Controller provides the interface to the following two areas of USB-relevant state:

- State change propagation
- Root hub

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The root hub presents to the hub driver the same standard states as other USB devices. The Host Controller supports these states and their transitions for the hub. For detailed discussions of USB states, including their interrelations and transitions, refer to Chapter 9.

The overall state of the Host Controller is inextricably linked with that of the root hub and of the overall USB. Any Host Controller state changes that are visible to attached devices must be reflected in the corresponding device state change information such that the resulting Host Controller and device states are consistent.

USB devices request a wakeup through the use of resume signaling (refer to Chapter 7). The Host Controller must notify the rest of the host of a resume event through a mechanism or mechanisms specific to that system's implementation. The Host Controller itself may cause a resume event through the same signaling method.

10.2.2 Serializer/Deserializer

The actual transmission of data across the physical USB takes places as a serial bit stream. A Serial Interface Engine (SIE), whether implemented as part of the host or a USB device, handles the serialization and deserialization of USB transmissions. On the host, this SIE is part of the Host Controller.

10.2.3 Frame and Microframe Generation

It is the Host Controller's responsibility to partition USB time into quantities called "frames" when operating with full-speed devices, and "microframes" when operating with high-speed devices. Frames and microframes are created by the Host Controller through issuing Start-of-Frame (SOF) tokens as shown in Figure 10-3. The SOF token is the first transmission in the (micro)frame period. Host controllers operating with high-speed devices generate SOF tokens at 125 µs intervals. Host controllers operating with full-speed devices generate SOF tokens at 1.00 ms intervals. After issuing an SOF token, the Host Controller is free to transmit other transactions for the remainder of the (micro)frame period. When the Host Controller is in its normal operating state, SOF tokens must be continuously generated at appropriate periodic rate, regardless of other bus activity or lack thereof. If the Host Controller enters a state where it is not providing power on the bus, it must not generate SOFs. When the Host Controller is not generating SOFs, it may enter a power-reduced state.

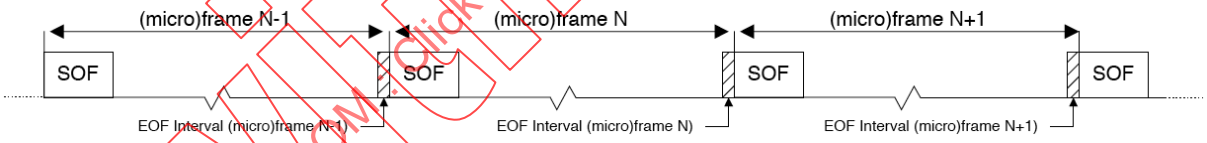


Figure 10-3. Frame and Microframe Creation

The SOF token holds the highest priority access to the bus. Babble circuitry in hubs electrically isolates any active transmitters during the End-of-microframe or End-of-Frame (EOF) interval, providing an idle bus for the SOF transmission.

The Host Controller maintains the current (micro)frame number that may be read by the USB System.

The following apply to the current (micro)frame number maintained by the host:

- Used to uniquely identify one (micro)frame from another
- Incremented at the end of every (micro)frame period
- Valid through the subsequent (micro)frame

Host controllers operating with full-speed devices maintain a current frame number (at least 11 bits) that increments at a 1 ms period. The host transmits the lower 11 bits of the current frame number in each SOF token transmission.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Host controllers operating with high-speed devices maintain a current microframe number (at least 14 bits) that increments at a 125 µs period. The host transmits bits 3 through 13 of the current microframe number in each SOF token transmission. This results in the same SOF packet value being transmitted for eight consecutive microframes before the SOF packet value increments.

When requested from the Host Controller, the current (micro)frame number is the (micro)frame number in existence at the time the request was fulfilled. The current (micro)frame number as returned by the host (Host Controller or HCD) is at least 32 bits, although the Host Controller itself is not required to maintain more than 11 bits when operating with full-speed devices or 14 bits when operating with high-speed devices.

The Host Controller shall cease transmission during the EOF interval. When the EOF interval begins, any transactions scheduled specifically for the (micro)frame that has just passed are retired. If the Host Controller is executing a transaction at the time the EOF interval is encountered, the Host Controller terminates the transaction.

10.2.4 Data Processing

The Host Controller is responsible for receiving data from the USB System and sending it to the USB and for receiving data from the USB and sending it to the USB System. The particular format used for the data communications between the USB System and the Host Controller is implementation specific, within the rules for transfer behavior described in Chapter 5.

10.2.5 Protocol Engine

The Host Controller manages the USB protocol level interface. It inserts the appropriate protocol information for outgoing transmissions. It also strips and interprets, as appropriate, the incoming protocol information.

10.2.6 Transmission Error Handling

The Host Controller must be capable of detecting the following transmission error conditions, which are defined from the host's point of view:

- Timeout conditions after a host-transmitted token or packet. These errors occur when the addressed endpoint is unresponsive or when the structure of the transmission is so badly damaged that the targeted endpoint does not recognize it.
- Data errors resulting in missing or invalid transmissions:
  - The Host Controller is unable to completely send or receive a packet for host specific reasons, for example, a transmission extending beyond EOF or a lack of resources available to the Host Controller.
  - An invalid CRC field on a received data packet.
- Protocol errors:
  - An invalid handshake PID, such as a malformed or inappropriate handshake
  - A false EOP
  - A bit stuffing error

For each bulk, control, and interrupt transaction, the host must maintain an error count tally. Errors result from the conditions described above, not as a result of an endpoint NAKing a request. This value reflects the number of times the transaction has encountered a transmission error. It is recommended that the error count not be incremented when there was an error due to host specific reasons (buffer underrun or overrun), and that whenever a transaction does not encounter a transmission error, the error count is reset to zero.

If the error count for a given transaction reaches three, the host retires the transfer. When a transfer is retired due to excessive errors, the last error type must be indicated. Isochronous

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

transactions are attempted only once, regardless of outcome, and, therefore, no error count is maintained for this type.

10.2.7 Remote Wakeup

If USB System wishes to place the bus in the Suspended state, it commands the Host Controller to stop all bus traffic, including SOFs. This causes all USB devices to enter the Suspended state. In this state, the USB System may enable the Host Controller to respond to bus wakeup events. This allows the Host Controller to respond to bus wakeup signaling to restart the host system.

10.2.8 Root Hub

The root hub provides the connection between the Host Controller and one or more USB ports. The root hub provides the same functionality for dealing with USB topology as other hubs (see Chapter 11), except that the hardware and software interface between the root hub and the Host Controller is defined by the specific hardware implementation.

10.2.8.1 Port Resets

Section 7.1.7.5 describes the requirements of a hub to ensure all upstream resume attempts are overpowered with a long reset downstream. Root hubs must provide an aggregate reset period of at least 50 ms. If the reset duration is controlled in hardware and the hardware timer is <50 ms, the USB System can issue several consecutive resets to accumulate the specified reset duration as described in Section 7.1.7.5.

10.2.9 Host System Interface

The Host Controller provides a high-speed bus-mastering interface to and from main system memory. The physical transfer between memory and the USB wire is performed automatically by the Host Controller. When data buffers need to be filled or emptied, the Host Controller informs the USB System.

10.3 Overview of Software Mechanisms

The HCD and the USBD present software interfaces based on different levels of abstraction. They are, however, expected to operate together in a specified manner to satisfy the overall requirements of the USB System (see Figure 10-2). The requirements for the USB System are expressed primarily as requirements for the USBDI. The division of duties between the USBD and the HCD is not defined. However, the one requirement of the HCDI that must be met is that it supports, in the specified operating system context, multiple Host Controller implementations.

The HCD provides an abstraction of the Host Controller and an abstraction of the Host Controller's view of data transfer across the USB. The USBD provides an abstraction of the USB device and of the data transfers between the client of the USBD and the function on the USB device. Overall, the USB System acts as a facilitator for transmitting data between the client and the function and as a control point for the USB-specific interfaces of the USB device. As part of facilitating data transfer, the USB System provides buffer management capabilities and allows the synchronization of the data transmittal to the needs of the client and the function.

The specific requirements for the USBDI are described later in this chapter. The exact functions that fulfill these requirements are described in the relevant operating system environment guide for the HCDI and the USBDI. The procedures involved in accomplishing data transfers via the USBDI are described in the following sections.

10.3.1 Device Configuration

Different operating system environments perform device configuration using different software components and different sequences of events. The USB System does not assume a specific operating system method. However, there are some basic requirements that must be fulfilled

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

by any USB System implementation. In some operating systems, existing host software provides these requirements. In others, the USB System provides the capabilities.

The USB System assumes a specialized client of the USBD, called a hub driver, that acts as a clearinghouse for the addition and removal of devices from a particular hub. Once the hub driver receives such notifications, it will employ additional host software and other USBD clients, in an operating system specific manner, to recognize and configure the device. This model, shown in Figure 10-4, is the basis of the following discussion.

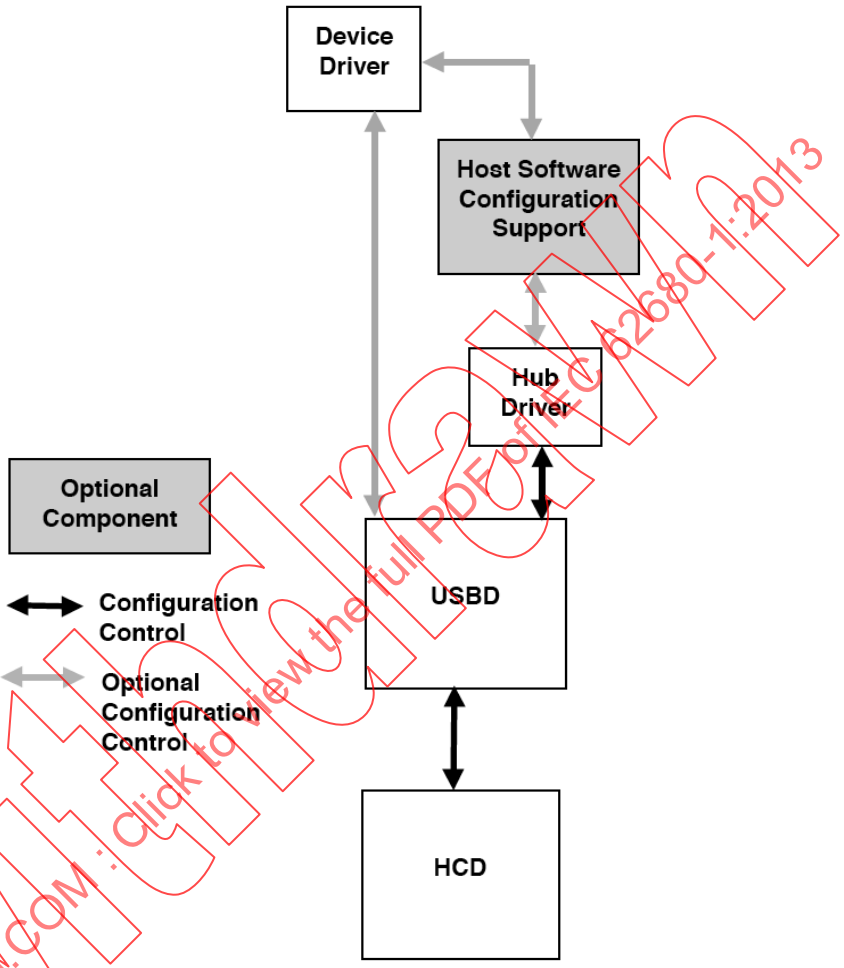


Figure 10-4. Configuration Interactions

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

When a device is attached, the hub driver receives a notification from the hub detecting the change. The hub driver, using the information provided by the hub, requests a device identifier from the USBD. The USBD in turn sets up the default pipe for that device and returns a device identifier to the hub driver.

The device is now ready to be configured for use. For each device, there are three configurations that must be complete before that device is ready for use:

1. Device Configuration: This includes setting up all of the device's USB parameters and allocating all USB host resources that are visible to the device. This is accomplished by setting the configuration value on the device. A limited set of configuration changes, such as alternate settings, is allowed without totally reconfiguring the device. Once the device is configured, it is, from its point of view, ready for use.
2. USB Configuration: In order to actually create a USBD pipe ready for use by a client, additional USB information, not visible to the device, must be specified by the client. This

information, known as the Policy for the pipe, describes how the client will use the pipe. This includes such items as the maximum amount of data the client will transfer with one IRP, the maximum service interval the client will use, the client's notification identification, and so on.

3. Function Configuration: Once configuration types 1 and 2 have been accomplished, the pipe is completely ready for use from the USB's point of view. However, additional vendor- or class-specific setup may be required before the client can actually use the pipe. This configuration is a private matter between the device and the client and is not standardized by the USBD.

The following paragraphs describe the device and USB configuration requirements.

The responsible configuring software performs the actual device configuration. Depending on the particular operating system implementation, the software responsible for configuration can include the following:

- The hub driver
- Other host software
- A device driver

The configuring software first reads the device descriptor, then requests the description for each possible configuration. It may use the information provided to load a particular client, such as a device driver, which initially interacts with the device. The configuring software, perhaps with input from that device driver, chooses a configuration for the device. Setting the device configuration sets up all of the endpoints on the device and returns a collection of interfaces to be used for data transfer by USBD clients. Each interface is a collection of pipes owned by a single client.

This initial configuration uses the default settings for interfaces and the default bandwidth for each endpoint. A USBD implementation may additionally allow the client to specify alternate interfaces when selecting the initial configuration. The USB System will verify that the resources required for the support of the endpoint are available and, if so, will allocate the bandwidth required. Refer to Section 10.3.2 for a discussion of resource management.

The device is now configured, but the created pipes are not yet ready for use. The USB configuration is accomplished when the client initializes each pipe by setting a Policy to specify how it will interact with the pipe. Among the information specified is the client's maximum service interval and notification information. Among the actions taken by the USB System, as a result of setting the Policy, is determining the amount of buffer working space required beyond the data buffer space provided by the client. The size of the buffers required is based upon the usage chosen by the client and upon the per-transfer needs of the USB System.

The client receives notifications when IRPs complete, successfully or due to errors. The client may also wake up independently of USB notification to check the status of pending IRPs.

The client may also choose to make configuration modifications, such as enabling an alternate setting for an interface or changing the bandwidth allocated to a particular pipe. In order to perform these changes, the interface or pipe, respectively, must be idle.

10.3.2 Resource Management

Whenever a pipe is setup by the USBD for a given endpoint, the USB System must determine if it can support the pipe. The USB System makes this determination based on the requirements stated in the endpoint descriptor. One of the endpoint requirements, which must be supported in order to create a pipe for an endpoint, is the bandwidth necessary for that endpoint's transfers. There are two stages to check for available bandwidth. First the

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

maximum execution time for a transaction is calculated. Then the (micro)frame schedule is consulted to determine if the indicated transaction will fit.

The allocation of the guaranteed bandwidth for isochronous and interrupt pipes, and the determination of whether a particular control or bulk transaction will fit into a given (micro)frame, can be determined by a software heuristic in the USB System. If the actual transaction execution time in the Host Controller exceeds the heuristically determined value, the Host Controller is responsible for ensuring that (micro)frame integrity is maintained (refer to Section 10.2.3). The following discussion describes the requirements for the USB System heuristic.

In order to determine if bandwidth can be allocated, or if a transaction can be fit into a particular (micro)frame, the maximum transaction execution time must be calculated. The calculation of the maximum transaction execution time requires that the following information be provided: (Note that an agent other than the client may provide some of this information.)

- Number of data bytes (*wMaxPacketSize*) to be transmitted.
- Transfer type.
- Depth in the topology. If less precision is allowed, the maximum topology depth may be assumed.

This calculation must include the bit transmission time, the signal propagation delay through the topology, and any implementation-specific delays, such as preparation or recovery time required by the Host Controller itself. Refer to Chapter 5 for examples of formulas that can be used for such calculations.

10.3.3 Data Transfers

The basis for all client-function communication is the interface: a bundle of related pipes associated with a particular USB device.

Exactly one client on the host manages a given interface. The client initializes each pipe of an interface by setting the Policy for that pipe. This includes the maximum amount of data to be transmitted per IRP and the maximum service interval for the pipe. A service interval is stated in milliseconds and describes the interval over which an IRP's data will be transmitted for an isochronous pipe. It describes the polling interval for an interrupt pipe. The client is notified when a specified request is completed. The client manages the size of each IRP such that its duty cycle and latency constraints are maintained. Additional Policy information includes the notification information for the client.

The client provides the buffer space required to hold the transmitted data. The USB System uses the Policy to determine the additional working space it will require.

The client views its data as a contiguous serial stream, which it manages in a similar manner to those streams provided over other types of bus technologies. Internally, the USB System may, depending on its own Policy and any Host Controller constraints, break the client request down into smaller requests to be sent across the USB. However, two requirements must be met whenever the USB System chooses to undertake such division:

- The division of the data stream into smaller chunks is not visible to the client.
- USB samples are not split across bus transactions.

When a client wishes to transfer data, it will send an IRP to the USBD. Depending on the direction of data transfer, a full or empty data buffer will be provided. When the request is complete (successfully or due to an error condition), the IRP and its status is returned to the client. Where relevant, this status is also provided on a per-transaction basis.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.3.4 Common Data Definitions

In order to allow the client to receive request results as directly as possible from its device, it is desirable to minimize the amount of processing and copying required between the device and the client. To facilitate this, some control aspects of the IRP are standardized such that different layers in the stack may directly use the information provided by the client. The particular format for this data is dependent on the actualization of the USBDI in the operating system. Some data elements may in fact not be directly visible to the client at all but are generated as a result of the client request.

The following data elements define the relevant information for a request:

- Identification of the pipe associated with the request. Identifying this pipe also describes information such as transfer type for this request.
- Notification identification for the particular client.
- Location and length of data buffer that is to be transmitted or received.
- Completion status for the request. Both the summary status and, as required, detailed per-transaction status must be provided.
- Location and length of working space. This is implementation-dependent.

The actual mechanisms used to communicate requests to the USBD are operating system-specific. However, beyond the requirements stated above for what request-related information must be available, there are also requirements on how requests will be processed. The basic requirements are described in Chapter 5. Additionally, the USBD provides a mechanism to designate a group of isochronous IRPs for which the transmission of the first transaction of each IRP will occur in the same (micro)frame. The USBD also provides a mechanism for designating an uninterruptable set of vendor- or class-specific requests to a default pipe. No other requests to that default pipe, including standard, class, or vendor request may be inserted in the execution flow for such an uninterruptable set. If any request in this set fails, the entire set is retired.

10.4 Host Controller Driver

The Host Controller Driver (HCD) is an abstraction of Host Controller hardware and the Host Controller's view of data transmission over the USB. The HCDI meets the following requirements:

- Provides an abstraction of the Host Controller hardware.
- Provides an abstraction for data transfers by the Host Controller across the USB interconnect.
- Provides an abstraction for the allocation (and de-allocation) of Host Controller resources to support guaranteed service to USB devices.
- Presents the root hub and its behavior according to the hub class definition. This includes supporting the root hub such that the hub driver interacts with the root hub exactly as it would for any hub. In particular, even though a root hub can be implemented in a combination of hardware and software, the root hub responds initially to the default device address (from a client perspective), returns descriptor information, supports having its device address set, and supports the other hub class requests. However, bus transactions may or may not need to be generated to accomplish this behavior given the close integration possible between the Host Controller and the root hub.

The HCD provides a software interface (HCDI) that implements the required abstractions. The function of the HCD is to provide an abstraction, which hides the details of the Host Controller hardware. Below the Host Controller hardware is the physical USB and all the attached USB devices.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The HCD is the lowest tier in the USB software stack. The HCD has only one client: the Universal Serial Bus Driver (USBD). The USBD maps requests from many clients to the appropriate HCD. A given HCD may manage many Host Controllers.

The HCDI is not directly accessible from a client. Therefore, the specific interface requirements for the HCDI are not discussed here.

10.5 Universal Serial Bus Driver

The USBD provides a collection of mechanisms that operating system components, typically device drivers, use to access USB devices. The only access to a USB device is that provided by the USBD. The USBD implementations are operating system-specific. The mechanisms provided by the USBD are implemented, using as appropriate and augmenting as necessary, the mechanisms provided by the operating system environment in which the USB runs. The following discussion centers on the basic capabilities required for all USBD implementations. For specifics of the USBD operation within a specific environment, see the relevant operating system environment guide for the USBD. A single instance of the USBD directs accesses to one or more HCDs that in turn connect to one or more Host Controllers. If allowed, how USBD instancing is managed is dependent upon the operating system environment. However, from the client's point of view, the USBD with which the client communicates manages all of the attached USB devices.

10.5.1 USBD Overview

Clients of USBD direct commands to devices or move streams of data to or from pipes. The USBD presents two groups of software mechanisms to clients: command mechanisms and pipe mechanisms.

Command mechanisms allow clients to configure and control USBD operation as well as to configure and generically control a USB device. In particular, command mechanisms provide all access to the device's default pipe.

Pipe mechanisms allow a USBD client to manage device specific data and control transfers. Pipe mechanisms do not allow a client to directly address the device's default pipe.

Figure 10-5 presents an overview of the USBD structure.

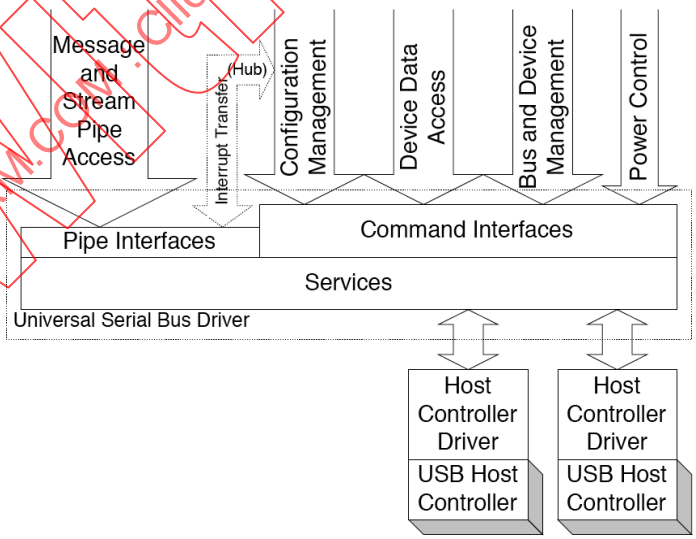


Figure 10-5. Universal Serial Bus Driver Structure

10.5.1.1 USBD Initialization

Specific USBD initialization is operating system-dependent. When a particular USB managed by USBD is initialized, the management information for that USB is also created. Part of this

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

management information is the default address device and its default pipe used to communicate to a newly reset device.

When a device is attached to a USB, it responds to a special address known as the default address (refer to Chapter 9) until its unique address is assigned by the bus enumerator. In order for the USB System to interact with the new device, the default device address and the device's default pipe must be available to the hub driver when a device is attached. During device initialization, the default address is changed to a unique address.

10.5.1.2 USBD Pipe Usage

Pipes are the method by which a device endpoint is associated with a Host Software entity. Pipes are owned by exactly one such entity on the host. Although the basic concept of a pipe is the same no matter who the owner, some distinction of capabilities provided to the USBD client occurs between two groups of pipes:

- Default pipes, which are owned and managed by the USBD
- All other pipes, which are owned and managed by clients of the USBD

Default pipes are never directly accessed by clients, although they are often used to fulfill some part of client requests relayed via command mechanisms.

10.5.1.2.1 Default Pipes

The USBD is responsible for allocating and managing appropriate buffering to support transfers on the default pipe that are not directly visible to the client such as setting a device address. For those transfers that are directly visible to the client, such as sending vendor and class commands or reading a device descriptor, the client must provide the required buffering.

10.5.1.2.2 Client Pipes

Any pipe not owned and managed by the USBD can be owned and managed by a USBD client. From the USBD viewpoint, a single client owns the pipe. In fact, a cooperative group of clients can manage the pipe, provided they behave as a single coordinated entity when using the pipe.

The client is responsible for providing the amount of buffering it needs to service the data transfer rate of the pipe within a service interval attainable by the client. Additional buffering requirements for working space are specified by the USB System.

10.5.1.3 USBD Service Capabilities

The USBD provides services in the following categories:

- Configuration via command mechanisms
- Transfer services via both command and pipe mechanisms
- Event notification
- Status reporting and error recovery

10.5.2 USBD Command Mechanism Requirements

USBD command mechanisms allow a client generic access to a USB device. Generally, these commands allow the client to make read or write accesses to one of potentially several device data and control spaces. The client provides as little as a device identifier and the relevant data or empty buffer pointer.

USBD command transfers do not require that the USB device be configured. Many of the device configuration facilities provided by the USBD are command transfers.

Following are the specific requirements on the command mechanisms provided.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.5.2.1 Interface State Control

USBD clients must be able to set a specified interface to any settable pipe state. Setting an interface state results in all of the pipes in that interface moving to that state. Additionally, all of the pipes in an interface may be reset or aborted.

10.5.2.2 Pipe State Control

USBD pipe state has two components:

- Host status
- Reflected endpoint status

Whenever the pipe status is reported, the value for both components will be identified. The pipe status reflected from the endpoint is the result of the endpoint being in a particular state. The USBD client manages the pipe state as reported by the USBD. For any pipe state reflected from the endpoint, the client must also interact with the endpoint to change the state.

A USBD pipe is in exactly one of the following states:

- Active: The pipe's Policy has been set and the pipe is able to transmit data. The client can query as to whether any IRPs are outstanding for a particular pipe. Pipes for which there are no outstanding IRPs are still considered to be in the Active state as long as they are able to accept new IRPs.
- Halted: An error has occurred on the pipe. This state may also be a reflection of the corresponding Halted endpoint on the device.

A pipe and endpoint are considered active when the device is configured and the pipe and/or endpoint is not stalled. Clients may manipulate pipe state in the following ways:

- Aborting a Pipe: All of the IRPs scheduled for a pipe are retired immediately and returned to the client with a status indicating they have been aborted. Neither the host state nor the reflected endpoint state of the pipe is affected.
- Resetting a Pipe: The pipe's IRPs are aborted. The host state is moved to Active. If the reflected endpoint state needs to be changed, that must be commanded explicitly by the USBD client.
- Clearing a Halted pipe: The pipe's state is cleared from *Halted* to *Active*.
- Halting a Pipe: The pipe's state is set to *Halted*.

10.5.2.3 Getting Descriptors

The USBDI must provide a mechanism to retrieve standard device, configuration, and string descriptors, as well as any class- or vendor-specific descriptors.

10.5.2.4 Getting Current Configuration Settings

The USBDI must provide a facility to return, for any specified device, the current configuration descriptor. If the device is not configured, no configuration descriptor is returned. This action is equivalent to returning the configuration descriptor for the current configuration by requesting the specific configuration descriptor. It does not, however, require the client to know the identifier for the current configuration. This will return all of the configuration information, including the following:

- All of the configuration descriptor information as stored on the device, including all of the alternate settings for all of the interfaces
- Indicators for which of the alternate settings for interfaces are active
- Pipe handles for endpoints in the active alternate settings for interfaces
- Actual *w*MaxPacketSize values for endpoints in the active alternate settings for interfaces

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Additionally, for any specified pipe, the USBDI must provide a facility to return the *wMaxPacketSize* that is currently being used by the pipe.

10.5.2.5 Adding Devices

The USBDI must provide a mechanism for the hub driver to inform USBD of the addition of a new device to a specified USB and to retrieve the USBD ID of the new USB device. The USBD tasks include assigning the device address and preparing the device's default pipe for use.

10.5.2.6 Removing Devices

The USBDI must provide a facility for the hub driver to inform the USBD that a specific device has been removed.

10.5.2.7 Managing Status

The USBDI must provide a mechanism for obtaining and clearing device-based status on a device, interface, or pipe basis.

10.5.2.8 Sending Class Commands

This USBDI mechanism is used by a client, typically a class-specific or adaptive driver, to send one or more class-specific commands to a device.

10.5.2.9 Sending Vendor Commands

This USBDI mechanism is used by a client to send one or more vendor-specific commands to a device.

10.5.2.10 Establishing Alternate Settings

The USBDI must provide a mechanism to change the alternate setting for a specified interface. As a result, the pipe handles for the previous setting are released and new pipe handles for the interface are returned. For this request to succeed, the interface must be idle; i.e., no data buffers may be queued for any pipes in the interface.

10.5.2.11 Establishing a Configuration

Configuring software requests a configuration by passing a buffer containing the configuration information to the USBD. The USBD requests resources for the endpoints in the configuration, and if all resource requests succeed, the USBD sets the device configuration and returns interface handles with corresponding pipe handles for all of the active endpoints. The default values are used for all alternate settings for interfaces.

Note: The interface implementing the configuration may require specific alternate settings to be identified.

10.5.2.12 Setting Descriptors

For devices supporting this behavior, the USBDI allows existing descriptors to be updated or new descriptors to be added.

10.5.3 USBD Pipe Mechanisms

This part of the USBDI offers clients the highest-speed, lowest overhead data transfer services possible. Higher performance is achieved by shifting some pipe management responsibilities from the USBD to the client. As a result, the pipe mechanisms are implemented at a more primitive level than the data transfer services provided by the USBD command mechanisms. Pipe mechanisms do not allow access to a device's default pipe.

USBD pipe transfers are available only after both the device and USB configuration have completed successfully. At the time the device is configured, the USBD requests the

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

resources required to support all device pipes in the configuration. Clients are allowed to modify the configuration, constrained by whether the specified interface or pipe is idle.

Clients provide full buffers to outgoing pipes and retrieve transfer status information following the completion of a request. The transfer status returned for an outgoing pipe allows the client to determine the success or failure of the transfer.

Clients provide empty buffers to incoming pipes and retrieve the filled buffers and transfer status information from incoming pipes following the completion of a request. The transfer status returned for an incoming pipe allows a client to determine the amount and the quality of the data received.

10.5.3.1 Supported Pipe Types

The four types of pipes supported, based on the four transfer types, are described in the following sections.

10.5.3.1.1 Isochronous Data Transfers

Each buffer queued for an isochronous pipe is required to be viewable as a stream of samples. As with all pipe transfers, the client establishes a Policy for using this isochronous pipe, including the relevant service interval for this client. Lost or missing bytes, which are detected on input, and transmission problems, which are noted on output, are indicated to the client.

The client queues a first buffer, starting the pipe streaming service. To maintain the continuous streaming transfer model used in all isochronous transfers, the client queues an additional buffer before the current buffer is retired.

The USBD is required to be able to provide a sample stream view of the client's data stream. In other words, using the client's specified method of synchronization, the precise packetization of the data is hidden from the client. Additionally, a given transaction is always contained completely within some client data buffer.

For an output pipe, the client provides a buffer of data. The USBD allocates the data across the (micro)frames for the service period using the client's chosen method of synchronization.

For an input pipe, the client must provide an empty buffer large enough to hold the maximum number of bytes the client's device will deliver in the service period. Where missing or invalid bytes are indicated, the USBD may leave the space that the bytes would have occupied in place in the buffer and identify the error. One of the consequences of using no synchronization method is that this reserved space is assumed to be the maximum packet size. The buffer-retired notification occurs when the IRP completes. Note that the input buffer need not be full when returned to the client.

The USBD may optionally provide additional views of isochronous data streams. The USBD is also required to be able to provide a packet stream view of the client's data stream.

10.5.3.1.2 Interrupt Transfers

The Interrupt out transfer originates in the client of the USBD and is delivered to the USB device. The Interrupt in transfer originates in a USB device and is delivered to a client of the USBD. The USB System guarantees that the transfers meet the maximum latency specified by the USB endpoint descriptor.

The client queues a buffer large enough to hold the interrupt transfer data (typically a single USB transaction). When all of the data is transferred, or if the error threshold is exceeded, the IRP is returned to the client.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.5.3.1.3 Bulk Transfers

Bulk transfers may originate either from the device or the client. No periodicity or guaranteed latency is assumed. When all of the data is transferred, or if the error threshold is exceeded, the IRP is returned to the client.

10.5.3.1.4 Control Transfers

All message pipes transfer data in both directions. In all cases, the client outputs a setup stage to the device endpoint. The optional data stage may be either input or output and the final status is always logically presented to the host. For details of the defined message protocol, refer to Chapter 8.

The client prepares a buffer specifying the command phase and any optional data or empty buffer space. The client receives a buffer-retired notification when all phases of the control transfer are complete, or an error notification, if the transfer is aborted due to transmission error.

10.5.3.2 USBD Pipe Mechanism Requirements

The following pipe mechanisms are provided.

10.5.3.2.1 Aborting IRPs

The USBDI must allow IRPs for a particular pipe to be aborted.

10.5.3.2.2 Managing Pipe Policy

The USBDI must allow a client to set and clear the Policy for an individual pipe or for an entire interface. Any IRPs made by the client prior to successfully setting a Policy are rejected by the USBD.

10.5.3.2.3 Queuing IRPs

The USBDI must allow clients to queue IRPs for a given pipe. When IRPs are returned to the client, the request status is also returned. A mechanism is provided by the USBD to identify a group of isochronous IRPs whose first transactions will all occur in the same (micro)frame.

10.5.4 Managing the USB via the USBD Mechanisms

Using the provided USBD mechanisms, the following general capabilities are supported by any USB System.

10.5.4.1 Configuration Services

Configuration services operate on a per-device basis. The configuring software tells the USBD when to perform device configuration. A hub driver has a special role in device management and provides at least the following capabilities:

- Device attach/detach recognition, driven by an interrupt pipe owned by the hub driver
- Device reset, accomplished by the hub driver by resetting the hub port upstream of the device
- Tells the USBD to perform device address assignment
- Power control

The USBDI additionally provides the following configuration facilities, which may be used by the hub driver or other configuring software available on the host:

- Device identification and access to configuration information (via access to descriptors on the device)
- Device configuration via command mechanisms

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

When the hub driver informs the USBD of a device attachment, the USBD establishes the default pipe for the new device.

10.5.4.1.1 Configuration Management

Configuration management services are provided primarily as a set of specific interface commands that generate USB transactions on the default pipe. The notable exception is the use of an additional interrupt pipe that delivers hub status directly to the hub driver.

Every hub initiates an interrupt transfer when there is a change in the state of one of the hub ports. Generally, the port state change will be the connection or removal of a downstream USB device. (Refer to Chapter 11 for more information.)

10.5.4.1.2 Initial Device Configuration

The device configuration process begins when a hub reports, via its status change pipe, the connection of a new USB device.

Configuration management services allow configuring software to select a USB device configuration from the set of configurations listed in the device. The USBD verifies that adequate power is available and the data transfer rates given for all endpoints in the configuration do not exceed the capabilities of the USB with the current schedule before setting the device configuration.

10.5.4.1.3 Modifying a Device Configuration

Configuration management services allow configuring software to replace a USB device configuration with another configuration from the set of configurations listed in the device. The operation succeeds if adequate power is available and the data transfer rates given for all endpoints in the new configuration fit within the capabilities of the USB with the current schedule. If the new configuration is rejected, the previous configuration remains.

Configuration management services allow configuring software to return a USB device to a Not Configured state.

10.5.4.1.4 Device Removal

Error recovery and/or device removal processing begins when a hub reports via its status change pipe that the USB device has been removed.

10.5.4.2 Power Control

There are two cooperating levels of power management for the USB: bus and device level management. This specification provides mechanisms for managing power on the USB bus. Device classes may define class-specific power control capabilities.

All USB devices must support the Suspended state (refer to Chapter 9). The device is placed into the Suspended state via control of the hub port to which the device is attached. Normal device operation ceases in the Suspend State; however, if the device is capable of wakeup signaling and the device is enabled for remote wakeup, it may generate resume signaling in response to external events.

The power management system may transition a device to the Suspended state or power-off the device in order to control and conserve power. The USB provides neither requirements nor commands for the device state to be saved and restored across these transitions. Device classes may define class-specific device state save-and-restore capabilities.

The USB System coordinates the interaction between device power states and the Suspended state.

It is recommended that while a device is not being used by the system (i.e., no transactions are being transmitted to or from the device besides SOF tokens), the device be suspended as

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

soon as possible by selectively suspending the port to which the device is attached. Suspending inactive devices reduces reliability issues due to high currents passing through a transceiver operating in high-speed mode in the presence of short circuit conditions described in Section 7.1.1. Some of these short circuit conditions are not detectable in the absence of transactions to the device. Suspending the unused device will place the transceiver interface into full-speed mode which has a greater reliability in the presence of short circuit conditions.

10.5.4.3 Event Notifications

USBD clients receive several kinds of event notifications through a number of sources:

- Completion of an action initiated by a client.
- Interrupt transfers over stream pipes can deliver notice of device events directly to USBD clients. For example, hubs use an interrupt pipe to deliver events corresponding to changes in hub status.
- Event data can be embedded by devices in streams.
- Standard device interface commands, device class commands, vendor-specific commands, and even general control transfers over message pipes can all be used to poll devices for event conditions.

10.5.4.4 Status Reporting and Error Recovery Services

The command and pipe mechanisms both provide status reporting on individual requests as they are invoked and completed.

Additionally, USB device status is available to USBD clients using the command mechanisms.

The USBD provides clients with pipe error recovery mechanisms by allowing pipes to be reset or aborted.

10.5.4.5 Managing Remote Wakeup Devices

The USB System can minimize the resume power consumption of a suspended USB tree. This is accomplished by explicitly enabling devices capable of resume signaling and controlling propagation of resume signaling via selectively suspending and/or disabling hub ports between the device and the nearest self-powered, awake hub.

In some error-recovery scenarios, the USB System will need to re-enumerate sub-trees. The sub-tree may be partially or completely suspended. During error-recovery, the USB System must avoid contention between a device issuing resume signaling and simultaneously driving reset down the port. Avoidance is accomplished via management of the devices' remote wakeup feature and the hubs' port features. The rules are as follows:

- Issue a SetDeviceFeature(DEVICE_REMOTE_WAKEUP) request to the leaf device, only just prior to selectively suspending any port between where the device is connected and the root port (via a SetPortFeature(PORT_SUSPEND) request).
- Do not reset a suspended port that has had a device enabled for remote wakeup without first enabling that port.
- Verify that after a remote wakeup, the devices in the subtree affected by the remote wakeup are still present. This will typically be done as part of determining which potential remote wakeup device was the source of the wakeup. This should be done to ensure that a suspended device is not disconnected (and possibly reconnected) or reset (e.g., by noise) during a suspend/resume process.

10.5.5 Passing USB Preboot Control to the Operating System

A single software driver owns the Host Controller. If the host system implements USB services before the operating system loads, the Host Controller must provide a mechanism that disables access by the preboot software and allows the operating system to gain control. Preboot USB configuration is not passed to the operating system. Once the operating system

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

gains control, it is responsible to fully configure the bus. If the operating system provides a mechanism to pass control back to the preboot environment, the bus will be in an unknown state. The preboot software should treat this event as a powerup.

10.6 Operating System Environment Guides

As noted previously, the actual interfaces between the USB System and host software are specific to the host platform and operating system. A companion specification is required for each combination of platform and operating system with USB support. These specifications describe the specific interfaces used to integrate the USB into the host. Each operating system provider for the USB System identifies a compatible Universal USB Specification revision.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11 Chapter 11 Hub Specification

This chapter describes the architectural requirements for the USB hub. It contains a description of the three principal sub-blocks: the Hub Repeater, the Hub Controller, and the Transaction Translator. The chapter also describes the hub's operation for error recovery, reset, and suspend/resume. The second half of the chapter defines hub request behavior and hub descriptors.

The hub specification supplies sufficient additional information to permit an implementer to design a hub that conforms to the USB specification.

11.1 Overview

Hubs provide the electrical interface between USB devices and the host. Hubs are directly responsible for supporting many of the attributes that make USB user friendly and hide its complexity from the user. Listed below are the major aspects of USB functionality that hubs must support:

- Connectivity behavior
- Power management
- Device connect/disconnect detection
- Bus fault detection and recovery
- High-, full-, and low-speed device support

A hub consists of three components: the Hub Repeater, the Hub Controller, and the Transaction Translator. The Hub Repeater is responsible for connectivity setup and tear-down. It also supports exception handling, such as bus fault detection and recovery and connect/disconnect detect. The Hub Controller provides the mechanism for host-to-hub communication. Hub-specific status and control commands permit the host to configure a hub and to monitor and control its individual downstream facing ports. The Transaction Translator responds to high-speed split transactions and translates them to full-/low-speed transactions with full-/low-speed devices attached on downstream facing ports.

11.1.1 Hub Architecture

Figure 11-1 shows a hub and the locations of its upstream and downstream facing ports. A hub consists of a Hub Repeater section, a Hub Controller section, and a Transaction Translator section. The hub must operate at high-speed when its upstream facing port is connected at high-speed. The hub must operate at full-speed when its upstream facing port is connected at full-speed.

The Hub Repeater is responsible for managing connectivity between upstream and downstream facing ports which are operating at the same speed. The Hub Repeater supports full-/low-speed connectivity and high-speed connectivity. The Hub Controller provides status and control and permits host access to the hub. The Transaction Translator takes high-speed split transactions and translates them to full-/low-speed transactions when the hub is operating at high-speed and has full-/low-speed devices attached. The operating speed of a device attached on a downstream facing port determines whether the Routing Logic connects a port to the Transaction Translator or hub repeater sections.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

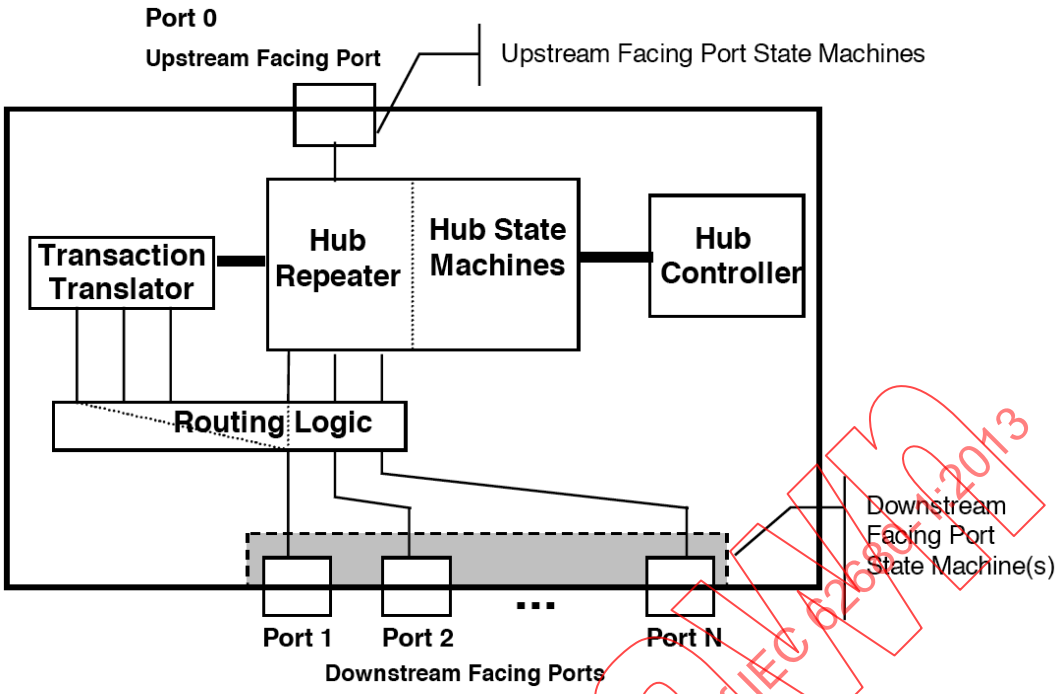


Figure 11-1. Hub Architecture

When a hub's upstream facing port is attached to an electrical environment that is operating at full-/low-speed, the hub's high-speed functionality is disallowed. This means that the hub will only operate at full-/low-speed and the transaction translator and high-speed repeater will not operate. In this electrical environment, the hub repeater must operate as a full-/low-speed repeater and the routing logic connects ports to the hub repeater.

When the hub upstream facing port is attached to an electrical environment that is operating at high-speed, the full-/low-speed hub repeater is not operational. In this electrical environment when a high-speed device is attached on downstream facing port, the routing logic will connect the port to the hub repeater and the hub repeater must operate as a high-speed repeater. In this case, when a full-/low-speed device is attached on a downstream facing port, the routing logic must connect the port to the transaction translator.

11.1.2 Hub Connectivity

Hubs exhibit different connectivity behavior depending on whether they are propagating packet traffic, or resume signaling, or are in the Idle state.

11.1.2.1 Packet Signaling Connectivity

The Hub Repeater contains one port that must always connect in the upstream direction (referred to as the upstream facing port) and one or more downstream facing ports. Upstream connectivity is defined as being towards the host, and downstream connectivity is defined as being towards a device. Figure 11-2 shows the packet signaling connectivity behavior for hubs in the upstream and downstream directions. A hub also has an Idle state, during which the hub makes no connectivity. When in the Idle state, all of the hub's ports are in the receive mode waiting for the start of the next packet.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

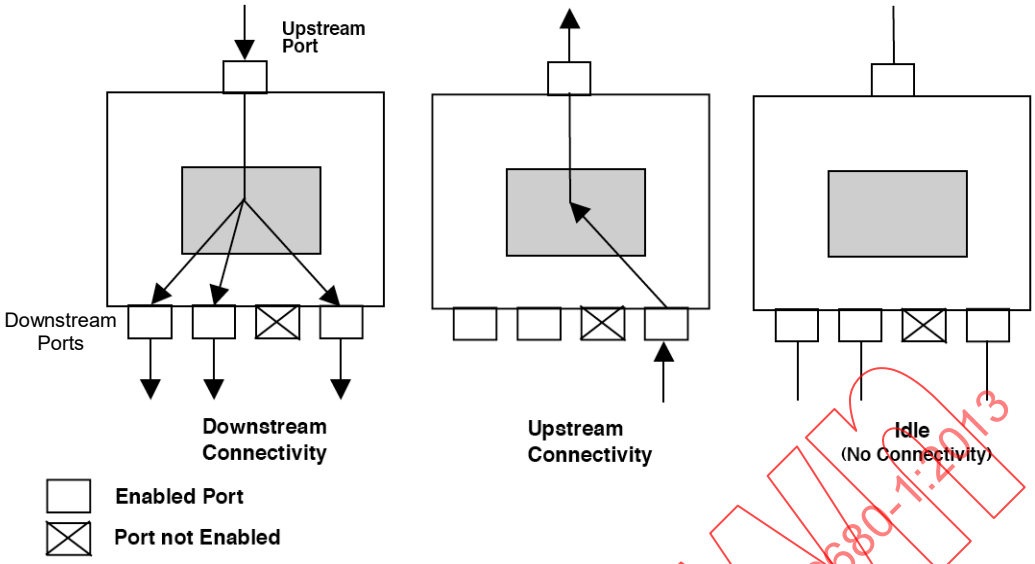


Figure 11-2. Hub Signaling Connectivity

If a downstream facing port is enabled (i.e., in a state where it can propagate signaling through the hub), and the hub detects the start of a packet on that port, connectivity is established in an upstream direction to the upstream facing port of that hub, but not to any other downstream facing ports. This means that when a device or a hub transmits a packet upstream, only those hubs in line between the transmitting device and the host will see the packet. Refer to Section 11.8.3 for optional behavior when a hub detects simultaneous upstream signaling on more than one port.

In the downstream direction, hubs operate in a broadcast mode. When a hub detects the start of a packet on its upstream facing port, it establishes connectivity to all enabled downstream facing ports. If a port is not enabled, it does not propagate packet signaling downstream.

11.1.2.2 Resume Connectivity

Hubs exhibit different connectivity behaviors for upstream- and downstream-directed resume signaling. A hub that is suspended reflects resume signaling from its upstream facing port to all of its enabled downstream facing ports. Figure 11-3 illustrates hub upstream and downstream resume connectivity.

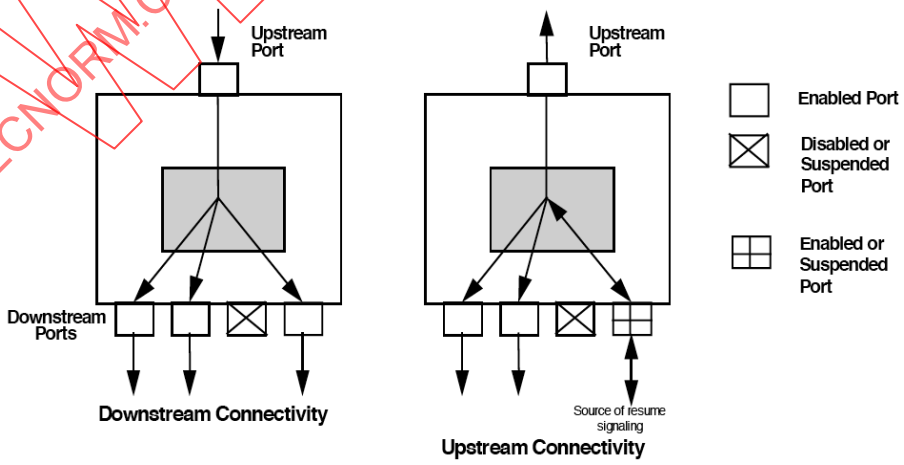


Figure 11-3. Resume Connectivity

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

If a hub is suspended and detects resume signaling from a selectively suspended or an enabled downstream facing port, the hub reflects that signaling upstream and to all of its enabled downstream facing ports, including the port that initiated the resume sequence. Resume signaling is not reflected to disabled or suspended ports. A detailed discussion of resume connectivity appears in Section 11.9.

11.1.2.3 Hub Fault Recovery Mechanisms

Hubs are the essential USB component for establishing connectivity between the host and other devices. It is vital that any connectivity faults, especially those that might result in a deadlock, be detected and prevented from occurring. Hubs need to handle connectivity faults only when they are in the repeater mode.

Hubs must also be able to detect and recover from lost or corrupted packets that are addressed to the Hub Controller. Because the Hub Controller is, in fact, another USB device, it must adhere to the same timeout rules as other USB devices, as described in Chapter 8.

11.2 Hub Frame/Microframe Timer

Each hub has a (micro)frame timer whose timing is derived from the hub's local clock and is synchronized to the host (micro)frame period by the host-generated Start-of-(micro)frame (SOF). The (micro)frame timer provides timing references that are used to allow the hub to detect a babbling device and prevent the hub from being disabled by the upstream hub. The hub (micro)frame timer must track the host (micro)frame period and be capable of remaining synchronized with the host even if two consecutive SOF tokens are missed by the hub.

The (micro)frame timer must lock to the host's (micro)frame timing for worst case clock accuracies and timing offsets between the host and hub. There are specific requirements for hubs when their upstream facing port is operating at high-speed and full-speed.

11.2.1 High-speed Microframe Timer Range

The range for a microframe timer must be from 59904 to 60096 high-speed bits.

The nominal microframe interval is 60000 high-speed bit times. The hub microframe timer range specified above is 60000 +/- 96 high-speed bit times in order to accommodate host accuracy, hub accuracy, repeater jittter, and hub quantization. The +/-96 full-speed bit time variation is calculated in Table 11-2.

Table 11-1. High-speed Microframe Timer Range Contributions

Source of Variation	Variation (ppm)	Variation (bits) Over One Microframe Interval	Comment
Host accuracy	+/- 500	+/- 30	
Hub accuracy	+/- 500	+/- 30	
Host jitter		+/- 2	
Hub chain jitter		+/- 20	Four hubs in series upstream of hub; 0 to 5 bits of jitter per hub
Quantization		+/-14	Bits need to round total variation to multiple of 16

11.2.2 Full-speed Frame Timer Range

The range of the frame timer must be from 11958 to 12042 full-speed bits.

The nominal frame interval is 12000 full-speed bit times. The hub frame timer range specified above is 12000 +/- 42 full-speed bit times in order to accommodate host accuracy and hub accuracy. The +/-42 full-speed bit time variation is calculated in Table 11-2.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 11-2. Full-speed Frame Timer Range Contributions

Source of Variation	Variation (ppm)	Variation (bits) Over One Frame Interval	Comment
Host accuracy	+/- 500	+/- 6	
Hub accuracy	+/- 3000	+/- 36	+/-6 bits due to hub accuracy (500 ppm) +/-30 bits due to 1.x parent hub accuracy (2500 ppm)

11.2.3 Frame/Microframe Timer Synchronization

A hub's (micro)frame timer is clocked by the hub's clock source and is synchronized to SOF packets that are derived from the host's (micro)frame timer. After a reset or resume, the hub's (micro)frame timer is not synchronized. Whenever the hub receives two consecutive SOF packets, its (micro)frame timer must be synchronized. Synchronized is synonymous with lock(ed). An example for a method of constructing a timer that properly synchronizes is as follows.

11.2.3.1 Example (Micro)frame Timer Synchronization Method

The hub maintains three timer values: (micro)frame timer (down counter), current (micro)frame (up counter), and next (micro)frame (register). After a reset or resume, a flag is set to indicate that the (micro)frame timer is not synchronized.

When the first SOF token is detected, the current (micro)frame timer resets and starts counting once per hub bit time. On the next SOF, if the timer has not rolled over, the value in the current (micro)frame timer is loaded into the next (micro)frame register and into the (micro)frame timer. The current (micro)frame timer is reset to zero and continues to count and the flag is set to indicate that the (micro)frame timer is locked. The (micro)frame timer rolls over when the count exceeds 60096 for high-speed or 12042 for full-speed (a test at 65535 for high-speed or 16383 for full-speed is adequate). If the current (micro)frame timer has rolled over, then an SOF was missed and the (micro)frame timer and next (micro)frame values are not loaded. When an SOF is missed, the flag indicating that the timer is not synchronized remains set.

Whenever the (micro)frame timer counts down to zero, the current value of the next (micro)frame register is loaded into the (micro)frame timer. When an SOF is detected, and the current (micro)frame timer has not rolled over, the value of the current (micro)frame timer is loaded into the (micro)frame timer and the next (micro)frame registers. The current (micro)frame timer is then reset to zero and continues to count. If the current (micro)frame timer has rolled over, then the value in the next (micro)frame register is loaded into the (micro)frame timer. This process can cause the (micro)frame timer to be updated twice in a single (micro)frame: once when the (micro)frame timer reaches zero and once when the SOF is detected.

11.2.3.2 EOF Advancement

The hub must advance its EOF points based on its SOF decode time in order to ensure that in the tiered topology, hubs farther away from the host will always have later EOF points than hubs nearer to the host. The magnitude of advance is implementation-dependent; the possible range of advance is derived below.

The synchronization circuit described above depends on successfully decoding an SOF packet identifier (PID). This means that the (micro)frame timer will be synchronized to a time that is later than the synchronization point in the SOF packet: later by at least 40 bit times for high-speed or 16 bit times for full-speed. Each implementation also takes some time to react to the SOF decode and set the appropriate timer/counter values. This reaction time is implementation-dependent but is assumed to be less than 192 bit times for high-speed and

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

four bit times for full-speed. Subsequent sections describe the actions that are controlled by the (micro)frame timer. These actions are defined at the EOF1, EOF2, and EOF. EOF1 and EOF2 are defined in later sections. These sections assume that the hub's (micro)frame timer will count to zero at the end of the (micro)frame (EOF). The circuitry described above will have the (micro)frame timer counting to zero after 40 to 192 for high-speed bit times or 16-20 full-speed bit times after the start of a (micro)frame (or end of previous (micro)frame). The timings and bit offsets in the later sections must be advanced to account for this delay (i.e., add 40-192 for high-speed or 16-20 bit times for full-speed to the EOF1 and EOF2 points).

Advancing the EOF points by the processing delay ensures that the spread between the EOFs is only due to the propagation delay. For example, for high-speed, the maximum spread between 2 EOF points anywhere on the USB is less than 216 bits (144 + 72). 144 bit times are due to 36 bit times of max latency through 4 repeaters. 72 bit times are due to five maximum cable and interconnect delays of 30 ns each. As can be seen in Figure 11-4 without EOF advancement, a hub with a larger tier number could have an EOF occuring earlier than a hub with a smaller tier number. In Figure 11-5 with EOF advancement ensures that in the tiered topology, hubs with larger tier numbers always have later EOF points than hubs with smaller tier numbers. Note: 13 bit times in the figures is an example maximum cable delay (approximately 30 ns).

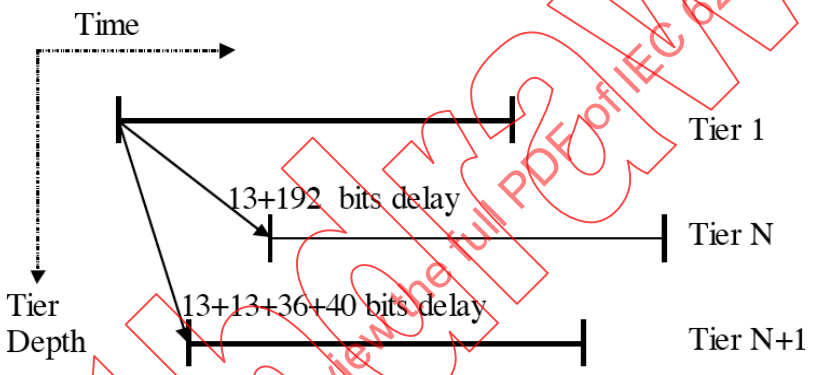


Figure 11-4. Example High-speed EOF Offsets Due to Propagation Delay Without EOF Advancement

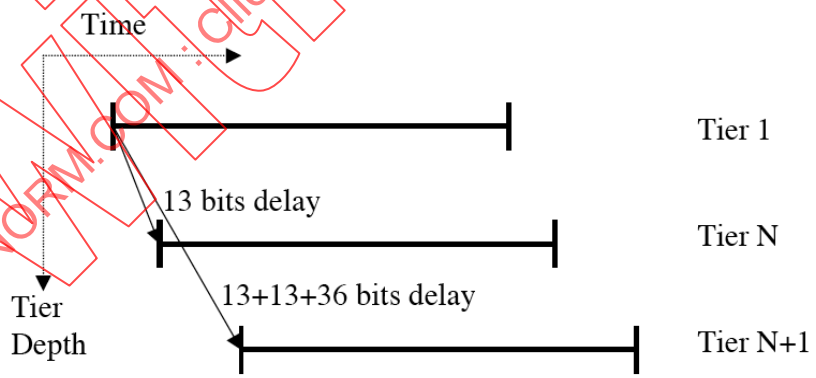


Figure 11-5. Example High-speed EOF Offsets Due to Propagation Delay With EOF Advancement

11.2.3.3 Effect of Synchronization on Repeater Behavior

The (micro)frame timer provides an indication to the hub Repeater state machine that the (micro)frame timer has synchronized to SOF and that the (micro)frame timer is capable of generating the EOF1 and EOF2 timing points. This signal is important after a global resume because of the possibility that a full-/low-speed device may have been detached, and a low-/full-speed device attached while the host was generating a long resume (several seconds) and the disconnect cannot be detected. The new device will bias D+ and D- to appear like a

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

K on the hub which would then be treated as an SOP and, unless inhibited, this SOP would propagate though the resumed hubs. Since the hubs would not have seen any SOFs at this point, the hubs would not be synchronized and, thus, unable to generate the EOF1 and EOF2 timing points. The only recovery from this would be for the host to reset and re-enumerate the section of the bus containing the changed device. This scenario is prevented by inhibiting any downstream facing port from establishing connectivity until the hub is locked after a resume.

11.2.4 Microframe Jitter Related to Frame Jitter

The period between the SOFs from the Transaction Translator must not vary by more than +/- 42 ns. The microframe timer count must be used by the Transaction Translator to generate SOFs to full-speed devices (and keepalives to low-speed devices) connected to it.

The SOF received at the upstream facing port of the hub is repeated with a local clock. The frequency of this clock may be a divided version of the bit rate. This could result in a quantization error and microframe-to-microframe jitter. The microframe-to-microframe jitter of a hub repeater must be between 0 and 5 bit times. This means that the latency through the repeater of consecutive SOFs must differ by less than 5 bits. A hub may register the SOF for internal use, e.g., microframe synchronization. This requires SOF PID detection. The circuitry used for internal registering of the SOF must have a jitter which is less than or equal to 16 bits. This means that the microframe timer count values between consecutive equally spaced SOFs must differ by less than or equal to 16 bits. The host controller frequency may drift over the period of a microframe resulting in microframe period jitter. The host controller source jitter for SOFs must be less than 4 bits. This means that the consecutive periods between SOFs must differ by less than 4 bits. These requirements ensure that the microframe period at the end of five hub tiers will have a jitter of less than 40 bits (4 from host controller + 4*5 from hub repeaters + 16 from the internal SOF registering). This means that the consecutive periods between SOFs as measured at any microframe timer will differ by less than 40 bits (83.3 ns at 480 Mbs). This is less than the +/- 42 ns variation allowed.

11.2.5 EOF1 and EOF2 Timing Points

The EOF1 and EOF2 are timing points that are derived from the hub's (micro)frame timer. Table 11-3 specifies the required host and hub EOF timing points for high-speed and full-speed operation.

Table 11-3. Hub and Host EOF1/EOF2 Timing Points

Label	Bit Times Before EOF for High-speed	Bit Times Before EOF for Full-speed	Notes
EOF1	560	32	End-of-(micro)frame point #1
EOF2	64	10	End-of-(micro)frame point #2

These timing points are used to ensure that devices and hubs do not interfere with the proper transmission of the SOF packet from the host. *These timing points have meaning only when the (micro)frame timer has been synchronized to the SOF.*

The host and hub (micro)frame markers, while all synchronized to the host's SOF, are subject to certain skews that dictate the placement of the EOF points. Figure 11-6 illustrates EOF2 timing point for high-speed operation. Figure 11-7 illustrates the EOF1 high-speed timing point. The numbers in the figures are in high-speed bit times.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

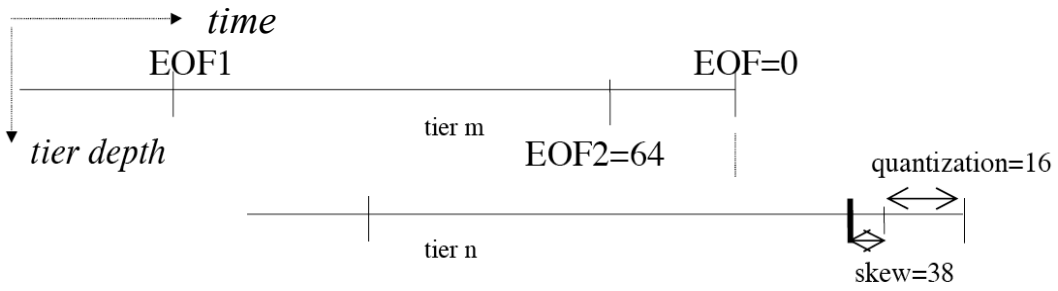


Figure 11-6. High-speed EOF2 Timing Point

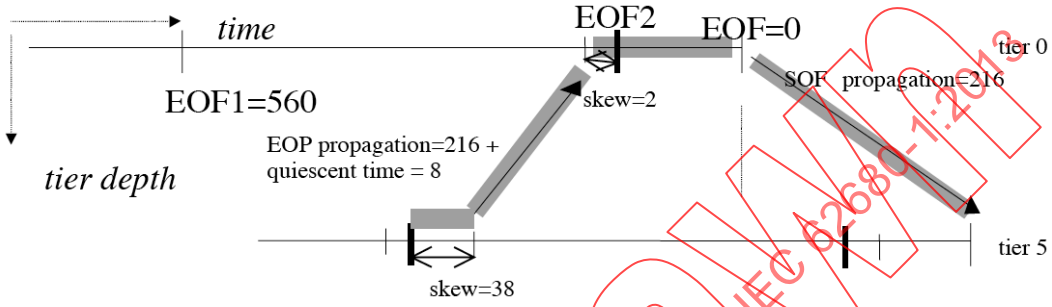


Figure 11-7. High-speed EOF1 Timing Point

At the EOF2 point, any port that has upstream connectivity will be disabled as a babbler. Hubs operating as a full-/low-speed repeater prevent becoming disabled by sending an end of packet to the upstream hub before that hub reaches its EOF2 point (i.e., at EOF1).

Figure 11-8 illustrates EOF timing points for full-/low-speed repeater operation.

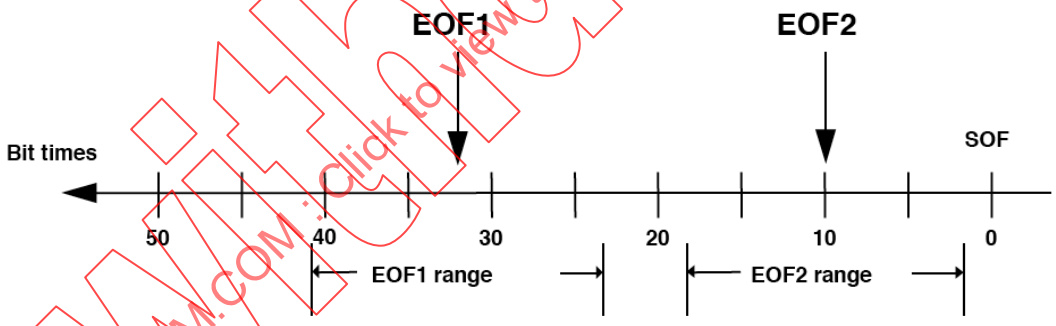


Figure 11-8. Full-speed EOF Timing Points

The hub operating as a full-/low-speed repeater is permitted to send the EOP if upstream connectivity is not established at EOF1 time. A full-speed repeater must send the EOP if connectivity is established from any downstream facing port at the EOF1 point.

A high-speed repeater must tear down upstream connectivity at the EOF1 point.

A high-speed repeater must tear down connectivity after the bus returns to the Idle state and the Elasticity buffer is emptied (as described in Section 11.7.2) rather than on decoding an EOP pattern as in full-/low-speed. Therefore, abrupt end of signaling (i.e, without a high-speed EOP) may cause malformed packets, and this must not affect repeater operation. The host controller design must be capable of processing such packets correctly.

11.2.5.1 High-speed EOF1 and EOF2 Timing Points

The EOF2 point is 64 bit times before EOF as shown in Figure 11-6, and the EOF1 point is 560 bit times before EOF as shown in Figure 11-7.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Although the hub is synchronized to the SOF, timing skew can accumulate between the host and a hub or between hubs. This timing skew represents the difference between different microframe timers on different hubs and the host. The total accumulated skew can be as much as 38 bit times. This is composed of ±2 bit times of (micro)frame host source jitter and 0 to 36 bit times of repeater jitter as derived earlier. This skew timing affects the placement of the EOF1 and EOF2 points.

Note: The hub skew timing assumes that the microframe interval will not be changed by the host after the microframe timers have synchronized.

EOF skew can be from –2 to + 38 bits, so all EOFs are within 256 bits (216 bits of EOF propagation delay + 40 bits of EOF skew) of each other.

Note: The EOF2 point is based on 16 bit times for quantization + 38 bit times of skew; therefore, the EOF2 point needs to located at least 54 bit times before EOF. The EOF2 point is set at 64 bit times to allow babble detection to be done with a divided (by 16) version of the bit clock. An upstream-directed packet ending before EOF1 must reach every upstream hub/host before it gets to its EOF2 point. This is achieved if the EOF1 point is located at least 544 bits before any upstream EOF (64 bits of EOF2 offset + 216 bits of EOP propagation delay + 8 bits of idle time + 216 bits of SOF propagation delay + 38 bits of EOF1 skew + 2 bits of EOF2 skew). The EOF1 point is set at 560 bit times to allow using a divided (by 16) version of the bit clock.

11.2.5.2 Full-speed EOF1 and EOF2 Timing Points

When the hub operates as a full-/low-speed repeater, the EOF1 point is 10 bit times before EOF and EOF1 is 32 bit times before EOF as shown in Figure 11-8.

The EOF2 point is defined to occur at least one bit time before the first bit of the SYNC for an SOP. The period allowed for an EOP is four full-speed bit times (the upstream facing port on a hub is always full-speed).

Although the hub is synchronized to the SOF, timing skew can accumulate between the host and a hub or between hubs. This timing skew represents the difference between different frame timers on different hubs and the host. The total accumulated skew can be as large as ±9 bit times. This is composed of ±1 bit times per frame of quantization error and ±1 bit per frame of wander. The quantization error occurs when the hub times the interval between SOFs and arrives at a value that is off by a fraction of a bit time but, due to quantization, is rounded to a full bit. Frame wander occurs when the host's frame timer is adjusted by the USB System Software so that the value sampled by the hub in a previous frame differs from the frame interval being used by the host. (Note: Such adjustment was permitted in the USB 1.0 and 1.1 specification but is no longer permitted.) These values accumulate over multiple frames because SOF packets can be lost and the hub cannot resynchronize its frame timer. This specification allows for the loss of two consecutive SOFs. During this interval, the quantization error accumulates to ±3 bit times, and the wander accumulates to ±1 ± 2 ± 3 = ±6 for a total of ±9 bit times of accumulated skew in three frames. This skew timing affects the placement of the EOF1 and EOF2 points as follows.

A hub must reach its EOF2 point one bit time before the end of the frame. In order to ensure this, a 9-bit time guard-band must be added so that the EOF2 point is set to occur when the hub's local frame timer reaches 10. A hub must complete its EOP before the hub to which it is attached reaches its EOF2 point. A hub may reach its EOF2 point nine bit times before bit time 10 (at bit time 19 before the SOF). To ensure that the EOP is completed by bit time 19, it must start before bit time 23. To ensure that the hub starts at bit time 23 with respect to another hub, a hub must set its EOF1 point nine bit times ahead of bit time 23 (at bit time 32). If a hub sets its timer to generate an EOP at bit time 32, that EOP may start as much as 9 bit times early (at bit time 41).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.3 Host Behavior at End-of-Frame

It is the responsibility of the USB host controller (the host) to not provoke a response from a device if the response would cause the device to be sending a packet at the EOF2 point. Furthermore, because a hub will terminate an upstream directed packet when the hub reaches its EOF1 point, the host should not start a transaction if a response from the device (data or handshake) would be pending or in process when a hub reaches its EOF1 point. The implications of these limitations are described in the following sections.

Note: The above requirements can be met if the host controller ensures that the last transaction will complete by its EOF1. The time consumed by a transaction (and consequently the latest start time of the transaction) can be evaluated by accumulating the various delay components in the transaction. The packet lengths should include all fields and account for bit-stuffing overhead as described in Chapter 7 and Chapter 8. Formulae for calculating transaction times are located in Section 5.11.3.

In defining the timing points below, the last bit interval in a (micro)frame is designated as bit time zero. Bit times in a (micro)frame that occur before the last have values that increase the further they are from bit time zero (earlier bit times have higher numbers). These bit time designations are used for convenience only and are not intended to imply a particular implementation. The only requirement of an implementation is that the relative time relationships be preserved.

Host controllers issuing high-speed transactions on a high-speed bus must meet the above requirements. Host controllers issuing full-/low-speed transactions on a full-/low-speed bus may also use the following three behaviors near EOF.

11.3.1 Full-/low-speed Latest Host Packet

Hubs are allowed to send an EOP on their upstream facing ports at the EOF1 point if there is no downstream-directed traffic in progress at that time. To prevent potential contention, the host is not allowed to start a packet if connectivity will not be established on all connections before a hub reaches its EOF1 point. This means that the host must not start a packet after bit time 42.

Note: Although there is as much as a six-bit time delay between the time the host starts a packet and all connections are established, this time need not be added to the packet start time as this phase delay exists for the SOF packet as well, causing all hub frame timers to be phase delayed with respect to the host by the propagation delay. There is only one bit time of phase delay between any two adjacent hubs and this has been accounted for in the skew calculations.

11.3.2 Full-/low-speed Packet Nullification

If a device is sending a packet (data or handshake) when a hub in the device's upstream path reaches its EOF1 point, the hub will send a full-speed EOP. Any packet that is truncated by a hub must be discarded.

A host implementation may discard any packet that is being received at bit time 41. Alternatively, a host implementation may attempt to maximize bus utilization by accepting a packet if the packet is predicted to start at or before bit time 41.

11.3.3 Full-/low-speed Transaction Completion Prediction

A device can send two types of packets: data and handshake. A handshake packet is always exactly 16 bit times long (sync byte plus PID byte.) The time from the end of a packet from the host until the first bit of the handshake must be seen at the host is 17 bit times. This gives a total allocation of 35 bit times from the end of data packet from the root (start of EOP) until it is predicted that the handshake will be completed (start of EOP) from the device. Therefore, if the host is sending a data packet for which the device can return a handshake (anything other than an isochronous packet), then if the host completes the data packet and

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

starts sending EOP before bit time 76, then the host can predict that the device will complete the handshake and start the EOP for the handshake on or before bit time 41. For a low-speed device, the 36 bit times from start of EOP from root to start of EOP from the device are low-speed bit times, which convert 1 to 8 into full-speed bit times. Therefore, if the host completes the low-speed data packet by bit time 329, then the low-speed device can be predicted to complete the handshake before bit time 41.

Note: If the host cannot accept a full-speed EOP as a valid end of a low-speed packet, then the low-speed EOP will need to complete before bit time 41, which will add 13 full-speed bit times to the low-speed handshake time.

As the host approaches the end of the frame, it must ensure that it does not require a device to send a handshake if that handshake cannot be completed before bit time 41. The host expects to receive a handshake after any valid, non-isochronous data packet. Therefore, if the host is sending a non-isochronous data packet when it reaches bit time 76 (329 for low-speed), then the host should start an abnormal termination sequence to ensure that the device will not try to respond. This abnormal termination sequence consists of 7 consecutive (non-bitstuffed) bits of 1 followed by an EOP. The abnormal termination sequence is sent at the speed of the current packet. Note: The intent of this sequence is to force a bitstuffing violation (and possibly other errors) at the receiver.

If the host is preparing to send an IN token, it may not send the token if the predicted packet from the device would not complete by bit time 41. The maximum valid length of the response from the device is known by the host and should be used in the prediction calculation. For a full-speed packet, the maximum interval between the start of the IN token and the end of a data packet is:

token_length + (packet_length + header + CRC) * 7/6 + 18

Where *token_length* is 34 bit times, *packet_length* is the maximum number of data bits in the packet, *header* is eight bits of sync and eight bits of PID, and CRC is 16 bits. The 7/6 multiplier accounts for the absolute worst case bit-stuff on the packet, and the 18 extra bits allow for worst case turn-around delay. For a low-speed device, the same calculation applies, but the result must be multiplied by 8 to convert to full-speed bit times, and an additional 20 full-speed bit times must be added to account for the low-speed prefix. This gives the maximum number of bit times between the start of the IN token and the end of the data packet, so the token cannot be sent if this number of bit times does not exist before the earliest EOF1 point (bit time 41). (For example, take the results of the above calculation and add 41. If the number of bits left in the frame is less than this value, the token may not be sent.)

The host is allowed to use a more conservative algorithm than the one given above for deciding whether or not to start a transaction. The calculation might also include the time required for the host to send the handshake when one is required, as there is no benefit in starting a transfer if the handshake cannot be completed.

11.4 Internal Port

The internal port is the connection between the Hub Controller and the Hub Repeater. Besides conveying the serial data to/from the Hub Controller, the internal port is the source of certain resume signals. Figure 11-9 illustrates the internal port state machine; Table 11-4 defines the internal port signals and events.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

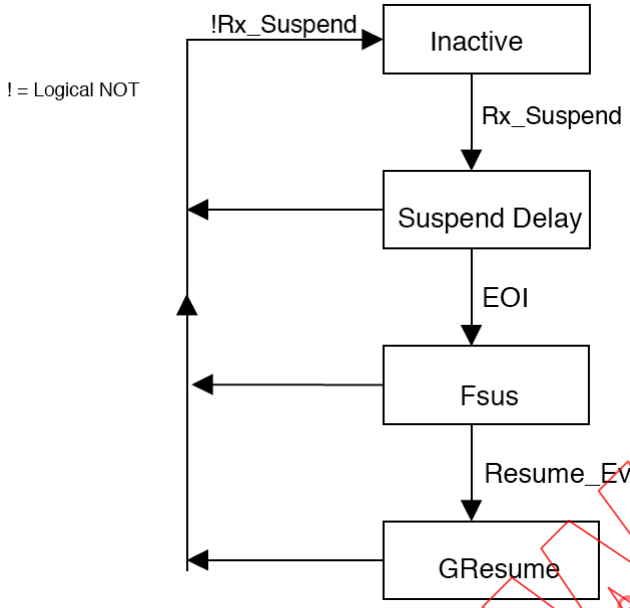


Figure 11-9. Internal Port State Machine

Table 11-4. Internal Port Signal/Event Definitions

Signal/Event Name	Event/Signal Source	Description
EOI	Internal	End of timed interval
Rx_Suspend	Receiver	Receiver is in the Suspend state
Resume_Event	Hub Controller	A resume condition exists in the Hub Controller

11.4.1 Inactive

This state is entered whenever the Receiver is not in the Suspend state.

11.4.2 Suspend Delay

This state is entered from the Inactive state when the Receiver transitions to the Suspend state.

This is a timed state with a 2 ms interval.

11.4.3 Full Suspend (Fsus)

This state is entered when the Suspend Delay interval expires.

11.4.4 Generate Resume (GResume)

This state is entered from the Fsus state when a resume condition exists in the Hub Controller. A resume condition exists if the C_PORT_SUSPEND bit is set in any port, or if the hub is enabled as a wakeup source and any bit is set in a Port Change field or the Hub Change field (as described in Figures 11-22 and 11-20, respectively).

In this state, the internal port generates signaling to emulate an SOP_FD to the Hub Repeater.

11.5 Downstream Facing Ports

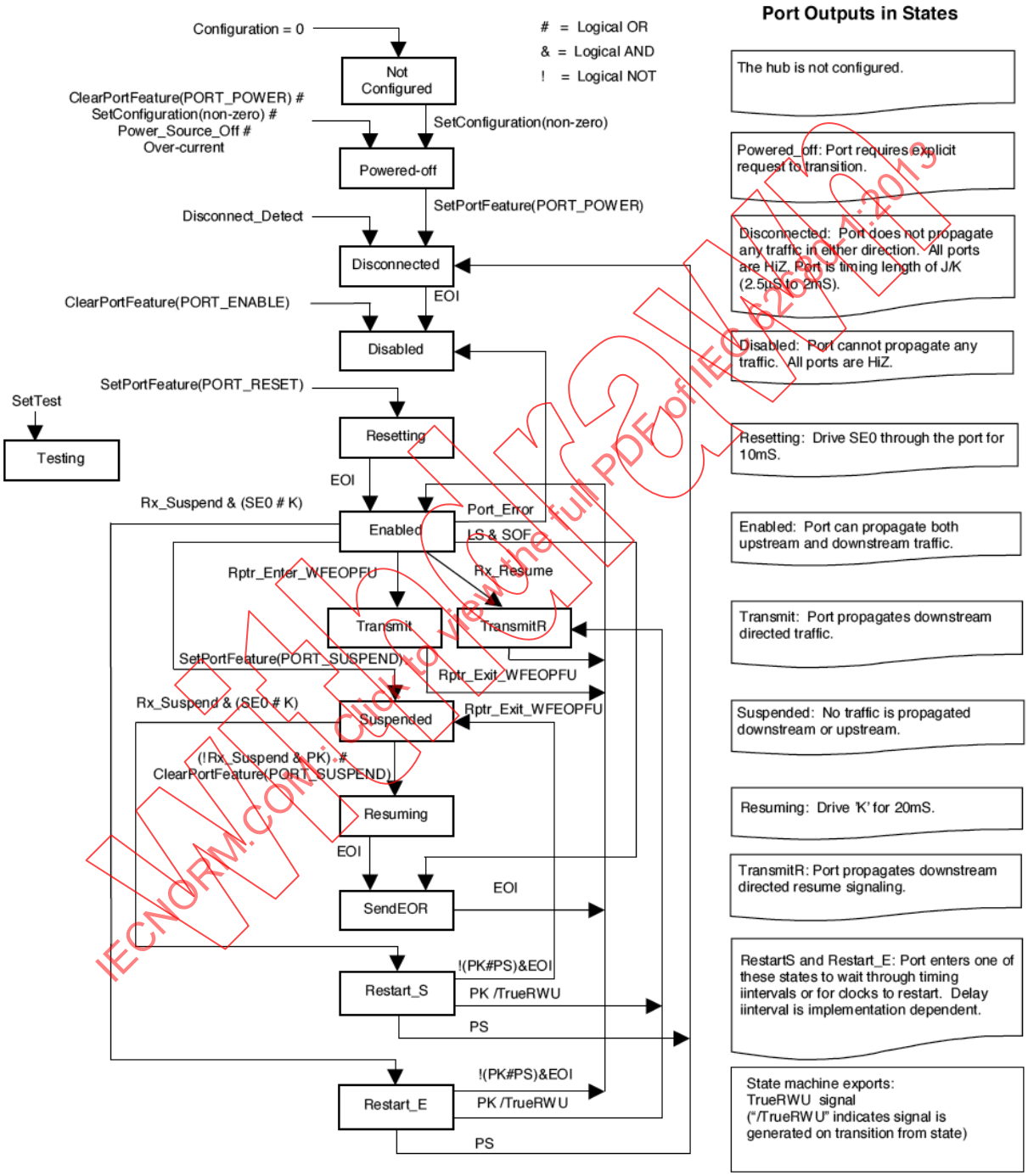
The following sections provide a functional description of a state machine that exhibits the correct behavior for a downstream facing port.

Figure 11-10 is an illustration of the downstream facing port state machine. The events and signals are defined in Table 11-5. Each of the states is described in Section 11.5.1. In the

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

diagram below, some of the entry conditions into states are shown without origin. These conditions have multiple origin states and the individual transitions lines are not shown so that the diagram can be simplified. The description of the entered state indicates from which states the transition is applicable.

Note: For the root hub, the signals from the upstream facing port state machines are implementation dependent.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 11-10. Downstream Facing Hub Port State Machine

Table 11-5. Downstream Facing Port Signal/Event Definitions

Signal/Event Name	Event/Signal Source	Description
Power_source_off	Implementation-dependent	Power to the port not available due to over-current or termination of source power (e.g., external power removed)
Over-current	Hub Controller	Over-current condition exists on the hub or the port
EOI	Internal	End of a timed interval or sequence
SE0	Internal	SE0 received on port
Disconnect_Detect	Internal	Disconnect seen at port
LS	Hub Controller	Low-speed device attached to this port
SOF	Hub Controller	SOF token received
TrueRWU	Internal	K lasting for at least TDDIS (see Table 7-13)
PK	Internal	K lasting for at least TDDIS
PS	Internal	SE0 lasting for at least TDDIS
K	Internal	'K' received on port
Rx_Resume	Receiver	Upstream Receiver in Resume state
Rx_Suspend	Receiver	Upstream Receiver in Suspend state
Rptr_Exit_WFEOPFU	Hub Repeater	Hub Repeater exits the WFEOPFU state
Rptr_Enter_WFEOPFU	Hub Repeater	Hub Repeater enters the WFEOPFU state
Port_Error	Internal	Error condition detected (see Section 11.8.1)
SetTest	Hub Controller	Logical OR of SetPortFeature(Test_SE0_NAK), SetPortFeature(Test_J), SetPortFeature(Test_K), SetPortFeature(Test_PRBS), SetPortFeature(Test_Force_Enable)
Configuration = 0	Hub Controller	Hub controller's configuration value is zero

11.5.1 Downstream Facing Port State Descriptions

11.5.1.1 Not Configured

A port transitions to and remains in this state whenever the value of the hub configuration is zero. While the port is in this state, the hub will drive an SE0 on the port (this behavior is optional on root hubs). No other active signaling takes place on the port when it is in this state.

11.5.1.2 Powered-off

This state is supported for all hubs.

A port transitions to this state in any of the following situations:

- From any state except Not Configured when the hub receives a ClearPortFeature(port_power) request for this port
- From any state when the hub receives a SetConfiguration() request with a configuration value other than zero
- From any state except Not Configured when power is lost to the port or an over-current condition exists

A port will enter this state due to an over-current condition on another port if that over-current condition may have caused the power supplied to this port to drop below specified limits for port power (see Section 7.2.1.2.1 and Section 7.2.4.1).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

If a hub was configured while the hub was self-powered, and then if external power is lost, the hub must place all ports in the Powered-off state. If the hub is configured while bus powered, then the hub need not change port status if the hub switched to externally applied power. However, if external power is subsequently lost, the hub must place ports in the Powered-off state.

In this state, the port's differential and single-ended transmitters and receivers are disabled.

Control of power to the port is covered in Section 11.11.

11.5.1.3 Disconnected

A port transitions to this state in any of the following situations:

- From the Powered-off state when the hub receives a SetPortFeature(PORT_POWER) request
- From any state except the Not Configured and Powered-off states when the port's disconnect timer times out
- From the Restart_S or Restart_E state at the end of the restart interval

In the Disconnected state, the port's differential transmitter and receiver are disabled and only connection detection is possible.

This is a timed state. While in this state, the timer is reset as long as the port's signal lines are in the SE0 or SE1 state. If another signaling state is detected, the timer starts. Unless the hub is suspended with clocks stopped, this timer's duration is 2.5 μs to 2 ms.

If the hub is suspended with its remote wakeup feature enabled, then on a transition to any state other than the SE0 state or SE1 state on a Disconnected port, the hub will start its clocks and time this event. The hub must be able to start its clocks and time this event within 12 ms of the transition. If a hub does not have its remote wakeup feature enabled, then transitions on a port that is in the Disconnected state are ignored until the hub is resumed.

11.5.1.4 Disabled

A port transitions to this state in any of the following situations:

- From the Disconnected state when the timer expires indicating a connection is detected on the port
- From any but the Powered-off, Disconnected, or Not Configured states on receipt of a ClearPortFeature(PORT_ENABLE) request
- From the Enabled state when an error condition is detected on the port

A port in the Disabled state will not propagate signaling in either the upstream or the downstream direction. While in this state, the duration of any SE0 received on the port is timed. If the port is using high-speed terminations when it enters this state, it switches to full-speed terminations. The port must not perform normal disconnect detection until at least 4 ms after entering this state.

11.5.1.5 Resetting

Unless it is in the Powered-off or Disconnected states, a port transitions to the Resetting state upon receipt of a SetPortFeature(PORT_RESET) request. The hub drives SE0 on the port during this timed interval. The duration of the Resetting state is nominally 10 ms to 20 ms (10 ms is preferred).

A hub in high-speed operation will use the high-speed terminations of the port when in this state.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.5.1.6 Enabled

A port transitions to this state in any of the following situations:

- At the end of the Resetting state
- From the Transmit state or the TransmitR state when the Hub Repeater exits the WFEOPFU state
- From the Suspended state if the upstream Receiver is in the Suspend state when a 'K' is detected on the port
- At the end of the SendEOR state
- From the Restart_E state when a persistent K or persistent SE0 has not been seen within 900 μs of entering that state

While in this state, the output of the port's differential receiver is available to the Hub Repeater so that appropriate signaling transitions can establish upstream connectivity.

A port which is using high-speed terminations in this state switches to full-speed terminations on Rx_Suspend (i.e., when the hub is suspended). The port must not perform normal disconnect detection until at least 1 ms after Rx_Suspend becomes active.

11.5.1.7 Transmit

This state is entered from the Enabled state on the transition of the Hub Repeater to the WFEOPFU state. While in this state, the port will transmit the data that is received on the upstream facing port.

For a low-speed port, this state is entered from the Enabled state if a full-speed PRE PID is received on the upstream facing port. While in this state, the port will retransmit the data that is received on the upstream facing port (after proper inversion).

In high-speed, this state is used for testing for disconnect at the port. The disconnect detection circuit is enabled after 32 bits of the same signaling level ('J' or 'K') have been transmitted down the port.

Note: Because of the timing skew in the repeater path to the downstream facing ports, all downstream facing ports may not be enabled for disconnect detection at the same instant in time.

11.5.1.8 TransmitR

This state is entered in either of the following situations:

- From the Enabled state if the upstream Receiver is in the Resume state
- From the Restart_S or Restart_E state if a PK is detected on the port

When in this state, the port repeats the resume 'K' at the upstream facing port to the downstream facing port. Depending on the speed of the port, two behaviors are possible on the K->SE0 transition at the upstream facing port at the end of the resume.

- Upstream facing port high-speed and downstream facing port full-/low-speed: After the K->SE0 transition, the port drives SE0 for 16 to 18 full-speed bit times followed by driving J for at least one full-speed bit time. Note: The timer in the Resume state of the upstream port receiver state machine which generates EOITR can be used to time this requirement at the downstream facing port(s). The pullup resistor and the latency of the Transaction Translator(TT) results in this Idle state being maintained for at least one low-speed bit time ensuring that a device sees the same end of resume behavior below the TT as it would below a USB 1.x hub.
- Upstream facing port and downstream facing port are the same speed: port continues to repeat the signaling which follows the K->SE0 transition.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A port operating in high-speed reverts to its high-speed terminations within 18 full-speed bit times after the K->SE0 transition as described in Section 7.1.7.7.

11.5.1.9 Suspended

A port enters the Suspended state:

- From the Enabled state when it receives a SetPortFeature(port_suspend) request
- From the Restart_S state when a persistent K or persistent SE0 has not been seen within 900 μs of entering that state

While a port is in the Suspended state, the port's differential transmitter is disabled. A high-speed port reverts from high-speed to full-speed terminations but its speed status continues to be high-speed. The port must not perform normal disconnect detection until at least 4 ms after entering this state.

An implementation must have a K/SE0 'noise' filter for a port that is in the suspended state. This filter can time the length of K/SE0 and, if the length of the K/SE0 is shorter than TDDIS, the port must remain in this state. If the hub is suspended with its clocks stopped, a transition to K/SE0 on a suspended port must cause the port to immediately transition to the Restart_S state.

11.5.1.10 Resuming

A port enters this state from the Suspended state in either of the following situations:

- If a 'K' is detected on the port and persists for at least 2.5 μs and the Receiver is not in the Suspended state. The transition from the Suspended state must happen within 900 μs of the J->K transition.
- When a ClearPortFeature(PORT_SUSPEND) request is received.

This is a timed state with a nominal duration of 20 ms (the interval may be longer under the conditions described in the note below). While in this state, the hub drives a 'K' on the port.

Note: A single timer is allowed to be used to time both the Resetting interval and the Resuming interval and that timer may be shared among multiple ports. When shared, the timer is reset when a port enters the Resuming state or the Resetting state. If shared, it may not be shared among more than ten ports as the cumulative delay could exceed the amount of time required to replace a device and a disconnect could be missed.

11.5.1.11 SendEOR

This state is entered from the Resuming state if the 20 ms timer expires. It is also entered from the Enabled state when an SOF (or other FS token) is received and a low-speed device is attached to this port.

This is a timed state which lasts for three low-speed bit times.

In this state, if the port is high-speed it will drive the bus to the Idle state for three low-speed bit times and then exit from this state to the Enabled state. It must also revert to its high-speed terminations within 18 full-speed bit times after the K->SE0 transition as described in Section 7.1.7.7.

If the port is full-speed or low-speed, the port must drive two low-speed bit times of SE0 followed by one low-speed bit time of Idle state and then exit from this state to the Enabled state.

Since the driven SE0 period should be of fixed length, the SendEOR timer, if shared, should not be reset. If the hub implementation shares the SendEOR timing circuits between ports, then for a port with a low-speed device attached, the Resuming state should not end until an

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

SOF (or other FS token) has been received (see Section 11.8.4.1 for Keep-alive generation rules).

11.5.1.12 Restart_S

A port enters the Restart_S state from the Suspended state when an SE0 or 'K' is seen at the port and the Receiver is in the Suspended state.

In this state, the port continuously monitors the bus state. If the bus is in the 'K' state for at least TDDIS, the port sets the C_PORT_SUSPEND bit, exits to the TransmitR, and generates a signal to the repeater called 'TrueRWU'. If the bus is in the 'SE0' state for at least TDDIS, the port exits to the Disconnected state. Either of these transitions must happen within 900 μs after entering the Restart_S state; otherwise, the port must transition back to the Suspended state.

11.5.1.13 Restart_E

A port enters the Restart_E state from the Enabled state when an 'SE0' or 'K' is seen at the port and the Receiver is in the Suspended state.

In this state, the port continuously monitors the bus state. If the bus is in the 'K' state for at least TDDIS, the port exits to the TransmitR state and generates a signal to the repeater called 'TrueRWU'. If the bus is in the 'SE0' state for at least TDDIS, the port exits to the Disconnected state. Either of these transitions must happen within 900 μs after entering the Restart_E state; otherwise the port must transition back to the Enabled state.

11.5.1.14 Testing

A port transitions to this state from any state when the port sees SetTest.

While in this state, the port executes the host command as decoded by the hub controller. If the command was a SetPortFeature(PORT_TEST, Test_Force_Enable), the port supports packet connectivity in the downstream direction in a manner identical to that when the port is in the Enabled state.

11.5.2 Disconnect Detect Timer

11.5.2.1 High-speed Disconnect Detection

High-speed disconnect detection is described in Section 7.1.7.3.

11.5.2.2 Full-/low-speed Disconnect Detection

Each port is required to have a timer used for detecting disconnect when a full-/low-speed device is attached to the port. This timer is used to constantly monitor the port's single-ended receivers to detect a disconnect event. The reason for constant monitoring is that a noise event on the bus can cause the attached device to detect a reset condition on the bus after 2.5 μs of SE0 or SE1 on the bus. If the hub does not place the port in the disconnect state before the device resets, then the device can be at the Default Address state with the port enabled. This can cause systems errors that are very difficult to isolate and correct.

This timer must be reset whenever the D+ and D- lines on the port are not in the SE0 or SE1 state or when the port is not in the Enabled, Suspended, Disabled, Restart-E, or Restart_S states. This timer must be reset for 4ms upon entry to the Suspended and Disabled states. This timer times an interval TDDIS. The range of TDDIS is 2.0 μs to 2.5 as defined in Table 7-13. When this timer expires, it generates the Disconnect_Detect signal to the port state machine.

This timer can also be used for filtering the K/SE0 signal in the Suspended, Restart_E, or Restart_S states as described in Section 11.5.1.

IECNORM.COM : Click to view the full PDF of iec 62680-1:2013

11.5.3 Port Indicator

Each downstream facing port of a hub can support an optional status indicator. The presence of indicators for downstream facing ports is specified by bit 7 of the *wHubCharacteristics* field of the hub class descriptor. Each port's indicator must be located in a position that obviously associates the indicator with the port. The indicator provides two colors: green and amber. This can be implemented as physically one LED with two color capability or two separate LEDs. A combination of hardware and software control is used to inform the user of the current status of the port or the device attached to the port and to guide the user through problem resolution. Colors and blinking are used to provide information to the user.

An external hub must automatically control the color of the indicator as specified in Figure 11-11. Automatic port indicator setting support for root hubs may be implemented with either hardware or software. The port indicator color selector value is zero (indicating automatic control) when the hub transitions to the configured device state. When the hub is suspended or not configured, port indicators must be off.

Table 11-6 identifies the mapping of color to port state when the port indicators are automatically controlled.

Table 11-6. Automatic Port State to Port Indicator Color Mapping

Power Switching	Downstream Facing Hub Port State			
	Powered-off	Disconnected, Disabled, Not Configured, Resetting, Testing	Enabled, Transmit, or TransmitR	Suspended, Resuming, SendEOR, Restart_E, or Restart_S
With	Off or amber if due to an over-current condition	Off	Green	Off
Without	Off	Off or amber if due to an over-current condition	Green	Off

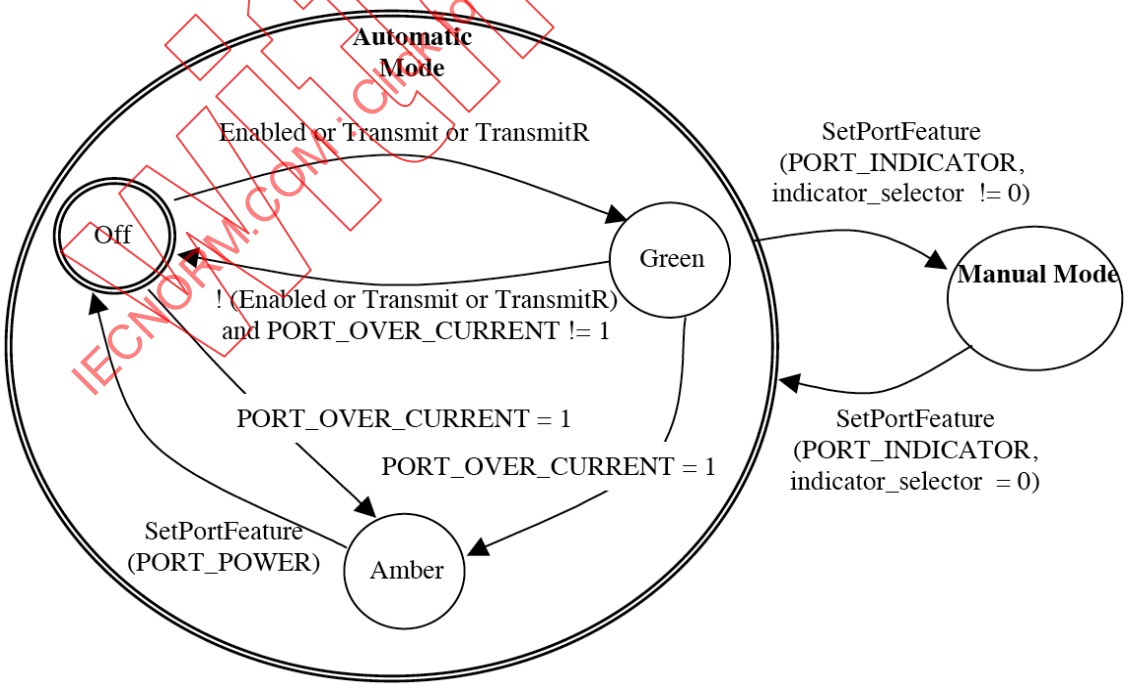


Figure 11-11. Port Indicator State Diagram

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

In **Manual Mode** the color of a port indicator (Amber, Green, or Off) is set by a system software USB Hub class request. In **Automatic Mode** the color of a port indicator is set by the port state information.

Table 11-7 defines port state as understood by the user.

Table 11-7. Port Indicator Color Definitions

Color	Definition
Off	Not operational
Amber	Error condition
Green	Fully operational
Blinking Off/Green	Software attention
Blinking Off/Amber	Hardware attention
Blinking Green/Amber	Reserved

Note that the indicators reflect the status of the port, not necessarily the device attached to it. Blinking of the indicator is used to draw the user's attention to the port, irrespective of its color.

Port indicators allow control by software. Host software forces the state of the indicator to draw attention to the port or to indicate the current state of the port.

See Section 11.24.2.7.1.10 for the specification of indicator requests.

11.5.3.1 Labeling

USB system software uses port numbers to reference an individual port with a ClearPortFeature or SetPortFeature request. If a vendor provides a labeling to identify individual downstream facing ports, then each port connector must be labeled with their respective port number.

11.6 Upstream Facing Port

The upstream facing port has four components: transmitter, transmitter state machine, receiver, and receiver state machine. The transmitter and its state machine are the Transmitter, while the receiver and its state machine are the Receiver. The Transmitter and Receiver operate in high-speed and full-speed depending on the current hub configuration.

11.6.1 Full-speed

Both the transmitter and receiver have differential and single-ended components. The differential transmitter and receiver can send/receive 'J' or 'K' to/from the bus while the single-ended components are used to send/receive SE0, suspend, and resume signaling. The single-ended components are also used to receive SE1. In this section, when it is necessary to differentiate the signals sent/received by the differential component of the transmitter/receiver from those of the single-ended components, DJ and DK will be used to denote the differential signal, while SJ, SK, SE0, and SE1 will be used for the single-ended signals.

When the Hub Repeater has connectivity in the upstream direction, the transmitter must not send or propagate SE1 signaling. Instead, the SE1 must be propagated as a DJ.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.6.2 High-speed

Both the transmitter and receiver have differential components only. These signals are called HJ and HK. The HS_Idle state is the idle state of the bus in high-speed.

It is assumed that the differential transmitter and receiver are turned off during suspend to minimize power consumption. The single-ended components are left on at all times, as they will take minimal power.

11.6.3 Receiver

The receiver state machine is responsible for monitoring the signaling state of the upstream connection to detect long-term signaling events such as bus reset, resume, and suspend. This state machine details the operation of the device state diagram shown in Figure 9-1 in the Default, Address, Configured, and Suspended state. The Suspend, Resume, and ReceivingSE0 states are only used when the upstream facing port is operating in full-speed mode with full-speed terminations. The ReceivingIS, ReceivingHJ, and ReceivingHK states are only used when the upstream facing port is operating in high-speed mode with high-speed terminations; so these states are categorized as the HS (high-speed) states, and all other states are categorized as nonHS in the description below.

Figure 11-12 illustrates the state transition diagram.

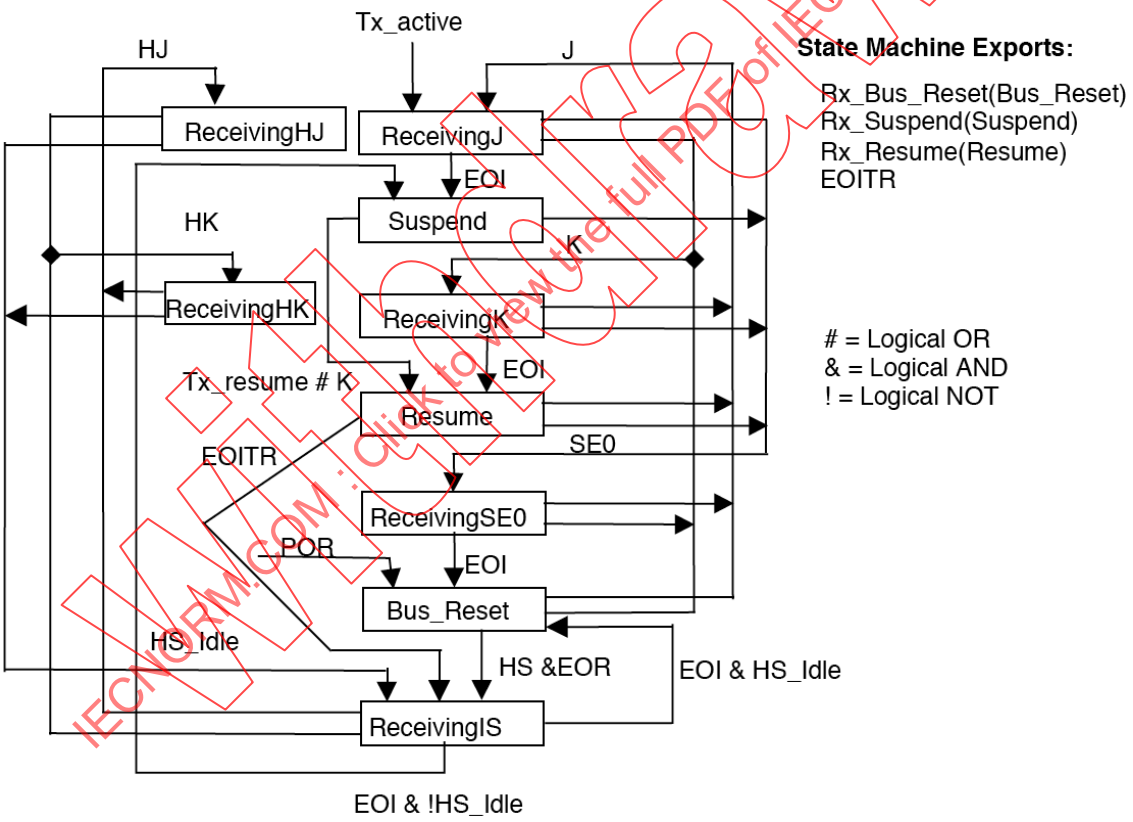


Figure 11-12. Upstream Facing Port Receiver State Machine

Table 11-8 defines the signals and events referenced in the figures.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 11-8. Upstream Facing Port Receiver Signal/Event Definitions

Signal/Event Name	Event/Signal Source	Description
HS	Internal	Port is operating in high-speed
Tx_active	Transmitter	Transmitter in the Active state
J	Internal	Receiving a 'J' (IDLE) or an 'SE1' on the upstream facing port
HJ	Internal	Receiving an HJ on the upstream facing port
EOI	Internal	End of timed interval
EOITR	Internal	Generated 24 full-speed bit times after the K->SE0 transition at the end of resume
HK, K	Internal	Receiving an HK, 'K' on the upstream facing port
Tx_resume	Transmitter	Transmitter is in the Sresume state
HS_Idle	Internal	Receiving an Idle state on the high-speed upstream facing port
SE0	Internal	Receiving an SE0 on the full-speed upstream facing port
EOR	Internal	End of Reset signaling from upstream
POR	Implementation-dependent	Power_On_Reset

11.6.3.1 ReceivingIS

This state is entered

- From the ReceivingHJ or ReceivingHK state when a SE0 is seen at the port and the port is in high-speed operation
- From the Resume state when a EOITR is seen and the port is in high-speed operation
- From the Bus Reset state at the End of Reset signaling from upstream when the port is in high-speed operation

This is a timed state with an interval of 3 ms. The timer is reset each time this state is entered.

11.6.3.2 ReceivingHJ

This state is entered from an HS state when a HJ is seen on the bus.

11.6.3.3 ReceivingJ

This state is entered from a nonHS state except the Suspend state if the receiver detects an SJ (or Idle) or SE1 condition on the bus or while the Transmitter is in the Active state.

This is a timed state with an interval of 3 ms. The timer is reset each time this state is entered.

The timer only advances if the Transmitter is in the Inactive state.

11.6.3.4 Suspend

This state is entered when:

- The 3 ms timer expires in the ReceivingJ
- The 3 ms timer expires in the ReceivingIS state and the port has removed its high-speed terminations and connected its D+ pull-up resistor and the resulting bus state is not SE0.

When the Receiver enters this state, the Hub Controller starts a 2 ms timer. If that timer expires while the Receiver is still in this state, then the Hub Controller is suspended. When the Hub Controller is suspended, it may generate resume signaling.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.6.3.5 ReceivingHK

This state is entered from an HS state when a HK is seen on the bus.

11.6.3.6 ReceivingK

This state is entered from any nonHS state except the Resume state when the receiver detects an SK condition on the bus and the Hub Repeater is in the WFSOP or WFSOPFU state.

This is a timed state with a duration of 2.5 μs to 100 μs. The timer is reset each time this state starts.

11.6.3.7 Resume

This state is entered:

- From the ReceivingK state when the timer expires
- From the Suspend state while the Transmitter is in the Sresume state or if there is a transition to the K state on the upstream facing port

If the hub enters this state when its timing reference is not available, the hub may remain in this state until the hub's timing reference becomes stable (timing references must stabilize in less than 10 ms). If this state is being held pending stabilization of the hub's clock, the Receiver must provide a K to the repeater for propagation to the downstream facing ports. When clocks are stable, the Receiver must repeat the incoming signals.

Note: Hub timing references will be stable in less than 10 ms since reset requirements already specify that they be stable in less than 10 ms and a hub must support reset from suspend.

11.6.3.8 ReceivingSE0

This state is entered from any nonHS state except Bus_Reset when the receiver detects an SE0 condition and the Hub Repeater is in the WFSOP or WFSOPFU state.

This is a timed state. The minimum interval for this state is 2.5 μs. The maximum depends on the hub but this interval must timeout early enough such that if the width of the SE0 on the upstream facing port is only 10 ms, the Receiver will enter the Bus_Reset state with sufficient time remaining in the 10 ms interval for the hub to complete its reset processing. Furthermore, if the hub is suspended when the Receiver enters this state, the hub must be able to start its clocks, time this interval, and complete its reset (chirp) protocol and processing in the Bus_Reset state within 10 ms. It is preferred that this interval be as long as possible given the constraints listed here. This will provide for the maximum immunity to noise on the upstream facing port and reduce the probability that the device will reset in the presence of noise before the upstream hub disables the port.

The timer is reset each time this state starts.

11.6.3.9 Bus_Reset

This state is entered:

- From the ReceivingSE0 state when the timer expires. As long as the port continues to receive SE0, the Receiver will remain in this state.
- This state is also entered while power-on-reset (POR) is being generated by the hub's local circuitry. The state machine cannot exit this state while POR is active.
- The 3 ms timer expires in the ReceivingIS state and the port has removed its high-speed terminations and connected its D+ pull-up resistor and the resulting bus state is still SE0.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

In this state, a high-speed capable port will implement the chirp signaling, handshake, and timing protocol as described in Section 7.1.7.5.

11.6.4 Transmitter

This state machine is used to monitor the upstream facing port while the Hub Repeater has connectivity in the upstream direction. The purpose of this monitoring activity is to prevent propagation of erroneous indications in the upstream direction. In particular, this machine prevents babble and disconnect events on the downstream facing ports of this hub from propagating and causing this hub to be disabled or disconnected by the hub to which it is attached. Figure 11-13 is the transmitter state transition diagram. Table 11-9 defines the signals and events referenced in Figure 11-13.

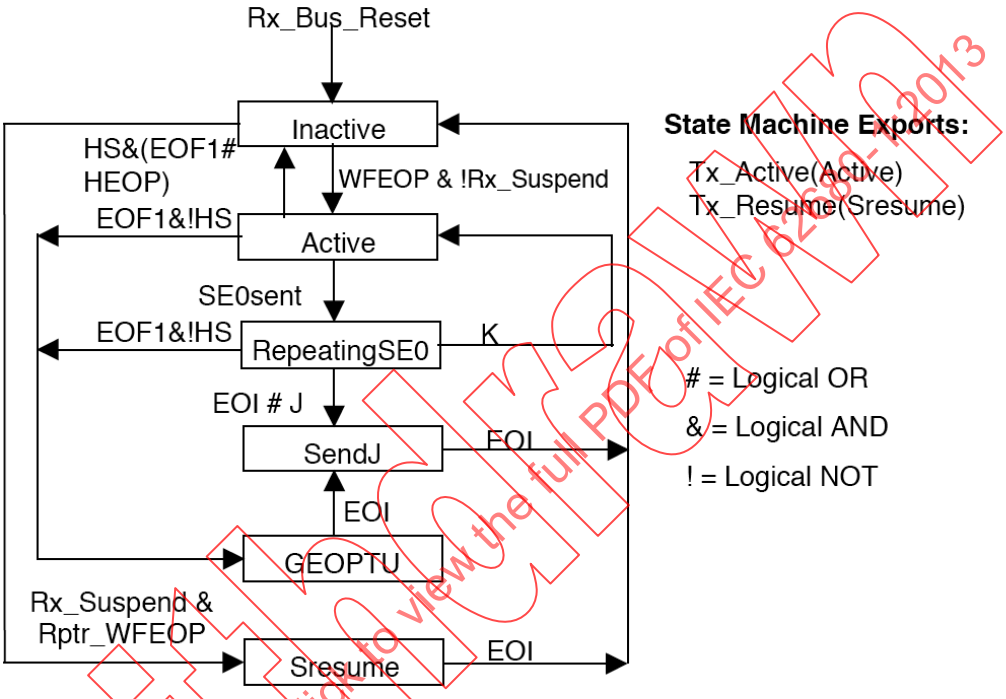


Figure 11-13. Upstream Facing Port Transmitter State Machine

Table 11-9. Upstream Facing Port Transmit Signal/Event Definitions

Signal/Event Name	Event/Signal Source	Description
Rx_Bus_Reset	Receiver	Receiver is in the Bus_Reset state
EOF1	(micro)frame Timer	Hub (micro)frame time has reached the EOF1 point or is between EOF1 and the end of the (micro)frame
J	Internal	Transmitter transitions to sending a 'J' and transmits a 'J'
Rptr_WFEOP	Hub Repeater	Hub Repeater is in the WFOEP state
K	Internal	Transmitter transmits a 'K'
SE0sent	Internal	At least one bit time of SE0 has been sent through the transmitter
Rx_Suspend	Receiver	Receiver is in Suspend state
HEOP	Repeater	Completion of packet transmission in upstream direction
HS	Internal	Upstream facing port is operating as high-speed port
EOI	Internal	End of timed interval

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.6.4.1 Inactive

This state is entered at the end of the SendJ state or while the Receiver is in the Bus_Reset state. This state is also entered at the end of the Sresume state. While the transmitter is in this state, both the differential and single-ended transmit circuits are disabled and placed in their high-impedance state.

When port is operating as a high-speed port, this state is entered from the Active state at EOF1 or after an HEOP from downstream.

11.6.4.2 Active

This state is entered from the Inactive state when the Hub Repeater transitions to the WFEOP state. This state is entered from the RepeatingSE0 state if the first transition after the SE0 is not to the J state. In this state, the data from a downstream facing port is repeated and transmitted on the upstream facing port.

11.6.4.3 RepeatingSE0

The port enters this state from the Active state when one bit time of SE0 has been sent on the upstream facing port. While in this state, the transmitter is still active and downstream signaling is repeated on the port. This is a timed state with a duration of 23 full-speed bit times.

11.6.4.4 SendJ

The port enters this state from the RepeatingSE0 state if either the bit timer reaches 23 or the repeated signaling changes from SE0 to 'J' or 'SE1'. This state is also entered at the end of the GEOPTU state. This state lasts for one full-speed bit time. During this state, the hub drives an SJ on the port.

11.6.4.5 Generate End of Packet Towards Upstream Port (GEOPTU)

The port enters this state from the Active or RepeatingSEO state if the frame timer reaches the EOF1 point.

In this state, the port transmits SE0 for two full-speed bit times.

11.6.4.6 Send Resume (Sresume)

The port enters this state from the Inactive state if the Receiver is in the Suspend state and the Hub Repeater transitions to the WFEOP state. This indicates that a downstream device (or the port to the Hub Controller) has generated resume signaling causing upstream connectivity to be established.

On entering this state, the hub will restart clocks if they had been turned off during the Suspend state. While in this state, the Transmitter will drive a 'K' on the upstream facing port. While the Transmitter is in this state, the Receiver is held in the Resume state. While the Receiver is in the Resume state, all downstream facing ports that are in the Enabled state are placed in the TransmitR state and the resume on this port is transmitted to those downstream facing ports.

The port stays in this state for at least 1 ms but for no more than 15 ms.

11.7 Hub Repeater

The Hub Repeater provides the following functions:

- Sets up and tears down connectivity on packet boundaries
- Ensures orderly entry into and out of the Suspend state, including proper handling of remote wakeups

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.7.1 High-speed Packet Connectivity

High-speed packet repeaters must reclock the packets in both directions. Reclocking means that the repeater extracts the data from the received stream and retransmits the stream using its own local clock. This is necessary in order to keep the jitter seen at a receiver within acceptable limits (see Chapter 7 for definition and limits on jitter).

Reclocking creates several requirements which can be best understood with the example repeater signal path shown in Figure 11-14.

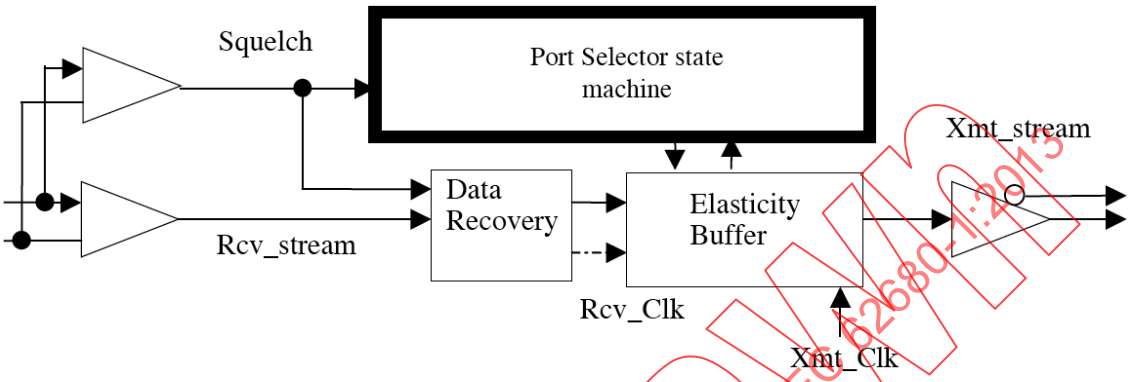


Figure 11-14. Example Hub Repeater Organization

11.7.1.1 Squelch Circuit

Because of squelch detection, the initial bits of the SYNC field may not be seen in the rest of the repeater. At most, 4 bits of the SYNC field may be sacrificed in the entire repeater path.

The squelch circuit may take at most 4 bit times to disable the repeater after the bus returns to the Idle state. This results in bits being added after the end of the packet. This is also known as EOP dribble and up to 4 random bits may get added after the packet by the entire repeater path.

11.7.1.2 Data Recovery Unit

The data recovery unit extracts the receive clock and receive data from this stream. Note that this is a conceptual model only; actual implementations (e.g., DLL) may achieve the reclocking by the local clock without separation of the receive clock and data.

11.7.1.3 Elasticity Buffer

The half-depth of the elasticity buffer in the repeater must be at least 12 bits.

The total latency of a packet through a repeater must be less than 36 bit times. This includes the latency through the elasticity buffer.

The elasticity buffer is used to handle the difference in frequency between the receive clock and the local clock and works as follows. The elasticity buffer is primed (filled with at least 12 bits) by the receive clock before the data is clocked out of it by the transmit clock. If the transmit clock is faster than the receive clock, the buffer will get emptied more quickly than it gets filled. If the transmit clock is slower, the buffer will get emptied slower than it gets filled. If the half-depth of the buffer is chosen to be equal to the maximum difference in clock rate over the length of a packet, bits will not be lost or added to the packet. The half-depth is calculated as follows.

The clock tolerance allowed is 500 ppm. This takes into account the effect of voltage, temperature, aging, etc. So the received clock and the local clock could be different by 1000 ppm. The longest packet has a data payload of 1 Kbytes. The maximum length of a packet is computed by adding the length of all the fields and assuming maximum bit-stuffing. This maximum length is 9644 bits (9624 bits of packet + 20 bits of EOP dribble). This means that

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

when the repeater is clocking out a packet with its local clock, it could get ahead of or fall behind the receive clock by 9.644 bits (1000 ppm*9644). This calculation yields 10 bits. The half-depth of the elasticity buffer in the repeater must be at least 12 bits to provide system timing margin.

11.7.1.4 High-Speed Port Selector State Machine

This state machine is used to establish connectivity on a valid packet and to keep the repeater from establishing connectivity from a port which is seeing noise. This state machine must implement the behavior shown in Figure 11-15. (Note: This state machine may be implemented on a per-port or per-hub basis.)

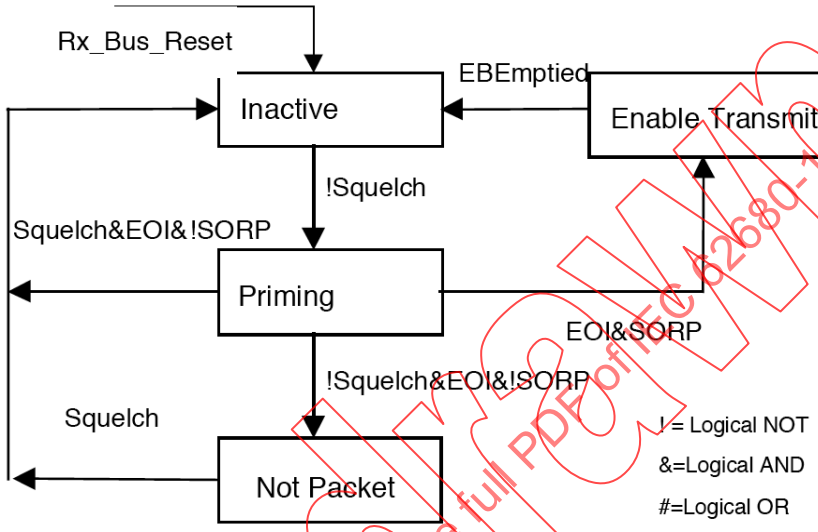


Figure 11-15. High-speed Port Selector State Machine

Table 11-10. High-speed Port Selector Signal/Event Definitions

Signal/Event Name	Event/Signal Source	Description
Rx_Bus_Reset	Internal	Receiver is in the Bus_reset state.
EBEmptied	Internal	All bits accumulated in the elasticity buffer have been transmitted.
EOI	Internal	End of interval of time needed for priming elasticity buffer
Squelch	Internal	Bus is in squelch state
SORP	Internal	Start Of Repeating Pattern; a 'JKJK' or 'KJKJ' pattern has been seen in data in elasticity buffer.

11.7.1.4.1 Inactive

This state is entered

- From the Enable Transmit state when all the bits accumulated in the elasticity buffer have been transmitted
- From the Priming state if squelch is seen and the elasticity buffer is primed without a SORP being seen
- From the Not Packet state when the squelch circuit indicates a squelch state on the port
- From on any state on Rx_Bus_Reset

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.7.1.4.2 Priming

This state is entered from the Inactive state when the squelch circuit indicates that valid signal levels have been observed at the port. This is a timed state and the priming interval is the time needed for the implementation to fill the elasticity buffer with at least 12 bits.

11.7.1.4.3 Enable Transmit

This state is entered from the Priming state when the Elasticity buffer priming interval has elapsed and the bits in the elasticity buffer include the SORP pattern.

In this state, the state machine generates a signal “start of high-speed packet” (SOHP) to the repeater state machine which allows the repeater to establish connectivity from this port to the upstream facing port (or downstream facing ports).

11.7.1.4.4 Not Packet

This state is entered from the Priming state when the Elasticity buffer priming interval has elapsed, and the bits in the elasticity buffer do not include the SORP pattern, and the squelch signal is not active.

11.7.2 Hub Repeater State Machine

The Hub repeater state machine in Figure 11-16 shows the states and transitions needed to implement the Hub Repeater. Table 11-11 defines the Hub Repeater signals and events. The following sections describe the states and the transitions.

11.7.2.1 High-speed Repeater Operation

Connectivity is setup on SOHP and torn down on HEOP. (HEOP is either the EBemptied signal from the port selector state machine ‘OR’ the EOI signal which causes the transition out of the SendEOR state in downstream facing port state machine.) Several of the state transitions below will occur when the HEOP is seen. When such a transition is indicated, the transition does not occur until after the hub has repeated the last bit in the elasticity buffer. Some of the transitions are triggered by an SOHP. Transitions of this type occur as soon as the hub detects the SOHP from the port selector state machine ensuring that a valid packet start has been seen.

11.7.2.2 Full-/low-speed Repeater Operation

Connectivity is setup on SOP and torn down on EOP. Several of the state transitions below will occur when the EOP is seen. When such a transition is indicated, the transition does not occur until after the hub has repeated the SE0-to-'J' transition and has driven 'J' for at least one bit time (bit time is determined by the speed of the port.) Some of the transitions are triggered by an SOP. Transitions of this type occur as soon as the hub detects the 'J'-to-'K' transition, ensuring that the initial edge of the SYNC field is preserved.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.7.2.3 Repeater State Machine

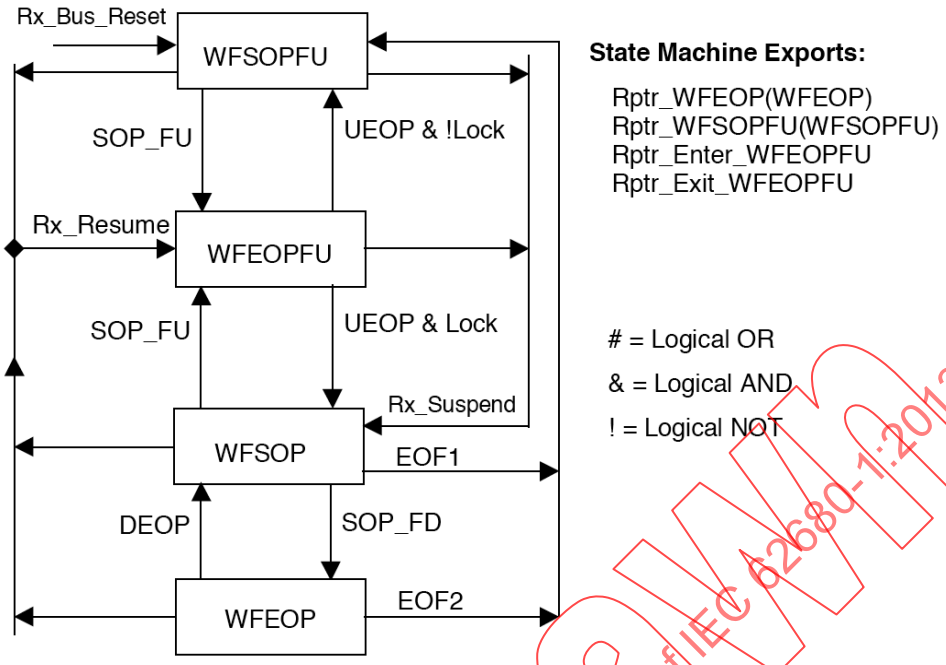


Figure 11-16. Hub Repeater State Machine

Table 11-11. Hub Repeater Signal/Event Definitions

Signal/Event Name	Event/Signal Source	Description
Rx_Bus_Reset	Receiver	Receiver is in the Bus_Reset state
HEOP	Internal (Port selector, Downstream port, Upstream port receiver)	Three sources of HEOP: EBEmptied signal from port selector state machine OR transition at EOI from SendEOR state in downstream facing port state machine OR EOITR from upstream facing port receiver state machine
UEOP	Internal	(HEOP)EOP received from the upstream facing port
DEOP	Internal	Generated when the Transmitter enters the (Inactive) SendJ state
EOF1	(Micro)frame Timer	(micro)frame timer is at the EOF1 point or between EOF1 and End-of-(micro)frame
EOF2	(Micro)frame Timer	(micro)frame timer is at the EOF2 point or between EOF2 and End-of-(micro)frame
Lock	(Micro)frame Timer	(micro)frame timer is locked
Rx_Suspend	Receiver	Receiver is in the Suspend state
Rx_Resume	Receiver	Receiver is in the Resume state
SOP_FD	Internal	(SOHP)SOP received from downstream facing port or Hub Controller. Generated (after SOHP identified) on the transition from the Idle to K state on a port.
SOP_FU	Internal	(SOHP)SOP received from upstream facing port. Generated (after SOHP identified) on the transition from the Idle to K state on the upstream facing port.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.7.3 Wait for Start of Packet from Upstream Port (WFSOPFU)

This state is entered in either of the following situations:

- From any other state when the upstream Receiver is in the Bus_Reset state
- From the WFSOP state if the (micro)frame timer is at or has passed the EOF1 point
- From the WFEOP state at the EOF2 point
- From the WFEOPFU if the (micro)frame timer is not synchronized (locked) when an (HEOP)EOP is received on the upstream facing port

In this state, the hub is waiting for an (SOHP)SOP on the upstream facing port, and transitions on downstream facing ports are ignored by the Hub Repeater. While the Hub Repeater is in this state, connectivity is not established.

This state is used during the End-of-(micro)frame (past the EOF1 point) to ensure that the hub will be able to receive the SOF when it is sent by the host.

11.7.4 Wait for End of Packet from Upstream Port (WFEOPFU)

The hub enters this state if the hub is in the WFSOP or WFSOPFU state and an (SOHP)SOP is detected on the upstream facing port. The hub also enters this state from the WFSOP, WFSOPFU, or WFEOP states when the Receiver enters the Resume state.

While in this state, connectivity is established from the upstream facing port to all enabled downstream facing ports. Downstream facing ports that are in the Enabled state are placed in the Transmit state on the transition to this state.

11.7.5 Wait for Start of Packet (WFSOP)

This state is entered in any of the following situations:

- From the WFEOP state when an (HEOP)EOP is detected from the downstream facing port
- From the WFEOPFU state if the (micro)frame timer is synchronized (locked) when an (HEOP)EOP is received from upstream
- From the WFSOPFU or WFEOPFU states when the upstream Receiver transitions to the Suspend state

A hub in this state is waiting for an (SOHP)SOP on the upstream facing port or any downstream facing port that is in the Enabled state. While the Hub Repeater is in this state, connectivity is not established.

11.7.6 Wait for End of Packet (WFEOP)

This state is entered from the WFSOP state when an (SOHP)SOP is received from a downstream facing port in the Enabled state.

In this state, the hub has connectivity established in the upstream direction and the signaling received on an enabled downstream facing port is repeated and driven on the upstream facing port. The upstream Transmitter is placed in the Active state on the transition to this state.

If the Hub Repeater is in this state when the EOF2 point is reached, the downstream facing port for which connectivity is established is disabled as a babble port.

Note: The full-speed Transmitter will send an EOP at EOF1, but the Repeater stays in this state until the device sends an (HEOP)EOP or the EOF2 point is reached.

11.8 Bus State Evaluation

A hub is required to evaluate the state of the connection on a port in order to make appropriate port state transitions. This section describes the appropriate times and means for several of these evaluations.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.8.1 Port Error

A Port Error can occur on a downstream facing port that is in the Enabled state. A Port Error condition exists when:

- The hub is in the WFEOP state with connectivity established upstream from the port when the (micro)frame timer reaches the EOF2 point.
- At the EOF2 point, the Hub Repeater is in the WFSOPFU state, and there is other than Idle state on the port.

If upstream-directed connectivity is established when the (micro)frame timer reaches the EOF1 point, the upstream Transmitter will (return to Inactive state) generate a full-speed EOP to prevent the hub from being disabled by the upstream hub. The connected port is then disabled if it has not ended the packet and returned to the Idle state before the (micro)frame timer reaches the EOF2 point.

11.8.2 Speed Detection

At the end of reset, the bus is in the Idle state for the speed recorded in the port status register. Speed detection is described in Section 7.1.7.5.

If the device connected at the downstream facing port is high-speed, the repeater (rather than the Transaction Translator) is used to signal between this port and the upstream facing port.

Due to connect and start-up transients, the hub may not be able to reliably determine the speed of the device until the transients have ended. The USB System Software is required to "debounce" the connection and provide a delay between the time a connection is detected and the device is used (see Section 7.1.7.3). At the end of the debounce interval, the device is expected to have placed its upstream facing port in the Idle state and be able to react to reset signaling. The USB System Software must send a SetPortFeature(PORT_RESET) request to the port to enable the port and make the attached device ready for use.

The downstream facing port monitors the state of the D+ and D- lines to determine if the connected device is low-speed. If so, the PORT_LOW_SPEED status bit is set to one to indicate a low-speed device. If not, the PORT_LOW_SPEED status bit is set to zero to indicate a full-/high-speed device. Upon exit from the reset process, the hub must set the PORT_HIGH_SPEED status bit according to the detected speed. The downstream facing port performs the required reset processing as defined in Section 7.1.7.5. At the end of the Resetting state, the hub will return the bus to the Idle state that is appropriate for the speed of the attached device and transition to the Enabled state.

11.8.3 Collision

If the Hub Repeater is in the WFEOP state and an (SOHP)SOP is detected on another enabled port, a Collision condition exists. There are two allowed behaviors for the hub in this instance. In either case, connectivity teardown at EOF1 and babble detection at EOF2 is required.

The first, and preferred, behavior is to ‘garble’ the message so that the host can detect the problem. The hub garbles the message by transmitting a (‘J’ or) 'K' on the upstream facing port. This (‘J’ or) 'K' should persist until packet traffic from all downstream facing ports ends. The hub should use the last (‘J’ or ‘K’) EOP to terminate the garbled packet. Babble detection is enabled during this garbled message.

A second behavior is to block the second packet and, when the first message ends, return the hub to the WFSOPFU or WFSOP state as appropriate. If the second stream is still active, the hub may reestablish connectivity upstream. This method is not preferred, as it does not convey the problem to the host. Additionally, if the second stream causes the hub to reestablish upstream connectivity as the host is trying to establish downstream connectivity, additional packets can be lost and the host cannot properly associate the problem.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Note: In high-speed repeaters, use of the SOHP to detect collisions would need replication of the datapath shown Figure 11-14 at every port. The unsquelch signal at a port can be used instead of the SOHP to detect collisions; in this case, the second behavior (blocking) described above must be used.

11.8.4 Low-speed Port Behavior

When a hub is configured for full-/low-speed operation, low-speed data is sent or received through the hub's upstream facing port at full-speed signaling even though the bit times are low-speed.

Full-speed signaling must not be transmitted to low-speed ports.

If a port is detected to be attached to a low-speed device, the hub port's output buffers are configured to operate at the slow slew rate (75-300 ns), and the port will not propagate downstream-directed packets unless they are prefaced with a PRE PID. When a PRE PID is received, the 'J' state must be driven on enabled low-speed ports within four bit times of receiving the last bit of the PRE PID.

Low-speed data follows the PID and is propagated to both low- and full-speed devices. Hubs continue to propagate downstream signaling to all enabled ports until a downstream EOP is detected, at which time all output drivers are turned off.

Full-speed devices will not misinterpret low-speed traffic because no low-speed data pattern can generate a valid full-speed PID.

When a low-speed device transmits, it does not preface its data packet with a PRE PID. Hubs will propagate upstream-directed packets of full-/low-speed using full-speed signaling polarity and edge rates.

For both upstream and downstream low-speed data, the hub is responsible for inverting the polarity of the data before transmitting to/from a low-speed port.

Although a low-speed device will send a low-speed EOP to properly terminate a packet, a hub may truncate a low-speed packet at the EOF1 point with a full-speed EOP. Thus, hubs must always be able to tear down connectivity in response to a full-speed EOP regardless of the data rate of the packet.

Because of the slow transitions on low-speed ports, when the D+ and D- signal lines are switching between the 'J' and 'K', they may both be below 2.0 V for a period of time that is longer than a full-speed bit time. A hub must ensure that these slow transitions do not result in termination of connectivity and must not result in an SE0 being sent upstream.

11.8.4.1 Low-speed Keep-alive

All hub ports to which low-speed devices are connected must generate a low-speed keep-alive strobe, generated at the beginning of the frame, which consists of a valid low-speed EOP (described in Section 7.1.13.2). The strobe must be generated at least once in each frame in which an SOF is received. This strobe is used to prevent low-speed devices from suspending if there is no other low-speed traffic on the bus. The hub can generate the keep-alive on any valid full-speed token packet. The following rules for generation of a low-speed keep-alive must be adhered to:

- A keep-alive must minimally be derived from each SOF. It is recommended that a keep-alive be generated on any valid full-speed token.
- The keep-alive must start by the eighth bit after the PID of the full-speed token.

11.9 Suspend and Resume

Hubs must support suspend and resume both as a USB device and in terms of propagating suspend and resume signaling. Hubs support both global and selective suspend and resume.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Global and selective suspend are defined in Section 7.1.7.6. Global suspend/resume refers to the entire bus being suspended or resumed without affecting any hub's downstream facing port states; selective suspend/resume refers to a downstream facing port of a hub being suspended or resumed without affecting the hub state. Global suspend/resume is implemented through the root port(s) at the host. Selective suspend/resume is implemented via requests to a hub. Device-initiated resume is called remote-wakeup (see Section 7.1.7.7).

If the hub upstream facing port is in (high-speed) full-speed, the required behavior is the same as that for a function with upstream facing port in (high-speed) full-speed and is described in Chapter 7.

When a downstream facing port operating at high-speed goes into the Suspended state, it switches to full-speed terminations but continues to have high-speed port status. In response to a remote wakeup or selective resume, this port will drive full-speed 'K' throughout its Resuming state. The requirements and timings are the same as for full-speed ports and described below. At the end of this signaling, the bus will be returned to the high-speed Idle state (using the SendEOR state). After this, the port will return to the Enabled state. The high-speed status of the port is maintained throughout the suspend-resume cycle.

Figure 11-17 and Figure 11-18 show the timing relationships for an example remote-wakeup sequence. This example illustrates a device initiating resume signaling through a suspended hub ('B') to an awake hub ('A'). Hub 'A' in this example times and completes the resume sequence and is the "Controlling Hub". The timings and events are defined in Section 7.1.7.7.

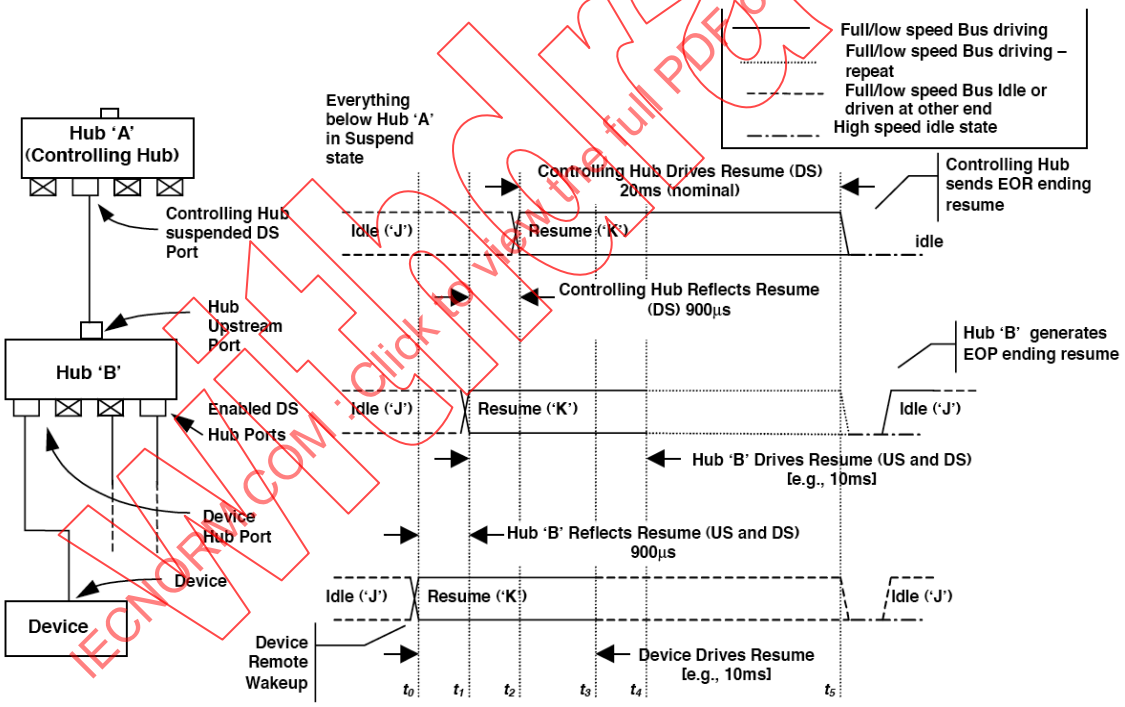


Figure 11-17. Example Remote-wakeup Resume Signaling With Full-/low-speed Device

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

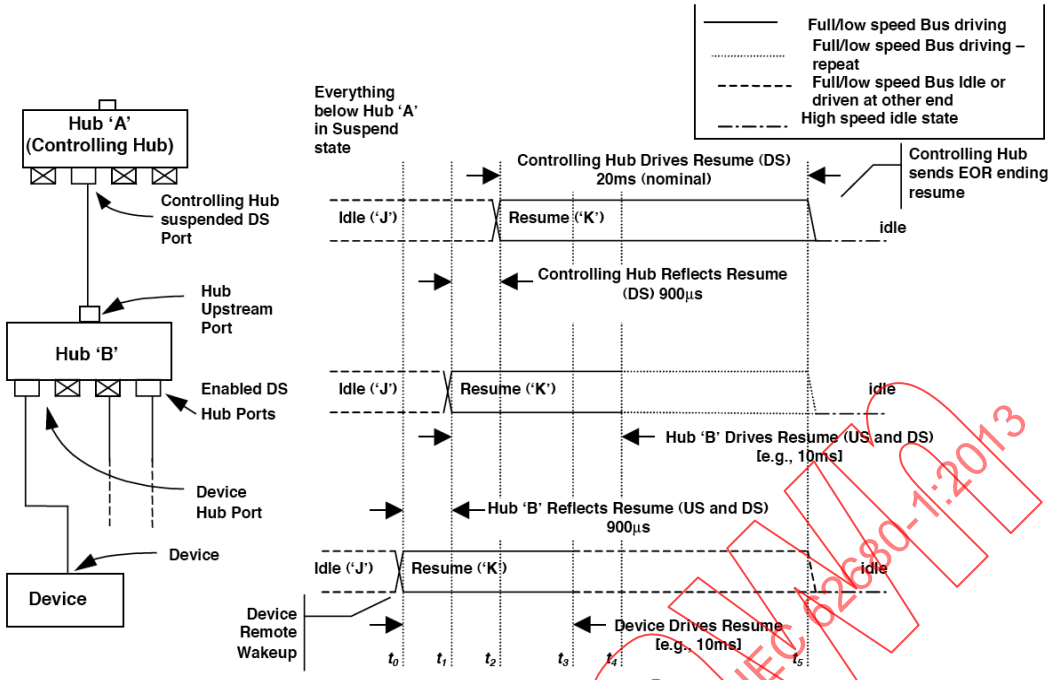


Figure 11-18. Example Remote-wakeup Resume Signaling With High-speed Device

Here is an explanation of what happens at each t_n:

t₀ Suspended device initiates remote-wakeup by driving a 'K' on the data lines.

t₁ Suspended hub 'B' detects the 'K' on its downstream facing port and wakes up enough within 900 µs to filter and then reflect the resume upstream and down through all enabled ports.

t₂ Hub 'A' is not suspended (implication is that the port at which 'B' is attached is selectively suspended), detects the 'K' on the selectively suspended port where 'B' is attached, and filters and then reflects the resume signal back to 'B' within 900 µs.

t₃ Device ceases driving 'K' upstream.

t₄ Hub 'B' ceases driving 'K' upstream and down all enabled ports and begins repeating upstream signaling to all enabled downstream facing ports.

t₅ Hub 'A' completes resume sequence, after appropriate timing interval, by driving a speed-appropriate end of resume downstream. (End of resume will be an Idle state for a high-speed device or a low-speed EOP for a full-/low-speed device.)

The hub reflection time is much smaller than the minimum duration a USB device will drive resume upstream. This relationship guarantees that resume will be propagated upstream and downstream without any gaps.

11.10 Hub Reset Behavior

Reset signaling to a hub is defined only in the downstream direction, which is at the hub's upstream facing port. Reset signaling required of the hub is described in Section 7.1.7.5.

A suspended hub must interpret the start of reset as a wakeup event; it must be awake and have completed its reset sequence by the end of reset signaling.

After completion of the reset sequence, a hub is in the following state:

- Hub Controller default address is 0.
- Hub status change bits are set to zero.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- Hub Repeater is in the WFSOPFU state.
- Transmitter is in the Inactive state.
- Downstream facing ports are in the Not Configured state and SE0 driven on all downstream facing ports.

11.11 Hub Port Power Control

Self-powered hubs may have power switches that control delivery of power downstream facing ports but it is not required. Bus-powered hubs are required to have power switches. A hub with power switches can switch power to all ports as a group/gang, to each port individually, or have an arbitrary number of gangs of one or more ports.

A hub indicates whether or not it supports power switching by the setting of the Logical Power Switching Mode field in *wHubCharacteristics*. If a hub supports per-port power switching, then the power to a port is turned on when a SetPortFeature(PORT_POWER) request is received for the port. Port power is turned off when the port is in the Powered-off or Not Configured states. If a hub supports ganged power switching, then the power to all ports in a gang is turned on when any port in a gang receives a SetPortFeature(PORT_POWER) request. The power to a gang is not turned off unless all ports in a gang are in the Powered-off or Not Configured states. Note, the power to a port is not turned on by a SetPortFeature(PORT_POWER) if both C_HUB_LOCAL_POWER and Local Power Status (in *wHubStatus*) are set to 1B at the time when the request is executed and the PORT_POWER feature would be turned on.

Although a self-powered hub is not required to implement power switching, the hub must support the Powered-off state for all ports. Additionally, the hub must implement the *PortPwrCtrlMask* (all bits set to 1B) even though the hub has no power switches that can be controlled by the USB System Software.

Note: To ensure compatibility with previous versions of USB Software, hubs must implement the Logical Power Switching Mode field in *wHubCharacteristics*. This is because some versions of SW will not use the SetPortFeature() request if the hub indicates in *wHubCharacteristics* that the port does not support port power switching. Otherwise, the Logical Power Switching Mode field in *wHubCharacteristics* would have become redundant as of this version of the specification.

The setting of the Logical Power Switching Mode for hubs with no power switches should reflect the manner in which over-current is reported. For example, if the hub reports over-current conditions on a per-port basis, then the Logical Power Switching Mode should be set to indicate that power switching is controlled on a per-port basis.

For a hub with no power switches, *bPwrOn2PwrGood* must be set to zero.

11.11.1 Multiple Gangs

A hub may implement any number of power and/or over-current gangs. A hub that implements more than one over-current and/or power switching gang must set both the Logical Power Switching Mode and the Over-current Reporting Mode to indicate that power switching and over-current reporting are on a per port basis (these fields are in *wHubCharacteristics*). Also, all bits in *PortPwrCtrlMask* must be set to 1B.

When an over-current condition occurs on an over-current protection device, the over-current is signaled on all ports that are protected by that device. When the over-current is signaled, all the ports in the group are placed in the Powered-off state, and the C_PORT_OVER-CURRENT field is set to 1B on all the ports. When port status is read from any port in the group, the PORT_OVER-CURRENT field will be set to 1B as long as the over-current condition exists. The C_PORT_OVER-CURRENT field must be cleared in each port individually.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

When multiple ports share a power switch, setting PORT_POWER on any port in the group will cause the power to all ports in the group to turn on. It will not, however, cause the other ports in that group to leave the Powered-off state. When all the ports in a group are in the Powered-off state or the hub is not configured, the power to the ports is turned off.

If a hub implements both power switching and over-current, it is not necessary for the over-current groups to be the same as the power switching groups.

If an over-current condition occurs and power switches are present, then all power switches associated with an over-current protection circuit must be turned off. If multiple over-current protection devices are associated with a single power switch then that switch will be turned off when any of the over-current protection circuits indicates an over-current condition.

11.12 Hub Controller

The Hub Controller is logically organized as shown in Figure 11-19.

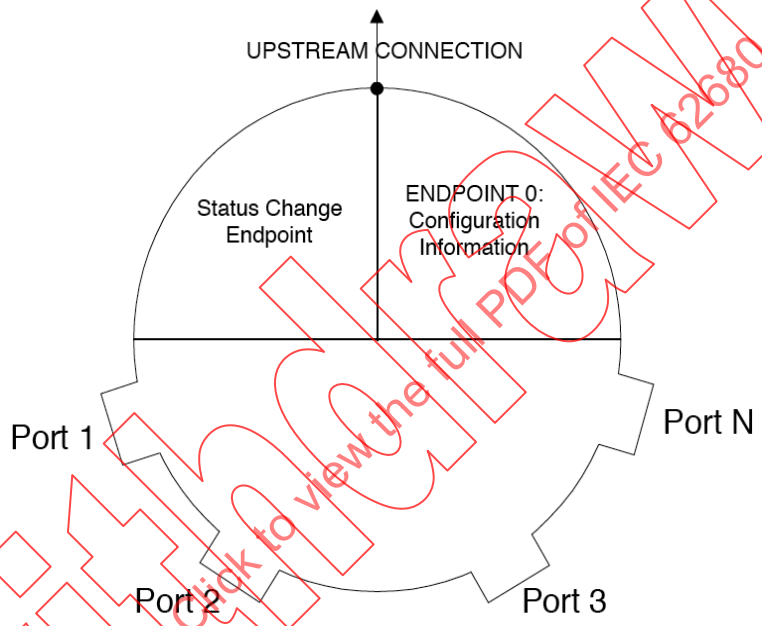


Figure 11-19. Example Hub Controller Organization

11.12.1 Endpoint Organization

The Hub Class defines one additional endpoint beyond Default Control Pipe, which is required for all hubs: the Status Change endpoint. The host system receives port and hub status change notifications through the Status Change endpoint. The Status Change endpoint is an interrupt endpoint. If no hub or port status change bits are set, then the hub returns an NAK when the Status Change endpoint is polled. When a status change bit is set, the hub responds with data, as shown in Section 11.12.4, indicating the entity (hub or port) with a change bit set. The USB System Software can use this data to determine which status registers to access in order to determine the exact cause of the status change interrupt.

11.12.2 Hub Information Architecture and Operation

Figure 11-20 shows how status, status change, and control information relate to device states. Hub descriptors and Hub/Port Status and Control are accessible through the Default Control Pipe. The Hub descriptors may be read at any time. When a hub detects a change on a port or when the hub changes its own state, the Status Change endpoint transfers data to the host in the form specified in Section 11.12.4.

Hub or port status change bits can be set because of hardware or Software events. When set, these bits remain set until cleared directly by the USB System Software through a

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

ClearPortFeature() request or by a hub reset. While a change bit is set, the hub continues to report a status change when polled until all change bits have been cleared by the USB System Software.

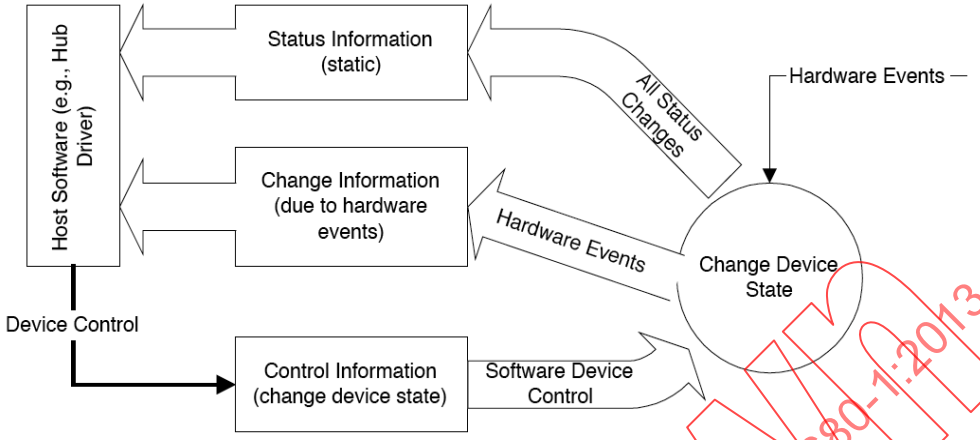


Figure 11-20. Relationship of Status, Status Change, and Control Information to Device States

The USB System Software uses the interrupt pipe associated with the Status Change endpoint to detect changes in hub and port status.

11.12.3 Port Change Information Processing

Hubs report a port's status through port commands on a per-port basis. The USB System Software acknowledges a port change by clearing the change state corresponding to the status change reported by the hub. The acknowledgment clears the change state for that port so future data transfers to the Status Change endpoint do not report the previous event. This allows the process to repeat for further changes (see Figure 11-21).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

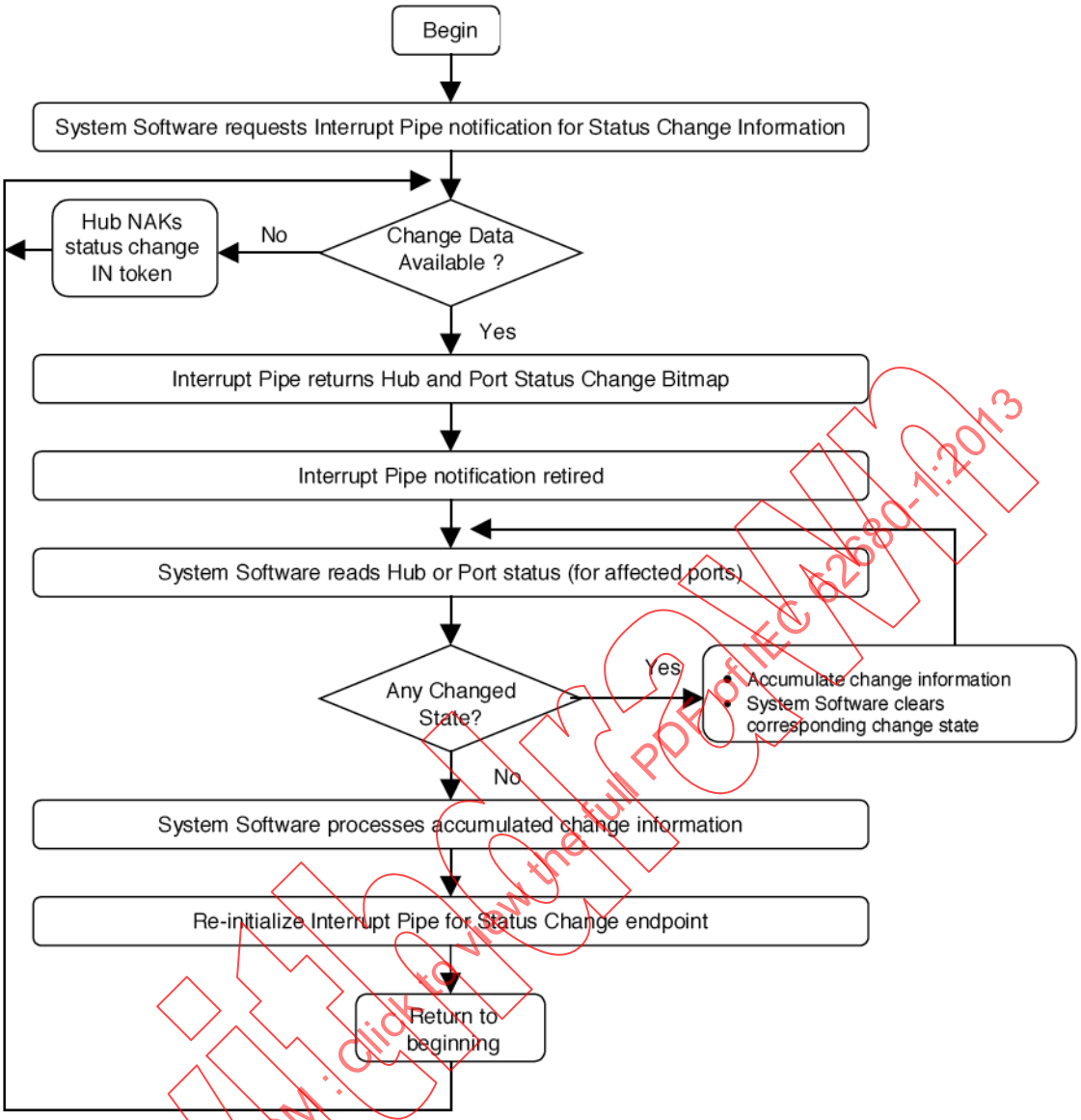


Figure 11-21. Port Status Handling Method

11.12.4 Hub and Port Status Change Bitmap

The Hub and Port Status Change Bitmap, shown in Figure 11-22, indicates whether the hub or a port has experienced a status change. This bitmap also indicates which port(s) has had a change in status. The hub returns this value on the Status Change endpoint. Hubs report this value in byte-increments. That is, if a hub has six ports, it returns a byte quantity, and reports a zero in the invalid port number field locations. The USB System Software is aware of the number of ports on a hub (this is reported in the hub descriptor) and decodes the Hub and Port Status Change Bitmap accordingly. The hub reports any changes in hub status in bit zero of the Hub and Port Status Change Bitmap.

The Hub and Port Status Change Bitmap size varies from a minimum size of one byte. Hubs report only as many bits as there are ports on the hub, subject to the byte-granularity requirement (i.e., round up to the nearest byte).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

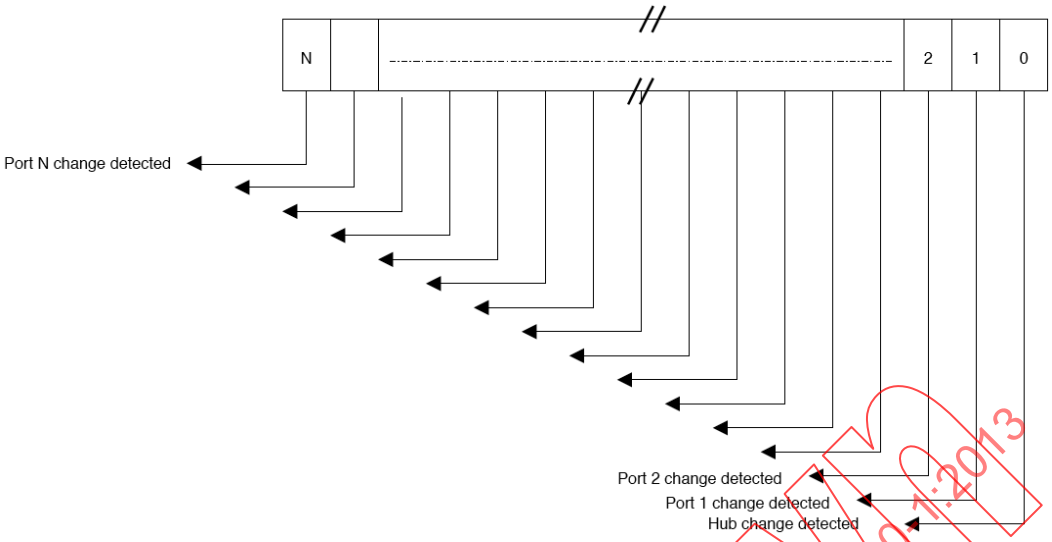


Figure 11-22. Hub and Port Status Change Bitmap

Any time the Status Change endpoint is polled by the host controller and any of the Status Changed bits are non-zero, the Hub and Port Status Change Bitmap is returned. Figure 11-23 shows an example creation mechanism for hub and port change bits.

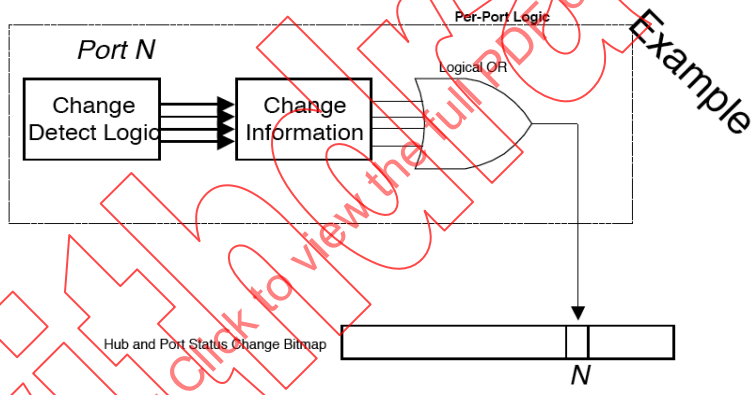


Figure 11-23. Example Hub and Port Change Bit Sampling

11.12.5 Over-current Reporting and Recovery

USB devices must be designed to meet applicable safety standards. Usually, this will mean that a self-powered hub implement current limiting on its downstream facing ports. If an over-current condition occurs, it causes a status and state change in one or more ports. This change is reported to the USB System Software so that it can take corrective action.

A hub may be designed to report over-current as either a port or a hub event. The hub descriptor field *wHubCharacteristics* is used to indicate the reporting capabilities of a particular hub (see Section 11.23.2). The over-current status bit in the hub or port status field indicates the state of the over-current detection when the status is returned. The over-current status change bit in the Hub or Port Change field indicates if the over-current status has changed.

When a hub experiences an over-current condition, it must place all affected ports in the Powered-off state. If a hub has per-port power switching and per-port current limiting, an over-current on one port may still cause the power on another port to fall below specified minimums. In this case, the affected port is placed in the Powered-off state and C_PORT_OVER_CURRENT is set for the port, but PORT_OVER_CURRENT is not set. If the hub has over-current detection on a hub basis, then an over-current condition on the hub will

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

cause all ports to enter the Powered-off state. However, in this case, neither C_PORT_OVER_CURRENT nor PORT_OVER_CURRENT is set for the affected ports.

Host recovery actions for an over-current event should include the following:

1. Host gets change notification from hub with over-current event.
2. Host extracts appropriate hub or port change information (depending on the information in the change bitmap).
3. Host waits for over-current status bit to be cleared to 0.
4. Host cycles power on to all of the necessary ports (e.g., issues a SetPortFeature(PORT_POWER) request for each port).
5. Host re-enumerates all affected ports.

11.12.6 Enumeration Handling

The hub device class commands are used to manipulate its downstream facing port state. When a device is attached, the device attach event is detected by the hub and reported on the status change interrupt. The host will accept the status change report and request a SetPortFeature(PORT_RESET) on the port. As part of the bus reset sequence, a speed detect is performed by the hub's port hardware.

The Get_Status(PORT) request invoked by the host will return a "not PORT_LOW_SPEED and PORT_HIGH_SPEED" indication for a downstream facing port operating at high-speed. The Get_Status(PORT) will report "PORT_LOW_SPEED" for a downstream facing port operating at low-speed. The Get_Status(PORT) will report "not PORT_LOW_SPEED and not PORT_HIGH_SPEED" for a downstream facing port operating at full-speed.

When the device is detached from the port, the port reports the status change through the status change endpoint and the port will be reconnected to the high-speed repeater. Then the process is ready to be repeated on the next device attach detect.

11.13 Hub Configuration

Hubs are configured through the standard USB device configuration commands. A hub that is not configured behaves like any other device that is not configured with respect to power requirements and addressing. If a hub implements power switching, no power is provided to the downstream facing ports while the hub is not configured. Configuring a hub enables the Status Change endpoint. The USB System Software may then issue commands to the hub to switch port power on and off at appropriate times.

The USB System Software examines hub descriptor information to determine the hub's characteristics. By examining the hub's characteristics, the USB System Software ensures that illegal power topologies are not allowed by not powering on the hub's ports if doing so would violate the USB power topology. The device status and configuration information can be used to determine whether the hub should be used as a bus or self-powered device. Table 11-12 summarizes the information and how it can be used to determine the current power requirements of the hub.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 11-12. Hub Power Operating Mode Summary

Configuration Descriptor		Hub Device Status (Self Power)	Explanation
MaxPower	bmAttributes (Self Powered)		
0	0	N/A	N/A This is an illegal set of information.
0	1	0	N/A A device which is only self-powered, but does not have local power cannot connect to the bus and communicate.
0	1	1	Self-powered only hub and local power supply is good. Hub status also indicates local power good, see Section 11.16.2.5. Hub functionality is valid anywhere depth restriction is not violated.
> 0	0	N/A	Bus-powered only hub. Downstream facing ports may not be powered unless allowed in current topology. Hub device status reporting Self Powered is meaningless in combination of a zeroed *bmAttributes.Self-Powered*.
> 0	1	0	This hub is capable of both self- and bus-powered operating modes. It is currently only available as a bus-powered hub.
> 0	1	1	This hub is capable of both self- and bus-powered operating modes. It is currently available as a self-powered hub.

A self-powered hub has a local power supply, but may optionally draw one unit load from its upstream connection. This allows the interface to function when local power is not available (see Section 7.2.1.2). When local power is removed (either a hub-wide over-current condition or local supply is off), a hub of this type remains in the Configured state but transitions all ports (whether removable or non-removable) to the Powered-off state. While local power is off, all port status and change information read as zero and all SetPortFeature() requests are ignored (request is treated as a no-operation). The hub will use the Status Change endpoint to notify the USB System Software of the hub event (see Section 11.24.2.6 for details on hub status).

The *MaxPower* field in the configuration descriptor is used to report to the system the maximum power the hub will draw from VBUS when the configuration is selected. For bus-powered hubs, the reported value must not include the power for any of external downstream facing ports. The external devices attaching to the hub will report their individual power requirements.

A compound device may power both the hub electronics and the permanently attached devices from VBUS. The entire load may be reported in the hubs' configuration descriptor with the permanently attached devices each reporting self-powered, with zero *MaxPower* in their respective configuration descriptors.

11.14 Transaction Translator

A hub has a special responsibility when it is operating in high-speed and has full-/low-speed devices connected on downstream facing ports. In this case, the hub must isolate the high-speed signaling environment from the full-/low-speed signaling environment. This function is performed by the Transaction Translator (TT) portion of the hub.

This section defines the required behavior of the transaction translator.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.14.1 Overview

Figure 11-24 shows an overview of the Transaction Translator. The TT is responsible for participating in high-speed split transactions on the high-speed bus via its upstream facing port and issuing corresponding full-/low-speed transactions on its downstream facing ports that are operating at full-/low-speed. The TT acts as a high-speed function on the high-speed bus and performs the role of a host controller for its downstream facing ports that are operating at full-/low-speed. The TT includes a high-speed handler to deal with high-speed transactions. The TT also includes a full-/low-speed handler that performs the role of a host controller on the downstream facing ports that are operating at full-/low-speed.

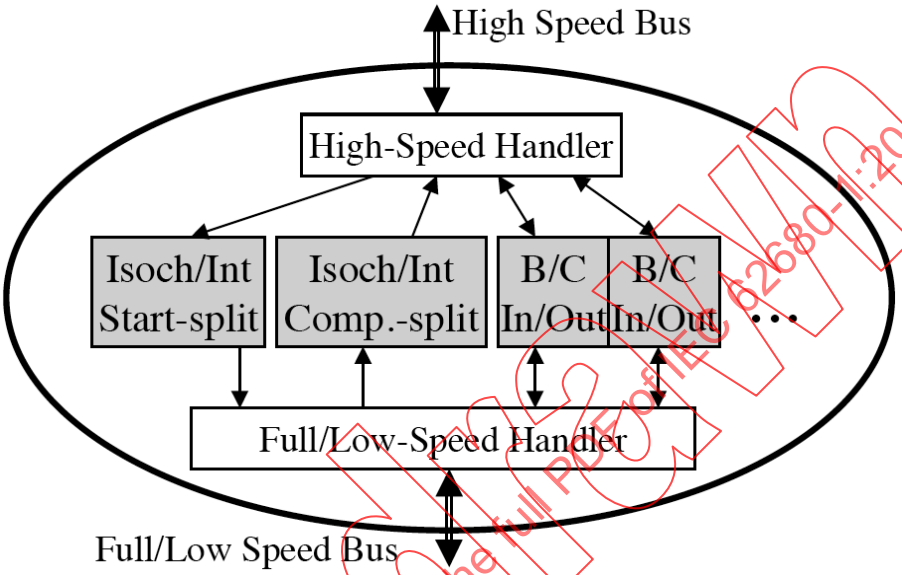


Figure 11-24. Transaction Translator Overview

The TT has buffers (shown in gray in the figure) to hold transactions that are in progress and tracks the state of each buffered transaction as it is processed by the TT. The buffers provide the connection between the high-speed and full-/low-speed handlers. The state tracking the TT does for each transaction depends on the specific USB transfer type of the transaction (i.e., bulk, control, interrupt, isochronous). The high-speed handler accepts high-speed start-split transactions or responds to high-speed complete-split transactions. The high-speed handler places the start-split transactions in local buffers for the full-/low-speed handler's use.

The buffered start-split transactions provide the full-/low-speed handler with the information that allows it to issue corresponding full-/low-speed transactions to full-/low-speed devices attached on downstream facing ports. The full-/low-speed handler buffers the results of these full-/low-speed transactions so that they can be returned with a corresponding complete-split transaction on the high-speed bus.

The general conversion between full-/low-speed transactions and the corresponding high-speed split transaction protocol is described in Section 8.4.2. More details about the specific transfer types for split transactions are described later in this chapter.

The high-speed handler of the TT operates independently of the full-/low-speed handler. Both handlers use the local transaction buffers to exchange information where required.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

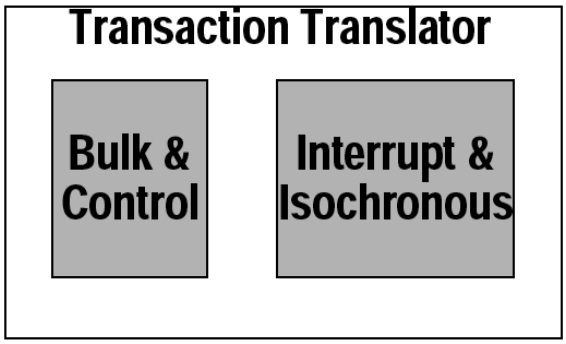


Figure 11-25. Periodic and Non-periodic Buffer Sections of TT

The TT has two buffer and state tracking sections (shown in gray in Figure 11-24 and Figure 11-25): periodic (for isochronous/interrupt full-/low-speed transactions) and non-periodic (for bulk/control full-/low-speed transactions). The requirements on the TT for these two buffer and state tracking sections are different. Each will be described in turn later in this chapter.

11.14.1.1 Data Handling Between High-speed and Full-/low-speed

The host converts transfer requests involving a full-/low-speed device into corresponding high-speed split transactions to the TT to which the device is attached.

Low-speed Preamble(PRE) packets are never used on the high-speed bus to indicate a low-speed transaction. Instead, a low-speed transaction is encoded in the split transaction token.

The host can have a single schedule of the transactions that need to be issued to devices. This single schedule can be used to hold both high-speed transactions and high-speed split transactions used for communicating with full-/low-speed devices.

11.14.1.2 Host Controller and TT Split Transactions

The host controller uses the split transaction protocol for initiating full-/low-speed transactions via the TT and then determining the completion status of the full-/low-speed transaction. This approach allows the host controller to start a full-/low-speed transaction and then continue with other high-speed transactions while avoiding having to wait for the slower transaction to proceed/complete at its speed. A high-speed split transaction has two parts: a start-split and a complete-split. Split transactions are only used between the host controller and a hub. No other high-/full-/low-speed devices ever participate in split transactions.

When the host controller sends a start-split transaction at high-speed, the split transaction is addressed to the TT for that device. That TT will accept the transaction and buffer it locally. The high-speed handler responds with an appropriate handshake to inform the host controller that the transaction has been accepted. Not all split transactions have a handshake phase to the start-split. The start-split transactions are kept temporarily in a TT transaction buffer.

The full-/low-speed handler processes start-split periodic transactions stored in the periodic transaction buffer (in order) as the downstream full-/low-speed bus is ready for the “next” transaction. The full-/low-speed handler accepts any result information from the downstream bus (in response to the full-/low-speed transaction) and accumulates it in a local buffer for later transmission to the host controller.

At an appropriate future time, the host controller sends a high-speed complete-split transaction to retrieve the status/data/result for appropriate full-/low-speed transactions. The high-speed handler checks this high-speed complete-split transaction with the response at the head of the appropriate local transaction buffer and responds accordingly. The specific split transaction sequences are defined for each USB transfer type in later sections.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.14.1.3 Multiple Transaction Translators

A hub has two choices for organizing transaction translators (TTs). A hub can have one TT for all downstream facing ports that have full-/low-speed devices attached or the hub can have one TT for each downstream facing port. The hub must report its organization in the hub class descriptor.

11.14.2 Transaction Translator Scheduling

As the high-speed handler accepts start-splits, the full-/low-speed transaction information and data for OUTs or the transaction information for INs accumulate in buffers awaiting their service on the downstream bus. The host manages the periodic TT transaction buffers differently than the non-periodic transaction buffers.

11.14.2.1 TT Isochronous/Interrupt (Periodic) Transaction Buffering

Periodic transactions have strict timing requirements to meet on a full-/low-speed bus (as defined by the specific endpoint and transfer type). Therefore, transactions must move across the high-speed bus, through the TT, across the full-/low-speed bus, back through the TT, and onto the high-speed bus in a timely fashion. An overview of the microframe pipeline of buffering in the TT is shown in Figure 11-26. A transaction begins as a start-split on the high-speed bus, is accepted by the high-speed handler, and is stored in the start-split transaction buffer. The full-/low-speed handler uses the next start-split transaction at the head of the start-split transaction buffer when it is time to issue the next periodic full-/low-speed transaction on the downstream bus. The results of the transaction are accumulated in the complete-split transaction buffer. The TT responds to a complete-split from the host and extracts the appropriate response from the complete-split transaction buffer. This completes the flow for a periodic transaction through the TT. This is called the periodic transaction pipeline.

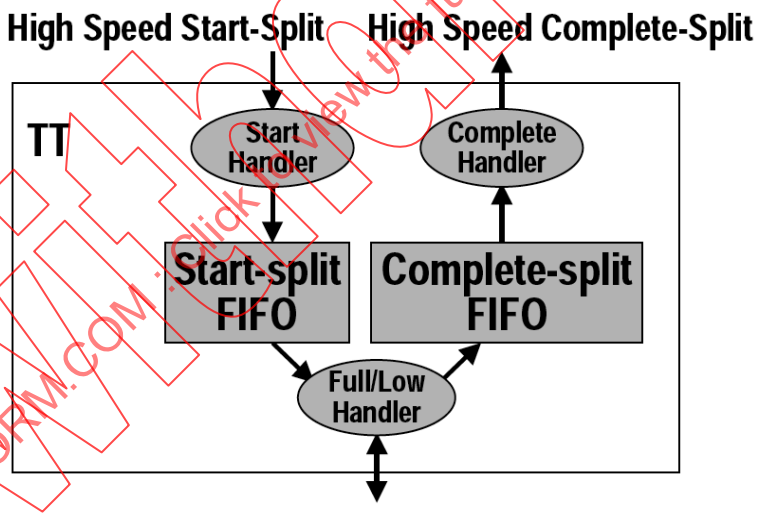


Figure 11-26. TT Microframe Pipeline for Periodic Split Transactions

The TT implements a traditional pipeline of transactions with its periodic transaction buffers. There is separate buffer space for start-splits and complete-splits. The host is responsible for filling the start-split transaction buffer and draining the complete-split transaction buffer. The host software manages the host controller to cause high-speed split transactions at the correct times to avoid over/under runs in the TT periodic transaction buffers. The host controller sends data “just in time” for full-/low-speed OUTs and retrieves response data from full-/low-speed INs to ensure that the periodic transaction buffer space required in the TT is the minimum possible. See Section 11.18 for more detailed information.

USB strictly defines the timing requirements of periodic transactions and the isochronous transport capabilities of the high-speed and full-/low-speed buses. This allows the host to accurately predict when data for periodic transactions must be moved on both the full-/low-

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

speed and high-speed buses, whenever a client requests a data transfer with a full-/low-speed periodic endpoint. Therefore, the host can “pipeline” data to/from the TT so that it moves in a timely manner with its target endpoint. Once the configuration of a full-/low-speed device with periodic endpoints is set, the host streams data to/from the TT to keep the device’s endpoints operating normally.

11.14.2.2 TT Bulk/Control (Non-Periodic) Transaction Buffering

Non-periodic transactions have no timing requirements, but the TT supports the maximum full-/low-speed throughput allowed. A TT provides a few transaction buffers for bulk/control full-/low-speed transactions. The host and TT use simple flow control (NAK) mechanisms to manage the bulk/control non-periodic transaction buffers. The host issues a start-split transaction, and if there is available buffer space, the TT accepts the transaction. The full-/low-speed handler uses the buffered information to issue the downstream full-/low-speed transaction and then uses the same buffer to hold any results (e.g., handshake or data or timeout). The buffer is then emptied with a corresponding high-speed complete-split and the process continues. Figure 11-27 shows an example overview of a TT that has two bulk/control buffers.

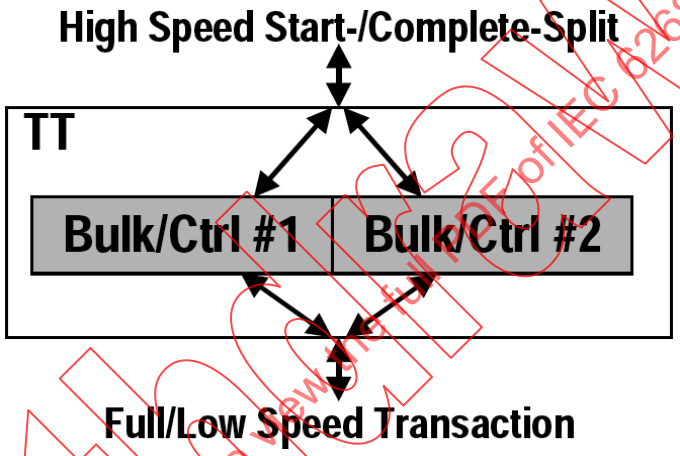


Figure 11-27. TT Nonperiodic Buffering

11.14.2.3 Full-/low-speed Handler Transaction Scheduling

The full-/low-speed handler uses a simple, scheduled priority scheme to service pending transactions on the downstream bus. Whenever the full-/low-speed handler finishes a transaction on the downstream bus, it takes the next start-split transaction from the start-split periodic transaction buffer (if any). If there are no available start-split periodic transactions in the buffer, the full-/low-speed handler may attempt a bulk/control transaction. If there are start-split transactions pending in the bulk/control buffer(s) and there is sufficient time left in the full-/low-speed 1 ms frame to complete the transaction, the full-/low-speed handler issues one of the bulk/control transactions (in round robin order). Figure 11-28 shows pseudo code for the full-/low-speed handler start-split transaction scheduling algorithm.

The TT also sequences the transaction pipeline based on the high-speed microframe timer to ensure that it does not start full-/low-speed periodic transactions too early or too late. The “Advance_pipeline” procedure in the pseudo code is used to keep the TT advancing the microframe “pipeline”. This procedure is described in more detail later in Figure 11-67.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

```
While (1) loop
   While (not end of microframe) loop
      -- process next start-split transaction
      If available periodic start-split transaction then
         Process next full-/low-speed periodic transaction
      Else if (available bulk/control transaction) and
              (fits in full-/low-speed 1 ms frame) then
         Process one transaction
      End if
   End loop

   Advance_Pipeline();  -- see description in Figure 11-67(below)
End loop
```

Figure 11-28. Example Full-/low-speed Handler Scheduling for Start-splits

As described earlier in this chapter, the TT derives the downstream bus's 1 ms SOF timer from the high-speed 125 μs microframe. This means that the host and the TT have the same 1 ms frame time for all TTs. Given the strict relationship between frames and the zeroth microframe, there is no need to have any explicit timing information carried in the periodic split transactions sent to the TT. See Section 11.18 for more information.

11.15 Split Transaction Notation Information

The following sections describe the details of the transaction phases and flow sequences of split transactions for the different USB transfer types: bulk/control, interrupt, and isochronous. Each description also shows detailed example host and TT state machines to achieve the required transaction definitions. The diagrams should not be taken as a required implementation, but to specify the required behavior. Appendix A includes example high-speed and full-speed transaction sequences with different results to clarify the relationships between the host controller, the TT, and a full-speed endpoint.

Low-speed is not discussed in detail since beyond the handling of the PRE packet (which is defined in Chapter 8), there are no packet sequencing differences between low- and full-speed.

For each data transfer direction, reference figures also show the possible flow sequences for the start-split and the complete-split portion of each split transaction transfer type.

The transitions on the flow sequence figures have labels that correspond to the transitions in the host and TT state machines. These labels are also included in the examples in Appendix A. The three character labels are of the form: < S | C >< T | D | H | E ><number>. S indicates that this is a start-split label. C indicates that this is a complete-split label. T indicates token phase; D indicates data phase; H indicates handshake phase; E indicates an error case. The number simply distinguishes different labels of the same case/phase in the same split transaction part.

The flow sequence figures further identify the visibility of transitions according to the legend in Figure 11-29. The flow sequences also include some indication of states required in the host or TT or actions taken. The legend shown in Figure 11-29 indicates how these are identified.

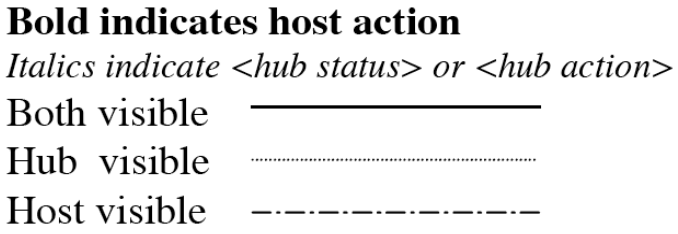


Figure 11-29. Flow Sequence Legend

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 11-30 shows the legend for the state machine diagrams. A circle with a three line border indicates a reference to another (hierarchical) state machine. A circle with a two line border indicates an initial state. A circle with a single line border is a simple state.

A diamond (joint) is used to join several transitions to a common point. A joint allows a single input transition with multiple output transitions or multiple input transitions and a single output transition. All conditions on the transitions of a path involving a joint must be true for the path to be taken. A path is simply a sequence of transitions involving one or more joints.

A transition is labeled with a block with a line in the middle separating the (upper) condition and the (lower) actions. The condition is required to be true to take the transition. The actions are performed if the transition is taken. The syntax for actions and conditions is VHDL. A circle includes a name in bold and optionally one or more actions that are performed upon entry to the state.

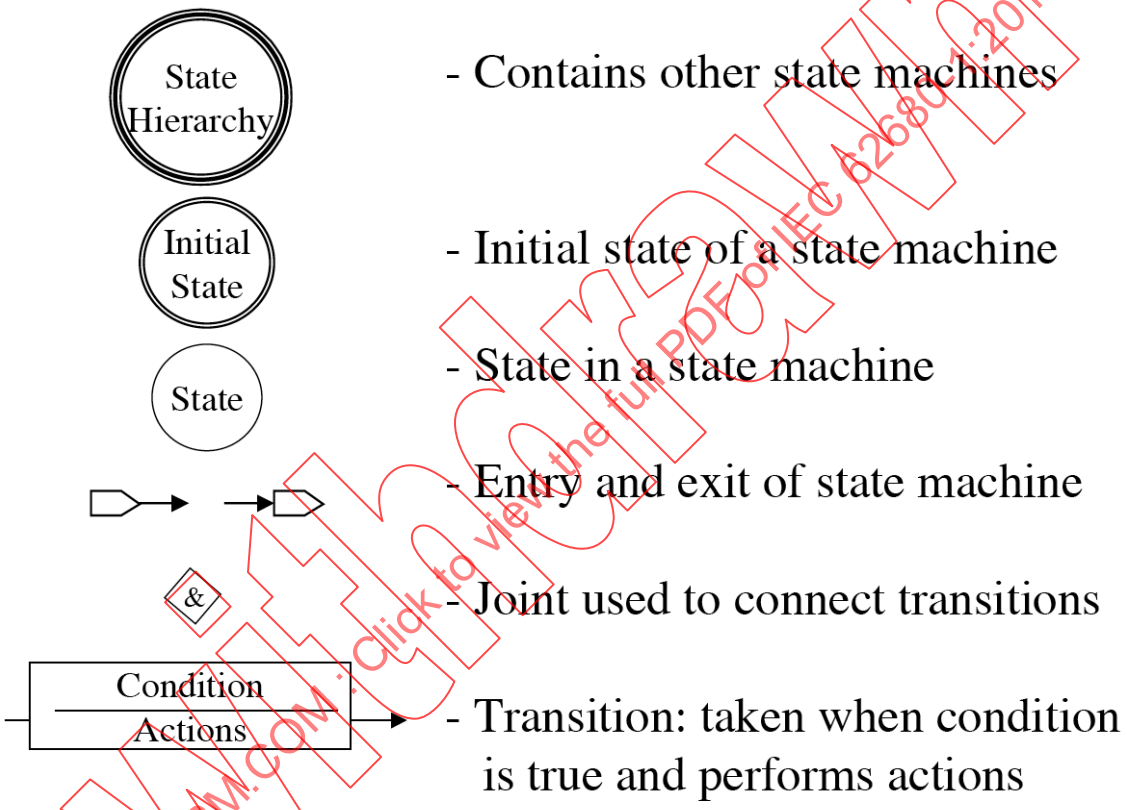


Figure 11-30. Legend for State Machines

The descriptions of the split transactions for the four transfer types refer to the status of the full-/low-speed transaction on the bus downstream of the TT. This status is used by the high-speed handler to determine its response to a complete-split transaction. The status is only visible within a TT implementation and is used in the specification purely for ease of explanation. The defined status values are:

- Ready – The transaction has completed on the downstream facing full-/low-speed bus with the result as follows:
  - Ready/NAK – A NAK handshake was received.
  - Ready /trans_err – The full-/low-speed transaction experienced a error in the transaction. Possible errors are: PID to PID_invert bits check failure, CRC5 check failure, incorrect PID, timeout, CRC16 check failure, incorrect packet length, bitstuffing error, false EOP.
  - Ready /ACK – An ACK handshake was received.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

  - Ready /Stall – A STALL handshake was received.
  - Ready /Data – A data packet was received and the CRC check passed. (bulk/control IN).
  - Ready /lastdata – A data packet was finished being received. (isochronous/interrupt IN).
  - Ready /moredata – A data packet was being received when the microframe timer occurred (isochronous/interrupt IN).
- Old – A complete-split has been received by the high-speed handler for a transaction that previously had a “ready” status. The possible status results are the same as for the Ready status. This is the initial state for a buffer before it has been used for a transaction.
- Pending – The transaction is waiting to be completed on the downstream facing full-/low-speed bus.

The figures use “old/x” and “ready/x” to indicate any of the old or ready status respectively.

The split transaction state machines in the remainder of this chapter are presented in the context of Figure 11-31. The host controller state machines are located in the host controller. The host controller causes packets to be issued downstream (labeled as HSD1) and it receives upstream packets (labeled as HSU2).

The transaction translator state machines are located in the TT. The TT causes packets to be issued upstream (labeled as HSU1) and it receives downstream packets (labeled as HSD2).

The host controller has commands that tell it what split transaction to issue next for an endpoint. The host controller tracks transactions for several endpoints. The TT has state in buffers that track transactions for several endpoints.

Appendix B includes some declarations that were used in constructing the state machines and may be useful in understanding additional details of the state machines. There are several pseudo-code procedures and functions for conditions and actions. Simple descriptions of them are also included in Appendix B.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

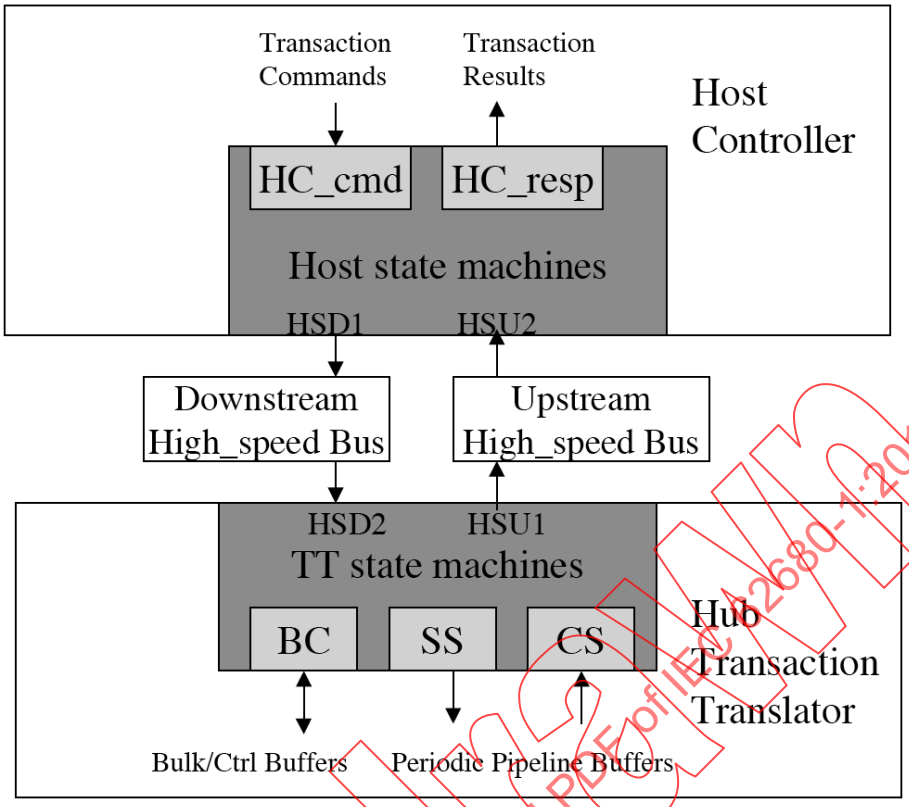


Figure 11-31. State Machine Context Overview

11.16 Common Split Transaction State Machines

There are several state machines common to all the specific split transaction types. These state machines are used in the host controller and transaction translator to determine the specific split transaction type (e.g., interrupt OUT start-split vs. bulk IN complete-split). An overview of the host controller state machine hierarchy is shown in Figure 11-32. The overview of the transaction translator state machine hierarchy is shown in Figure 11-33. Each of the labeled boxes in the figures show an individual state machine. Boxes contained in another box indicate a state machine contained within another state machine. All the state machines except the lowest level ones are shown in the remaining figures in this section. The lowest level state machines are shown in later sections describing the specific split transaction type.

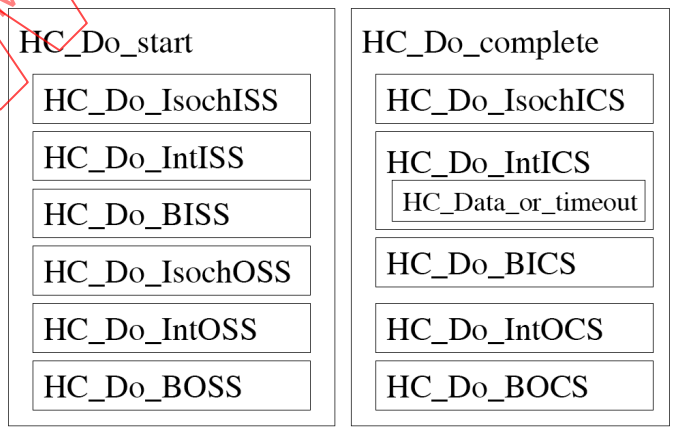


Figure 11-32. Host Controller Split Transaction State Machine Hierarchy Overview

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

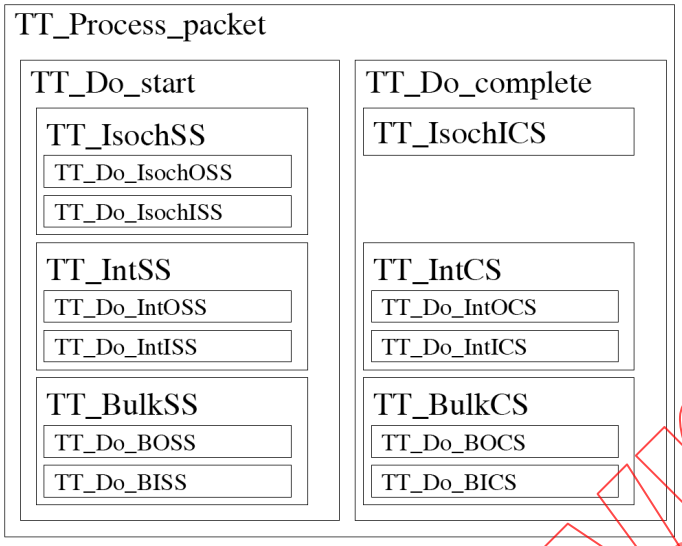


Figure 11-33. Transaction Translator State Machine Hierarchy Overview

11.16.1 Host Controller State Machine

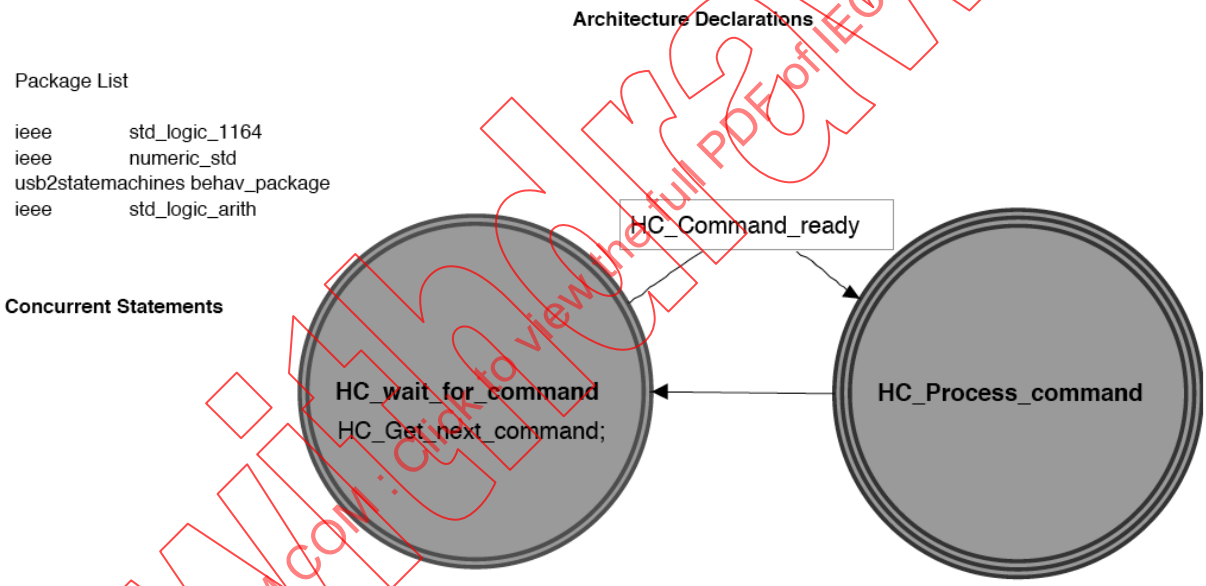


Figure 11-34. Host Controller

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.1.1 HC_Process_command State Machine

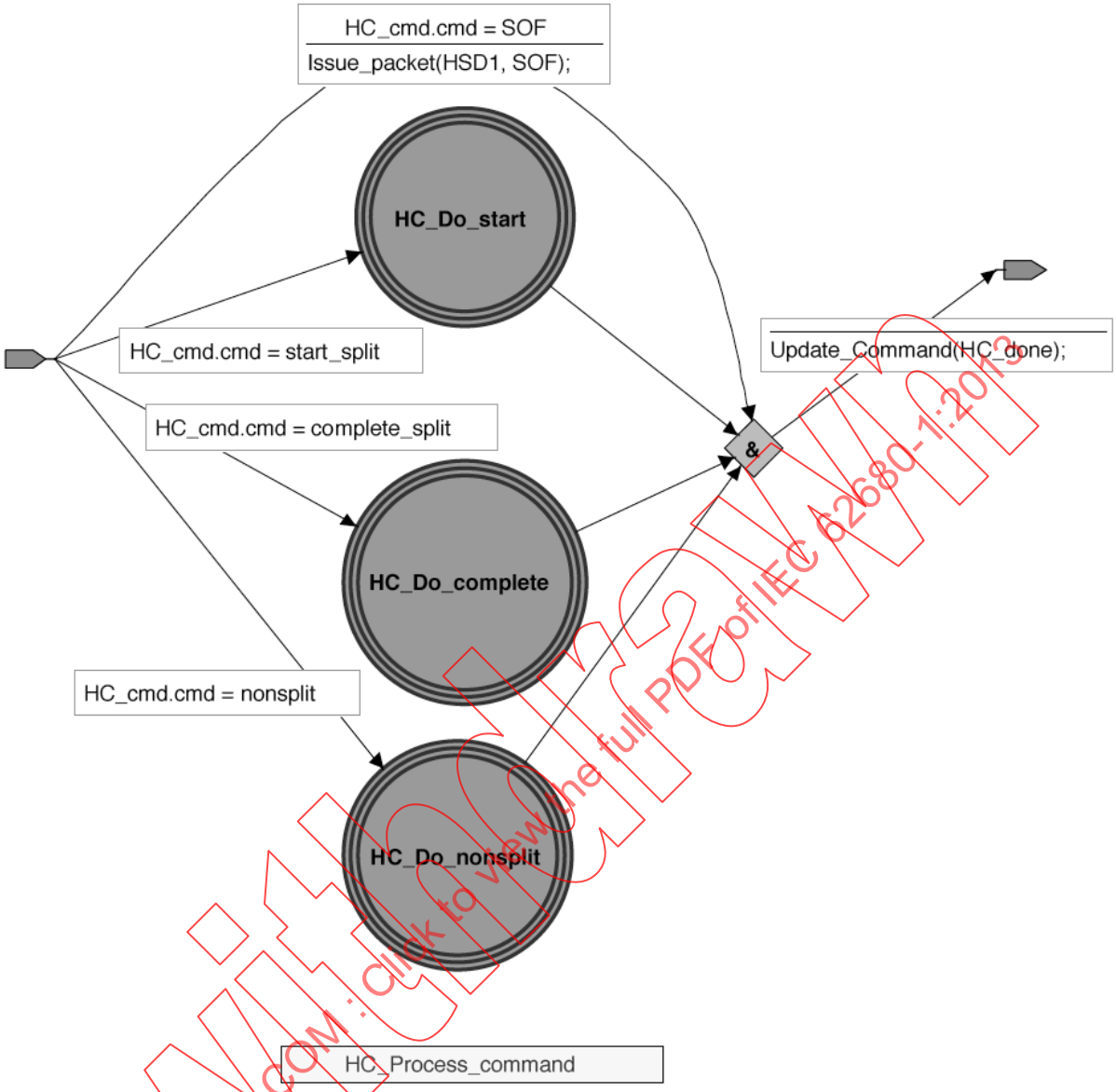


Figure 11-35. HC_Process_Command

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.1.1.1 HC_Do_start State Machine

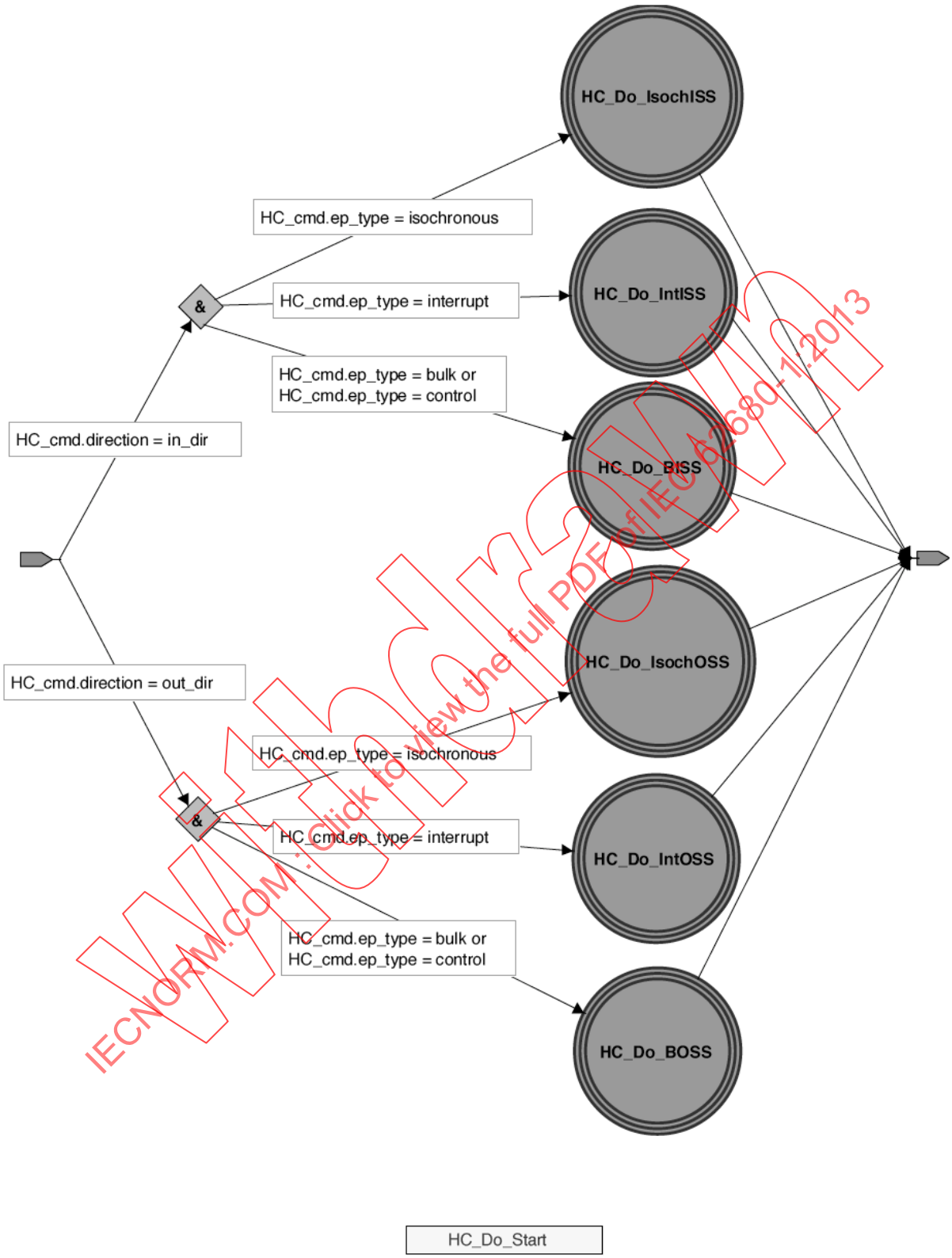
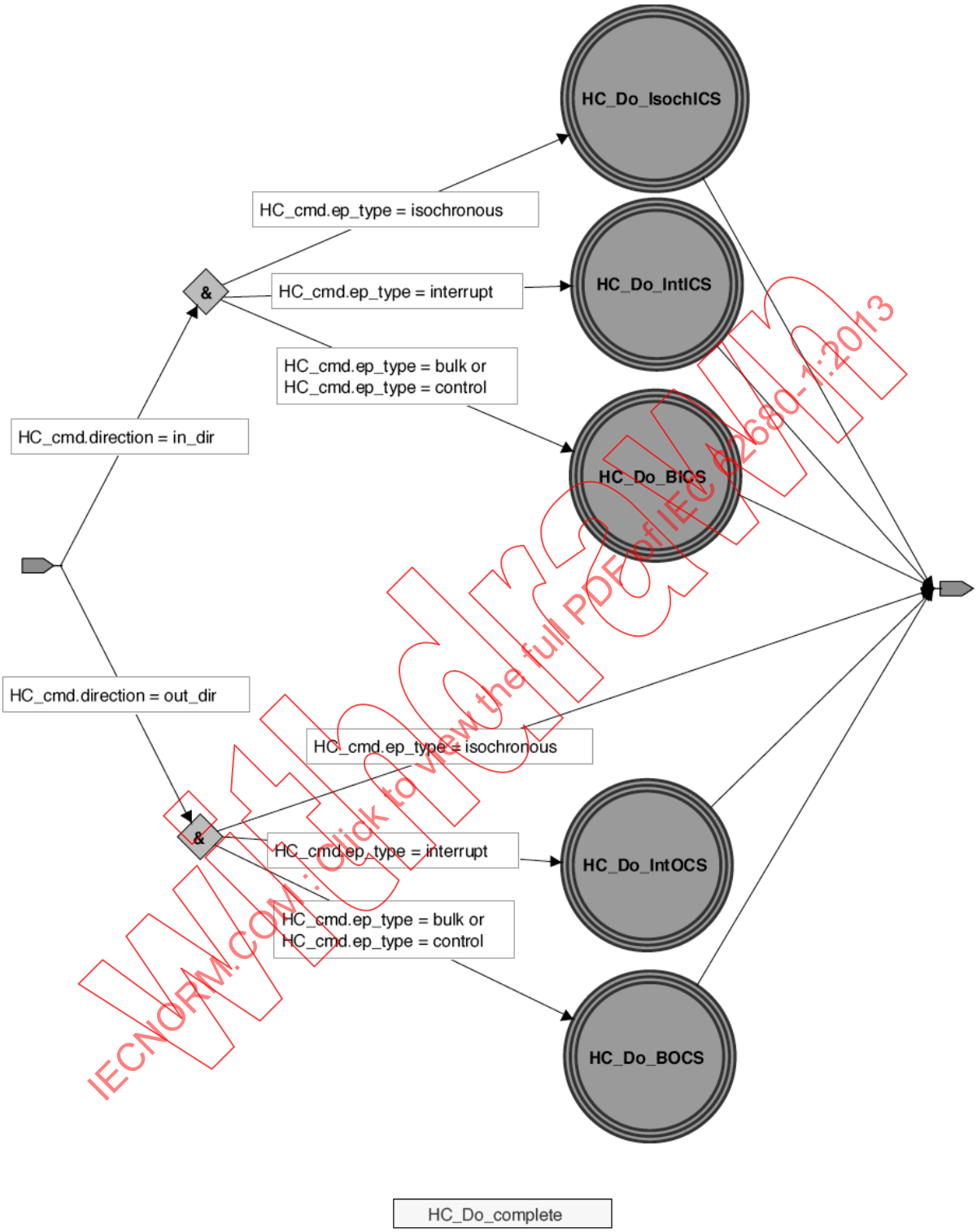


Figure 11-36. HC_Do_Start

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.16.1.1.2 HC_Do_complete State Machine



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 11-37. HC_Do_Complete

11.16.2 Transaction Translator State Machine

Architecture Declarations

Package List

ieee std_logic_1164
ieee numeric_std
usb2statemachines behav_package

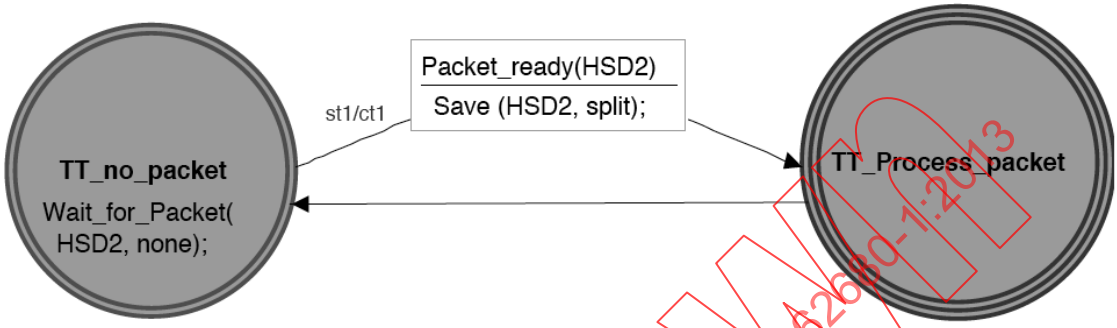


Figure 11-38. Transaction Translator

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.2.1 TT_Process_packet State Machine

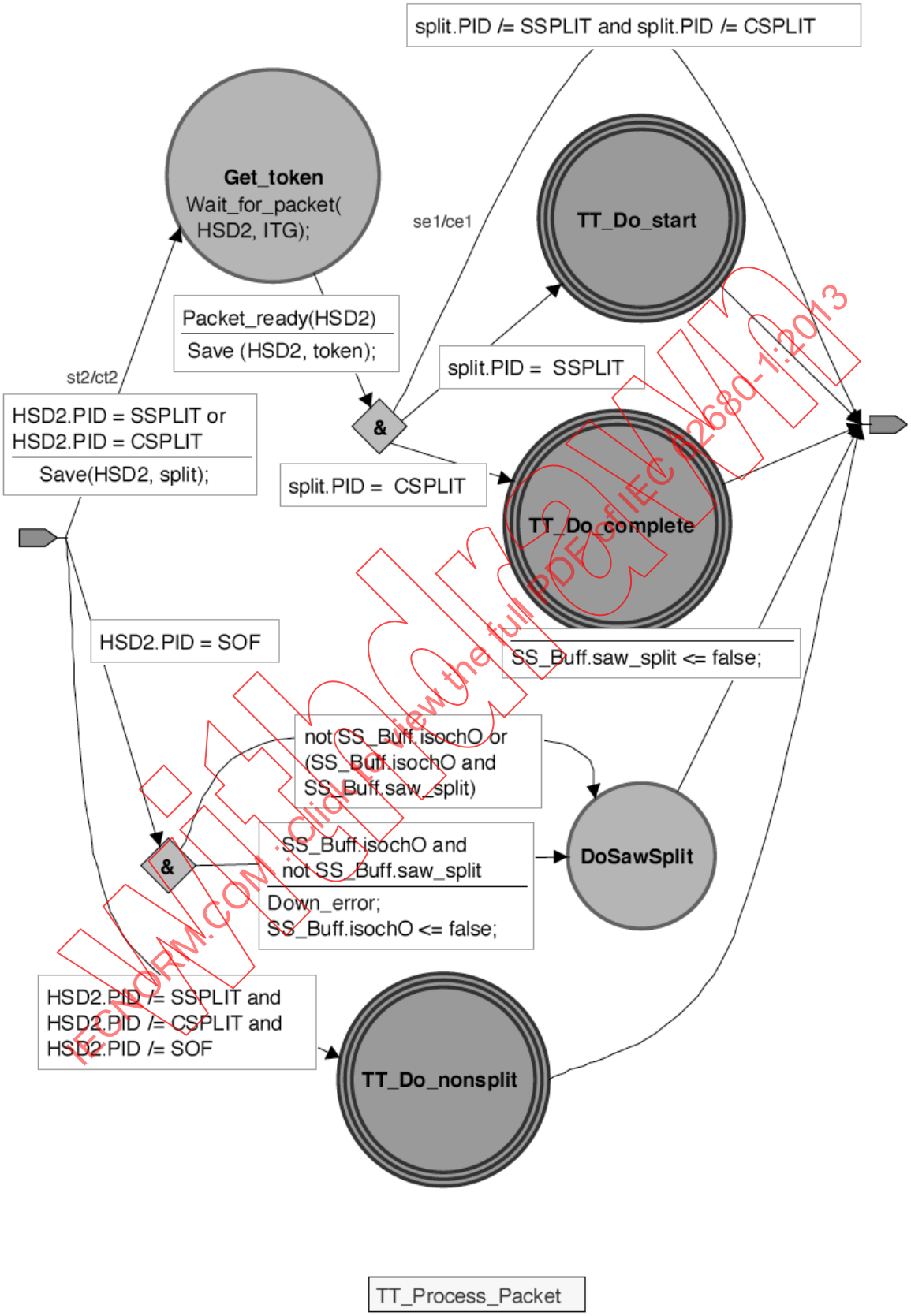


Figure 11-39. TT_Process_Packet

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.2.1.1 TT_Do_Start State Machine

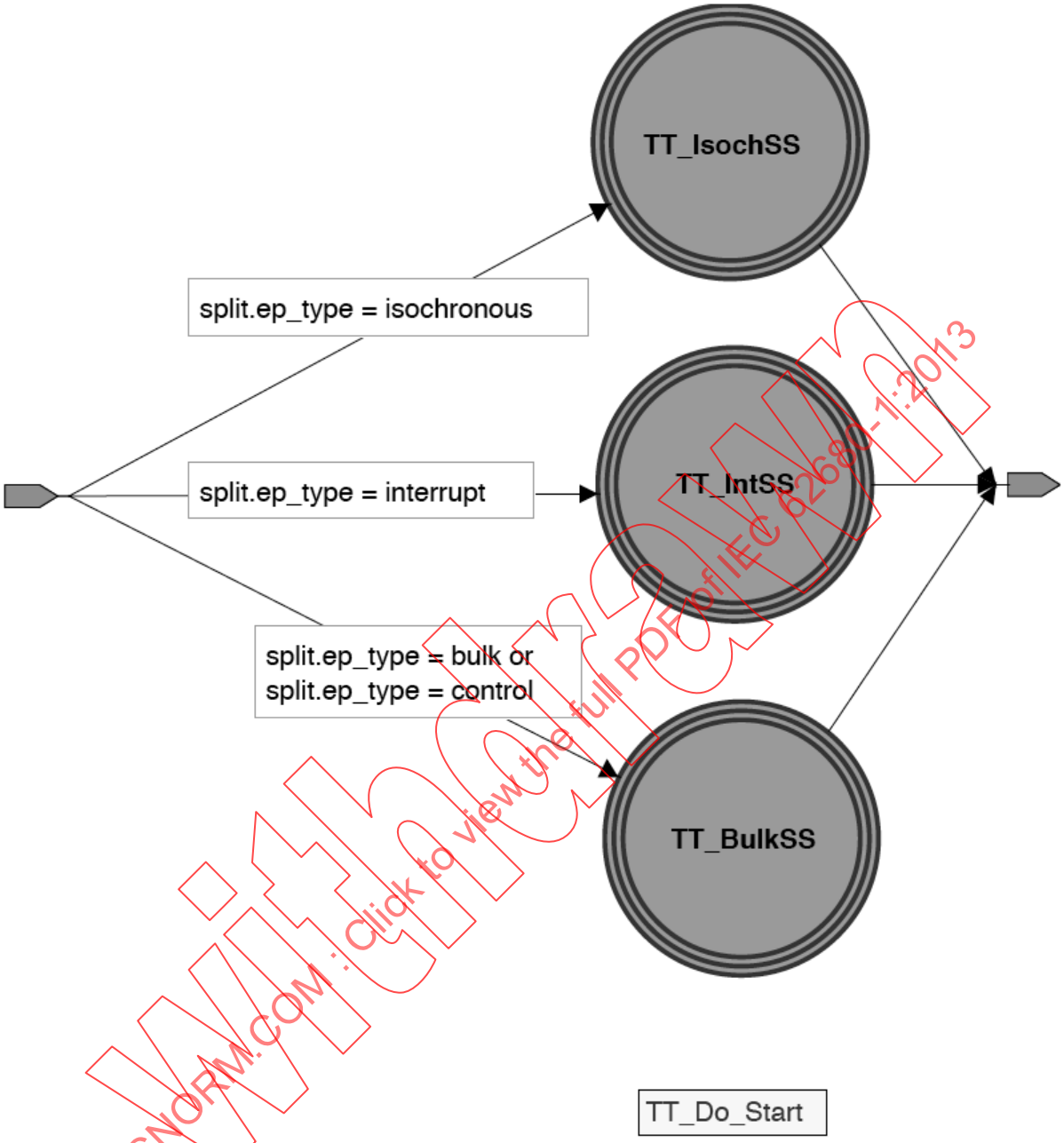


Figure 11-40. TT_Do_Start

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.16.2.1.2 TT_Do_Complete State Machine

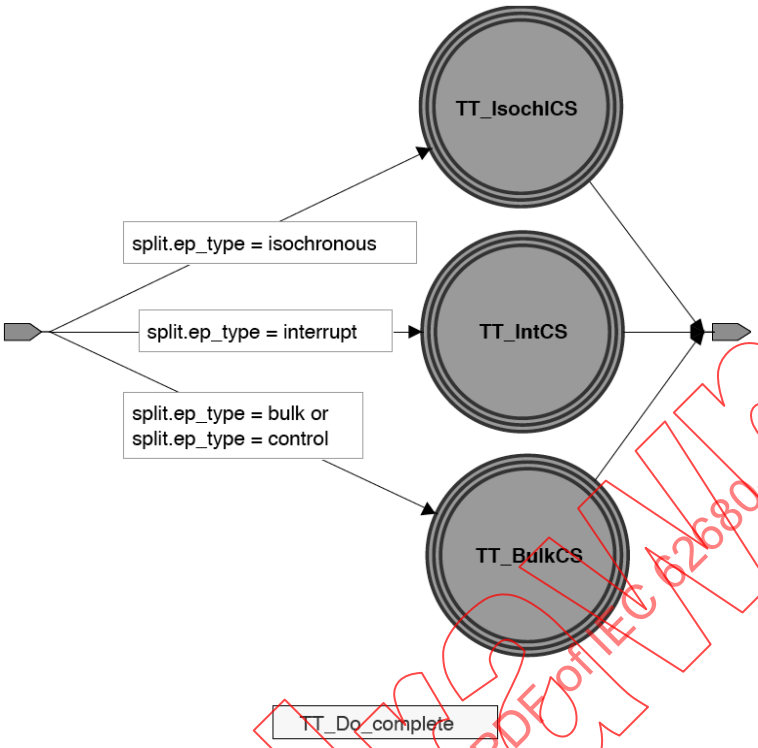


Figure 11-41. TT_Do_Complete

11.16.2.1.3 TT_BulkSS State Machine

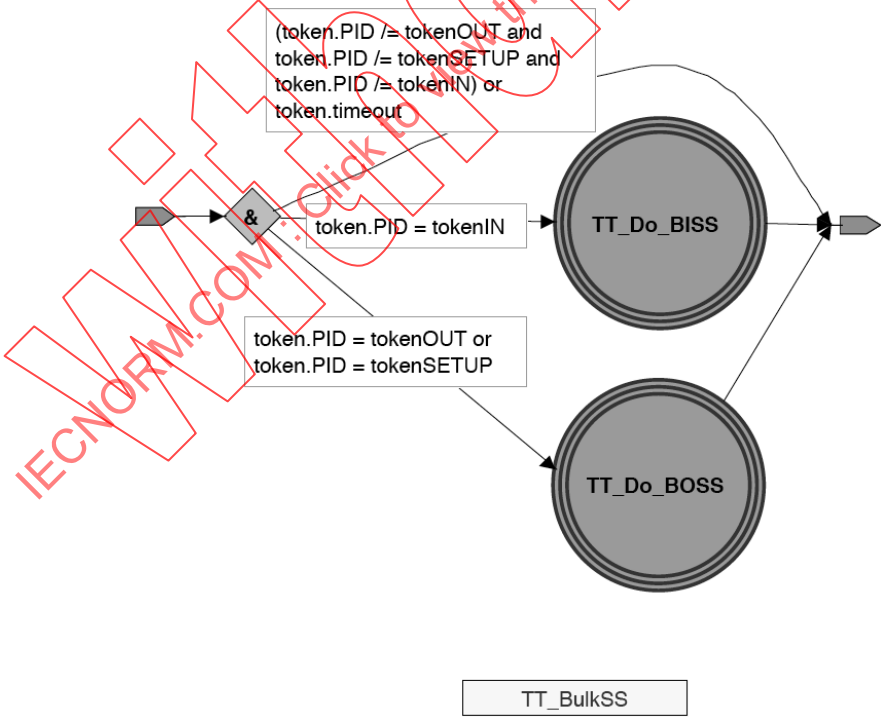


Figure 11-42. TT_BulkSS

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.16.2.1.4 TT_BulkCS State Machine

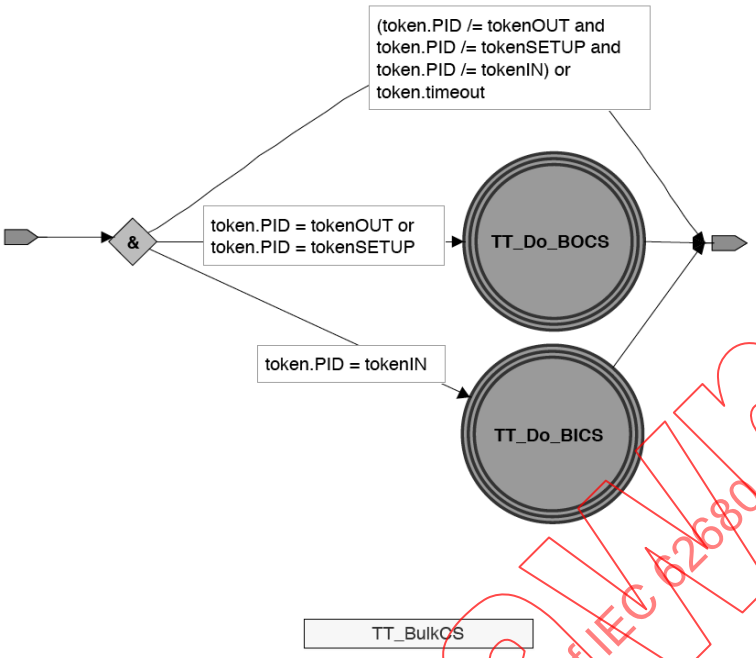


Figure 11-43. TT_BulkCS

11.16.2.1.5 TT_IntSS State Machine

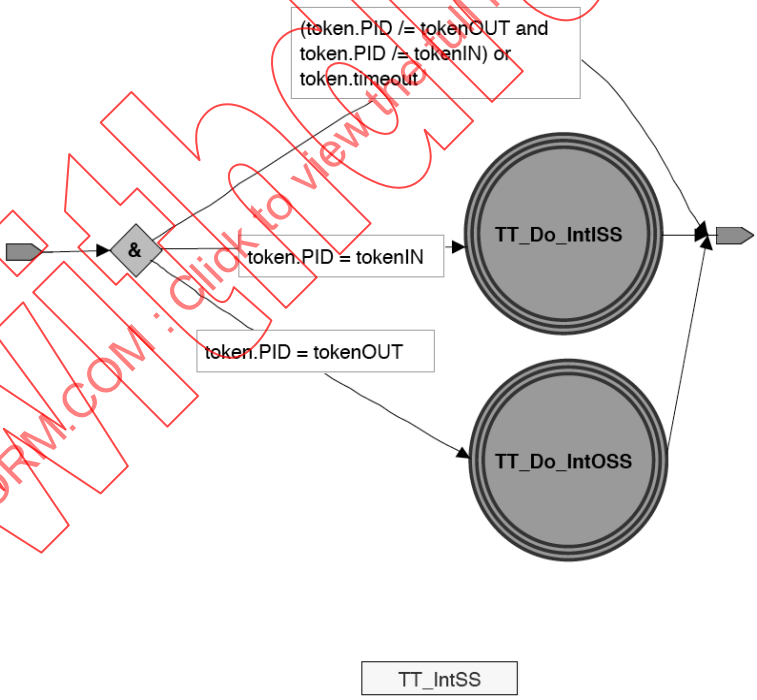


Figure 11-44. TT_IntSS

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.2.1.6 TT_IntCS State Machine

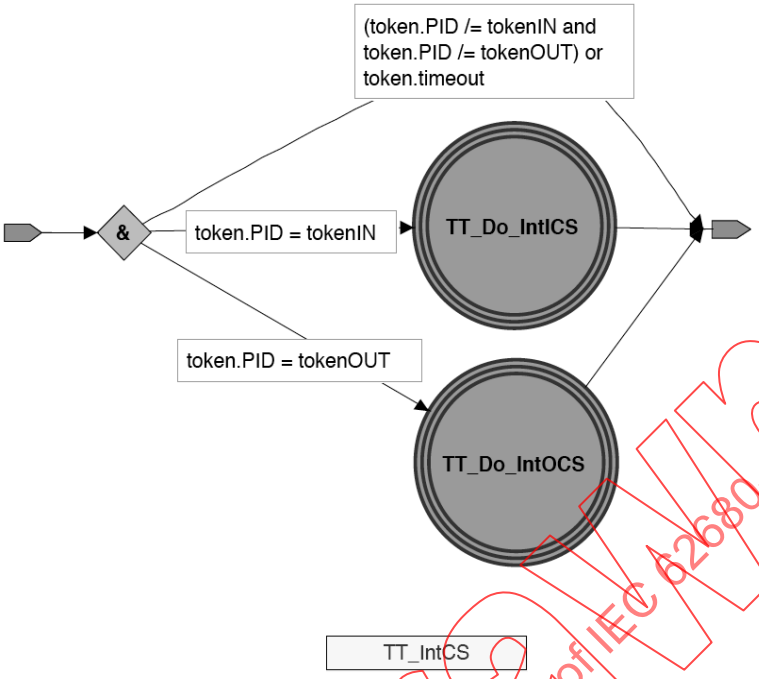


Figure 11-45. TT_IntCS

11.16.2.1.7 TT_IsochSS State Machine

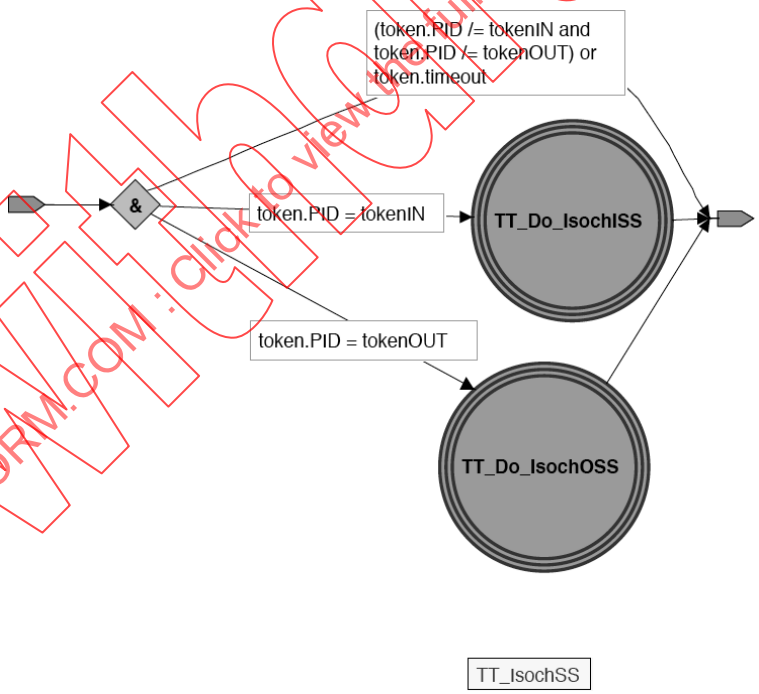


Figure 11-46. TT_IsochSS

11.17 Bulk/Control Transaction Translation Overview

Each TT must have at least two bulk/control transaction buffers. Each buffer holds the information for a start- or complete-split transaction and represents a single full-/low-speed transaction that is awaiting (or has completed) transfer on the downstream bus. The buffer is used to hold the transaction information from the start-split (and data for an OUT) and then the handshake/result of the full-/low-speed transaction (and data for an IN). This buffer is

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

filled and emptied by split transactions from the high-speed bus via the high-speed handler. The buffer is also updated by the full-/low-speed handler while the transaction is in progress on the downstream bus.

The high-speed handler must accept a start-split transaction from the host controller for a bulk/control endpoint whenever the high-speed handler has appropriate space in a bulk/control buffer.

The host controller attempts a start-split transaction according to its bulk/control high-speed transaction schedule. As soon as the high-speed handler responds to a complete-split transaction with the results from the corresponding buffer, the next start-split for some (possibly other) full-/low-speed endpoint can be saved in the buffer.

There is no method to control the start-split transaction accepted next by the high-speed handler. Sequencing of start-split transactions is simply determined by available TT buffer space and the current state of the host controller schedule (e.g., which start-split transaction is next that the host controller tries as a normal part of processing high-speed transactions).

The host controller does not need to segregate split transaction bulk (or control) transactions from high-speed bulk (control) transactions when building its schedule. The host controller is required to track whether a transaction is a normal high-speed transaction or a high-speed split transaction.

The following sections describe the details of the transaction phases, flow sequences, and state machines for split transactions used to support full-/low-speed bulk and control OUT and IN transactions. There are only minor differences between bulk and control split transactions. In the figures, some areas are shaded to indicate that they do not apply for control transactions.

11.17.1 Bulk/Control Split Transaction Sequences

The state machine figures show the transitions required for high-speed split transactions for full-/low-speed bulk/control transfer types for a single endpoint. These figures must not be interpreted as showing any particular specific timing. They define the required sequencing behavior of different packets of a bulk/control split transaction. In particular, other high-speed or split transactions for other endpoints occur before or after these split transaction sequences.

Figure 11-47 shows a sample code algorithm that describes the behavior of the transitions labeled with *Is_new_SS*, *Is_old_SS* and *Is_no_space* shown in the figures for both bulk/control IN and OUT start-split transactions buffered in the TT for any endpoint. This algorithm ensures that the TT only buffers a single bulk/control split transaction for any endpoint. The complete-split protocol definition requires an endpoint has only a single result buffered in the TT at any time. Note that the “buffer match” test is different for bulk and control endpoints. A buffer match test for a bulk transaction must include the direction of the transaction in the test since bulk endpoints are unidirectional. A control transaction must not use direction as part of the match test.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

```
procedure Compare_buffs IS
      variable match:boolean:=FALSE;
begin
--
--  Is_new_SS is true when BC_buff.status == NEW_SS
--  Is_old_SS is true when BC_buff.status == OLD_SS
--  Is_no_space is true when BC_buff.status == NO_SPACE
--
       -- Assume nospace and intialize index to 0.
       BC_buff.status := NO_SPACE;
       BC_buff.index  := 0;

       FOR i IN 0 to num_buffs-1 LOOP
           IF NOT match THEN
               -- Re-use buffer with same Device Address/End point.
               IF (token.endpt = cam(i).store.endpt AND
                   token.dev_addr = cam(i).store.dev_addr AND
                   ((token.direction = cam(i).store.direction AND
                     split.ep_type /= CONTROL) OR
                    split.ep_type = CONTROL)) THEN

                 -- If The buffer is already pending/ready this must be a retry.
                 IF (cam(i).match.state = READY OR cam(i).match.state = PENDING) THEN
                     BC_buff.status := OLD_SS;
                 ELSE
                     BC_buff.status := NEW_SS;
                 END IF;
                 BC_buff.index := i;
                 match := TRUE;

               -- Otherwise use the buffer if it's old.
               ELSIF (cam(i).match.state = OLD) THEN
                 BC_buff.status := NEW_SS;
                 BC_buff.index := i;
               END IF;
           END IF;
       END LOOP;

    end Compare_buffs;
```

Figure 11-47. Sample Algorithm for Compare_buffs

Figure 11-48 shows the sequence of packets for a start-split transaction for the full-/low-speed bulk OUT transfer type. The block labeled SSPLIT represents a split transaction token packet as described in Chapter 8. It is followed by an OUT token packet (or SETUP token packet for a control setup transaction). If the high-speed handler times out after the SSPLIT or OUT token packets, and does not receive the following OUT/SETUP or DATA0/1 packets, it will not respond with a handshake as indicated by the dotted line transitions labeled “se1” or “se2”. This causes the host to subsequently see a transaction error (timeout) (labeled “se2” and indicated with a dashed line). If the high-speed handler receives the DATA0/1 packet and it fails the CRC check, it takes the transition “se2” which causes the host to timeout and follow the “se2” transition.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

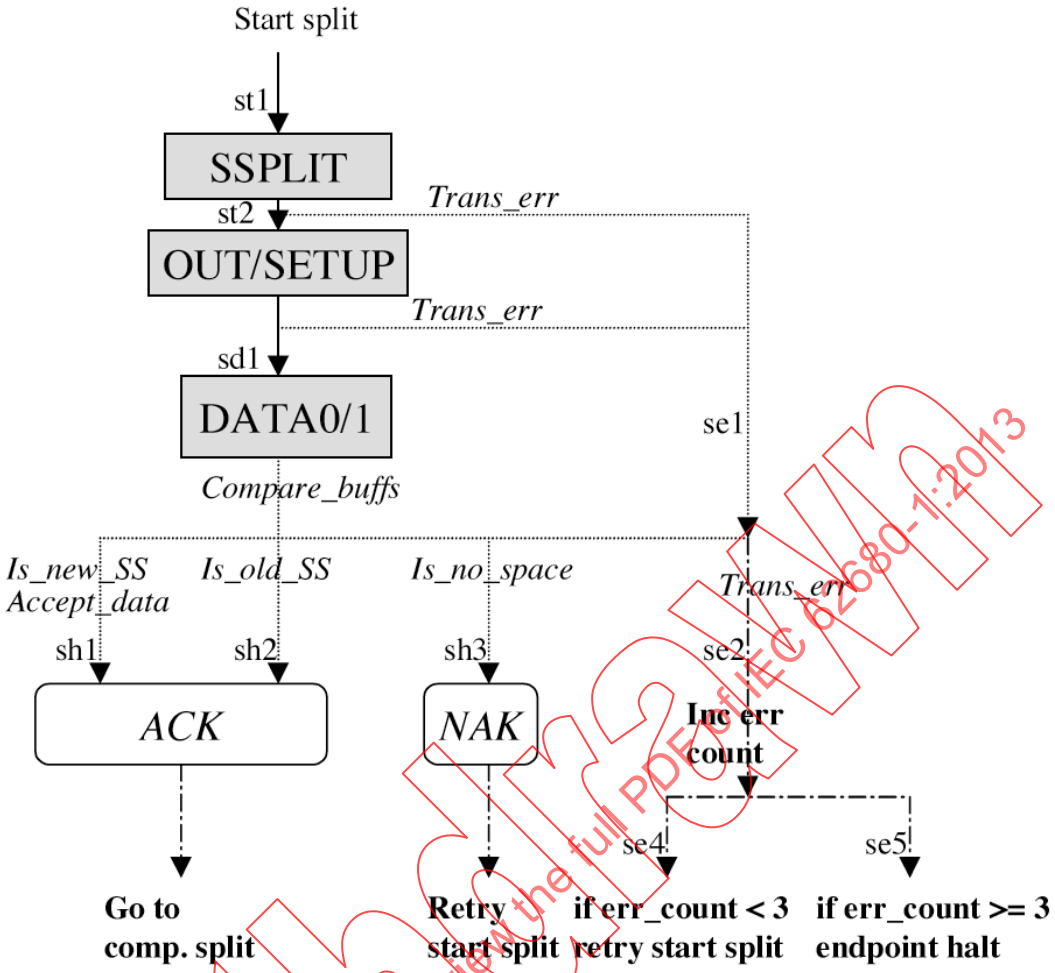


Figure 11-48. Bulk/Control OUT Start-split Transaction Sequence

The host must keep retrying the start-split for this endpoint until the err_count reaches three for this endpoint before continuing on to some other start-split for this endpoint. However, the host can issue other start-splits for other endpoints before it retries the start-split for this endpoint. The err_count is used to count how many errors have been experienced during attempts to issue a particular transaction for a particular endpoint.

If there is no space in the transaction buffers to hold the start-split, the high-speed handler responds with a NAK via transition “sh3”. This will cause the host to retry this start-split at some future time based on its normal schedule. The host does not increase its err_count for a NAK handshake response. Once the host has received a NAK response to a start-split, it can skip other start-splits for this TT for bulk/control endpoints until it finishes a bulk/control complete-split.

If there is buffer space for the start-split, the high-speed handler takes transition “sh1” and responds with an ACK. This tells the host it must try a complete-split the next time it attempts to process a transaction for this full-/low-speed endpoint. After receiving an ACK handshake, the host must not issue a further start-split for this endpoint until the corresponding complete-split has been completed.

If the high-speed handler already has a start-split for this full-/low-speed endpoint pending or ready, it follows transition “sh2” and also responds with an ACK, but ignores the data. This handles the case where an ACK handshake was smashed and missed by the host controller and now the host controller is retrying the start-split; e.g., a high-speed handler transition of “sh1” but a host transition of “se2”.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

In the host controller error cases, the host controller implements the “three strikes and you’re out” mechanism. That is, it increments an error count (err_count) and, if the count is less than three (transition “se4”), it will retry the transaction. If the err_count is greater or equal to three (transition “se5”), the host controller does endpoint halt processing and does not retry the transaction. If for some reason, a host memory or non-USB bus delay (e.g.,a system memory “hold off”) occurs that causes the transaction to not be completed normally, the err_count must not be incremented. Whenever a transaction completes normally, the err_count is reset to zero.

The high-speed handler in the TT has no immediate knowledge of what the host sees, so the “se2”, “se4”, and “se5” transitions show only host visibility.

This packet flow sequence showing the interactions between the host and hub is also represented by host and high-speed handler state machine diagrams in the next section. Those state machine diagrams use the same labels to correlate transitions between the two representations of the split transaction rules.

Figure 11-49 shows the corresponding flow sequence for the complete-split transaction for the full-/low-speed bulk/control OUT transfer type. The notation “ready/x” or “old/x” indicates that the transaction status of the split transaction is any of the ready or old states. After a full-/low-speed transaction is run on the downstream bus, the transaction status is updated to reflect the result of the transaction. The possible result status is: nak, stall, ack. The “x” means any of the NAK, ACK, STALL full-/low-speed transaction status results. Each status result reflects the handshake response from the full-/low-speed transaction.

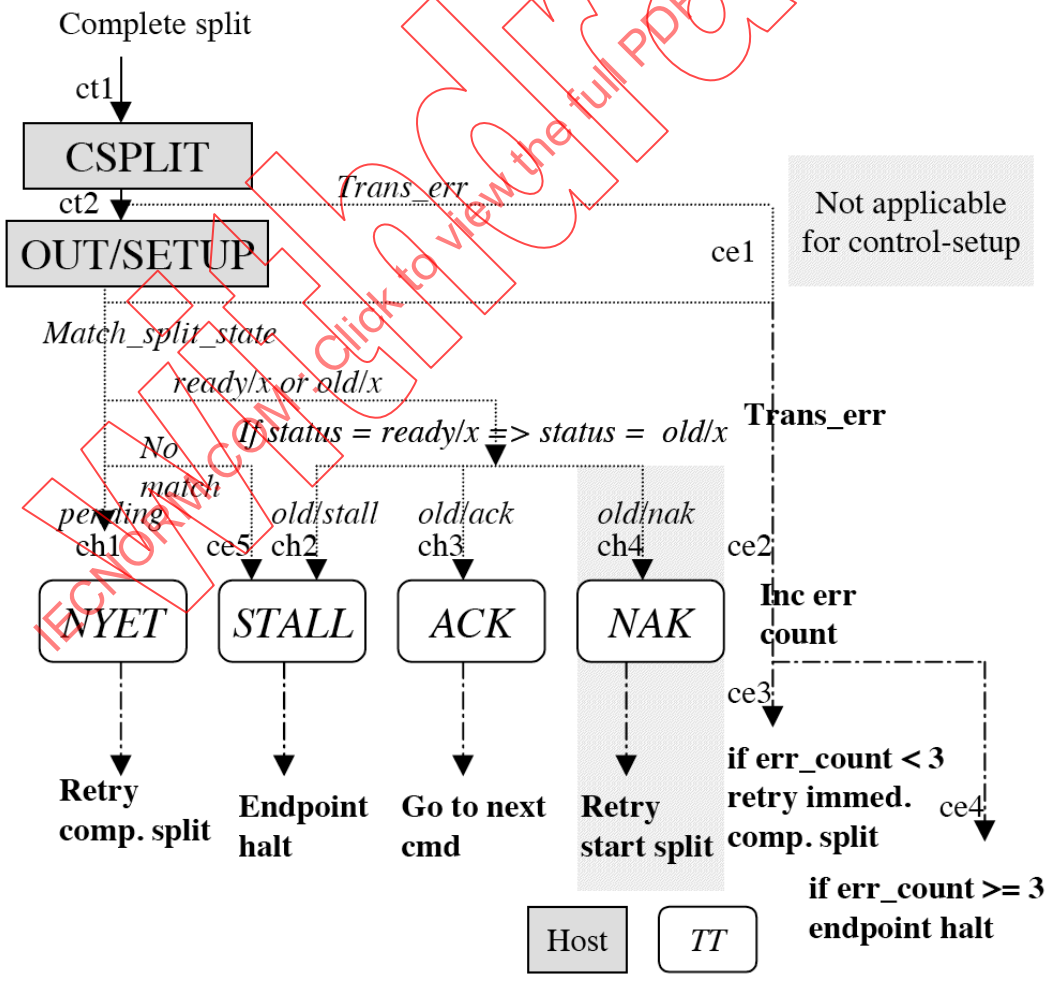


Figure 11-49. Bulk/Control OUT Complete-split Transaction Sequence

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

There is no timeout response status for a transaction because the full-/low-speed handler must perform a local retry of a full-/low-speed bulk or control transaction that experiences a transaction error. It locally implements a “three strikes and you’re out” retry mechanism. This means that the full-/low-speed transaction will resolve to one of a NAK, STALL or ACK handshake results. If the transaction experiences a transaction error three times, the full-/low-speed handler will reflect this as a stall status result. The full-/low-speed handler must not do a local retry of the transaction in response to an ACK, NAK, or STALL handshake.

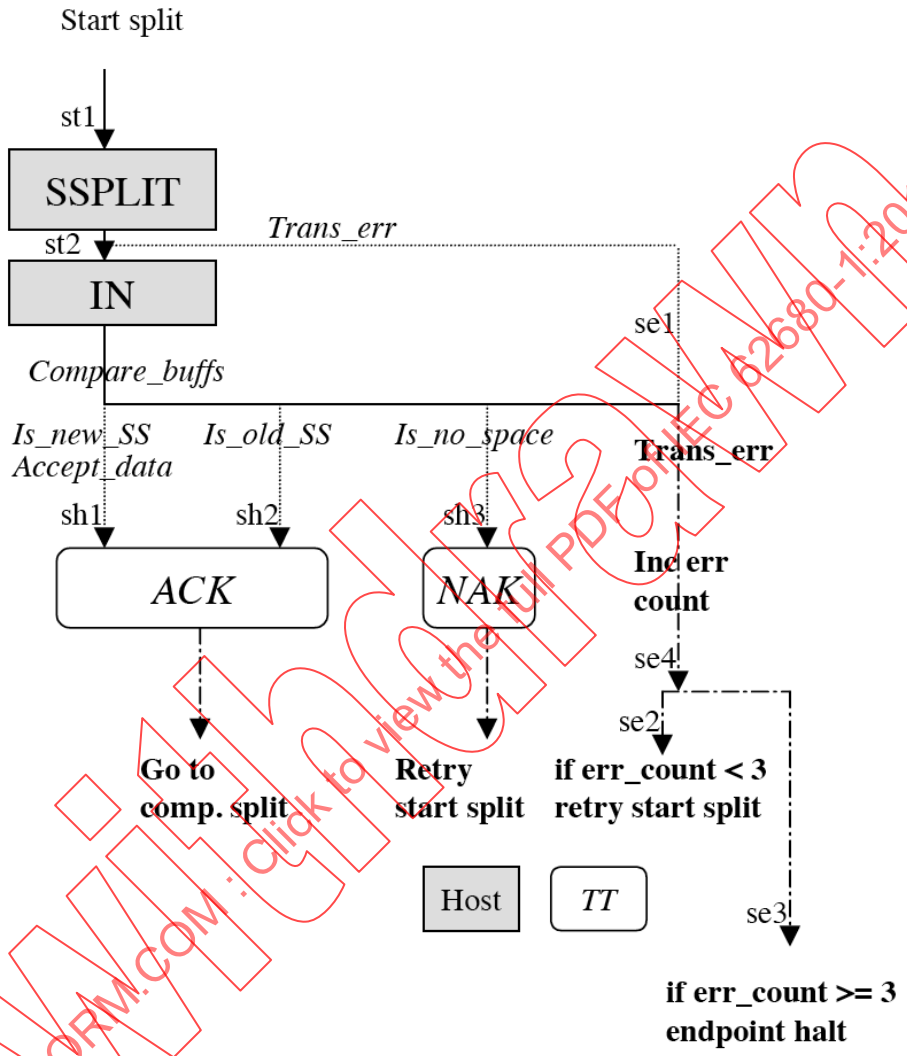


Figure 11-50. Bulk/Control IN Start-split Transaction Sequence

If the high-speed handler receives the complete-split token packet (and the token packet) while the full-/low-speed transaction has not been completed (e.g., the transaction status is “pending”), the high-speed handler responds with a NYET handshake. This causes the host to retry the complete-split for this endpoint some time in the future.

If the high-speed handler receives a complete-split token packet (and the token packet) and finds no local buffer with a corresponding transaction, the TT responds with a STALL to indicate a protocol violation.

Once the full-/low-speed handler has finished a full-/low-speed transaction, it changes the transaction status from pending to ready and saves the transaction result. This allows the high-speed handler to respond to the complete-split transaction with something besides NYET. Once the high-speed handler has seen a complete-split, it changes the transaction status

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

from ready/x to old/x. This allows the high-speed handler to reuse its local buffer for some other bulk/control transaction after this complete-split is finished.

If the host times out the transaction or does not receive a valid handshake, it immediately retries the complete-split before going on to any other bulk/control transactions for this TT. The normal “three strikes” mechanism applies here also for the host; i.e., the err_count is incremented. If for some reason, a host memory or non-USB bus delay (e.g., a system memory “hold off”) occurs that causes the transaction to not be completed normally, the err_count must not be incremented.

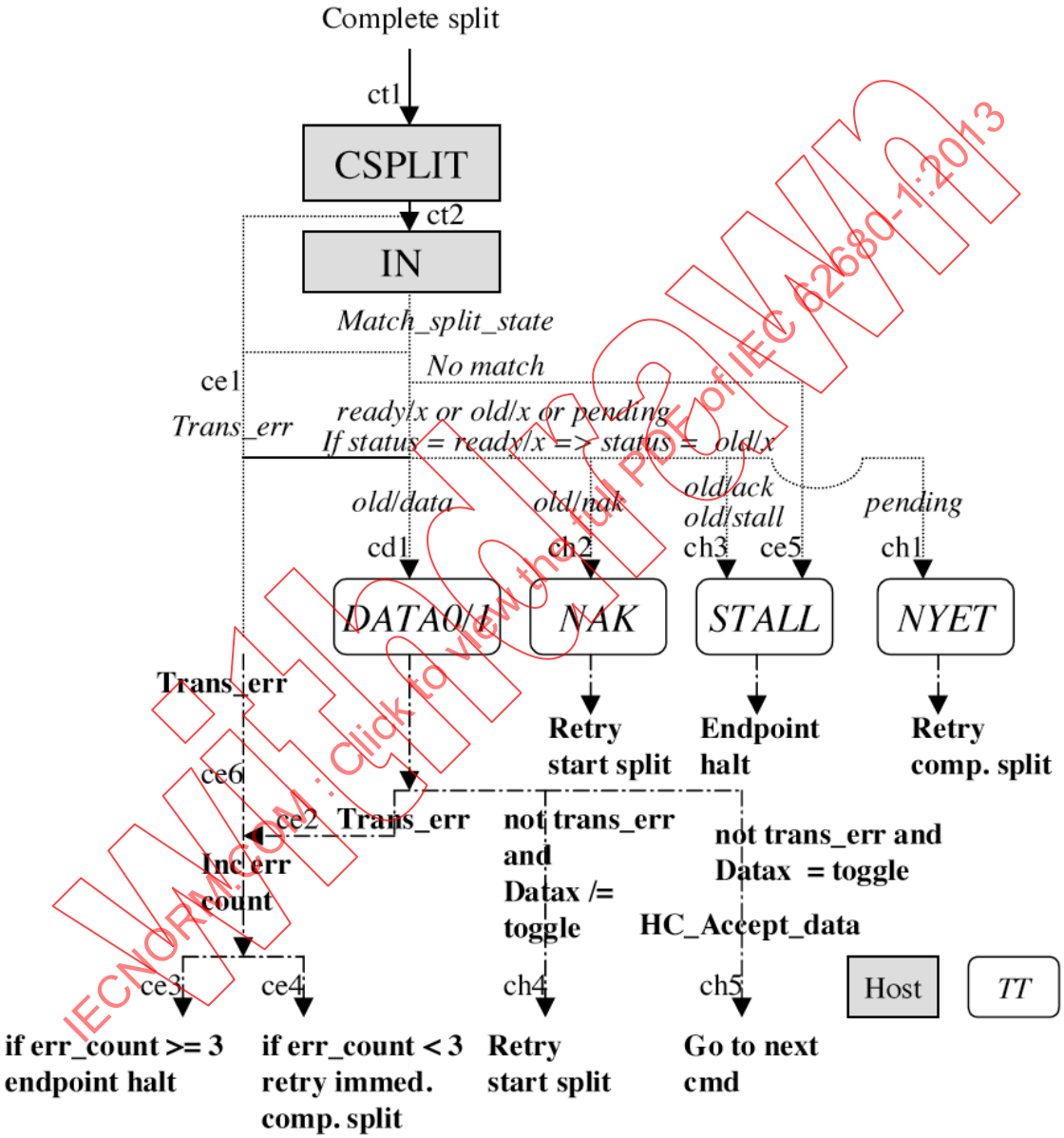


Figure 11-51. Bulk/Control IN Complete-split Transaction Sequence

If the host receives a STALL handshake, it performs endpoint halt processing and will not issue any more split transactions for this full-/low-speed endpoint until the halt condition is removed.

If the host receives an ACK, it records the results of the full-/low-speed transaction and advances to the next split transaction for this endpoint. The next transaction will be issued at some time in the future according to normal scheduling rules.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

If the host receives a NAK, it will retry the start-split transaction for this endpoint at some time in the future according to normal scheduling rules. The host must not increment the err_count in this case.

The host must keep retrying the current start-split until the err_count reaches three for this endpoint before proceeding to the next split transaction for this endpoint. However, the host can issue other start-splits for other endpoints before it retries the start-split for this endpoint.

After the host receives a NAK, ACK, or STALL handshake in response to a complete-split transaction, it may subsequently issue a start-split transaction for the same endpoint. The host may choose to instead issue a start-split transaction for a different endpoint that is not awaiting a complete-split response.

The shaded case shown in the figure indicates that a control setup transaction should never encounter a NAK response since that is not allowed for full-/low-speed transactions.

Figure 11-50 and Figure 11-51 show the corresponding flow sequences for bulk/control IN split transactions.

11.17.2 Bulk/Control Split Transaction State Machines

The host and TT state machines for bulk/control IN and OUT split transactions are shown in the following figures. The transitions for these state machines are labeled the same as in the flow sequence figures.

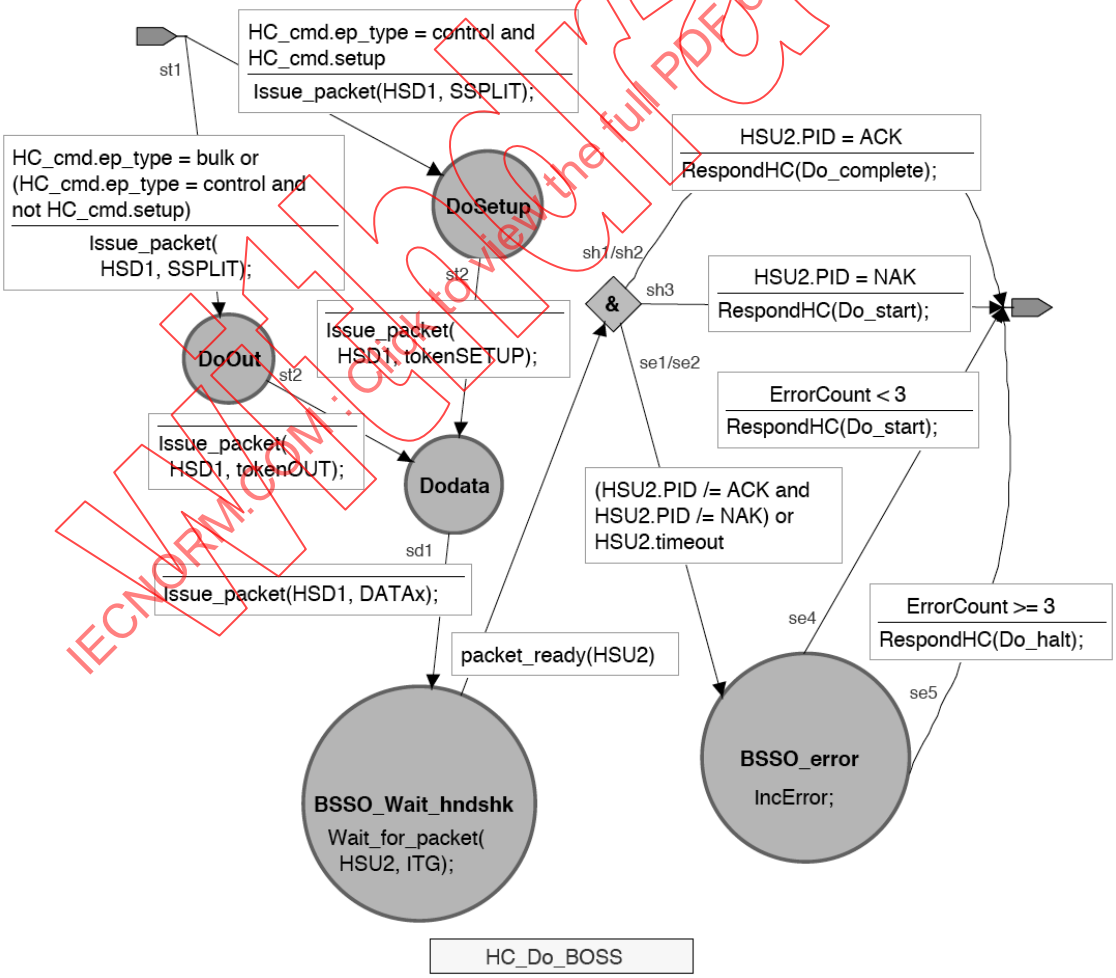


Figure 11-52. Bulk/Control OUT Start-split Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

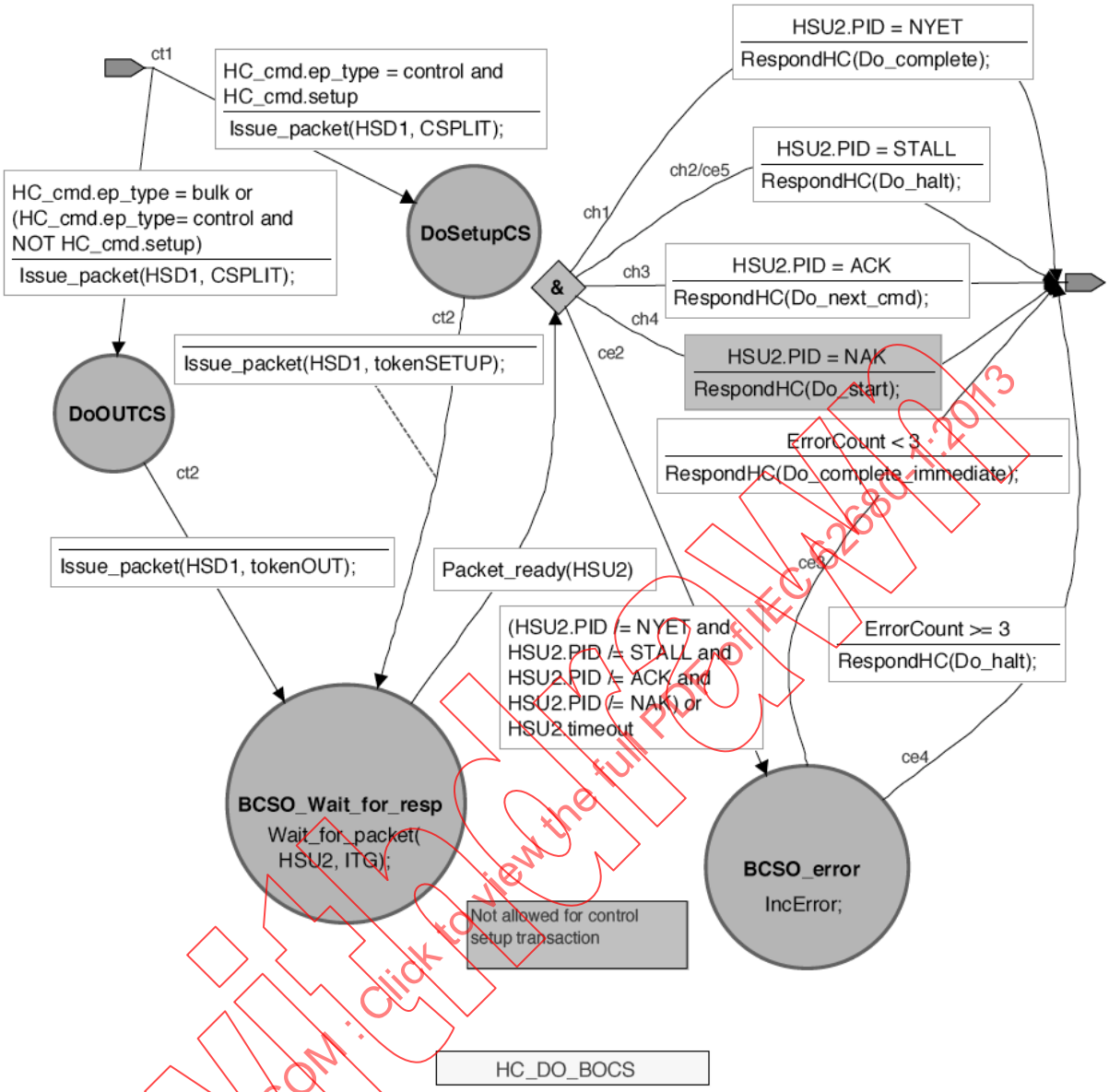


Figure 11-53. Bulk/Control OUT Complete-split Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

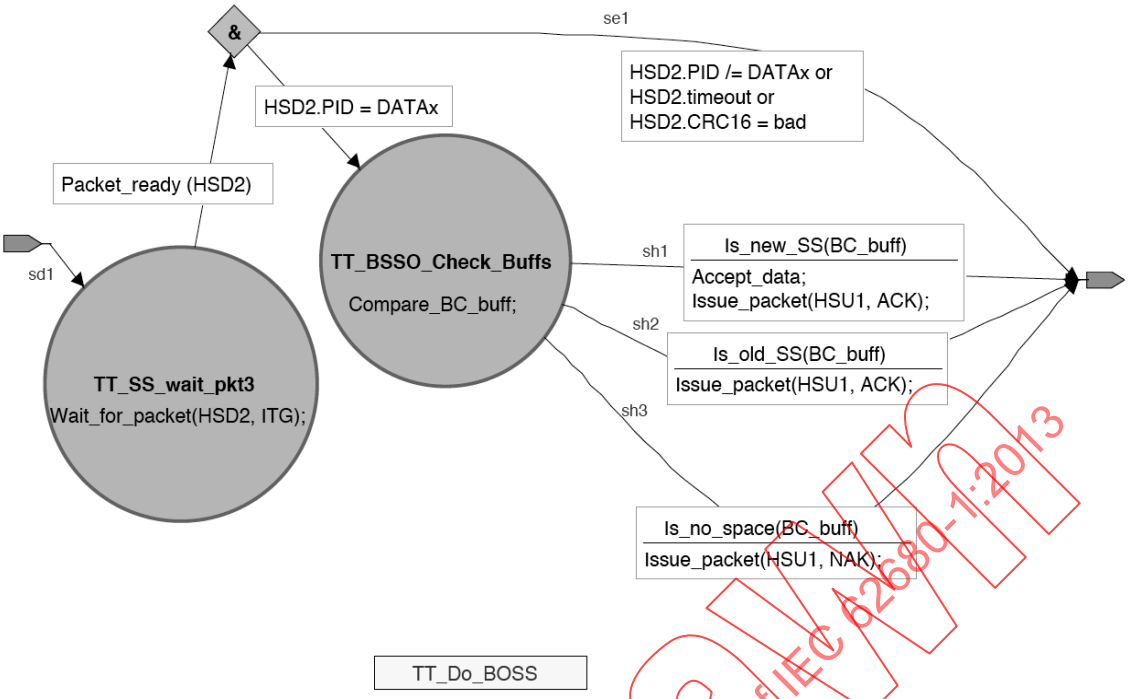


Figure 11-54. Bulk/Control OUT Start-split Transaction TT State Machine

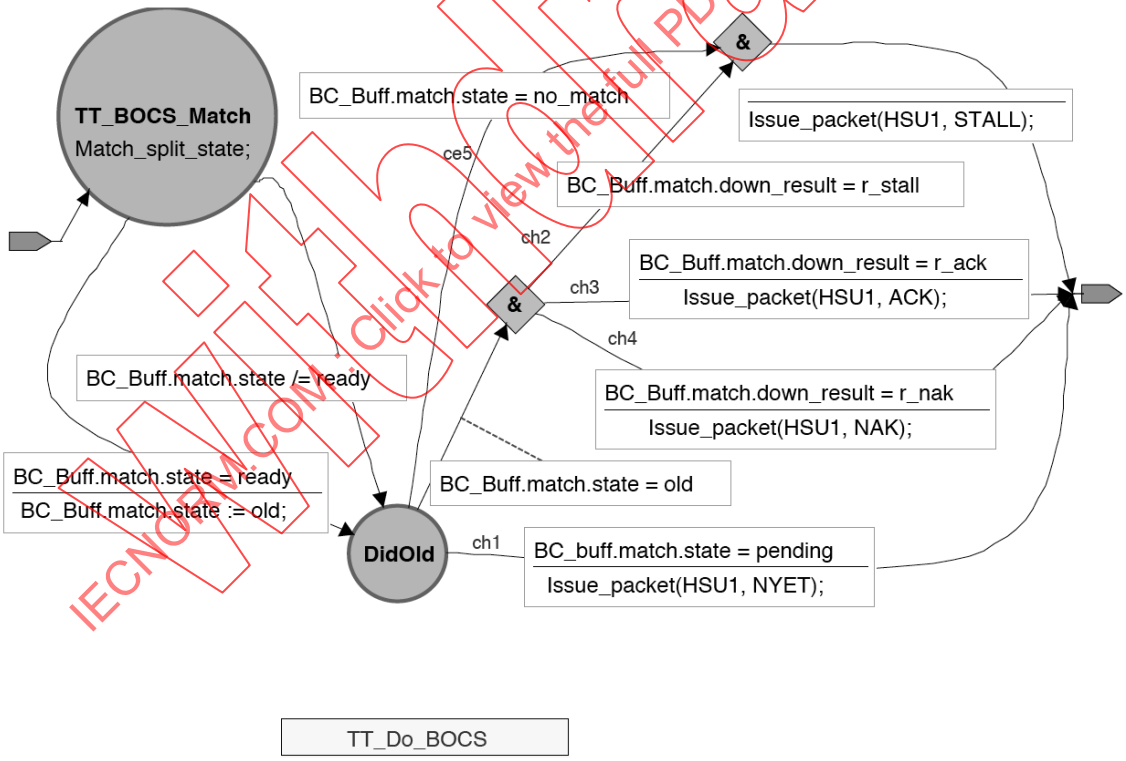


Figure 11-55. Bulk/Control OUT Complete-split Transaction TT State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

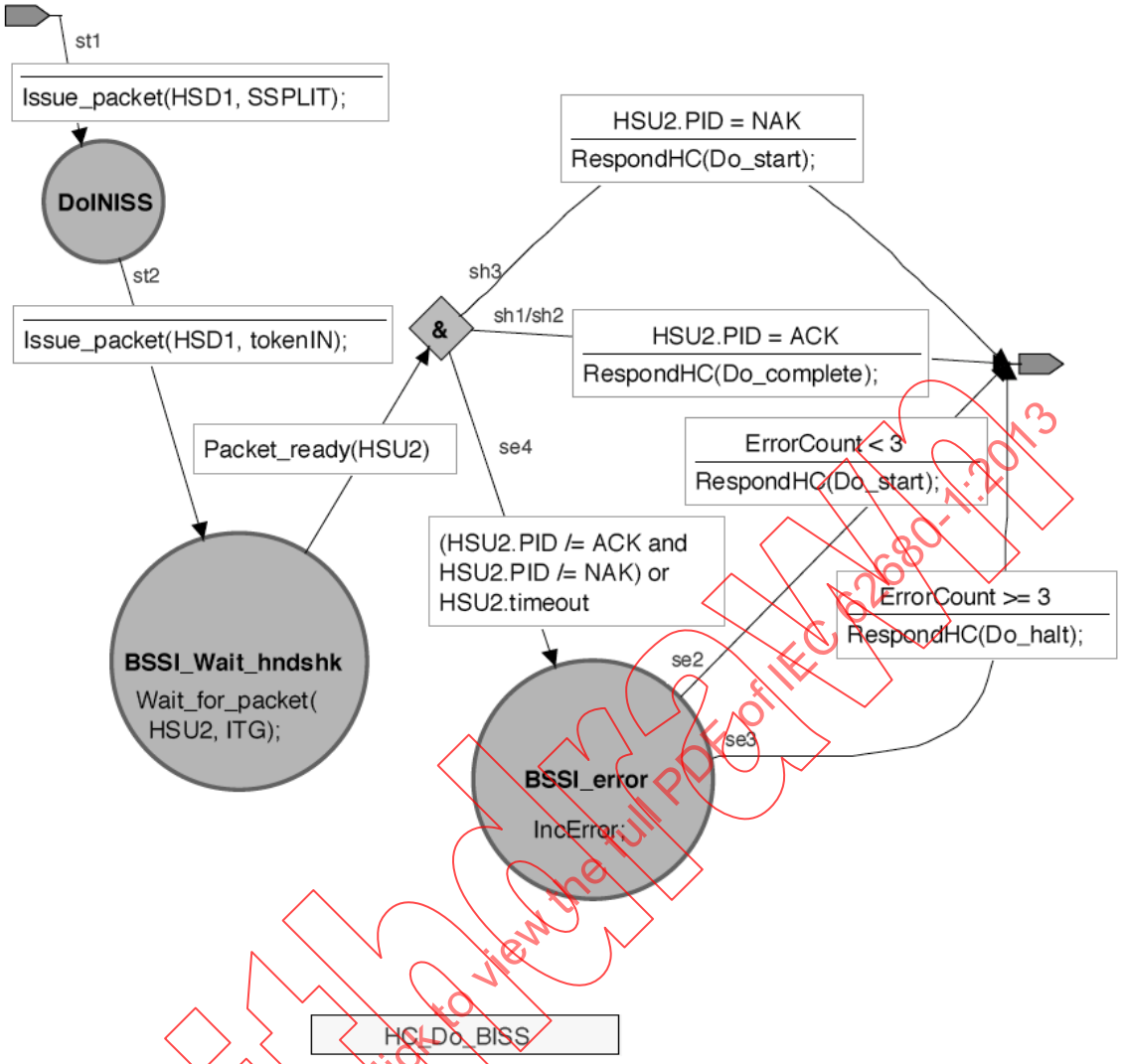


Figure 11-56. Bulk/Control IN Start-split Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

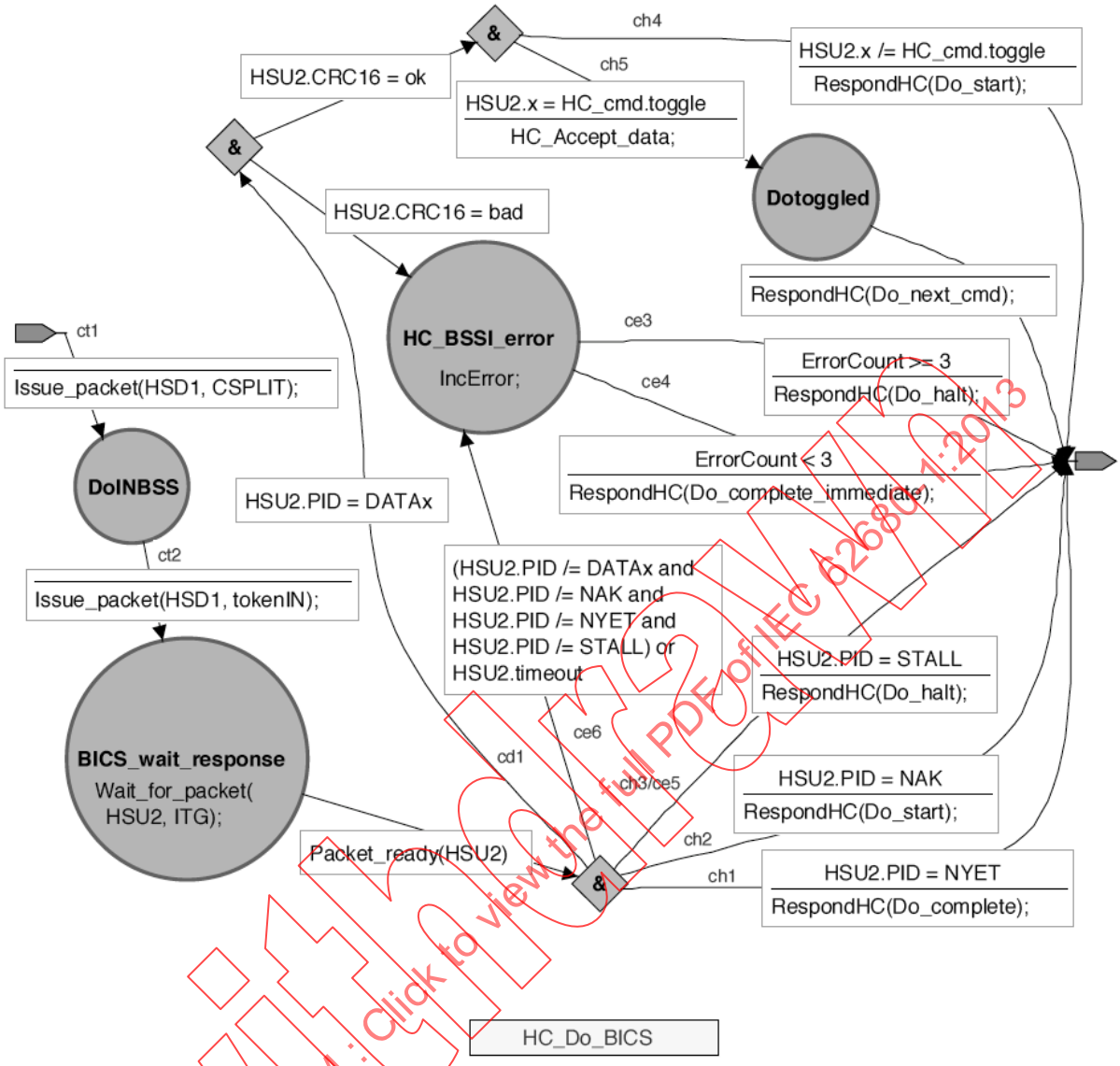


Figure 11-57. Bulk/Control IN Complete-split Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

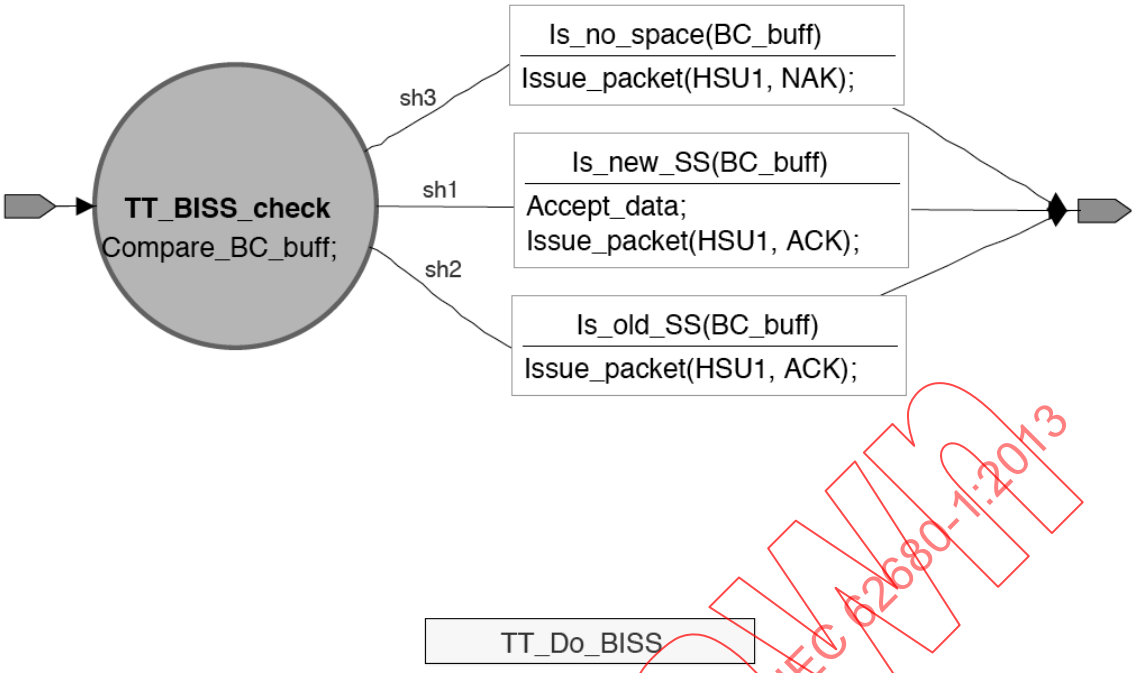


Figure 11-58. Bulk/Control IN Start-split Transaction TT State Machine

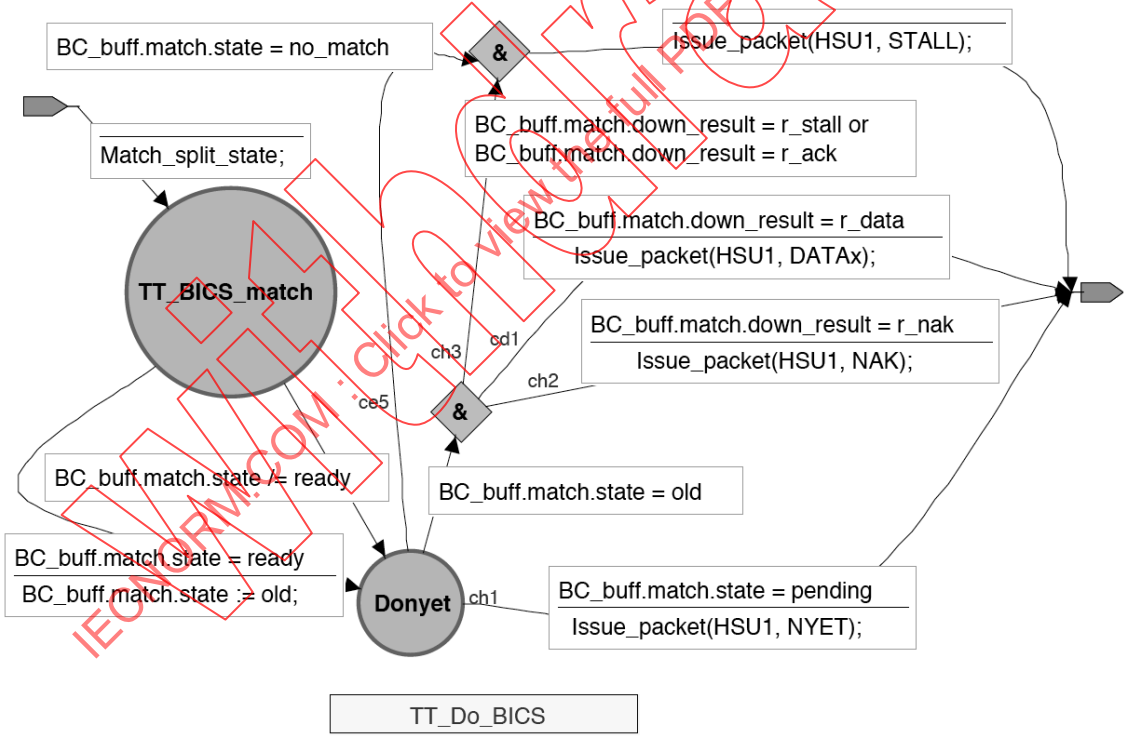


Figure 11-59. Bulk/Control IN Complete-split Transaction TT State Machine

11.17.3 Bulk/Control Sequencing

Once the high-speed handler has received a start-split for an endpoint and saved it in a local buffer, it responds with an ACK split transaction handshake. This tells the host controller to do a complete-split transaction next time this endpoint is polled.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

As soon as possible (subject to scheduling rules described previously), the full-/low-speed handler issues the full-/low-speed transaction and saves the handshake status (for OUT) or data/handshake status (for IN) in the same buffer.

Some time later (according to the host controller schedule), this endpoint will be polled for the complete-split transaction. The high-speed handler responds to the complete-split to return the full-/low-speed endpoint status for this transaction (as recorded in the buffer). If the host controller polls for the complete-split transaction for this endpoint before the full-/low-speed handler has finished processing this transaction on the downstream bus, the high-speed handler responds with a NYET handshake. This tells the host controller that the transaction is not yet complete. In this case, the host controller will retry the complete-split again at some later time.

When the full-/low-speed handler finally finishes the full-/low-speed transaction, it saves the data/status in the buffer to be ready for the next host controller complete-split transaction for this endpoint. When the host sends the complete-split, the high-speed handler responds with the indicated data/status as recorded in the buffer. The buffer transaction status is updated from ready to old so the high-speed handler is ready for either a retry or a new start-split transaction for this (or some other) full-/low-speed endpoint.

If there is an error on the complete-split transaction, the host controller will retry the complete-split transaction for this bulk/control endpoint "immediately" before proceeding to some other bulk/control split transaction. The host controller may issue other periodic split transactions or other non-split transactions before doing this complete-split transaction retry.

If there is a bulk/control transaction in progress on the downstream facing bus when the EOF time occurs, the TT must adhere to the definition in Section 11.3 for its behavior on the downstream facing bus. This will cause an increase in the error count for this transaction. The normal retry rules will determine if the transaction will be retried or not on the downstream facing bus.

11.17.4 Bulk/Control Buffering Requirements

The TT must provide at least two transactions of non-periodic buffering to allow the TT to deliver maximum full-/low-speed throughput on a downstream bus when the high-speed bus is idle.

As the high-speed bus becomes busier, the throughput possible on downstream full-/low-speed buses will decrease.

A TT may provide more than two transactions of non-periodic buffering and this can improve throughput for downstream buses for specific combinations of device configurations.

11.17.5 Other Bulk/Control Details

When a bulk/control split transaction fails, it can leave the associated TT transaction buffer in a busy (ready/x) state. This buffer state will not allow the buffer to be reused for other bulk/control split transactions. Therefore, as part of endpoint halt processing for full-/low-speed endpoints connected via a TT, the host software must use the Clear_TT_Buffer request to the TT to ensure that the buffer is not in the busy state.

Appendix A shows examples of packet sequences for full-/low-speed bulk/control transactions and their relationship with start-splits and complete-splits in various normal and error conditions.

11.18 Periodic Split Transaction Pipelining and Buffer Management

There are requirements on the behavior of the host and the TT to ensure that the microframe pipeline correctly sequences full-/low-speed isochronous/interrupt transactions on downstream facing full-/low-speed buses. The host must determine the microframes in which a start-split and complete-split transaction must be issued on high-speed to correctly

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

sequence a corresponding full-/low-speed transaction on the downstream facing bus. This is called “scheduling” the split transactions.

In the following descriptions, the 8 microframes within each full-speed (1 ms.) frame are referred to as microframe Y_0, Y_1, Y_2, …, Y_7. This notation means that the first microframe of each full-speed frame is labeled Y_0. The second microframe is labeled Y_1, etc. The last microframe of each full-speed frame is labeled Y_7. The labels repeat for each full-speed frame.

This section describes details of the microframe pipeline that affect both full-speed isochronous and full-/low-speed interrupt transactions. Then the split transaction rules for interrupt and isochronous are described.

Bulk/control transactions are not scheduled with this mechanism. They are handled as described in the previous section.

11.18.1 Best Case Full-Speed Budget

A microframe of time allows at most 187.5 raw bytes of signaling on a full-speed bus. In order to estimate when full-/low-speed transactions appear on a downstream bus, the host must calculate a best case full-speed budget. This budget tracks in which microframes a full-/low-speed transaction appears. The best case full-speed budget assumes that 188 full-speed bytes occur in each microframe. Figure 11-60 shows how a 1 ms frame subdivided into microframes of budget time. This estimate assumes that no bit stuffing occurs to lengthen the time required to move transactions over the bus.

The maximum number of bytes in a 1 ms frame is calculated as:

1157 maximum_periodic_bytes_per_frame = 12 Mb/s * 1 ms / 8 bits_per_byte * 6 data_bits / 7 bit-stuffed_data_bits * 90% maximum_periodic_data_per_frame

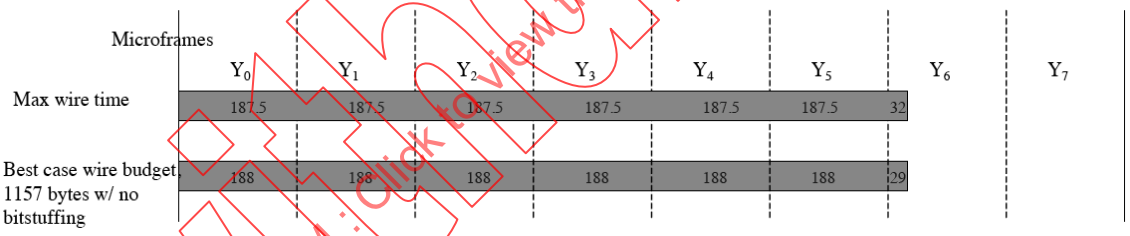


Figure 11-60. Best Case Budgeted Full-speed Wire Time With No Bit Stuffing

11.18.2 TT Microframe Pipeline

The TT implements a microframe pipeline of split transactions in support of a full-/low-speed bus. Start-split transactions are scheduled a microframe before the earliest time that their corresponding full-/low-speed transaction is expected to start. Complete-split transactions are scheduled in microframes that the full-/low-speed transaction can finish.

When a full-/low-speed device is attached to the bus and configured, the host assigns some time on the full-/low-speed bus at some budgeted time, based on the endpoint requirements of the configured device.

The effects of bit stuffing can delay when the full-/low-speed transaction actually runs. The results of other previous full-/low-speed transactions can cause the transaction to run earlier or later on the full-/low-speed bus.

The host always uses the maximum data payload size for a full-/low-speed endpoint in doing its budgeting. It does not attempt to schedule the actual data payloads that may be used in specific transactions to full-/low-speed endpoints. The host must include the maximum duration interpacket gap, bus turnaround times, and “TT think time”. The TT requires some

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

time to proceed to the next full-/low-speed transaction. This time is called the “TT think time” and is specified in the hub descriptor field *wHubCharacteristics* bit 5 and 6.

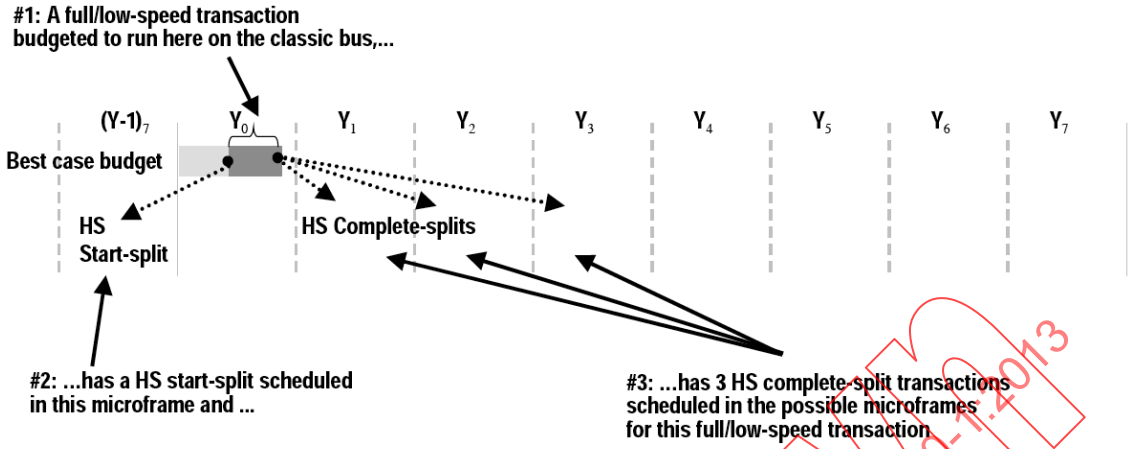


Figure 11-61. Scheduling of TT Microframe Pipeline

Figure 11-61 shows an example of a new endpoint that is assigned some portion of a full-/low-speed frame and where its start- and complete-splits are generally scheduled. The act of assigning some portion of the full-/low-speed frame to a particular transaction is called determining the budget for the transaction. More precise rules for scheduling and budgeting are presented later. The start-split for this example transaction is scheduled in microframe $Y-1_7$, the transaction is budgeted to run in microframe Y_0, and complete-splits are scheduled for microframes Y_1, Y_2, and Y_3. Section 11.18.4 describes the scheduling rules more completely.

The host must determine precisely when start- and complete- splits are scheduled to avoid overruns or underruns in the periodic transaction buffers provided by the TT.

11.18.3 Generation of Full-speed Frames

The TT must generate SOFs on the full-speed bus to establish the 1 ms frame clock within the defined jitter tolerances for full-speed devices. The TT has its own frame clock that is synchronized to the microframe SOFs on the high-speed bus. The SOF that reflects a change in the frame number it carries is identified as the zeroth microframe SOF. The zeroth high-speed microframe SOF corresponds to the full-speed SOF on the TT’s downstream facing bus. The TT must adhere to all timing/jitter requirements of a host controller related to frames as defined in other parts of this specification.

The TT must stop issuing full-speed SOFs after it detects 250 μs of high-speed idle. This is required to ensure that the full-/low-speed downstream facing bus enters suspend no more than 250 μs after the high-speed bus enters suspend.

The TT must generate a full-speed SOF on the downstream facing bus based on its frame timer. The generation of the full-speed SOF must occur within +/-3 full-speed bit time from the occurrence of the zeroth high-speed SOF. See Section 11.22.1 for more information about TT SOF generation.

11.18.4 Host Split Transaction Scheduling Requirements

Scheduling of split transactions is done by the host (typically in software) based on a best-case estimate of how the full-/low-speed transactions can be run on the downstream facing bus. This best-case estimate is called the best case budget. The host is free to issue the split transactions anytime within the scheduled microframe, but each split transaction must be issued sometime within the scheduled microframe. This description of the scheduling requirements applies to the split transactions for a single full-/low-speed transaction at a time.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

1. The host must never schedule a start-split in microframe Y_6. Some error conditions may result in the host controller erroneously issuing a start-split in this microframe. The TT response to this start-split is undefined.
2. The host must compute the start-split schedule by determining the best case budget for the transaction and:
   a. For isochronous OUT full-speed transactions, for each microframe in which the transaction is budgeted, the host must schedule a 188 (or the remaining data size) data byte start-split transaction. The start-split transaction must be scheduled in the microframe before the data is budgeted to begin on the full-speed bus. The start-split transactions must use the beginning/middle/end/all split transaction token encodings corresponding to the piece of the full-speed data that is being sent on the high-speed bus. For example, if only a single start-split is required, an “all” encoding is used. If multiple start-splits are required, a “beginning” encoding is used for the first start-split and an “end” encoding is used for the final start-split. If there are more than two start-splits required, the additional start-splits that are not the first or last use a “middle” encoding. A zero length full-speed data payload must only be scheduled with an “all” start-split. A start-split transaction for a beginning, middle, or end start-split must always have a non-zero length data payload. Figure 11-62 shows an example of an isochronous OUT that would appear to have budgeted a zero length data payload in a start-split (end). This example instead must be scheduled with a start-split(all) transaction.

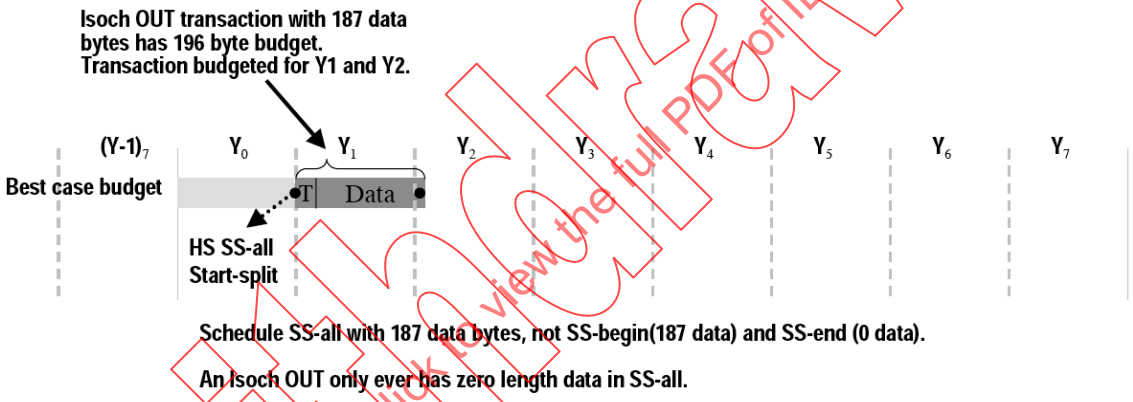


Figure 11-62. Isochronous OUT Example That Avoids a Start-split-end With Zero Data

   b. For isochronous IN and interrupt IN/OUT full-/low-speed transactions, a single start-split must be scheduled in the microframe before the transaction is budgeted to start on the full-/low-speed bus.
3. The host never schedules more than one complete-split in any microframe for the same full-/low-speed transaction.
   a. For isochronous OUT full-speed transactions, the host must never schedule a complete-split. The TT response to a complete-split for an isochronous OUT is undefined.
   b. For interrupt IN/OUT full-/low-speed transactions, the host must schedule a complete-split transaction in each of the two microframes following the first microframe in which the full-/low-speed transaction is budgeted. An additional complete-split must also be scheduled in the third following microframe unless the full-/low-speed transaction was budgeted to start in microframe Y_6. Figure 11-63 shows an example with only two complete-splits.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

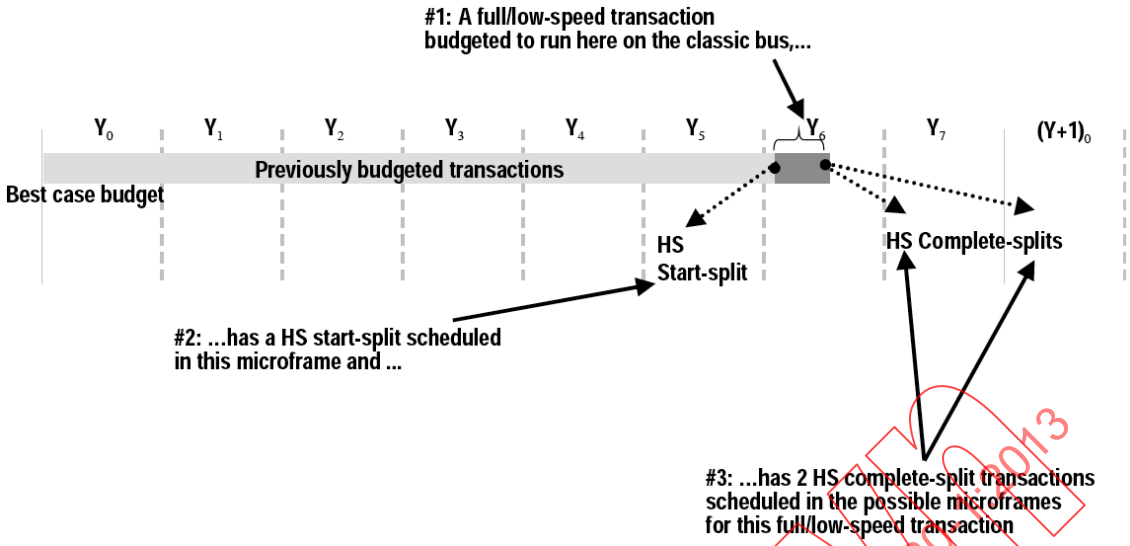


Figure 11-63. End of Frame TT Pipeline Scheduling Example

c. For isochronous IN full-speed transactions, for each microframe in which the full-speed transaction is budgeted, a complete-split must be scheduled for each following microframe. Also, determine the last microframe in which a complete-split is scheduled, call it L. If L is less than Y_6, schedule additional complete-splits in microframe L+1 and L+2.

   If L is equal to Y_6, schedule one complete-split in microframe Y_7. Also, schedule one complete-split in microframe Y_0 of the next frame, unless the full-speed transaction was budgeted to start in microframe Y_0.

   If L is equal to Y_7, schedule one complete-split in microframe Y_0 of the next frame, unless the full-speed transaction was budgeted to start in microframe Y_0. Figure 11-64 and Figure 11-65 show examples of the cases for L= Y_6 and L=Y_7.

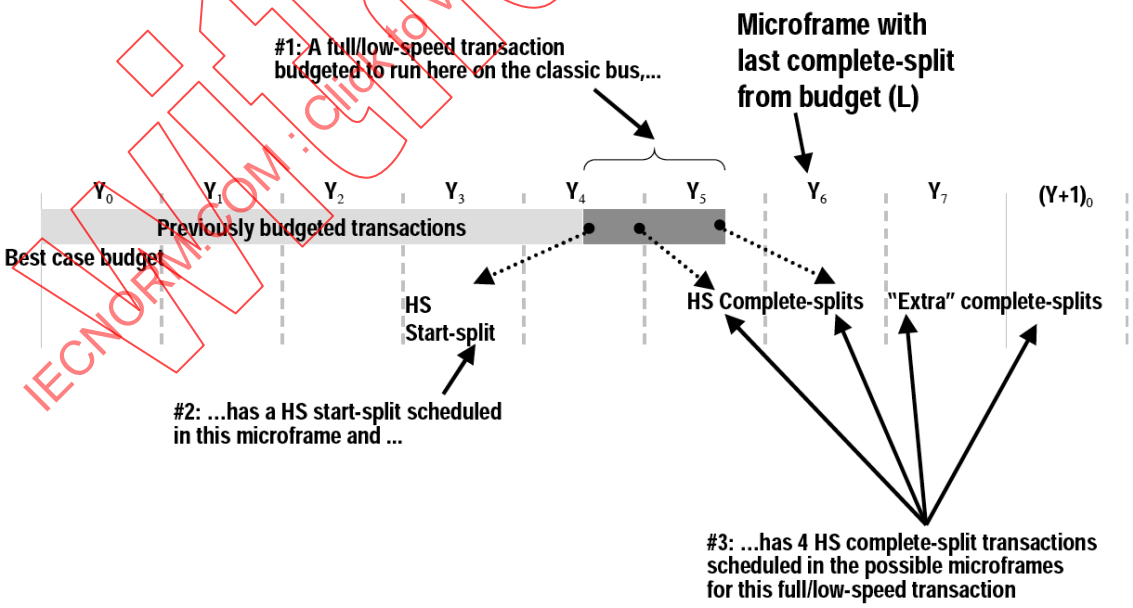


Figure 11-64. Isochronous IN Complete-split Schedule Example at L=Y_6

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

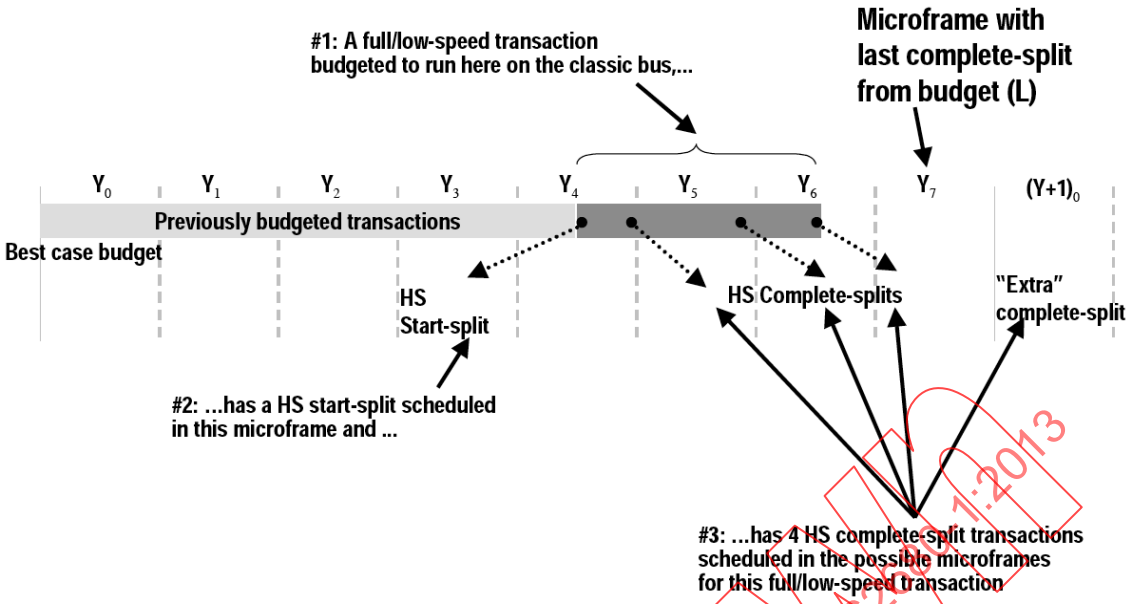


Figure 11-65. Isochronous IN Complete-split Schedule Example at L=Y_7

4. The host must never issue more than 16 start-splits in any high-speed microframe for any TT.
5. The host must only issue a split transaction in the microframe in which it was scheduled.
6. As precisely identified in the flow sequence and state machine figures, the host controller must immediately retry a complete-split after a high-speed transaction error (“trans_err”).

The “pattern” of split transactions scheduled for a full-/low-speed transaction can be computed once when each endpoint is configured. Then the pattern does not change unless some change occurs to the collection of currently configured full-/low-speed endpoints attached via a TT.

Finally, for all periodic endpoints that have split transactions scheduled within a particular microframe, the host must issue complete-split transactions in the same relative order as the corresponding start-split transactions were issued.

11.18.5 TT Response Generation

The approach used for full-speed isochronous INs and interrupt INs/OUTs ensures that there is always an opportunity for the TT to return data/results whenever it has something to return from the full-/low-speed transaction. Then whenever the full-/low-speed handler starts the full-/low-speed transaction, it simply accumulates the results in each microframe and then returns it in response to a complete-split from the host. The TT acts similar to an isochronous device in that it uses the microframe boundary to "carve up" the full-/low-speed data to be returned to the host. The TT does not do any computation on how much data to return at what time. In response to the "next" high-speed complete-split, the TT simply returns the endpoint data it has received from the full-/low-speed bus in a microframe.

Whenever the TT has data to return in response to a complete-split for an interrupt full-/low-speed or isochronous full-speed transaction, it uses either a DATA0/1 or MDATA for the data packet PID.

If the full-/low-speed handler completes the full-/low-speed isochronous/interrupt IN transaction during a microframe with a valid CRC16, it uses the DATA0/1 PID for the data packet of the complete-split transaction. This indicates that this is the last data of the full-/low-speed transaction. A DATA0 PID is always used for isochronous transactions. For interrupt transactions, a DATA0/1 PID is used corresponding to the full-/low-speed data packet PID received.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

If the full-/low-speed handler completes the full-/low-speed isochronous/interrupt IN transaction during a microframe with a bad CRC16, it uses the ERR response to the complete-split transaction and does not return the data received from the full-/low-speed device.

If the TT is still receiving data on the downstream facing bus at the microframe boundary, the TT will respond with either an MDATA PID or a NYET for the corresponding complete-split. If the TT has received more than two bytes of the data field of the full-/low-speed data packet, it will respond with an MDATA PID. Further, the data packet that will be returned in the complete-split must contain the data received from the full-/low-speed device minus the last two bytes. The last two bytes must not be included since they could be the CRC16 field, but the TT will not know this until the next microframe. The CRC16 field received from the full-/low-speed device is never returned in a complete-split data packet for isochronous/interrupt transactions. If less than three data bytes of the full-/low-speed data packet have been received at the end of a microframe, the TT must respond with a NYET to the corresponding high-speed complete-split. Both of these responses indicate to the host that more data is being received and another complete-split transaction is required.

When the host controller receives a DATA0/1 PID for interrupt or isochronous IN complete-splits (and ACK, NAK, STALL, ERR for interrupt IN/OUT complete-splits), it stops issuing any remaining complete-splits that might be scheduled for that endpoint for this full-/low-speed transaction.

If the TT has not started the full-/low-speed transaction when it receives a complete-split, the TT will not find an entry in the complete-split pipeline stage. When this happens, the protocol state machines show that the TT responds with a NYET (e.g., the “no match” case). This NYET response tells the host that there are no results available currently, but the host should continue with other scheduled split transactions for this endpoint in subsequent microframes.

In general, there will be two (or more) complete-split transactions scheduled for a periodic endpoint. However, for interrupt endpoints, the maximum size of the full-/low-speed transaction guarantees that it can never require more than two complete-split transactions. Two complete-split transactions are only required when the transaction spans a microframe boundary. In cases where the full-/low-speed transaction actually starts and completes in the same microframe, only a single complete-split will return data; any other earlier complete-splits will have a NYET response.

For isochronous IN transactions, more complete-split transactions may be scheduled based on the length of the full-speed transaction. A full-speed isochronous IN transaction can be up to 1023 data bytes, which can require portions of up to 8 microframes of time on the downstream facing bus (with the worst alignment in the frame and worst case bit stuffing). Such a maximum sized full-speed transaction can require 8 complete-split transactions. If the device generates less data, the host will stop issuing complete-splits after the one that returns the final data from the device for a frame.

11.18.6 TT Periodic Transaction Handling Requirements

The TT has two methods it must use to react to timing related events that affect the microframe pipeline: current transaction abort and freeing pending start-splits. These methods must be used to manage the microframe pipeline.

The TT must also react (as described in Section 11.22.1) when its microframe or frame timer loses synchronization with the high-speed bus.

The TT must not issue too many full-/low-speed transactions in any microframe.

Each of these requirements are described below.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.18.6.1 Abort of Current Transaction

When a current transaction is in progress on the downstream facing bus and it is no longer appropriate for the TT to continue the transaction, the transaction is “aborted.”

The TT full-/low-speed handler must abort the current full-/low-speed transaction:

1. For all periodic transaction types, if the full-speed frame EOF time occurs
2. If the transaction is an interrupt transaction and the start-split for the transaction was received in some microframe (call it X) and the TT microframe timer indicates the X+4 microframe

Note that no additional abort handling is required for isochronous transactions besides the generic IN/OUT handling described below. Abort has different processing requirements with regards to the downstream facing bus for IN and OUT transactions. For any type of transaction, the TT must not generate a complete-split response for an aborted transaction; e.g., no entry is made in the complete-split pipeline stage for an aborted transaction.

1. At the time the TT decides to abort an IN transaction, the TT must not issue the handshake packet for the transaction if the handshake has not already been started on the downstream facing bus. The TT may choose to not issue the IN token packet, if possible. If the transaction is in the data phase (e.g., in the middle of the target device generated DATA packet), the TT simply awaits the completion of that packet and ignores any data received and must not respond with a full-/low-speed handshake. The TT must not make an entry in the complete-split pipeline stage. This processing will cause a NYET response to the corresponding complete-split on the high-speed bus.
2. At the time the TT decides to abort an OUT transaction, the TT may choose to not issue the TOKEN or DATA packets, if possible. If the TT is in the middle of the DATA packet, it must stop issuing data bytes as soon as possible and force a bit-stuffing error on the downstream facing bus. In any case, the TT must not make an entry in the complete-split pipeline stage. This processing will cause a NYET response to the corresponding complete-split on the high-speed bus.

11.18.6.2 Free of Pending Start-splits

A start-split can be buffered in the start-split pipeline stage that is no longer appropriate to cause a full-/low-speed transaction on the downstream facing bus. Such a start-split transaction must be “freed” from the start-split pipeline stage. This means the start-split is simply ignored by the TT and the TT must respond to a corresponding complete-split with a NYET. For example, no entry is made in the complete-split pipeline stage for the freed start-split.

A start-split in the start-split pipeline must be freed:

1. If the full-speed frame EOF time occurs, except for start-splits received in $(Y\text{-}1)_7$
2. If the start-split transaction was received in some microframe (call it X) and the TT microframe timer indicates the X+4 microframe

If the TT receives a periodic start-split transaction in microframe Y_6, its behavior is undefined. This is a host scheduling error.

11.18.6.3 Maximum Full-/low-speed Transactions per Microframe

The TT must not start a full-/low-speed transaction unless it has space available in the complete-split pipeline stage to hold the results of the transaction. If there is not enough space, the TT must wait to issue the transaction until there is enough space. The maximum number of normally operating full-speed transactions that can ever be completed in a microframe is 16.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.18.7 TT Transaction Tracking

Figure 11-66 shows the TT microframe pipeline of transactions. The 8 high-speed microframes that compose a full-/low-speed frame are labeled with Y_0 through Y_7 assuming the microframe timer has occurred at the point in time shown by the arrow (e.g., time "NOW").

As shown in the figure, a start-split high-speed transaction that the high-speed handler receives in microframe Y_0 (e.g., a start-split "B") can run on the full-/low-speed bus during microframe times Y_1 or Y_2 or Y_3. This variation in starting on the full-/low-speed bus is due to bit stuffing and bulk/control reclamation that can occur on the full-/low-speed bus. Once the full-/low-speed transaction finishes, its complete-split transactions (if they are required) will run on the high-speed bus during microframes Y_2, Y_3, or Y_4.

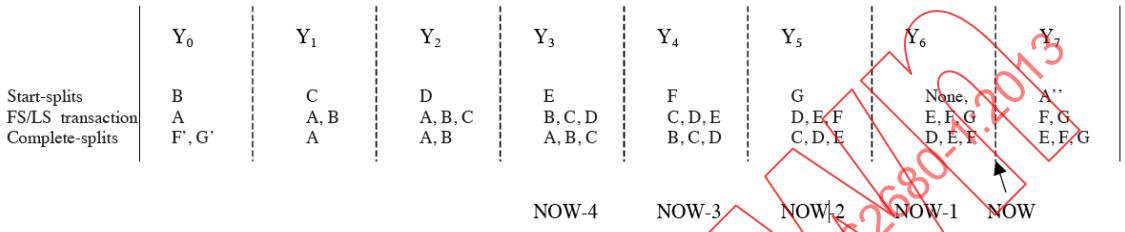


Figure 11-66. Microframe Pipeline

When the microframe timer indicates a new microframe, the high-speed handler must mark any start-splits in the start-split pipeline stage it received in the previous microframe as "pending" so that they can be processed on the full-/low-speed bus as appropriate. This prevents the full-/low-speed transactions from running on the downstream bus too early.

At the beginning of each microframe (call it "NOW"), the high-speed handler must free (as defined in Section 11.18.6.2) any start-split transactions from the start-split pipeline stage that are still pending from microframe NOW-4 (or earlier) and ignore them. If the transaction is in progress on the downstream facing bus, the transaction must be aborted (with full-/low-speed methods as defined in Chapter 8). This is described in more detail in the previous sections. This ensures that even if the full-/low-speed bus has encountered a babble condition on the bus (or other delay condition), the TT keeps its periodic transaction pipeline running on time (e.g., transactions do not run too late). This also ensures that when the last scheduled complete-split transaction is received by the TT, the full-/low-speed transaction has been completed (either successfully or by being aborted).

Finally, at the beginning of each microframe, the high-speed handler must change any complete-split transaction responses in the complete-split pipeline stage from microframe NOW-2 to the free state so that their space can be reused for responses in this microframe.

This algorithm is shown in pseudo code in Figure 11-67. This pseudo-code corresponds to the Advance_pipeline procedure identified previously.

```
-- Clean up start-split state in case full-/low-speed bus fell behind
while start-splits in pending state received by TT before microframe-4 loop
   Free start-split entry
End loop

-- Clean up complete-split pipeline in case no complete-splits were received
While complete-split transaction states from (microframe-2) loop
   Free complete-split response transaction entry
End loop

-- Enable full-/low-speed transactions received in previous microframe
While start-split transactions from (previous_microframe) loop
   Set start-split entry to pending status
End loop
```

Figure 11-67. Advance_Pipeline Pseudocode

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.18.8 TT Complete-split Transaction State Searching

A host must issue complete-split transactions in a microframe for a set of full-/low-speed endpoints in the same relative order as the start-splits were issued in a microframe for this TT. However, errors on start- or complete-splits can cause the high-speed handler to receive a complete-split transaction that does not “match” the expected next transaction according to the TT’s transaction pipeline.

The TT has a pipeline of complete-split transaction state that it is expecting to use to respond to complete-split transactions. Normally the host will issue the complete-split that the high-speed handler is expecting next and the complete-split will correspond to the entry at the front of the complete-split pipeline.

However, when errors occur, the complete-split transaction that the high-speed handler receives might not match the entry at the front of the complete-split pipeline. This can happen for example, when a start-split is damaged on the high-speed bus and the high-speed handler does not receive it successfully. Or the high-speed handler might have a match, but the matching entry is located after the state for other expected complete-splits that the high-speed handler did not receive (due to complete-split errors on the high-speed bus).

The high-speed handler must respond to a complete-split transaction with the results of a full-/low-speed transaction that it has completed. This means that the high-speed handler must search to find the correct state tracking entry among several possible complete-split response entries. This searching takes time. The high-speed handler only needs to search the complete-split responses accumulated during the previous microframe. There only needs to be at most 1 microframe of complete-split response entries; the microframe of responses that have already been accumulated and are awaiting to be returned via high-speed complete-splits.

The split transaction protocol is defined to allow the high-speed handler to timeout the first high-speed complete-split transaction while it is searching for the correct response. This allows the high-speed handler time to complete its search and respond correctly to the next (retried) complete-split.

The following interrupt and isochronous flow sequence figures show these cases with the transitions labeled “Search not complete in time” and “No split response found”.

The high-speed handler matches the complete-split transaction with the correct entry in the complete-split pipeline stage and advances the pipeline appropriately. There are five cases the TT must handle correctly:

1. If the high-speed complete-split token and first entry of the complete-split pipeline match, the high-speed handler responds with the indicated data/status. This case occurs the first time the TT receives a complete-split.
2. Same as above, but this is a retry of a complete-split that the TT has already received due to the host controller not receiving the (previous) response information.
3. If the complete-split transaction matches some other entry in the complete-split pipeline besides the first, the high-speed handler advances the complete-split pipeline (e.g., frees response information for previous complete-split entries) and responds with the information for the matching entry. This case can happen due to normal or missed previous complete-split transactions. An example abnormal case could be that the host controller was unsuccessful in issuing a complete-split transaction to the high-speed handler and has done endpoint halt processing for that endpoint. This means the next complete-split will not match the first entry of the complete-split pipeline stage.
4. The high-speed handler can also receive a complete-split before it has started a full-/low-speed transaction. If there is not an entry in the complete-split pipeline, the high-speed handler responds with a NYET handshake to inform the host that it has no status information. When the host issues the last scheduled complete-split for this endpoint for

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

this frame, it must interpret the NYET as an error condition. This stimulates the normal "three strikes" error handling. If there have been more than three errors, the host halts this endpoint. If there have been less than three errors, the host continues processing the scheduled transactions of this endpoint (e.g., a start-split will be issued as the next transaction for this endpoint at the next scheduled time for this endpoint). Note that a NYET response is possible in this case due to a transaction error on the start-split or a host (or TT) scheduling error.

5. The high-speed handler can timeout its first high-speed complete-split transaction while it is searching the complete-split pipeline stage for a matching entry. However, the high-speed handler must respond correctly to the subsequent complete-split transaction. If the high-speed handler did not respond correctly for an interrupt IN after it had acknowledged the full-/low-speed transaction, the endpoint software and the device would lose data synchronization and more catastrophic errors could occur.

The host controller must issue the complete-split transactions in the same relative order as the original corresponding start-split transactions.

11.19 Approximate TT Buffer Space Required

A transaction translator requires buffer and state tracking space for its periodic and non-periodic portions.

The TT microframe pipeline requires less than:

- 752 data bytes for the start-split stage
- 2x 188 data bytes for the complete-split stage
- 16x 4x transaction status (<4 bytes for each transaction) for start-split stage
- 16x 2x transaction status (<4 bytes for each transaction) for complete-split stage

There are, at most, 4 microframes of buffering required for the start-split stage of the pipeline and, at most, 2 microframes of buffering for the complete-split stage of the pipeline. There are, at most, 16 full-speed (minimum sized) transactions possible in any microframe.

The non-periodic portion of the TT requires at least:

- 2x (64 data + 4 transaction status) bytes

Different implementations may require more or less buffering and state tracking space.

11.20 Interrupt Transaction Translation Overview

The flow sequence and state machine figures show the transitions required for high-speed split transactions for full-/low-speed interrupt transfer types for a single endpoint. These figures must not be interpreted as showing any particular specific timing. In particular, high-speed or full-/low-speed transactions for other endpoints may occur before or after these split transactions. Specific details are described as appropriate.

In contrast to bulk/control processing, the full-/low-speed handler must not do local retry processing on the full-/low-speed bus in response to a transaction error for full-/low-speed interrupt transactions.

11.20.1 Interrupt Split Transaction Sequences

The interrupt IN and OUT flow sequence figures use the same notation and have descriptions similar to the bulk/control figures.

In contrast to bulk/control processing, the full-speed handler must not do local retry processing on the full-speed bus in response to a transaction errors (including timeout) of an interrupt transaction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

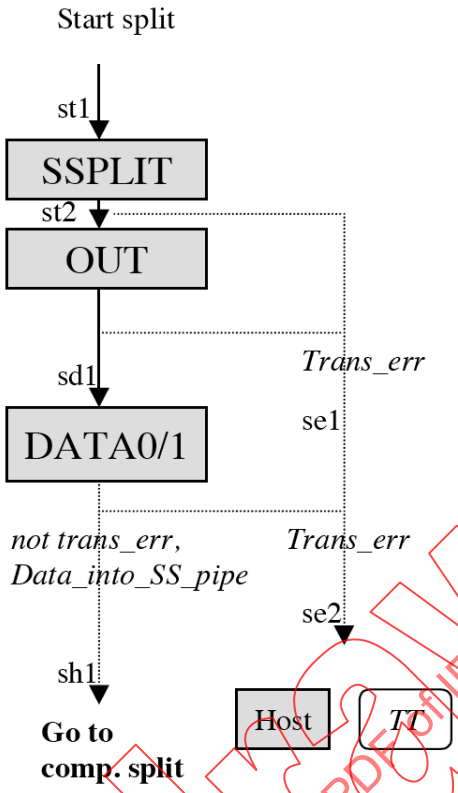


Figure 11-68. Interrupt OUT Start-split Transaction Sequence

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

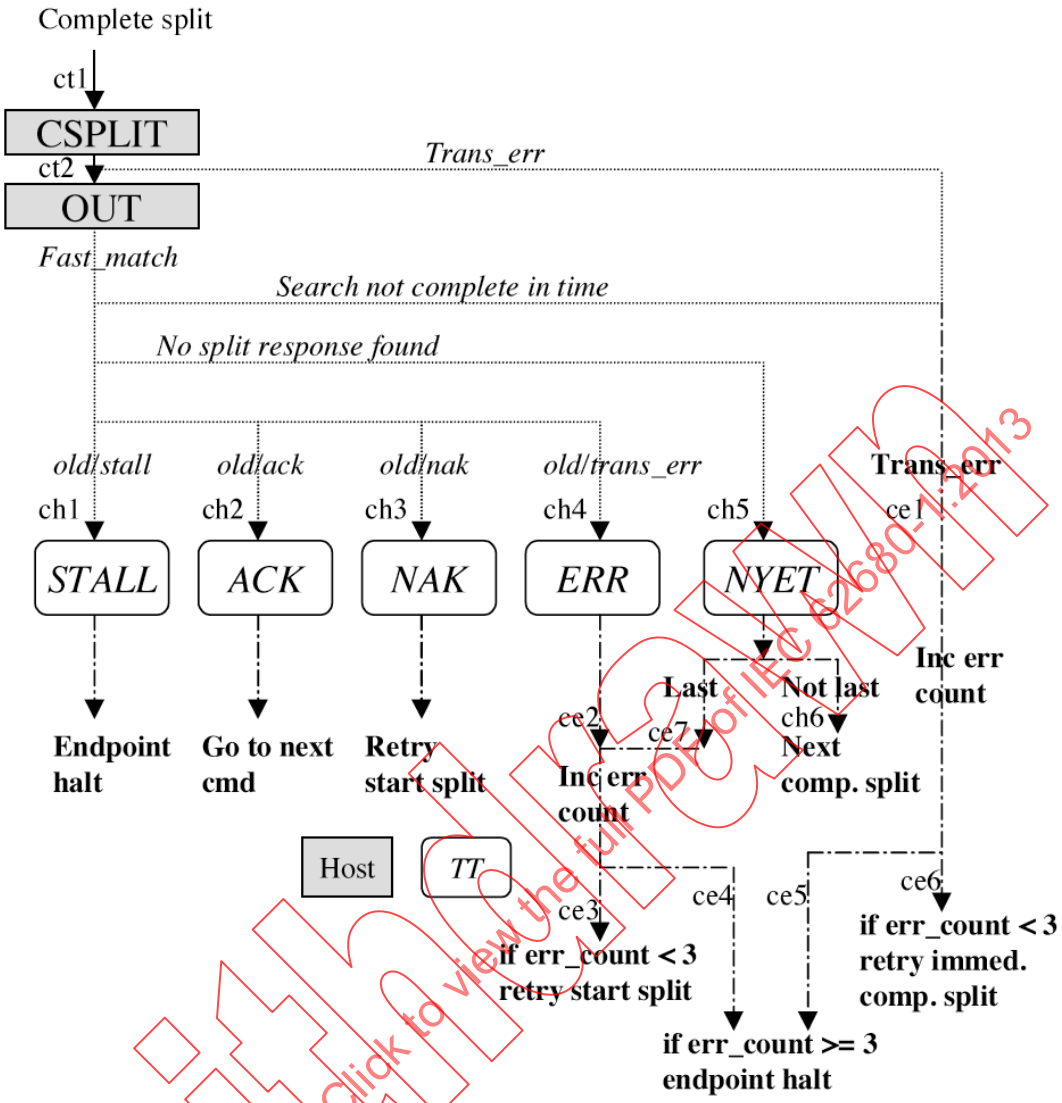


Figure 11-69. Interrupt OUT Complete-split Transaction Sequence

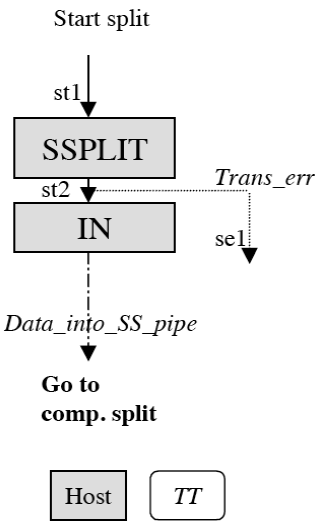


Figure 11-70. Interrupt IN Start-split Transaction Sequence

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

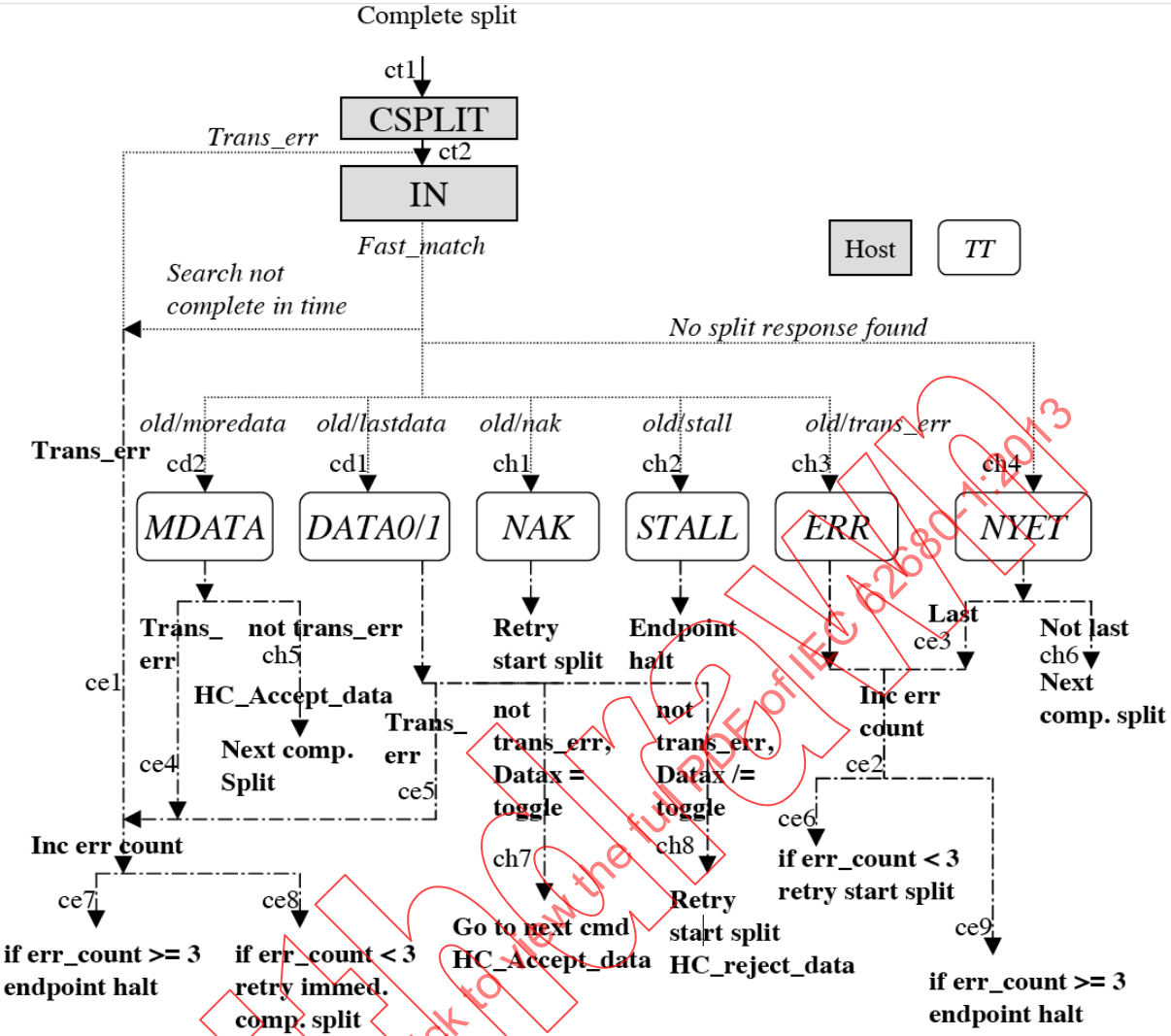


Figure 11-71. Interrupt IN Complete-split Transaction Sequence

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.20.2 Interrupt Split Transaction State Machines

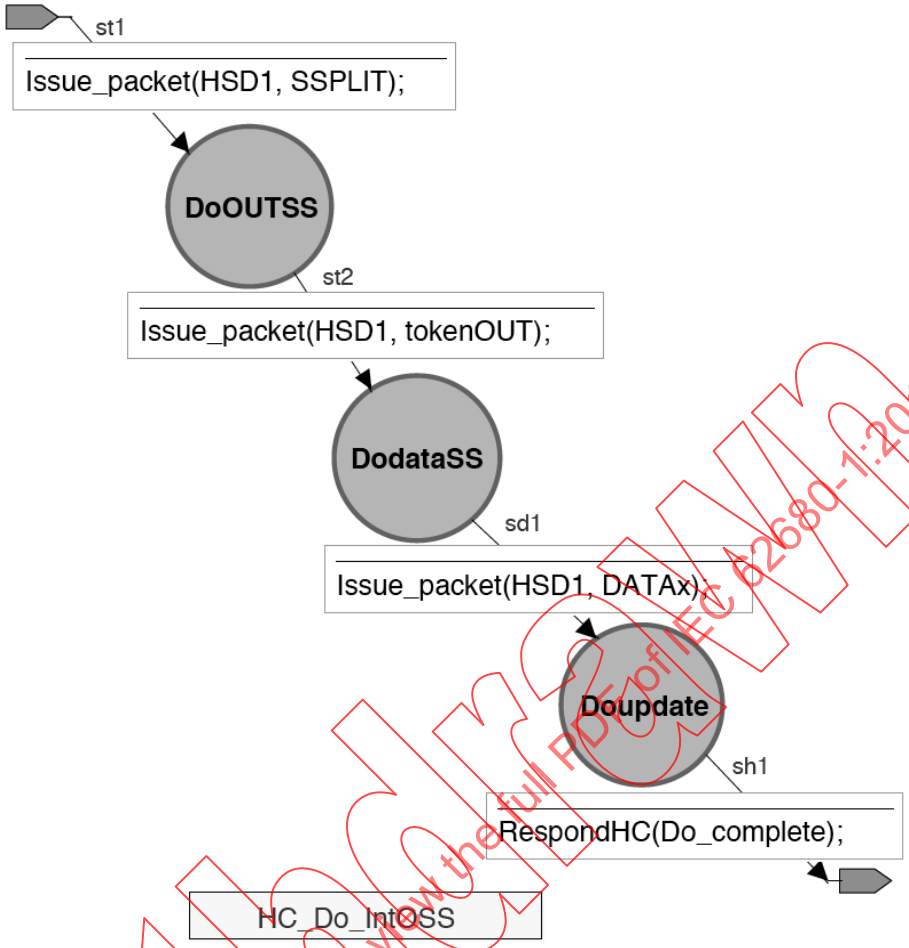


Figure 11-72. Interrupt OUT Start-split Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

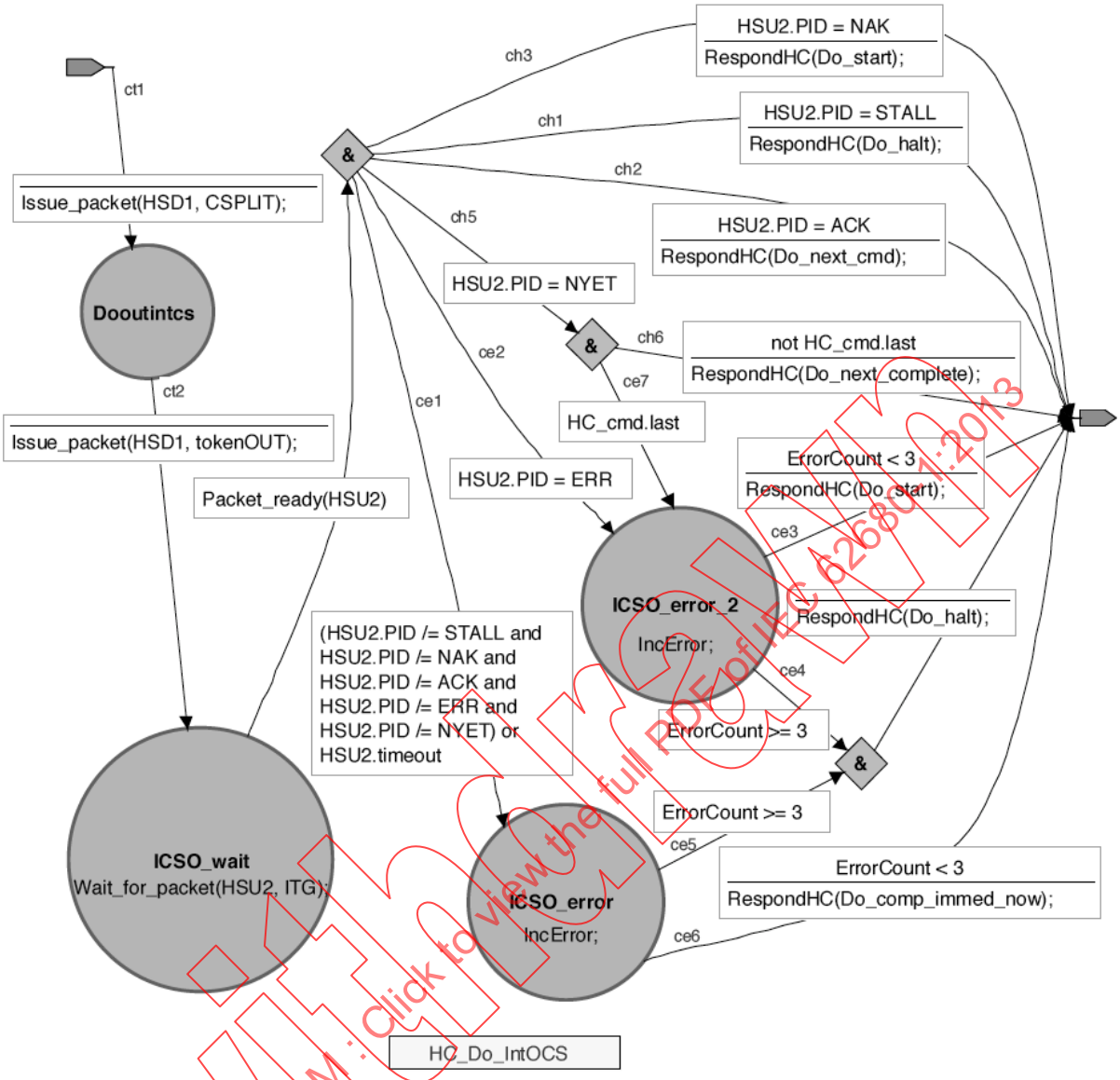


Figure 11-73. Interrupt OUT Complete-split Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

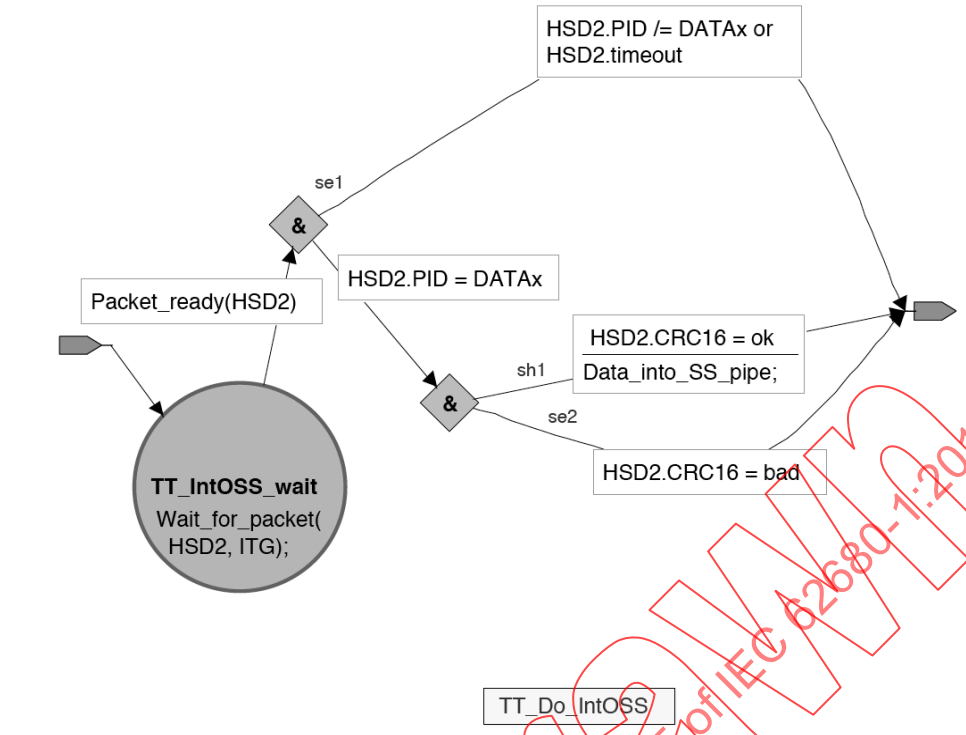


Figure 11-74. Interrupt OUT Start-split Transaction TT State Machine

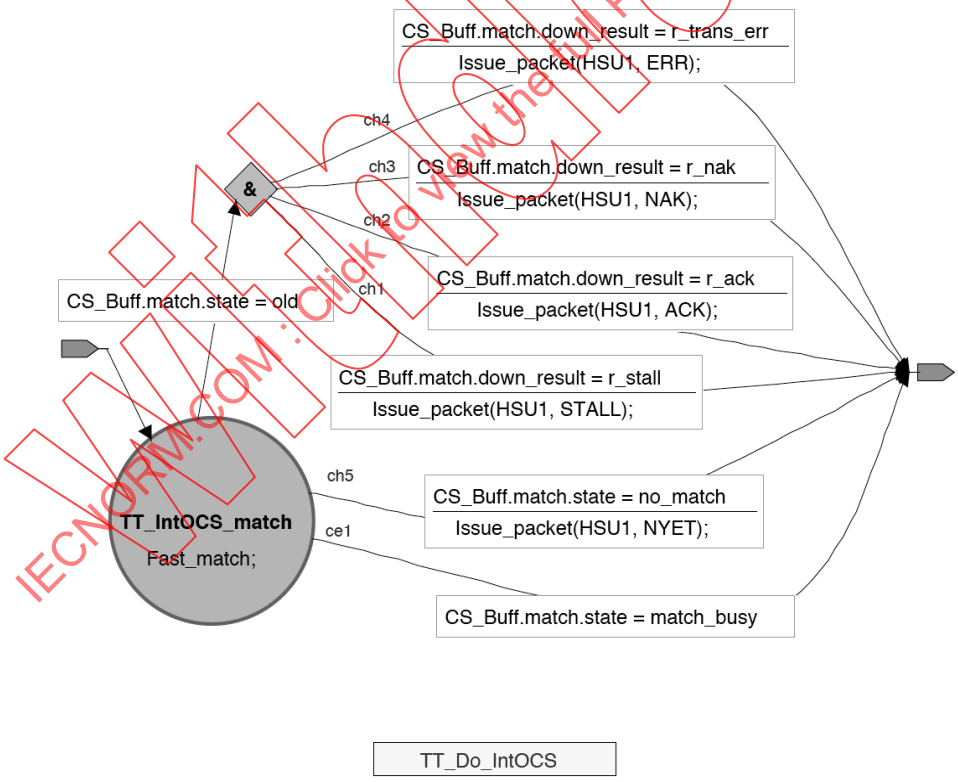


Figure 11-75. Interrupt OUT Complete-split Transaction TT State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

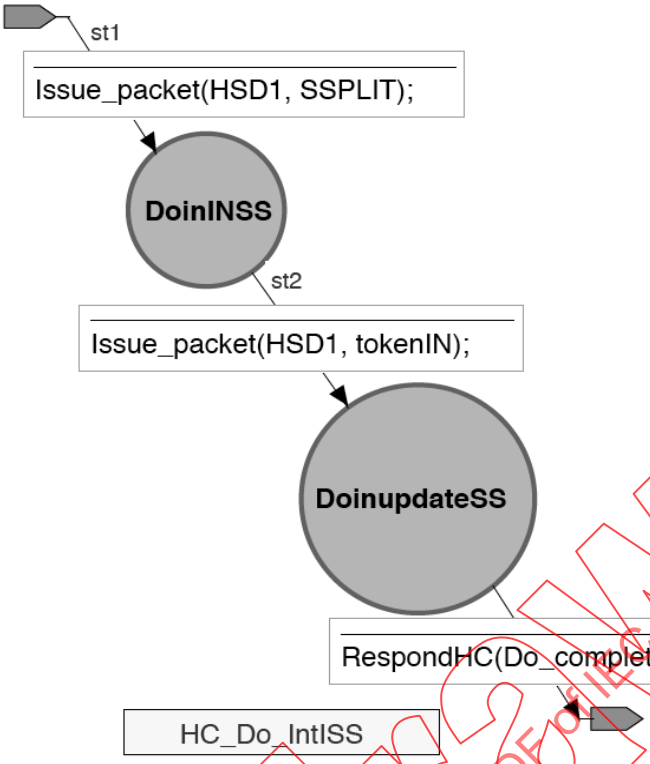


Figure 11-76. Interrupt IN Start-split Transaction Host State Machine

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

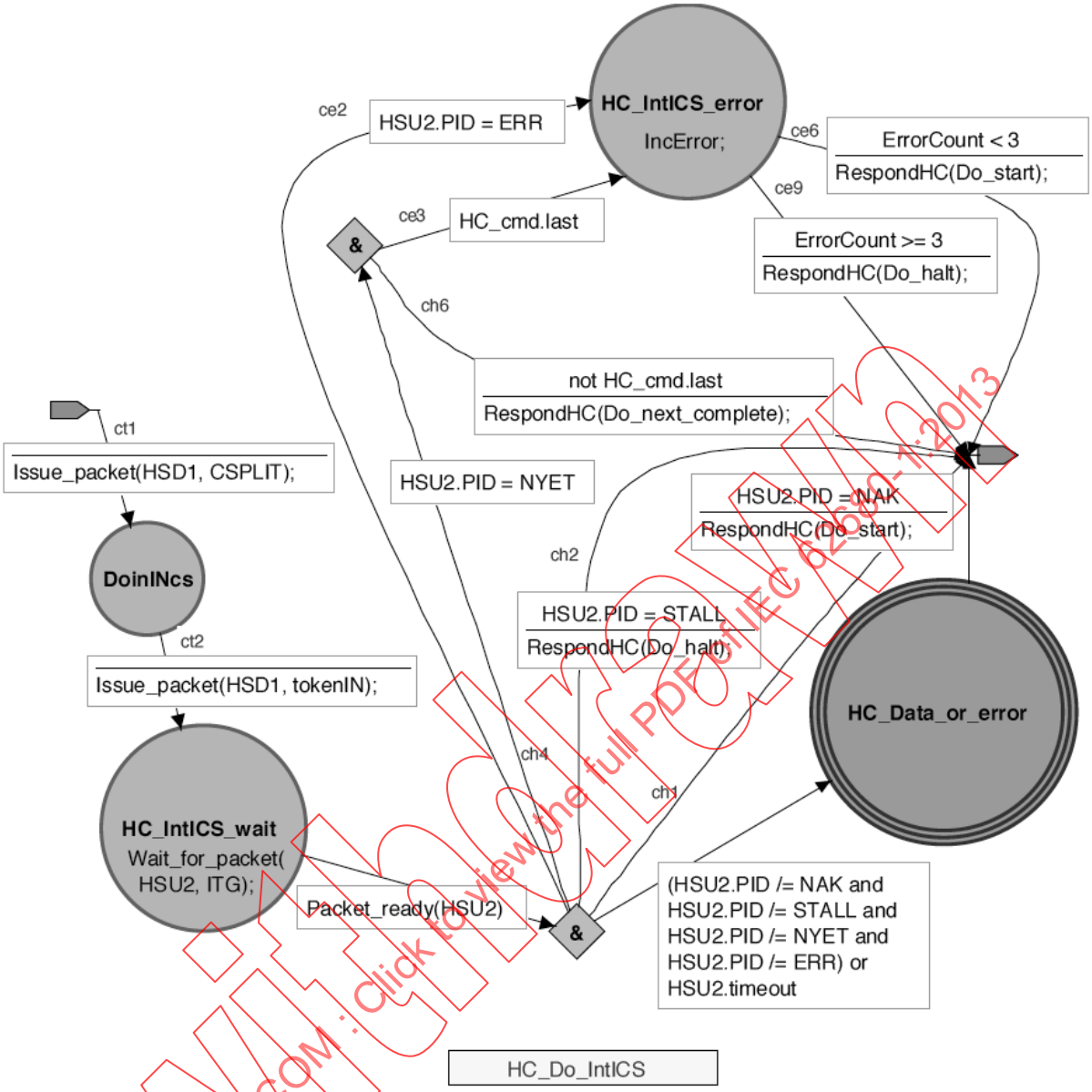


Figure 11-77. Interrupt IN Complete-split Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

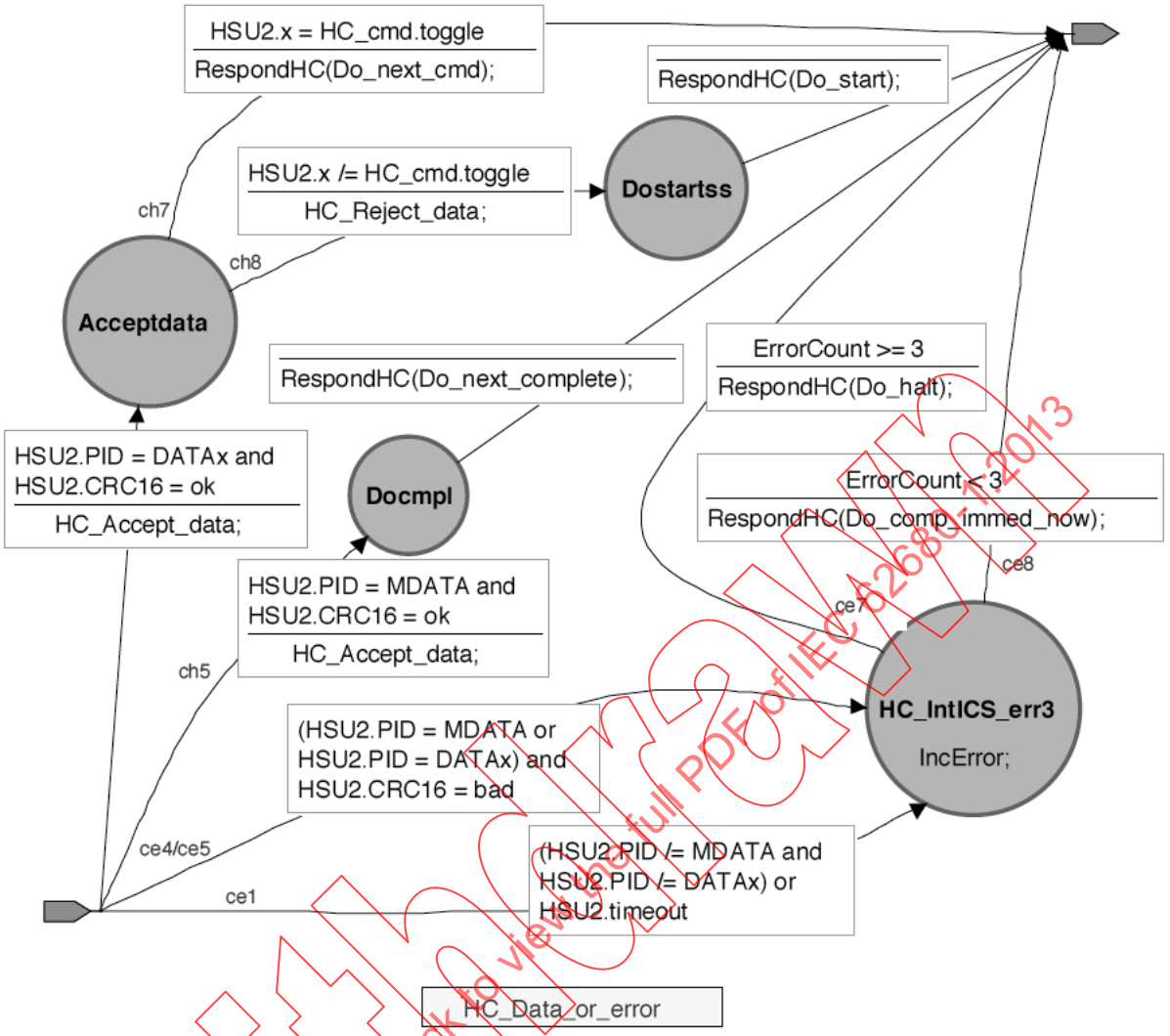


Figure 11-78. HC_Data_or_Error State Machine

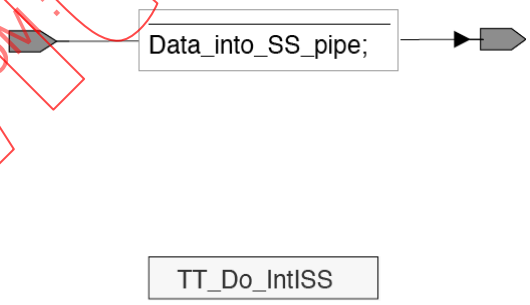


Figure 11-79. Interrupt IN Start-split Transaction TT State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

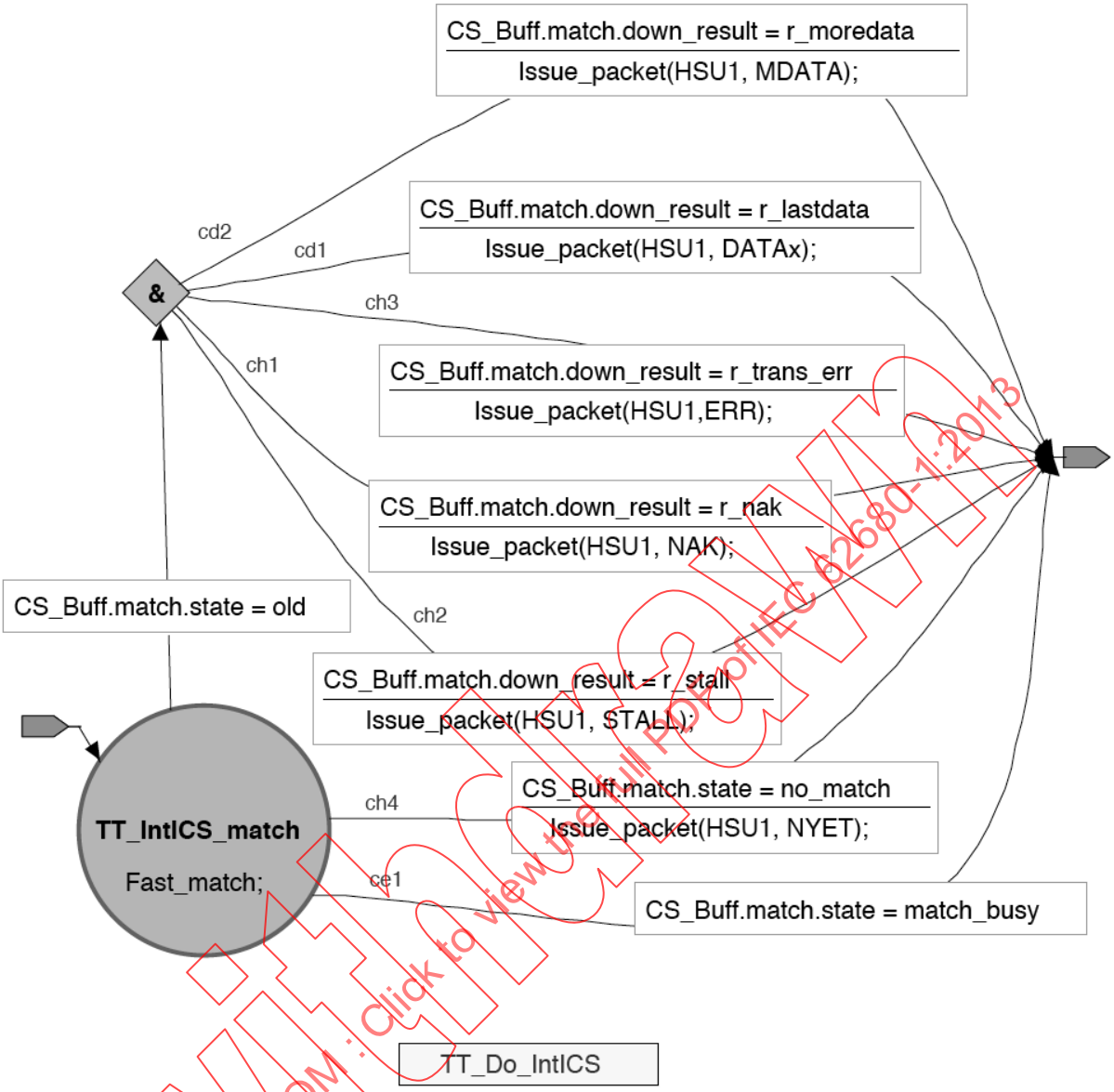


Figure 11-80. Interrupt IN Complete-split Transaction TT State Machine

11.20.3 Interrupt OUT Sequencing

Interrupt OUT split transactions are scheduled by the host controller as normal high-speed transactions with the start- and complete-splits scheduled as described previously.

When there are several full-/low-speed transactions allocated for a given microframe, they are saved by the high-speed handler in the TT in the start-split pipeline stage. The start-splits are saved in the order they are received until the end of the microframe. At the end of the microframe, these transactions are available to be issued by the full-/low-speed handler on the full-/low-speed bus in the order they were received.

In a following microframe (as described previously), the full-/low-speed handler issues the transactions that had been saved in the start-split pipeline stage on the downstream facing full-/low-speed bus. Some transactions could be leftover from a previous microframe since the high-speed schedule was built assuming best case bit stuffing and the full-/low-speed transactions could be taking longer on the full-/low-speed bus. As the full-/low-speed handler issues transactions on the downstream facing full-/low-speed bus, it saves the results in the

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

periodic complete-split pipeline stage and then advances to the next transaction in the start-split pipeline.

In a following microframe (as described previously), the host controller issues a high-speed complete-split transaction and the TT responds appropriately.

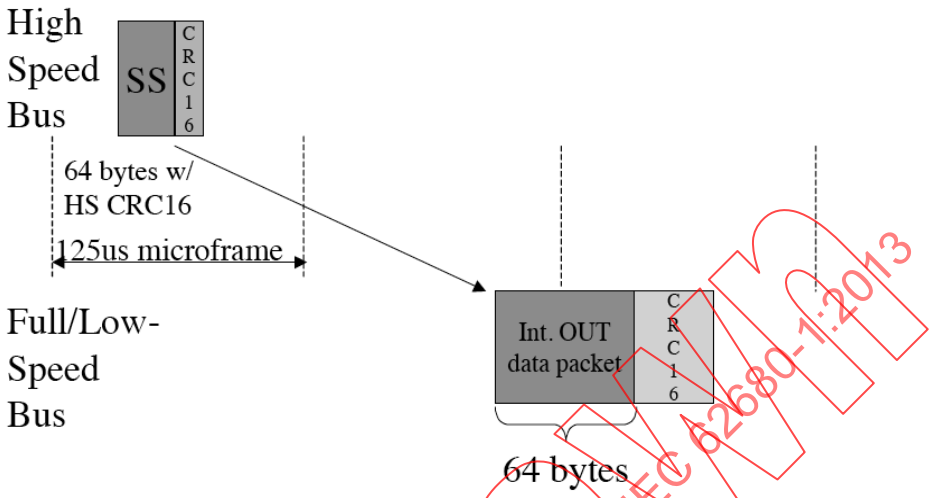


Figure 11-81. Example of CRC16 Handling for Interrupt OUT

The start-split transaction for an interrupt OUT transaction includes a normal CRC16 field for the high-speed data packet of the data phase of the start-split transaction. However, the data payload of the data packet contains only the data payload of the corresponding full-/low-speed data packet; i.e., there is only a single CRC16 in the data packet of the start-split transaction. The TT high-speed handler must check the CRC on the start-split and ignore the start-split if there is a failure in the CRC check of the data packet. If the start-split has a CRC check failure, the full-speed transaction must not be started on the downstream bus. Figure 11-81 shows an example of the CRC16 handling for an interrupt OUT transaction and its start-split.

11.20.4 Interrupt IN Sequencing

When the high-speed handler receives an interrupt start-split transaction, it saves the packet in the start-split pipeline stage. In this fashion, it accumulates some number of start-split transactions for a following microframe.

At the beginning of the next microframe (as described previously), these transactions are available to be issued by the full-/low-speed handler on the downstream full-/low-speed bus in the order they were saved in the start-split pipeline stage. The full-/low-speed handler issues each transaction on the downstream facing bus. The full-/low-speed handler responds to the full-/low-speed transaction with an appropriate handshake as described in Chapter 8. The full-/low-speed handler saves the results of the transaction (data, NAK, STALL, trans_err) in the complete-split pipeline stage.

During following microframes, the host controller issues high-speed complete-split transactions to retrieve the data/handshake from the high-speed handler. When the high-speed handler receives s complete-split transaction, the TT returns whatever data it has received during a microframe. If the full-/low-speed transaction was started and completed in a single microframe, the TT returns all the data for the transaction in the complete-split response occurring in the following microframe. If the full-/low-speed CRC check passes, the appropriate DATA0/1 PID for the data packet is used. If the full-/low-speed CRC check fails, an ERR handshake is used and there is no data packet as part of the complete-split transaction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

If the full-/low-speed transaction spanned a microframe, the TT requires two complete-splits (in two subsequent microframes) to return all the data for the full-/low-speed transaction. The data packet PID for the first complete-split must be an MDATA to tell the host controller that another complete-split is required for this endpoint. This MDATA response is made without performing a CRC check (since the CRC16 field has not yet been received on the full-/low-speed bus). The complete-split in the next microframe must use a DATA0/1 PID if the CRC check passes. If the CRC check fails, an ERR handshake response is made instead and there is no data packet as part of the complete-split transaction. Since full-speed interrupt transactions are limited to 64 data bytes or less (and low-speed interrupt transactions are limited to 8 data bytes or less), no full-/low-speed interrupt transaction can span more than a single microframe boundary; i.e., no more than two microframes are ever required to complete the transaction.

The complete-split transaction for an interrupt IN transaction must not include the CRC16 field received from the full-/low-speed data packet (i.e., only a high-speed CRC16 field is used in split transactions). The TT must use a high-speed CRC16 on each complete-split data packet. If the full-speed handler detects a failed CRC check, it must use an ERR handshake response in the complete-split transaction to reflect that error to the high-speed host controller. The host controller must check the CRC16 on each returned complete-split data packet. A CRC failure (or ERR handshake) on any (partial) complete-split is reflected as a CRC failure on the total full-/low-speed transaction. This means that for a case where a full-/low-speed interrupt spans a microframe boundary, the host controller can accept the first complete-split without errors, then the second complete-split can indicate that the data from the first complete-split must be rejected as if it were never received by the host controller. Figure 11-82 shows an example of an interrupt IN and its CRC16 handling with corresponding complete-split responses.

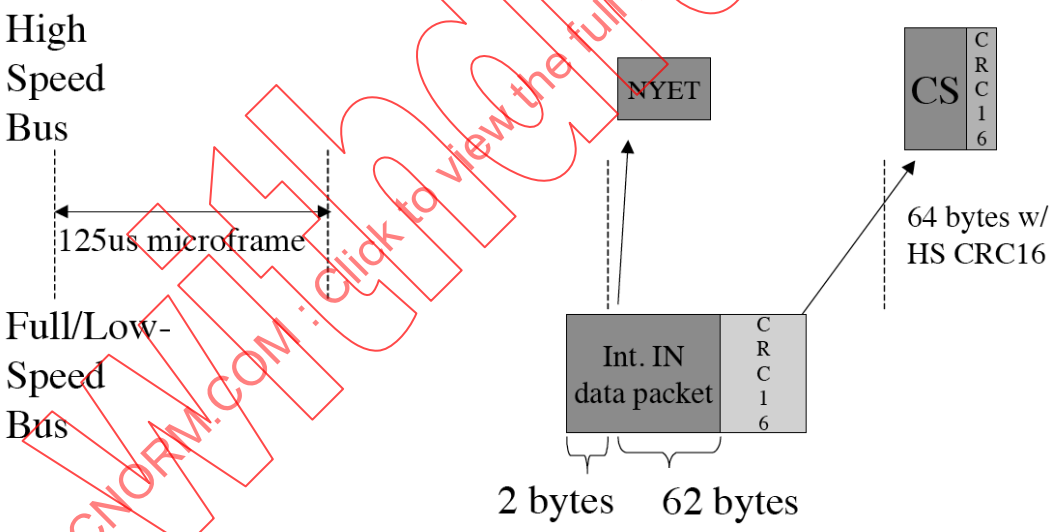


Figure 11-82. Example of CRC16 Handling for Interrupt IN

11.21 Isochronous Transaction Translation Overview

Isochronous split transactions are handled by the host by scheduling start- and complete-split transactions as described previously. Isochronous IN split transactions have more than two schedule entries:

- One entry for the start-split transaction in the microframe before the earliest the full-speed transaction can occur
- Other entries for the complete-splits in microframes after the data can occur on the full-speed bus (similar to interrupt IN scheduling)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Furthermore, isochronous transactions are split into microframe sized pieces; e.g., a 300 byte full-speed transaction is budgeted multiple high-speed split transactions to move data to/from the TT. This allows any alignment of the data for each microframe.

Full-speed isochronous OUT transactions issued by a TT do not have corresponding complete-split transactions. They must only have start-split transaction(s).

The host controller must preserve the same order for the complete-split transactions (as for the start-split transactions) for IN handling.

Isochronous INs have start- and complete- split transactions. The “first” high-speed split transaction for a full-speed endpoint is always a start-split transaction and the second (and others as required) is always a complete-split no matter what the high-speed handler of the TT responds.

The full-/low-speed handler recombines OUT data in its local buffers to recreate the single full-speed data transaction and handle the microframe error cases. The full-/low-speed handler splits IN response data on microframe boundaries.

Microframe buffers always advance no matter what the interactions with the host controller or the full-speed handler.

11.21.1 Isochronous Split Transaction Sequences

The flow sequence and state machine figures show the transitions required for high-speed split transactions for a full-speed isochronous transfer type for a single endpoint. These figures must not be interpreted as showing any particular specific timing. In particular, high-speed or full-speed transactions for other endpoints may occur before or after these split transactions. Specific details are described as appropriate.

In contrast to bulk/control processing, the full-speed handler must not do local retry processing on the full-speed bus in response to transaction errors (including timeout) of an isochronous transaction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

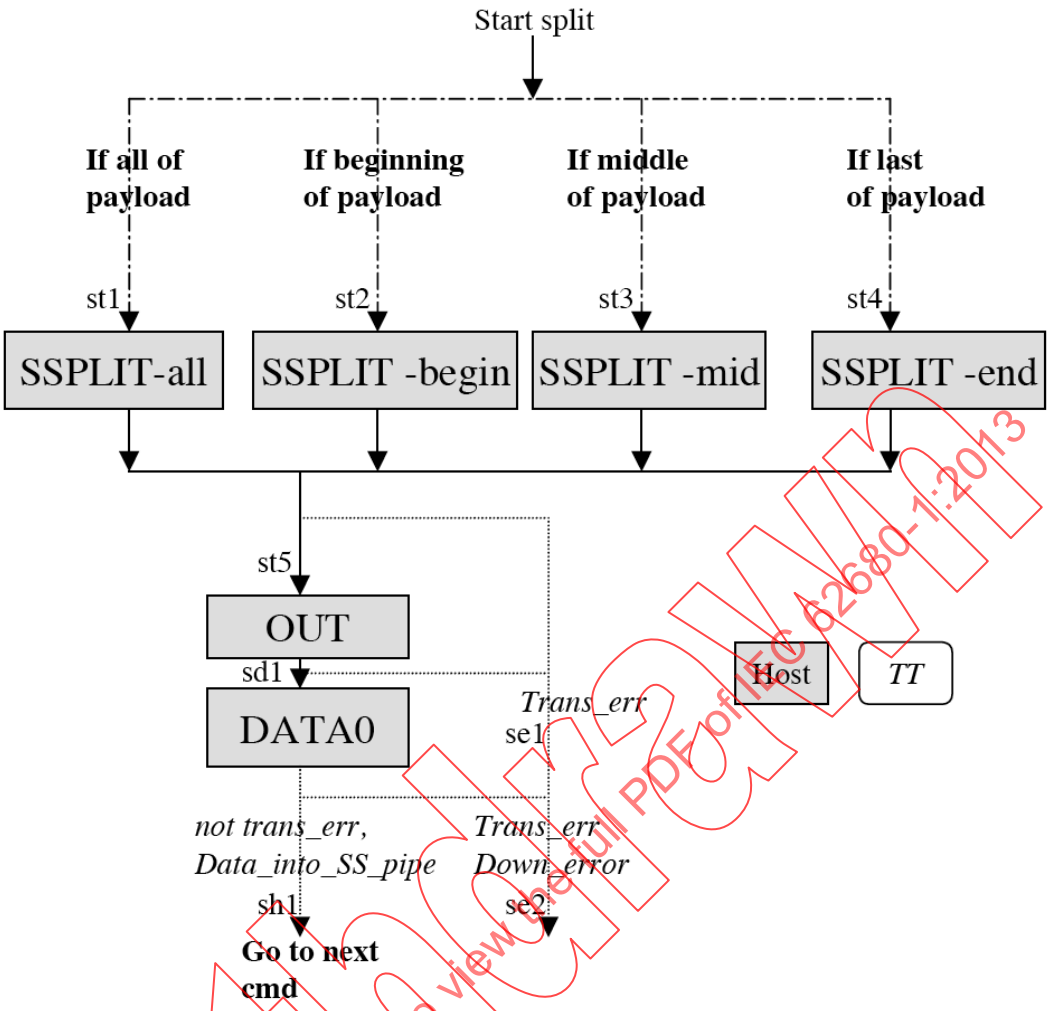


Figure 11-83. Isochronous OUT Start-split Transaction Sequence

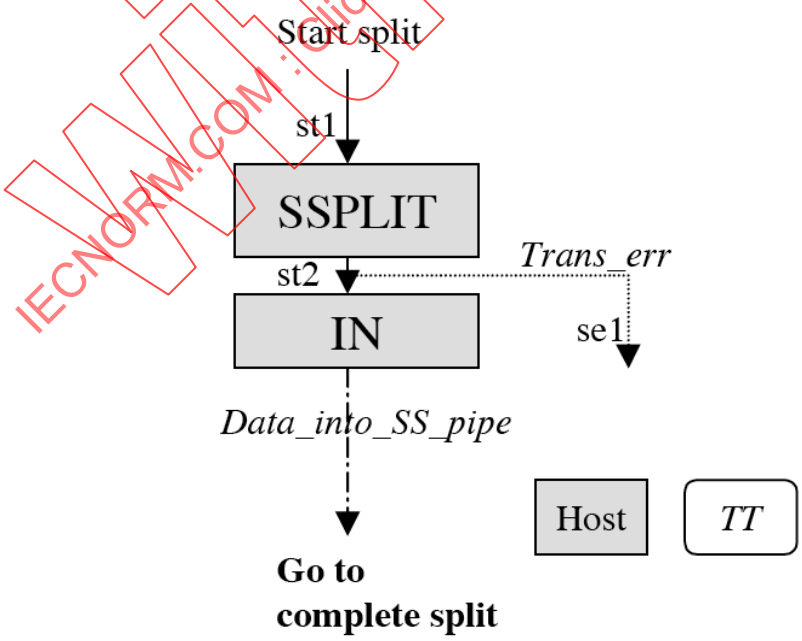


Figure 11-84. Isochronous IN Start-split Transaction Sequence

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

In Figure 11-85, the high-speed handler returns an ERR handshake for a “transaction error” of the full-speed transaction.

The high-speed handler returns an NYET handshake when it cannot find a matching entry in the complete-split pipeline stage. This handles the case where the host controller issued the first high-speed complete-split transaction, but the full-/low-speed handler has not started the transaction yet or has not yet received data back from the full-speed device. This can be due to a delay from starting previous full-speed transactions.

The transition labeled "TAdvance" indicates that the host advances to the next transaction for this full-speed endpoint.

The transition labeled "DAdvance" indicates that the host advances to the next data area of the current transaction for the current full-speed endpoint.

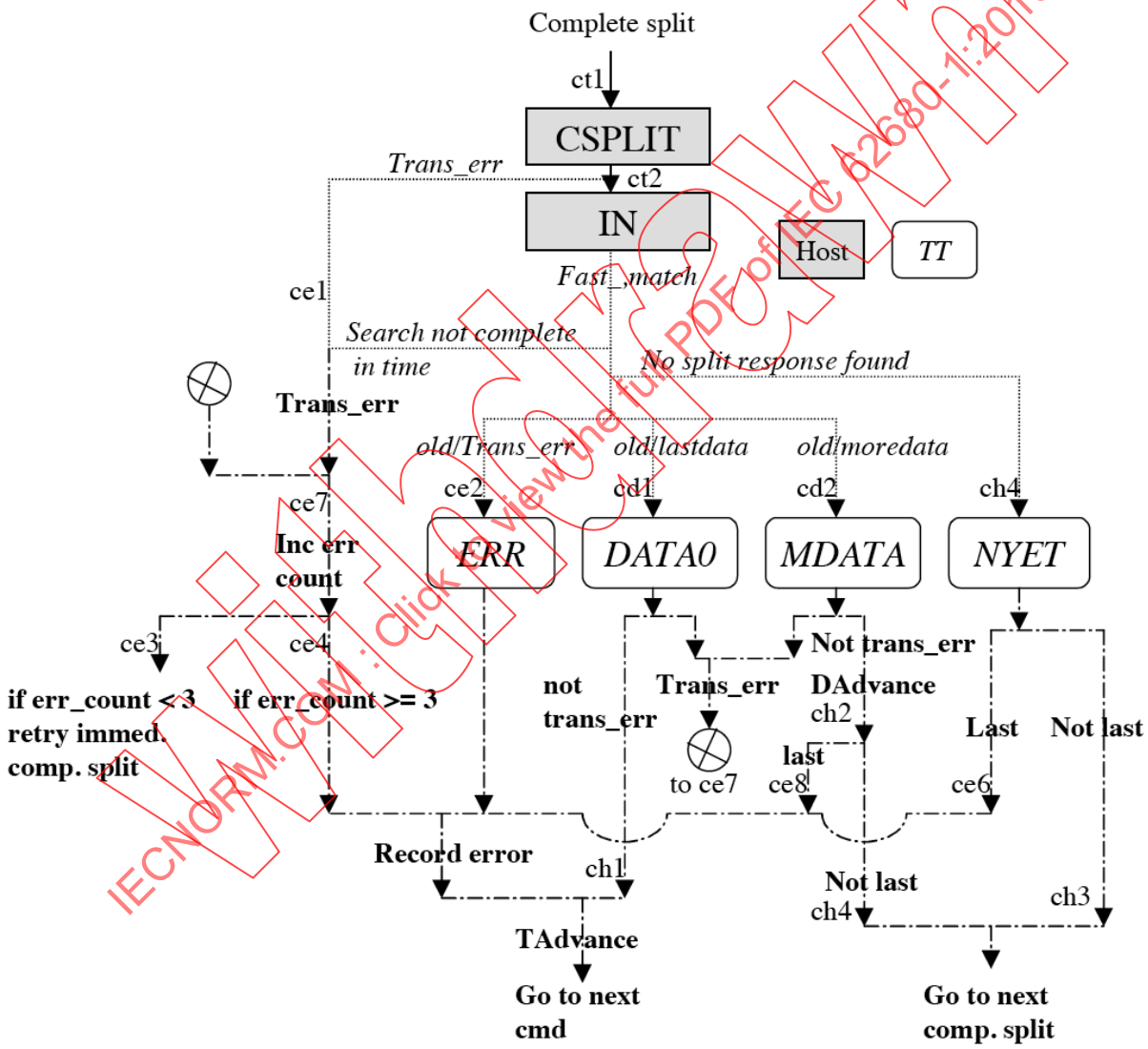


Figure 11-85. Isochronous IN Complete-split Transaction Sequence

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.21.2 Isochronous Split Transaction State Machines

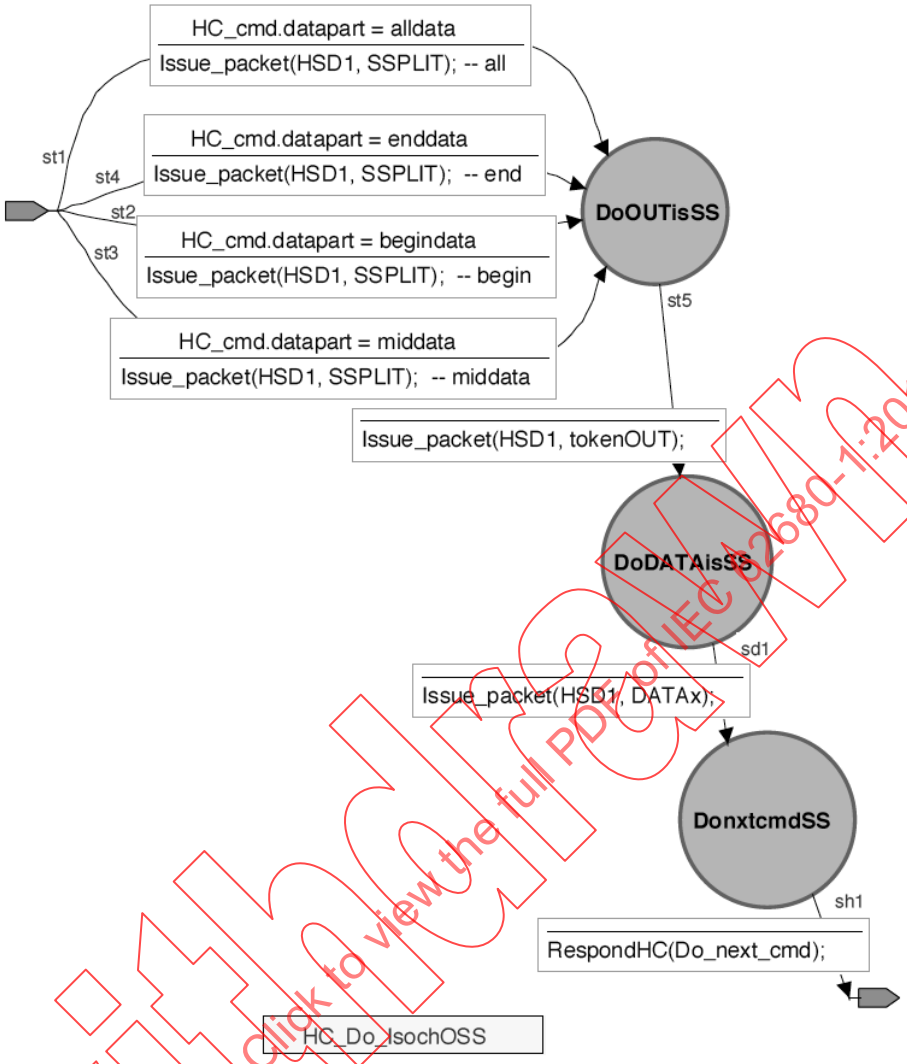


Figure 11-86. Isochronous OUT Start-split Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

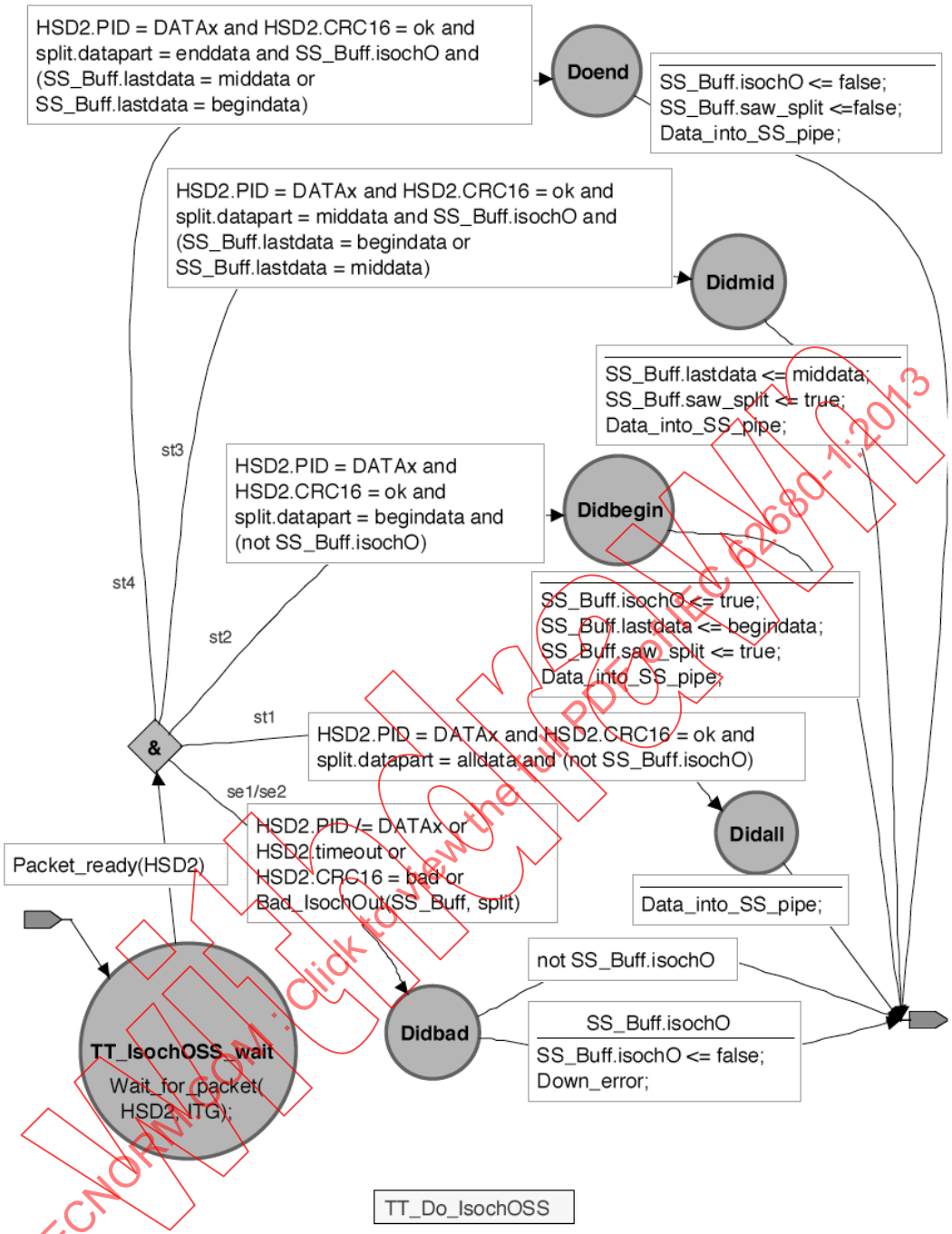


Figure 11-87. Isochronous OUT Start-split Transaction TT State Machine

There is a condition in Figure 11-87 on transition se1/se2 labeled “Bad_IsochOut”. This condition is true when none of the conditions on transitions st1 through st4 are true. The action labeled “Down_error” records an error to be indicated on the downstream facing full-speed bus for the transaction corresponding to this start-split.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

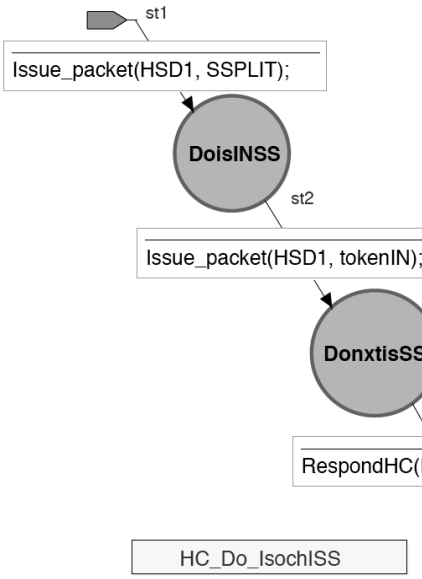


Figure 11-88. Isochronous IN Start-split Transaction Host State Machine

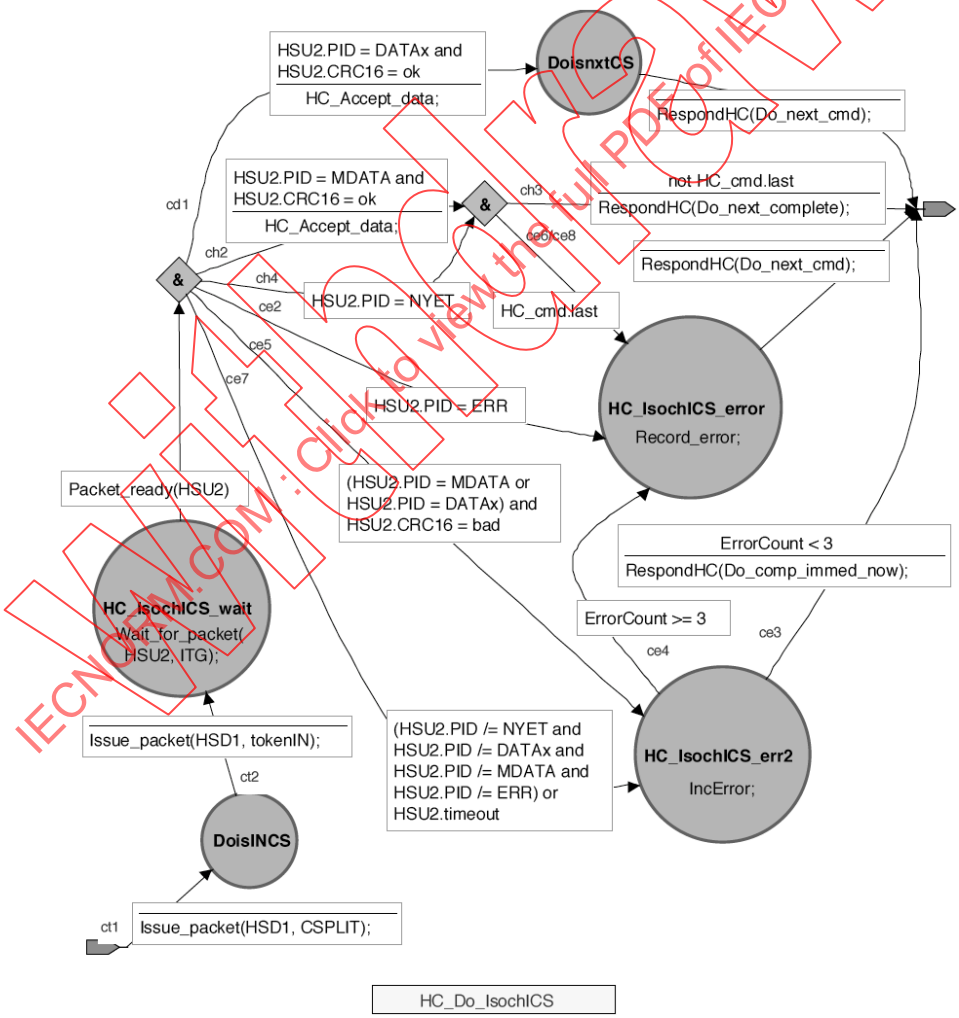


Figure 11-89. Isochronous IN Complete-split Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

In Figure 11-89, the transition “ce8” occurs when the high-speed handler responds with an MDATA to indicate there is more data for the full-speed transaction, but the host controller knows that this is the last scheduled complete-split for this endpoint for this frame. If a DATA0 response from the high-speed handler is not received before the last scheduled complete-split, the host controller records an error and proceeds to the next transaction for this endpoint (in the next frame).

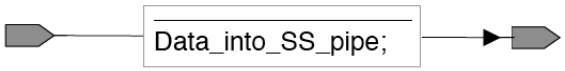




Figure 11-90. Isochronous IN Start-split Transaction TT State Machine

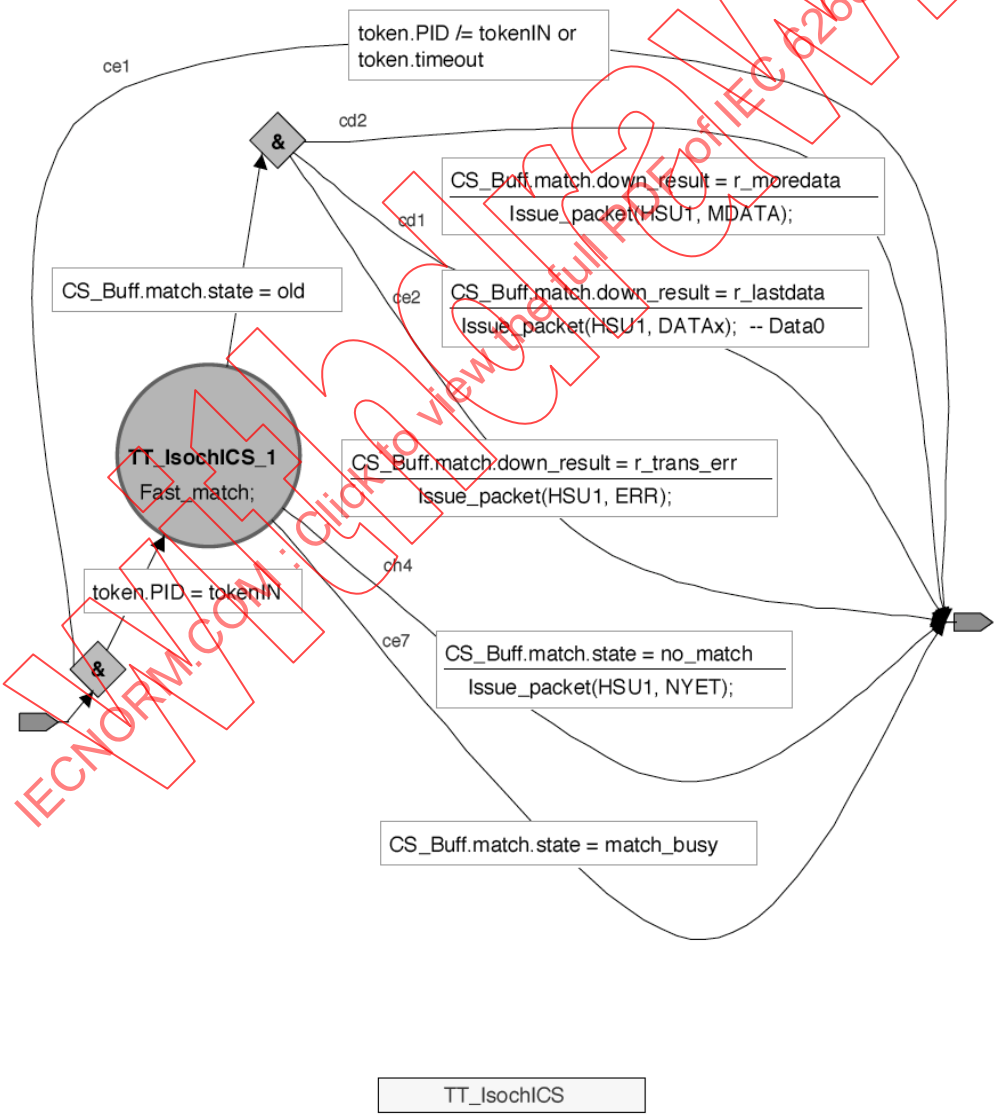


Figure 11-91. Isochronous IN Complete-split Transaction TT State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.21.3 Isochronous OUT Sequencing

The host controller and TT must ensure that errors that can occur in split transactions of an isochronous full-speed transaction translate into a detectable error. For isochronous OUT split transactions, once the high-speed handler has received an “SSPLIT-begin” start-split transaction token packet, the high-speed handler must track start-split transactions that are received for this endpoint. The high-speed handler must track that a start-split transaction is received each and every microframe until an “SSPLIT-end” split transaction token packet is received for this endpoint. If a microframe passes without the high-speed handler receiving a start-split for this full-speed endpoint, it must ensure that the full-speed handler forces a bitstuff error on the full-speed transaction. Any subsequent “SPLIT-middle” or “SPLIT-end” start-splits for the same endpoint must be ignored until the next non “SPLIT-middle” and non “SPLIT-end” is received (for any endpoint supported by this TT).

The start-split transaction for an isochronous OUT transaction must not include the CRC16 field for the full-speed data packet. For a full-speed transaction, the host would compute the CRC16 of the data packet for the full data packet (e.g., a 1023 byte data packet uses a single CRC16 field that is computed once by the host controller). For a split transaction, any isochronous OUT full-speed transaction is subdivided into multiple start-splits, each with a data payload of 188 bytes or less. For each of these start-splits, the host computes a high-speed CRC16 field for each start-split data packet. The TT high-speed handler must check each high-speed CRC16 value on each start-split. The TT full-speed handler must locally generate the CRC16 value for the complete full-speed data packet. Figure 11-92 shows an example of a full-speed isochronous OUT packet and the high-speed start-splits with their CRC16 fields.

If there is a CRC check failure on the high-speed start-split, the high-speed handler must indicate to the full-speed handler that there was an error in the start-split for the full-speed transaction. If the transaction has been indicated as having a CRC failure (or if there is a missed start-split), the full-speed handler uses the defined mechanism for forcing a downstream corrupted packet. If the first start-split has a CRC check failure, the full-speed transaction must not be started on the downstream bus.

Additional high-speed start-split transactions for the same endpoint must be ignored after a CRC check fails, until the high-speed handler receives either an “SSPLIT-end” start-split transaction token packet for that endpoint or a start-split for a different endpoint.

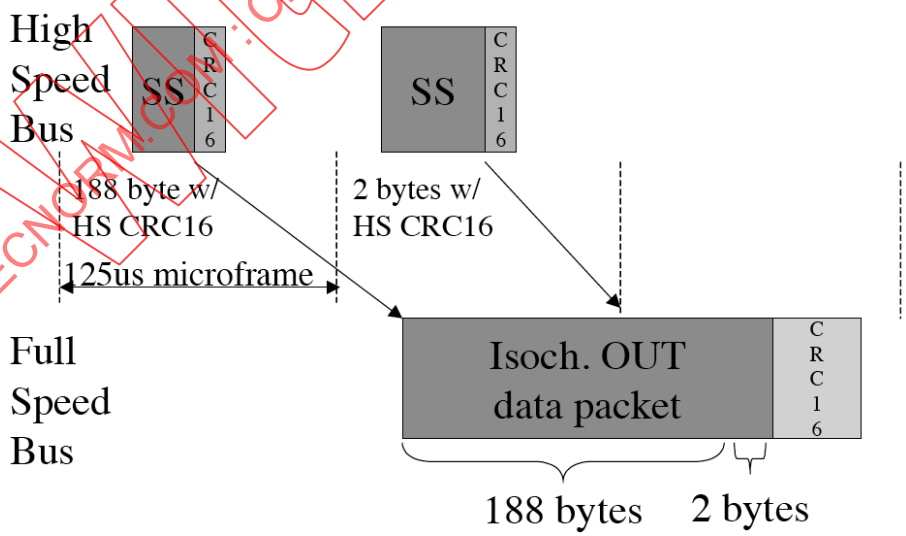


Figure 11-92. Example of CRC16 Isochronous OUT Data Packet Handling

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.21.4 Isochronous IN Sequencing

The complete-split transaction for an isochronous IN transaction must not include the CRC16 field for the full-speed data packet (e.g., only a high-speed CRC16 field is used in split transactions). The TT must not pass the full-speed value received from the device and instead only use high-speed CRC16 values for complete-split transactions. If the full-speed handler detects a failed CRC check at the end of the data packet (e.g., after potentially several complete-split transactions on high-speed), the handler must use an ERR handshake response to reflect that error to the high-speed host controller. The host controller must check the CRC16 on each returned high-speed complete-split. A CRC failure (or ERR handshake) on any (partial) complete-split is reflected by the host controller as a CRC failure on the total full-speed transaction. Figure 11-93 shows an example of the relationships of the full-speed data packet and the high-speed complete-splits and their CRC16 fields.

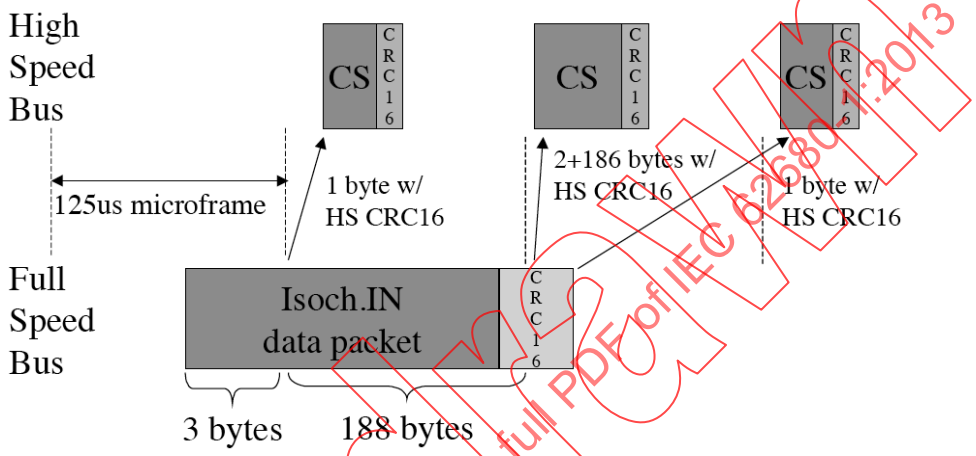


Figure 11-93. Example of CRC16 Isochronous IN Data Packet Handling

11.22 TT Error Handling

The TT has the same requirements for handling errors as a host controller or hub. In particular:

- If the TT is receiving a packet at EOF2 of the downstream facing bus, it must disable the downstream facing port that is currently transmitting.
- If the TT is transmitting a packet near EOF1 of the downstream facing bus, it must force an abnormal termination sequence as defined in Section 11.3.3 and stop transmitting.
- If the TT is going to transmit a non-periodic full-/low-speed transaction, it must determine that there is sufficient time remaining before EOF1 to complete the transaction. This determination is based on normal sequencing of the packets in the transaction. Since the TT has no information about data payload size for INs, it must use the maximum allowed size allowed for the transfer type in its determination. Periodic transactions do not need to be included in this test since the microframe pipeline is maintained separately.

11.22.1 Loss of TT Synchronization With HS SOFs

The hub has a timer it uses for (micro)frame maintenance. It has a 1 ms frame timer when operating at full-/low-speed for enforcing EOF with downstream connected devices. It has a 125 μs microframe timer when operating at high-speed for enforcing EOF with high-speed devices. It also uses the 125 μs microframe timer to create a 1 ms frame timer for enforcing EOF with downstream full-/low-speed devices when operating at high-speed. The hub (micro)frame timer must always stay synchronized with host generated SOFs to keep the bus operating correctly

In normal hub repeater (full- or high-speed) operation (e.g., not involving a TT), the (micro)frame timer loses synchronization whenever it has missed SOFs for three consecutive

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

microframes. While timer synchronization is lost, the hub does not establish upstream connectivity. Downstream connectivity is established normally, even when timer synchronization is lost. When the timer is synchronized, the hub allows upstream connectivity to be established when required. The hub is responsible for ensuring that there is no signaling being repeated/transmitted upstream from a device after the EOF2 point in any (micro)frame. The hub must not establish upstream connectivity if it has lost (micro)frame timer synchronization since it no longer knows accurately where the EOF2 point is.

11.22.2 TT Frame and Microframe Timer Synchronization Requirements

When the hub is operating at high-speed and has full-/low-speed devices connected on its downstream facing ports (e.g., a TT is active), the hub has additional responsibilities beyond enforcement of the (high-speed) EOF2 point on its upstream facing port in every microframe. The TT must also generate full-speed SOFs downstream and ensure that the TT operates correctly (in bridging high-speed and full-/low-speed operation).

A high-speed operating hub synchronizes its microframe timer to 125 μs SOFs. However, in order to generate full-speed downstream SOFs, it must also have a 1 ms frame timer. It generates this 1 ms frame timer by recognizing zeroth microframe SOFs, e.g., a high-speed SOF when the frame number value changes compared to SOF of the immediately previous microframe.

In order to create the 1 ms frame timer, the hub must successfully receive a zeroth microframe SOF after its microframe timer is synchronized. In order to recognize a zeroth microframe SOF, the hub must successfully receive SOFs for two consecutive microframes where the frame number increments by 1 (mod 2^11). When the hub has done this, it knows that the second SOF is a zeroth microframe SOF and thereby establishes a 1 ms frame timer starting time. Note that a hub can synchronize both timers with as few as two SOFs if the SOFs are for microframe 7 and microframe 0, i.e., if the second SOF is a zeroth microframe SOF.

Once the hub has synchronized its 1 ms frame timer, it can keep that timer synchronized as long as it keeps its 125 μs microframe timer synchronized (since it knows that every 8 microframes from the zeroth microframe SOF is a 1 ms frame). In particular, the hub can keep its frame timer synchronized even if it misses zeroth microframe SOFs (as long as the microframe timer stays synchronized).

So in summary, the hub can synchronize its 125 μs microframe timer after receiving SOFs of two consecutive microframes. It synchronizes its 1 ms frame timer when it receives a zeroth microframe SOF (and the microframe timer is synchronized). The 125 μs microframe timer loses synchronization after three SOFs for consecutive microframes have been missed. This also causes the 1 ms frame timer to lose synchronization at the same time.

The TT must only generate full-speed SOFs downstream when its 1 ms frame timer is synchronized.

Correct internal operation of the TT is dependent on both timers. The TT must accurately know when microframes occur to enforce its microframe pipeline abort/free rules. It knows this based on a synchronized microframe timer (for generally incrementing the microframe number) and a synchronized frame timer (to know when the zeroth microframe occurs).

Since loss of microframe timer synchronization immediately causes loss of frame timer synchronization, the TT stops normal operation once the microframe timer loses synchronization. In an error free environment, microframe timer synchronization can be restored after receiving the two SOFs for the next two consecutive microframes (e.g., synchronization is restored at least 250 μs after synchronization loss). As long as SOFs are not missed, frame timer synchronization will be restored in less than 1 ms after microframe synchronization. Note that frame timer synchronization can be restored in a high-speed operating case in much less time (0.250-1.250 ms) than the 2-3 ms required in full-speed

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

operation. Once the frame timer is synchronized, SOFs can be issued on downstream facing full-speed ports for the beginning of the next frame.

Once the hub detects loss of microframe timer synchronization, its TT(s):

1. Must respond to periodic complete-splits with any responses buffered in the periodic pipeline (only good for at most 1 microframe of complete-splits).
2. Must abort any buffered periodic start-split transactions in the periodic pipeline.
3. Must ignore any high-speed periodic start-splits.
4. Must stop issuing full-speed SOFs on downstream facing full-speed ports (and low-speed keep-alives on low-speed ports).
5. Must not start issuing subsequent periodic full-/low-speed transactions on downstream facing full-/low-speed ports.
6. Must respond to high-speed start-split bulk/control transactions.
7. Buffered bulk/control results must respond to high-speed complete-split transactions.
8. Pending bulk/control transactions must not be issued to full-/low-speed downstream facing ports. The TT buffers used to hold bulk/control transactions must be preserved until the microframe timer is re-synchronized. (Or until a Clear_TT_Buffer request is received for the transaction).

Note that in any case a TT must not issue transactions of any speed on downstream facing ports when its upstream facing port is suspended.

A TT only restores normal operation on downstream facing full-/low-speed ports after both microframe and frame timers are synchronized. Figure Figure 11-93 summarizes the relationship between high-speed SOFs and the TT frame and microframe timer synchronization requirements on start-splits.

For suspend sequencing of a hub, a hub will first lose microframe/frame timer synchronization at the same time. This will cause its TT(s) to stop issuing SOFs (which should be the only transactions keeping the downstream facing full-/low-speed ports out of suspend). Then the hub (along with any downstream devices) will enter suspend.

Upon a resume, the hub will first restore its microframe timer synchronization (after high-speed transactions continue). Then in less than 1 ms (assuming no errors), the frame timer will be synchronized and the TT can start normal operation (including SOFs/keep-alives on downstream facing full-/low-speed ports).

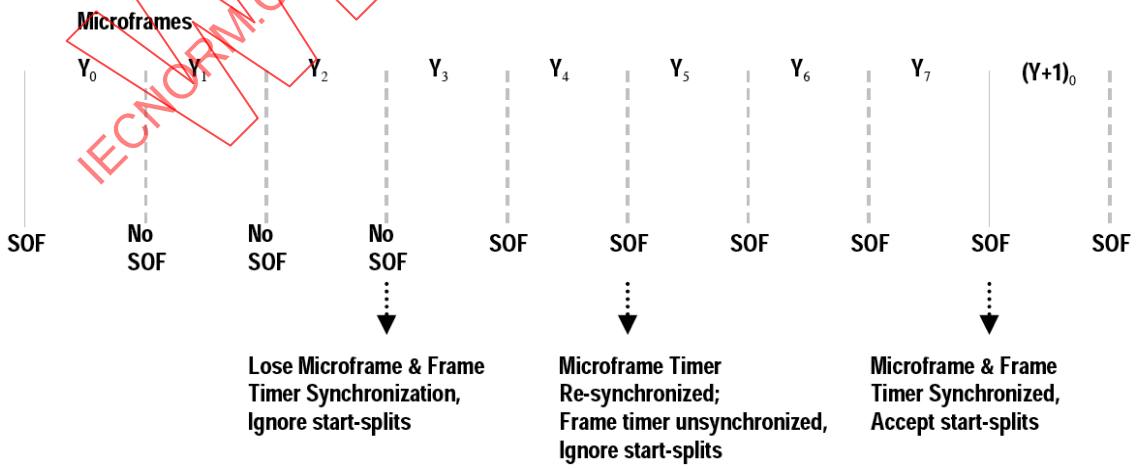


Figure 11-94. Example Frame/Microframe Synchronization Events

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.23 Descriptors

Hub descriptors are derived from the general USB device framework. Hub descriptors define a hub device and the ports on that hub. The host accesses hub descriptors through the hub's default pipe.

The USB specification (refer to Chapter 9) defines the following descriptors:

- Device
- Configuration
- Interface
- Endpoint
- String (optional)

The hub class defines additional descriptors (refer to Section 11.23.2). In addition, vendor-specific descriptors are allowed in the USB device framework. Hubs support standard USB device commands as defined in Chapter 9.

11.23.1 Standard Descriptors for Hub Class

The hub class pre-defines certain fields in standard USB descriptors. Other fields are either implementation-dependent or not applicable to this class.

A hub returns different descriptors based on whether it is operating at high-speed or full-/low-speed. A hub can report three different sets of the descriptors: one descriptor set for full-/low-speed operation and two sets for high-speed operation.

A hub operating at full-/low-speed has a device descriptor with a bDeviceProtocol field set to zero(0) and an interface descriptor with a bInterfaceProtocol field set to zero(0). The rest of the descriptors are the same for all speeds.

A hub operating at high-speed can have one of two TT organizations: single TT or multiple TT. All hubs must support the single TT organization. A multiple TT hub has an additional interface descriptor (with a corresponding endpoint descriptor). The first set of descriptors shown below must be provided by all hubs. A hub that has a single TT must set the bDeviceProtocol field of the device descriptor to one(1) and the interface descriptor bInterfaceProtocol field set to 0.

A multiple TT hub must set the bDeviceProtocol field of the device descriptor to two (2). The first interface descriptor has the bInterfaceProtocol field set to one(1). Such a hub also has a second interface descriptor where the bInterfaceProtocol is set to two(2). When the hub is configured with an interface protocol of one(1), it will operate as a single TT organized hub. When the hub is configured with an interface protocol of two(2), it will operate as a multiple TT organized hub. The TT organization must not be changed while the hub has full-/low-speed transactions in progress.

Note: For the descriptors and fields shown below, the bits in a field are organized in a little-endian fashion; that is, bit location 0 is the least significant bit and bit location 7 is the most significant bit of a byte value.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Full-/Low-speed Operating Hub

Device Descriptor (full-speed information):

bLength	12H
bDescriptorType	1
bcdUSB	0200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
bNumConfigurations	1

Device_Qualifier Descriptor (high-speed information):

bLength	0AH
bDescriptorType	6
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	1 (for single TT) or 2 (for multiple TT)
bMaxPacketSize0	64
bNumConfigurations	1

Configuration Descriptor (full-speed information):

bLength	09H
bDescriptorType	2
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in full-/low-speed configuration

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	i

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

Other_Speed_Configuration Descriptor (High-speed information):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1 (for single TT) or 2 (for multiple TT)
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in high-speed configuration

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0 (for single TT) 1 (for multiple TT)
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Interface Descriptor (present if multiple TT hub):

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	2
iInterface	i

Endpoint Descriptor (present if multiple TT hub):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

High-speed Operating Hub with Single TT

Device Descriptor (High-speed information):

bLength	12H
bDescriptorType	1
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	1
bMaxPacketSize0	64
bNumConfigurations	1

Device_Qualifier Descriptor (full-speed information):

bLength	0AH
bDescriptorType	6
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
bNumConfigurations	1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Configuration Descriptor (high-speed information):

bLength	09H
bDescriptorType	2
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in this configuration

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0 (single TT)
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

Other_Speed_Configuration Descriptor (full-speed information):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in high-speed configuration

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

High-speed Operating Hub with Multiple TTs

Device Descriptor (High-speed information):

bLength	12H
bDescriptorType	1
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	2 (multiple TTs)
bMaxPacketSize0	64
bNumConfigurations	1

Device_Qualifier Descriptor (full-speed information):

bLength	0AH
bDescriptorType	6
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
bNumConfigurations	1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Configuration Descriptor (high-speed information):

bLength	09H
bDescriptorType	2
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in this configuration

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	1 (single TT)
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	1
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	2 (multiple TTs)
iInterface	i

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Endpoint Descriptor:

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

Other_Speed_Configuration Descriptor (full-speed information):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in high-speed configuration

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

The hub class driver retrieves a device configuration from the USB System Software using the GetDescriptor() device request. The only endpoint descriptor that is returned by the GetDescriptor() request is the Status Change endpoint descriptor.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.23.2 Class-specific Descriptors

11.23.2.1 Hub Descriptor

Table 11-13 outlines the various fields contained by the hub descriptor.

Table 11-13. Hub Descriptor

Offset	Field	Size	Description
0	bDescLength	1	Number of bytes in this descriptor, including this byte
1	bDescriptorType	1	Descriptor Type, value: 29H for hub descriptor
2	bNbrPorts	1	Number of downstream facing ports that this hub supports
3	wHubCharacteristics	2	D1...D0: Logical Power Switching Mode 00: Ganged power switching (all ports' power atonce) 01: Individual port power switching 1X: Reserved. Used only on 1.0 compliant hubs that implement no power switching D2: Identifies a Compound Device 0: Hub is not part of a compound device. 1: Hub is part of a compound device. D4...D3: Over-current Protection Mode 00: Global Over-current Protection. The hub reports over-current as a summation of all ports' current draw, without a breakdown of individual port over-current status. 01: Individual Port Over-current Protection. The hub reports over-current on a per-port basis. Each port has an over-current status. 1X: No Over-current Protection. This option is allowed only for bus-powered hubs that do not implement over-current protection. D6...D5: TT Think TIme 00: TT requires at most 8 FS bit times of inter transaction gap on a full-/low-speed downstream bus. 01: TT requires at most 16 FS bit times. 10: TT requires at most 24 FS bit times. 11: TT requires at most 32 FS bit times. D7: Port Indicators Supported 0: Port Indicators are not supported on its downstream facing ports and the PORT_INDICATOR request has no effect. 1: Port Indicators are supported on its downstream facing ports and the PORT_INDICATOR request controls the indicators. See Section 11.5.3. D15...D8: Reserved

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Offset	Field	Size	Description
5	bPwrOn2PwrGood	1	Time (in 2 ms intervals) from the time the power-on sequence begins on a port until power is good on that port. The USB System Software uses this value to determine how long to wait before accessing a powered-on port.
6	bHubContrCurrent	1	Maximum current requirements of the Hub Controller electronics in mA.
7	DeviceRemovable	Variable, depend-ing on number of ports on hub	Indicates if a port has a removable device attached. This field is reported on byte-granularity. Within a byte, if no port exists for a given location, the field representing the port characteristics returns 0. Bit value definition: 0B - Device is removable. 1B - Device is non-removable This is a bitmap corresponding to the individual ports on the hub: Bit 0: Reserved for future use. Bit 1: Port 1 Bit 2: Port 2 .... Bit *n*: Port *n* (implementation-dependent, up to a maximum of 255 ports).
Variable	*PortPwrCtrlMask*	Variable, dependin g on number of ports on hub	This field exists for reasons of compatibility with software written for 1.0 compliant devices. All bits in this field should be set to 1B. This field has one bit for each port on the hub with additional pad bits, if necessary, to make the number of bits in the field an integer multiple of 8.

11.24 Requests

11.24.1 Standard Requests

Hubs have tighter constraints on request processing timing than specified in Section 9.2.6 for standard devices because they are crucial to the "time to availability" of all devices attached to USB. The worst case request timing requirements are listed below (apply to both Standard and Hub Class requests):

1. Completion time for requests with no data stage: 50 ms
2. Completion times for standard requests with data stage(s)
   - Time from setup packet to first data stage: 50 ms
   - Time between each subsequent data stage: 50 ms
   - Time between last data stage and status stage: 50 ms

As hubs play such a crucial role in bus enumeration, it is recommended that hubs average response times be less than 5 ms for all requests.

Table 11-14 outlines the various standard device requests.

Table 11-14. Hub Responses to Standard Device Requests

bRequest	Hub Response
CLEAR_FEATURE	Standard
GET_CONFIGURATION	Standard
GET_DESCRIPTOR	Standard
GET_INTERFACE	Undefined. Hubs are allowed to support only one interface.
GET_STATUS	Standard

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

bRequest	Hub Response
SET_ADDRESS	Standard
SET_CONFIGURATION	Standard
SET_DESCRIPTOR	Optional
SET_FEATURE	Standard
SET_INTERFACE	Undefined. Hubs are allowed to support only one interface.
SYNCH_FRAME	Undefined. Hubs are not allowed to have isochronous endpoints.

Optional requests that are not implemented shall return a STALL in the Data stage or Status stage of the request.

11.24.2 Class-specific Requests

The hub class defines requests to which hubs respond, as outlined in Table 11-15. Table 11-16 defines the hub class request codes. All requests in the table below except SetHubDescriptor() are mandatory.

Table 11-15. Hub Class Requests

Request	bmRequestType	bRequest	wValue	wIndex	wLength	Data
ClearHubFeature	00100000B	CLEAR_ FEATURE	Feature Selector	Zero	Zero	None
ClearPortFeature	00100011B	CLEAR_ FEATURE	Feature Selector	Selector, Port	Zero	None
ClearTTBuffer	00100011B	CLEAR_TT_BUFFER	Dev_Addr, EP_Num	TT_port	Zero	None
GetHubDescriptor	10100000B	GET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor
GetHubStatus	10100000B	GET_ STATUS	Zero	Zero	Four	Hub Status and Change Status
GetPortStatus	10100011B	GET_ STATUS	Zero	Port	Four	Port Status and Change Status
ResetTT	00100011B	RESET_TT	Zero	Port	Zero	None
SetHubDescriptor	00100000B	SET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor
SetHubFeature	00100000B	SET_ FEATURE	Feature Selector	Zero	Zero	None
SetPortFeature	00100011B	SET_ FEATURE	Feature Selector	Selector, Port	Zero	None
GetTTState	10100011B	GET_TT_STATE	TT_Flags	Port	TT State Length	TT State
StopTT	00100011B	STOP_TT	Zero	Port	Zero	None

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 11-16. Hub Class Request Codes

bRequest	Value
GET_ STATUS	0
CLEAR_ FEATURE	1
Reserved (USED IN PREVIOUS SPECIFICATIONS FOR GET_STATE)	2
SET_ FEATURE	3
Reserved for future use	*4-5*
GET_DESCRIPTOR	6
SET_DESCRIPTOR	7
CLEAR_TT_BUFFER	8
RESET_TT	9
GET_TT_STATE	10
STOP_TT	11

Table 11-17 gives the valid feature selectors for the hub class. See Section 11.24.2.6 and Section 11.24.2.7 for a description of the features.

Table 11-17. Hub Class Feature Selectors

	Recipient	Value
C_HUB_LOCAL_POWER	Hub	0
C_HUB_OVER_CURRENT	Hub	1
PORT_CONNECTION	Port	0
PORT_ENABLE	Port	1
PORT_SUSPEND	Port	2
PORT_OVER_CURRENT	Port	3
PORT_RESET	Port	4
PORT_POWER	Port	8
PORT_LOW_SPEED	Port	9
C_PORT_CONNECTION	Port	16
C_PORT_ENABLE	Port	17
C_PORT_SUSPEND	Port	18
C_PORT_OVER_CURRENT	Port	19
C_PORT_RESET	Port	20
PORT_TEST	Port	21
PORT_INDICATOR	Port	22

11.24.2.1 Clear Hub Feature

This request resets a value reported in the hub status.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00100000B	CLEAR_FEATURE	Feature Selector	Zero	Zero	None

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Clearing a feature disables that feature; refer to Table 11-17 for the feature selector definitions that apply to the hub as a recipient. If the feature selector is associated with a status change, clearing that status change acknowledges the change. This request format is used to clear either the C_HUB_LOCAL_POWER or C_HUB_OVER_CURRENT features.

It is a Request Error if *wValue* is not a feature selector listed in Table 11-17 or if *wIndex* or *wLength* are not as specified above.

If the hub is not configured, the hub's response to this request is undefined.

11.24.2.2 Clear Port Feature

This request resets a value reported in the port status.

bmRequestType	bRequest	wValue	wIndex		wLength	Data
00100011B	CLEAR_FEATURE	Feature Selector	Selector	Port	Zero	None

The port number must be a valid port number for that hub, greater than zero. The port field is located in bits 7..0 of the *wIndex* field.

Clearing a feature disables that feature or starts a process associated with the feature; refer to Table 11-17 for the feature selector definitions. If the feature selector is associated with a status change, clearing that status change acknowledges the change. This request format is used to clear the following features:

- PORT_ENABLE
- PORT_SUSPEND
- PORT_POWER
- PORT_INDICATOR
- C_PORT_CONNECTION
- C_PORT_RESET
- C_PORT_ENABLE
- C_PORT_SUSPEND
- C_PORT_OVER_CURRENT

Clearing the PORT_SUSPEND feature causes a host-initiated resume on the specified port. If the port is not in the Suspended state, the hub should treat this request as a functional no-operation.

Clearing the PORT_ENABLE feature causes the port to be placed in the Disabled state. If the port is in the Powered-off state, the hub should treat this request as a functional no-operation.

Clearing the PORT_POWER feature causes the port to be placed in the Powered-off state and may, subject to the constraints due to the hub's method of power switching, result in power being removed from the port. Refer to Section 11.11 on rules for how this request is used with ports that are gang-powered.

The selector field identifies the port indicator selector when clearing a port indicator. The selector field is in bits 15..8 of the *wIndex* field.

It is a Request Error if *wValue* is not a feature selector listed in Table 11-17, if *wIndex* specifies a port that does not exist, or if *wLength* is not as specified above. It is not an error for this request to try to clear a feature that is already cleared (hub should treat as a functional no-operation).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

If the hub is not configured, the hub's response to this request is undefined.

11.24.2.3 Clear TT Buffer

This request clears the state of a Transaction Translator(TT) bulk/control buffer after it has been left in a busy state due to high-speed errors. This request is only defined for non-periodic endpoints; e.g., if it is issued for a periodic endpoint, the response is undefined. After successful completion of this request, the buffer can again be used by the TT with high-speed split transactions for full-/low-speed transactions to attached full-/low-speed devices.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00100011B	CLEAR_TT_BUFFER	Device_Address, Endpoint_Number	TT_port	Zero	None

If the hub supports a TT per port, then wIndex must specify the port number of the TT that encountered the high-speed errors (e.g., with the busy TT buffer). If the hub provides only a single TT, then *wIndex* must be set to one.

The device_address, endpoint_number, and endpoint_type of the full-/low-speed endpoint (as specified in the corresponding split transaction) that may have a busy TT buffer must be specified in the wValue field. The specific bit fields used are shown in Table 11-18.

It is a Request Error if *wIndex* specifies a port that does not exist, or if *wLength* is not as specified above. It is not an error for this request to try to clear a buffer for a transaction that is not buffered by the TT (should treat as a functional no-operation).

If the hub is not configured, the hub's response to this request is undefined.

Table 11-18. wValue Field for Clear_TT_Buffer

Bits	Field
3..0	Endpoint Number
10..4	Device Address
12..11	Endpoint Type
14..13	Reserved, must be zero
15	Direction, 1 = IN, 0 = OUT

11.24.2.4 Get Bus State

Previous versions of USB defined a GetBusState request. This request is no longer defined.

11.24.2.5 Get Hub Descriptor

This request returns the hub descriptor.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10100000B	GET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero	Descriptor Length	Descriptor

The GetDescriptor() request for the hub class descriptor follows the same usage model as that of the standard GetDescriptor() request (refer to Chapter 9). The standard hub descriptor is denoted by using the value *bDescriptorType* defined in Section 11.23.2.1. All hubs are required to implement one hub descriptor, with descriptor index zero.

If *wLength* is larger than the actual length of the descriptor, then only the actual length is returned. If *wLength* is less than the actual length of the descriptor, then only the first

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

wLength bytes of the descriptor are returned; this is not considered an error even if *wLength* is zero.

It is a Request Error if *wValue* or *wIndex* are other than as specified above.

If the hub is not configured, the hub's response to this request is undefined.

11.24.2.6 Get Hub Status

This request returns the current hub status and the states that have changed since the previous acknowledgment.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10100000B	GET_ STATUS	Zero	Zero	Four	Hub Status and Change Status

The first word of data contains wHubStatus (refer to Table 11-19). The second word of data contains wHubChange (refer to Table 11-20).

It is a Request Error if *wValue*, *wIndex*, or *wLength* are other than as specified above.

If the hub is not configured, the hub's response to this request is undefined.

Table 11-19. Hub Status Field, wHubStatus

Bit	Description
0	**Local Power Source:** This is the source of the local power supply. This field indicates whether hub power (for other than the SIE) is being provided by an external source or from the USB. This field allows the USB System Software to determine the amount of power available from a hub to downstream devices. 0 = Local power supply good 1 = Local power supply lost (inactive)
1	**Over-current:** If the hub supports over-current reporting on a hub basis, this field indicates that the sum of all the ports' current has exceeded the specified maximum and all ports have been placed in the Powered-off state. If the hub reports over-current on a per-port basis or has no over-current detection capabilities, this field is always zero. For more details on over-current protection, see Section 7.2.1.2.1. 0 = No over-current condition currently exists. 1 = A hub over-current condition exists.
2-15	**Reserved** These bits return 0 when read.

There are no defined feature selector values for these status bits and they can neither be set nor cleared by the USB System Software.

Table 11-20. Hub Change Field, wHubChange

Bit	Description
0	**Local Power Status Change:** (c_hub_local_power) This field indicates that a change has occurred in the hub's Local Power Source field in *wHubStatus*. This field is initialized to zero when the hub receives a bus reset. 0 = No change has occurred to Local Power Status. 1 = Local Power Status has changed.
1	**Over-Current Change**: (c_hub_over_current) This field indicates if a change has occurred in the Over-Current field in *wHubStatus*. This field is initialized to zero when the hub receives a bus reset. 0 = No change has occurred to the Over-Current Status. 1 = Over-Current Status has changed.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Bit	Description
2-15	**Reserved** These bits return 0 when read.

Hubs may allow setting of these change bits with SetHubFeature() requests in order to support diagnostics. If the hub does not support setting of these bits, it should either treat the SetHubFeature() request as a Request Error or as a functional no-operation. When set, these bits may be cleared by a ClearHubFeature() request. A request to set a feature that is already set or to clear a feature that is already clear has no effect and the hub will not fail the request.

11.24.2.7 Get Port Status

This request returns the current port status and the current value of the port status change bits.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10100011B	GET_STATUS	Zero	Port	Four	Port Status and Change Status

The port number must be a valid port number for that hub, greater than zero.

The first word of data contains wPortStatus (refer to Table 11-21). The second word of data contains wPortChange (refer to Table 11-20).

The bit locations in the wPortStatus and wPortChange fields correspond in a one-to-one fashion where applicable.

It is a Request Error if *wValue* or *wLength* are other than as specified above or if *wIndex* specifies a port that does not exist.

If the hub is not configured, the behavior of the hub in response to this request is undefined.

11.24.2.7.1 Port Status Bits

Table 11-21. Port Status Field, wPortStatus

Bit	Description
0	**Current Connect Status:** (port_connection) This field reflects whether or not a device is currently connected to this port. 0 = No device is present. 1 = A device is present on this port.
1	**Port Enabled/Disabled:** (port_enable) Ports can be enabled by the USB System Software only. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by the USB System Software. 0 = Port is disabled. 1 = Port is enabled.
2	**Suspend:** (port_suspend) This field indicates whether or not the device on this port is suspended. Setting this field causes the device to suspend by not propagating bus traffic downstream. This field may be reset by a request or by resume signaling from the device attached to the port. 0 = Not suspended. 1 = Suspended or resuming.
3	**Over-current:** (port_over_current) If the hub reports over-current conditions on a per-port basis, this field will indicate that the current drain on the port exceeds the specified maximum. For more details, see Section 7.2.1.2.1. 0 = All no over-current condition exists on this port. 1 = An over-current condition exists on this port.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Bit	Description
4	**Reset**: (port_reset) This field is set when the host wishes to reset the attached device. It remains set until the reset signaling is turned off by the hub. 0 = Reset signaling not asserted. 1 = Reset signaling asserted.
5-7	**Reserved** These bits return 0 when read.
8	**Port Power:** (port_power) This field reflects a port's logical, power control state. Because hubs can implement different methods of port power switching, this field may or may not represent whether power is applied to the port. The device descriptor reports the type of power switching implemented by the hub. 0 = This port is in the Powered-off state. 1 = This port is not in the Powered-off state.
9	**Low- Speed Device Attached:** (port_low_speed) This is relevant only if a device is attached. 0 = Full-speed or High-speed device attached to this port (determined by bit 10). 1 = Low-speed device attached to this port.
10	**High-speed Device Attached:** (PORT_HIGH_SPEED) This is relevant only if a device is attached. 0 = Full-speed device attached to this port. 1 = High-speed device attached to this port.
11	**Port Test Mode** : (PORT_TEST) This field reflects the status of the port's test mode. Software uses the SetPortFeature() and ClearPortFeature() requests to manipulate the port test mode. 0 = This port is not in the Port Test Mode. 1 = This port is in Port Test Mode.
12	**Port Indicator Control**: (PORT_INDICATOR) This field is set to reflect software control of the port indicator. For more details see Sections 11.5.3, 11.24.2.7.1.10, and 11.24.2.13. 0 = Port indicator displays default colors. 1 = Port indicator displays software controlled color.
13-15	**Reserved** These bits return 0 when read.

11.24.2.7.1.1 PORT_CONNECTION

When the Port Power bit is one, this bit indicates whether or not a device is attached. This field reads as one when a device is attached; it reads as zero when no device is attached. This bit is reset to zero when the port is in the Powered-off state or the Disconnected states. It is set to one when the port is in the Powered state, a device attach is detected (see Section 7.1.7.3), and the port transitions from the Disconnected state to the Disabled state.

SetPortFeature(PORT_CONNECTION) and ClearPortFeature(PORT_CONNECTION) requests shall not be used by the USB System Software and must be treated as no-operation requests by hubs.

11.24.2.7.1.2 PORT_ENABLE

This bit is set when the port is allowed to send or receive packet data or resume signaling.

This bit may be set only as a result of a SetPortFeature(PORT_RESET) request. When the hub exits the Resetting state or, if present, the Speed_eval state, this bit is set and bus traffic may be transmitted to the port. This bit may be cleared as the result of any of the following:

- The port being in the Powered-off state
- Receipt of a ClearPortFeature(PORT_ENABLE) request
- Port_Error detection
- Disconnect detection
- When the port enters the Resetting state as a result of receiving the SetPortFeature(PORT_RESET) request

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The hub response to a SetPortFeature(PORT_ENABLE) request is not specified. The preferred behavior is that the hub respond with a Request Error. This may not be used by the USB System Software. The ClearPortFeature(PORT_ENABLE) request is supported as specified in Section 11.5.1.4.

11.24.2.7.1.3 PORT_SUSPEND

This bit is set to one when the port is selectively suspended by the USB System Software. While this bit is set, the hub does not propagate downstream-directed traffic to this port, but the hub will respond to resume signaling from the port. This bit can be set only if the port's PORT_ENABLE bit is set and the hub receives a SetPortFeature(PORT_SUSPEND) request. This bit is cleared to zero on the transition from the SendEOP state to the Enabled state, or on the transition from the Restart_S state to the Transmit state, or on any event that causes the PORT_ENABLE bit to be cleared while the PORT_SUSPEND bit is set.

The SetPortFeature(PORT_SUSPEND) request may be issued by the USB System Software at any time but will have an effect only as specified in Section 11.5.

11.24.2.7.1.4 PORT_OVER-CURRENT

This bit is set to one while an over-current condition exists on the port. This bit is cleared when an over-current condition does not exist on the port.

If the voltage on this port is affected by an over-current condition on another port, this bit is set and remains set until the over-current condition on the affecting port is removed. When the over-current condition on the affecting port is removed, this bit is reset to zero if an over-current condition does not exist on this port.

Over-current protection is required on self-powered hubs (it is optional on bus-powered hubs) as outlined in Section 7.2.1.2.1.

The SetPortFeature(PORT_OVER_CURRENT) and ClearPortFeature(PORT_OVER_CURRENT) requests shall not be used by the USB System Software and may be treated as no-operation requests by hubs.

11.24.2.7.1.5 PORT_RESET

This bit is set while the port is in the Resetting state. A SetPortFeature(PORT_RESET) request will initiate the Resetting state if the conditions in Section 11.5.1.5 are met. This bit is set to zero while the port is in the Powered-off state.

The ClearPortFeature(PORT_RESET) request shall not be used by the USB System Software and may be treated as a no-operation request by hubs.

11.24.2.7.1.6 PORT_POWER

This bit reflects the current power state of a port. This bit is implemented on all ports whether or not actual port power switching devices are present.

While this bit is zero, the port is in the Powered-off state. Similarly, anything that causes this port to go to the Power-off state will cause this bit to be set to zero.

A SetPortFeature(PORT_POWER) will set this bit to one unless both C_HUB_LOCAL_POWER and Local Power Status (in *wHubStatus*) are set to one in which case the request is treated as a functional no-operation.

This bit may be cleared under the following curcumstances:

- Hub receives a ClearPortFeature(PORT_POWER).
- An over-current condition exists on the port.
- An over-current condition on another port causes the power on this port to be shut off.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The SetPortFeature(PORT_POWER) and ClearPortFeature(PORT_POWER) requests may be issued by the USB System Software whenever the port is not in the Not Configured state, but will have an effect only as specified in Section 11.11.

11.24.2.7.1.7 PORT_LOW_SPEED

This bit has meaning only when the PORT_ENABLE bit is set. This bit is set to one if the attached device is low-speed. If this bit is set to zero, the attached device is either full- or high-speed as determined by bit 10 (PORT_HIGH_SPEED, see below).

The SetPortFeature(PORT_LOW_SPEED) and ClearPortFeature(PORT_LOW_SPEED) requests shall not be used by the USB System Software and may be treated as no-operation requests by hubs.

11.24.2.7.1.8 PORT_HIGH_SPEED

This bit has meaning only when the PORT_ENABLE bit is set and the PORT_LOW_SPEED bit is set to zero. This bit is set to one if the attached device is high-speed. The bit is set to zero if the attached device is full-speed.

The SetPortFeature(PORT_HIGH_SPEED) and ClearPortFeature(PORT_HIGH_SPEED) requests shall not be used by the USB System Software and may be treated as no-operation requests by hubs.

11.24.2.7.1.9 PORT_TEST

When the Port Test Mode bit is set to a one (1B), the port is in test mode. The specific test mode is specified in the SetPortFeature(PORT_TEST) request by the test selector. The hub provides no standard mechanism to report the specific test mode; therefore, system software must track which test selector was used. Refer to Section 7.1.20 for details on each test mode. See Section 11.24.2.13 for more information about using SetPortFeature to control test mode.

This field may only be set as a result of a SetPortFeature(PORT_TEST) request. A port PORT_TEST request is only valid to a port that is in the Disabled, Disconnected, or Suspended states. If the port is not in one of these states, the hub must respond with a request error.

This field may be cleared as a result of resetting the hub.

11.24.2.7.1.10 PORT_INDICATOR

When the Port Indicator port status is set to a (1B), the port indicator selector is non-zero. The specific indicator mode is specified in the SetPortFeature(PORT_INDICATOR) request by the indicator selector. The GetPortStatus(PORT_INDICATOR) provides no standard mechanism to report a specific indicator mode; therefore, system software must track which indicator mode was used. Refer to Sections 11.5.3 and 11.24.2.13 for details on each indicator mode.

This field may only be set as a result of a SetPortFeature(PORT_INDICATOR) request.

This field may be cleared as a result of a SetPortFeature(PORT_INDICATOR) request with Indicator Selector = Default or a ClearPortFeature(PORT_INDICATOR) request.

This feature must be set when host software detects an error on a port that requires user intervention. This feature must be utilized by system software if it determines that any of the following conditions are true:

- A high power device is plugged into a low power port.
- A defective device is plugged into a port (Babble conditions, excessive errors, etc.).
- An overcurrent condition occurs which causes software or hardware to set the indicator.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The PORT_OVER_CURRENT status bit will set the default port indicator color to amber. Setting the PORT_POWER feature, sets the indicator to off.

This feature is also used when host software determines that a port requires user attention. Many error conditions can be resolved if the user moves a device from one port to another that has the proper capabilities.

A typical scenario is when a user plugs a high power device in to a bus-powered hub. If there is an available high power port, then the user can be directed to move the device from the low-power port to the high power port.

1. Host software would cycle the PORT_INDICATOR feature of the low-power port to blink the indicator and display a message to the user to unplug the device from the port with the blinking indicator.
2. Using the C_PORT_CONNECTION status change feature, host software can determine when the user physically removed the device from the low-power port.
3. Host software would next issue a ClearPortFeature(PORT_INDICATOR) to the low-power port (restoring the default color), begin cycling the PORT_INDICATOR of the high-power port, and display a message telling the user to plug the device into the port with the blinking indicator.
4. Using the C_PORT_CONNECTION status change feature host software can determine when the user physically inserted the device onto the high power port.

Host software must cycle the PORT_INDICATOR feature to blink the current color at approximately 0.5 Hz rate with a 30-50% duty cycle.

11.24.2.7.2 Port Status Change Bits

Port status change bits are used to indicate changes in port status bits that are not the direct result of requests. Port status change bits can be cleared with a ClearPortFeature() request or by a hub reset. Hubs may allow setting of the status change bits with a SetPortFeature() request for diagnostic purposes. If a hub does not support setting of the status change bits, it may either treat the request as a Request Error or as a functional no-operation. Table 11-22 describes the various bits in the *wPortChange* field.

Table 11-22. Port Change Field, wPortChange

Bit	Description
0	**Connect Status Change:** (c_port_connection) Indicates a change has occurred in the port's Current Connect Status. The hub device sets this field as described in Section 11.24.2.7.2.1. 0 = No change has occurred to Current Connect status. 1 = Current Connect status has changed.
1	**Port Enable/Disable Change:** (c_port_enable) This field is set to one when a port is disabled because of a Port_Error condition (see Section 11.8.1).
2	**Suspend Change:** (c_port_suspend) This field indicates a change in the host-visible suspend state of the attached device. It indicates the device has transitioned out of the Suspend state. This field is set only when the entire resume process has completed. That is, the hub has ceased signaling resume on this port. 0 = No change. 1 = Resume complete.
3	**Over-Current Indicator Change:** (c_port_over_current) This field applies only to hubs that report over-current conditions on a per-port basis (as reported in the hub descriptor). 0 = No change has occurred to Over-Current Indicator. 1 = Over-Current Indicator has changed. If the hub does not report over-current on a per-port basis, then this field is always zero.
4	**Reset Change:** (c_port_reset) This field is set when reset processing on this port is complete. 0 = No change. 1 = Reset complete.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Bit	Description
5-15	**Reserved** These bits return 0 when read.

11.24.2.7.2.1 C_PORT_CONNECTION

This bit is set when the PORT_CONNECTION bit changes because of an attach or detach detect event (see Section 7.1.7.3). This bit will be cleared to zero by a ClearPortFeature(C_PORT_CONNECTION) request or while the port is in the Powered-off state.

11.24.2.7.2.2 C_PORT_ENABLE

This bit is set when the PORT_ENABLE bit changes from one to zero as a result of a Port Error condition (see Section 11.8.1). This bit is not set on any other changes to PORT_ENABLE.

This bit may be set if, on a SetPortFeature(PORT_RESET), the port stays in the Disabled state because an invalid idle state exists on the bus (see Section 11.8.2).

This bit will be cleared by a ClearPortFeature(C_PORT_ENABLE) request or while the port is in the Powered-off state.

11.24.2.7.2.3 C_PORT_SUSPEND

This bit is set on the following transitions:

- On transition from the Resuming state to the SendEOP state
- On transition from the Restart_S state to the Transmit state

This bit will be cleared by a ClearPortFeature(C_PORT_SUSPEND) request, or while the port is in the Powered-off state.

11.24.2.7.2.4 C_PORT_OVER-CURRENT

This bit is set when the PORT_OVER_CURRENT bit changes from zero to one or from one to zero. This bit is also set if the port is placed in the Powered-off state due to an over-current condition on another port.

This bit will be cleared when the port is in the Not Configured state or by a ClearPortFeature(C_PORT_OVER_CURRENT) request.

11.24.2.7.2.5 C_PORT_RESET

This bit is set when the port transitions from the Resetting state (or, if present, the Speed_eval state) to the Enabled state.

This bit will be cleared by a ClearPortFeature(C_PORT_RESET) request, or while the port is in the Powered-off state.

11.24.2.8 Get_TT_State

This request returns the internal state of the transaction translator in a vendor specific format. A TT receiving this request must have first been stopped via the Stop_TT request. This request is provided for debugging purposes.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10100011B	Get_TT_State	TT_Flags	TT_Port	TT State Length	TT State

The TT_Flags bits 7..0 are reserved for future USB definition and must be set to zero. The TT_Flags bits 15..8 are for vendor specific usage.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

The TT state returned in the data stage of the control transfer for this request is shown in Table 11-23.

Table 11-23. Format of Returned TT State

Offset	Field	Size (bytes)	Comments
0	TT_Return_Flags	4	Bits 15..0 are reserved for future USB definition and must be set to zero. Bits 31..16 are for vendor specific usage.
4	TT_specific_state	Implementation dependent	

If the hub supports multiple TTs, then *wIndex* must specify the port number of the TT that will return TT_state. If the hub provides only a single TT, then Port must be set to one.

The state of the TT after processing this request is undefined.

It is a Request Error, if *wIndex* specifies a port that does not exist. If *wLength* is larger than the actual length of this request, then only the actual length is returned. If *wLength* is less than the actual length of this request, then only the first *wLength* bytes of this request are returned; this is not considered an error even if *wLength* is zero.

If the hub is not configured, the hub's response to this request is undefined.

11.24.2.9 Reset_TT

This request returns the transaction translator in a hub to a known state.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00100011B	RESET_TT	Zero	TT_Port	Zero	None

Under some circumstances, a Transaction Translator (TT) in a hub may be in an unknown state such that it is no longer functioning correctly. The Reset_TT request allows the TT to be returned to the state it is in immediately after the hub is configured. Reset_TT only resets the TT internal data structures (buffers) and pipeline and its related state machines. After the reset is completed, the TT can resume its normal operation. Reset of the TT is de-coupled from the other parts of the hub (including downstream facing ports of the hub, the hub repeater, the hub controller, etc). Other parts of the hub are not reset and can continue their normal operation. The downstream facing ports are not reset, so that when the TT resumes its normal operation, the corresponding attached devices continue to work; i.e., a new enumeration process is not required. The working of downstream FS/LS devices are disrupted only during the reset time of the TT to which they belong.

If the hub supports multiple TTs, then *wIndex* must specify the port number of the TT that is to be reset. If the hub provides only a single TT, then Port must be set to one. For a single TT Hub, the Hub can ignore the Port number.

It is a Request Error, if *wIndex* specifies a port that does not exist, or if *wLength* is not as specified above.

If the hub is not configured, the hub's response to this request is undefined.

11.24.2.10 Set Hub Descriptor

This request overwrites the hub descriptor.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00100000B	SET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero	Descriptor Length	Descriptor

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The SetDescriptor request for the hub class descriptor follows the same usage model as that of the standard SetDescriptor request (refer to Chapter 9). The standard hub descriptor is denoted by using the value *bDescriptorType* defined in Section 11.23.2.1. All hubs are required to implement one hub descriptor with descriptor index zero.

This request is optional. This request writes data to a class-specific descriptor. The host provides the data that is to be transferred to the hub during the data transfer phase of the control transaction. This request writes the entire hub descriptor at once.

Hubs must buffer all the bytes received from this request to ensure that the entire descriptor has been successfully transmitted from the host. Upon successful completion of the bus transfer, the hub updates the contents of the specified descriptor.

It is a Request Error if wIndex is not zero or if wLength does not match the amount of data sent by the host. Hubs that do not support this request respond with a STALL during the Data stage of the request.

If the hub is not configured, the hub's response to this request is undefined.

11.24.2.11 Stop_TT

This request stops the normal execution of the transaction translator so that the internal TT state can be retrieved via Get_TT_State. This request is provided for debugging purposes.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00100011B	STOP_TT	Zero	TT_Port	Zero	None

The only standardized method to restart a TT after a Stop_TT request is via the Reset_TT request.

If the hub supports multiple TTs, then *wIndex* must specify the port number of the TT that is being stopped. If the hub provides only a single TT, then Port must be set to one. For a single TT Hub, the Hub can ignore the Port number.

It is a Request Error, if *wIndex* specifies a port that does not exist, or if *wLength* is not as specified above.

If the hub is not configured, the hub's response to this request is undefined.

11.24.2.12 Set Hub Feature

This request sets a value reported in the hub status.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00100000B	SET_ FEATURE	Feature Selector	Zero	Zero	None

Setting a feature enables that feature; refer to Table 11-17 for the feature selector definitions that apply to the hub as recipient. Status changes may not be acknowledged using this request.

It is a Request Error if *wValue* is not a feature selector listed in Table 11-17 or if *wIndex* or *wLength* are not as specified above.

If the hub is not configured, the hub's response to this request is undefined.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.24.2.13 Set Port Feature

This request sets a value reported in the port status.

bmRequestType	bRequest	wValue	wIndex		wLength	Data
00100011B	SET_ FEATURE	Feature Selector	Selector	Port	Zero	None

The port number must be a valid port number for that hub, greater than zero. The port number is in the least significant byte (bits 7..0) of the *wIndex* field. The most significant byte of *wIndex* is zero, except when the feature selector is PORT_TEST.

Setting a feature enables that feature or starts a process associated with that feature; see Table 11-17 for the feature selector definitions that apply to a port as a recipient. Status change may not be acknowledged using this request. Features that can be set with this request are:

- PORT_RESET
- PORT_SUSPEND
- PORT_POWER
- PORT_TEST
- PORT_INDICATOR
- C_PORT_CONNECTION*
- C_PORT_RESET*
- C_PORT_ENABLE*
- C_PORT_SUSPEND*
- C_PORT_OVER_CURRENT*

* Denotes features that are not required to be set by this request.

Setting the port_suspend feature causes bus traffic to cease on that port and, consequently, the device to suspend. Setting the reset feature PORT_RESET causes the hub to signal reset on that port. When the reset signaling is complete, the hub sets the C_PORT_RESET status change and immediately enables the port. Also see Section 11.24.2.7.1 for further details.

When the feature selector is PORT_TEST, the most significant byte (bits 15..8) of the *wIndex* field is the selector identifying the specific test mode. Table 11-24 lists the test selector definitions. Refer to Section 7.1.20 for definitions of each test mode. Test mode of a downstream facing port can only be used in a well defined sequence of hub states. This sequence is defined as follows:

1) All enabled downstream facing ports of the hub containing the port to be tested must be (selectively) suspended via the SetPortFeature(PORT_SUSPEND) request. Each downstream facing port of the hub must be in the disabled, disconnected, or suspended state (see Figure 11-9).
2) A SetPortFeature(PORT_TEST) request must be issued to the downstream facing port to be tested. Only a single downstream facing port can be in test_mode at a time. The transition to test mode must be complete no later than 3 ms after the completion of the status stage of the request.
3) The downstream facing port under test can now be tested.
4) During test_mode, a port disconnect or resume status change on one of the suspended ports (not including the port under test) must cause a status change (C_PORT_CONNECTION or C_PORT_SUSPEND) report (See Section 11.12.3 and

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.24.2.7.2) from the hub. Note: Other status changes may or may not be supported in a hub with a downstream facing port in test mode. The reporting of these status changes can allow a test application to restore normal operation of a root hub without requiring a non-USB keyboard or mouse for user input. For example. a USB device attached to the root hub can be disconnected to notify the test application to restore normal root hub operation.

5) During test_mode, the state of the hub downstream facing ports must not be changed by the host (i.e., hub class requests other than the Get_Port_Status() request must not be issued by the host). Note: The hub must also be reset before a SetPortFeature(PORT_TEST) can be used to place the port into another test mode.

6) After the test is completed, the hub (with the port under test) must be reset by the host or user. This must be accomplished by manipulating the port of the parent hub to which the hub under test is attached. This manipulation can consist of one of the following:

   a) Issuing a SetPortFeature(PORT_RESET) to port of the parent hub to which the hub under test is attached.

   b) Issuing a ClearPortFeature(PORT_POWER) and SetPortFeature(PORT_POWER) to cycle power of a parent hub that supports per port power control.

   c) Disconnecting and re-connecting the hub under test from its parent hub port.

   d) For a root hub under test, a reset of the Host Controller may be required as there is no parent hub of the root hub.

7) Behavior of the hub under test and its downstream facing ports is undefined if these requirements are not met.

Table 11-24. Test Mode Selector Codes

Value	Test Mode Description
0H	Reserved
1H	Test_J
2H	Test_K
3H	Test_SE0_NAK
4H	Test_Packet
5H	Test_Force_Enable
06H-3FH	Reserved for Standard Test selections
40H-BFH	Reserved
C0H-FFH	Reserved for Vendor-Unique test selections

When the feature selector is PORT_INDICATOR, the most significant byte of the *wIndex* field is the selector identifying the specific indicator mode. Table 11-25 lists the indicator selector definitions. Refer to Sections 11.5.3 and 11.24.2.7.1.10 for indicator details. The hub will respond with a request error if the request contains an invalid indicator selector.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 11-25. Port Indicator Selector Codes

Value	Port Indicator Color	Port Indicator Mode
0	Color set automatically, as defined in Table 11-6	Automatic
1	Amber	Manual
2	Green	
3	Off	
4-FFH	Reserved	Reserved

The hub must meet the following requirements:

- If the port is in the Powered-off state, the hub must treat a SetPortFeature(PORT_RESET) request as a functional no-operation.
- If the port is not in the Enabled or Transmitting state, the hub must treat a SetPortFeature(PORT_SUSPEND) request as a functional no-operation.
- If the port is not in the Powered-off state, the hub must treat a SetPortFeature(PORT_POWER) request as a functional no-operation.

It is a Request Error if *wValue* is not a feature selector listed in Table 11-17, if *wIndex* specifies a port that does not exist, or if *wLength* is not as specified above.

If the hub is not configured, the hub's response to this request is undefined.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Appendix A Transaction Examples

This appendix contains transaction examples for different split transaction cases. The cases are for bulk/control OUT and SETUP, bulk/control IN, interrupt OUT, interrupt IN, isochronous OUT, and isochronous IN.

A.1 Bulk/Control OUT and SETUP Transaction Examples

Legend:

(S): Start Split

(C): Complete Split

Summary of cases for bulk/control OUT and SETUP transaction

- Normal cases

Case	Reference Figure	Similar Figure
No smash	Figure A-1	
HS SSPLIT smash		Figure A-2
HS SSPLIT 3 strikes smash		Figure A-3
HS OUT/SETUP(S) smash		Figure A-2
HS OUT/SETUP(S) 3 strikes smash		Figure A-3
HS DATA0/1 smash	Figure A-2	
HS DATA0/1 3 strikes smash	Figure A-3	
HS ACK(S) smash	Figure A-4 Figure A-5	
HS ACK(S) 3 strikes smash	Figure A-6	
HS CSPLIT smash	Figure A-7	
HS CSPLIT 3 strikes smash	Figure A-8	
HS OUT/SETUP(C) smash		Figure A-7
HS OUT/SETUP(C) 3 strikes smash		Figure A-8
HS ACK(C) smash	Figure A-9	
HS ACK(C) 3 strikes smash	Figure A-10	
FS/LS OUT/SETUP smash		Figure A-11
FS/LS OUT/SETUP 3 strikes smash		Figure A-12
FS/LS DATA0/1 smash	Figure A-11	
FS/LS DATA0/1 3 strikes smash	Figure A-12	
FS/LS ACK smash	Figure A-13	
FS/LS ACK 3 strikes smash	Figure A-14	

- No buffer(on hub) available cases

Case	Reference Figure	Similar Figure
No smash(HS NAK(S))	Figure A-15	
HS NAK(S) smash	Figure A-16	
HS NAK(S) 3 strikes smash	Figure A-17	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- CS(Complete-split transaction) earlier cases

Case	Reference Figure	Similar Figure
No smash(HS NYET)	Figure A-18	
HS NYET smash	Figure A-19 Figure A-20	
HS NYET 3 strikes smash	Figure A-21	

- Device busy cases

Case	Reference Figure	Similar Figure
No smash(HS NAK(C))	Figure A-22	
HS NAK(C) smash		Figure A-9
HS NAK(C) 3 strikes smash		Figure A-10
FS/LS NAK smash		Figure A-13
FS/LS NAK 3 strikes smash		Figure A-14

- Device stall cases

Case	Reference Figure	Similar Figure
No smash	Figure A-23	
HS STALL(C) smash		Figure A-9
HS STALL(C) 3 strikes smash		Figure A-10
FS/LS STALL smash		Figure A-13
FS/LS STALL 3 strikes smash		Figure A-14

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

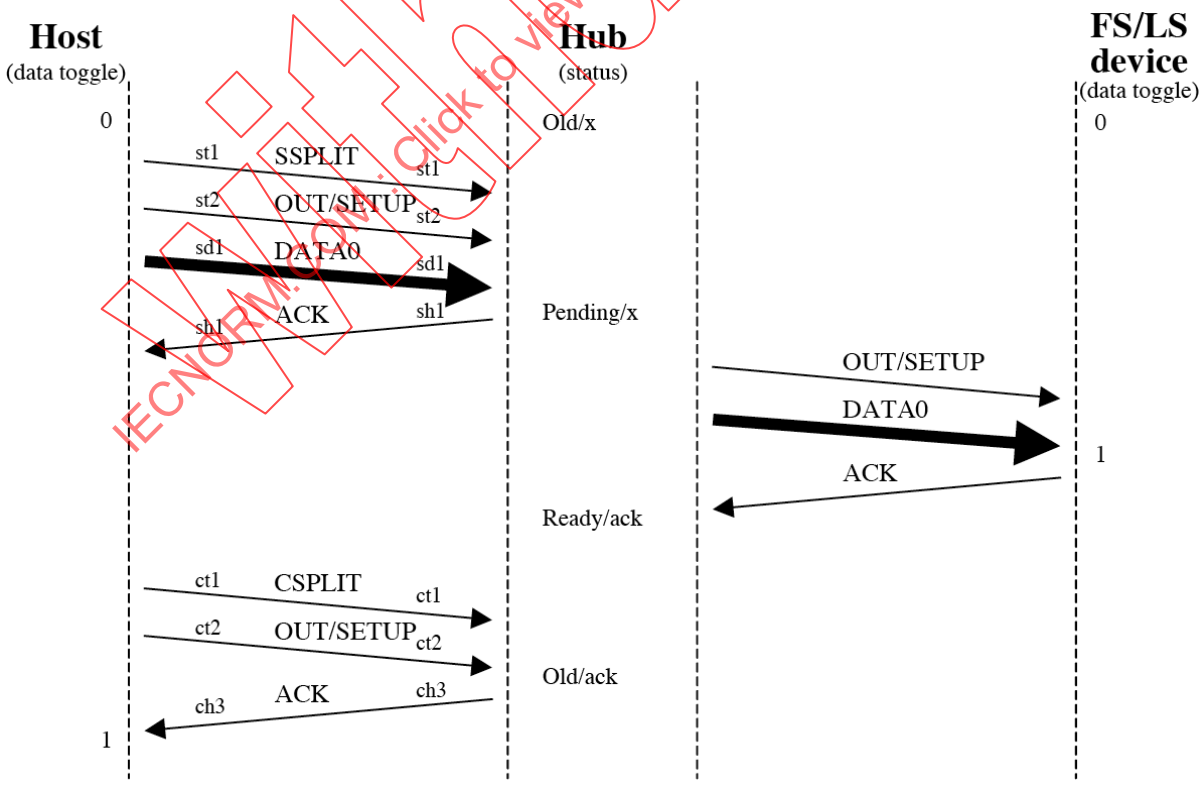


Figure A-1. Normal No Smash

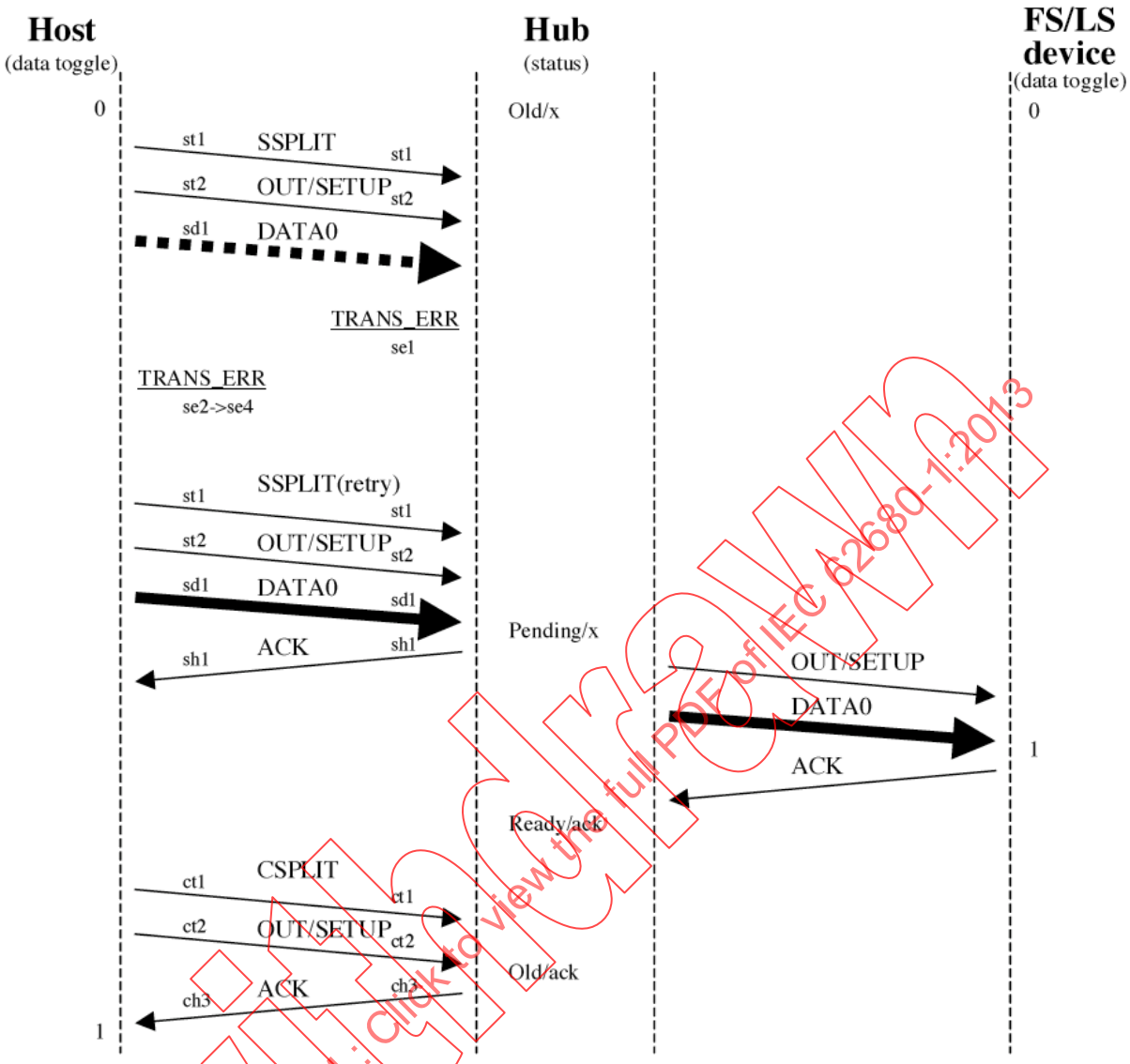


Figure A-2. Normal HS DATA0/1 Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

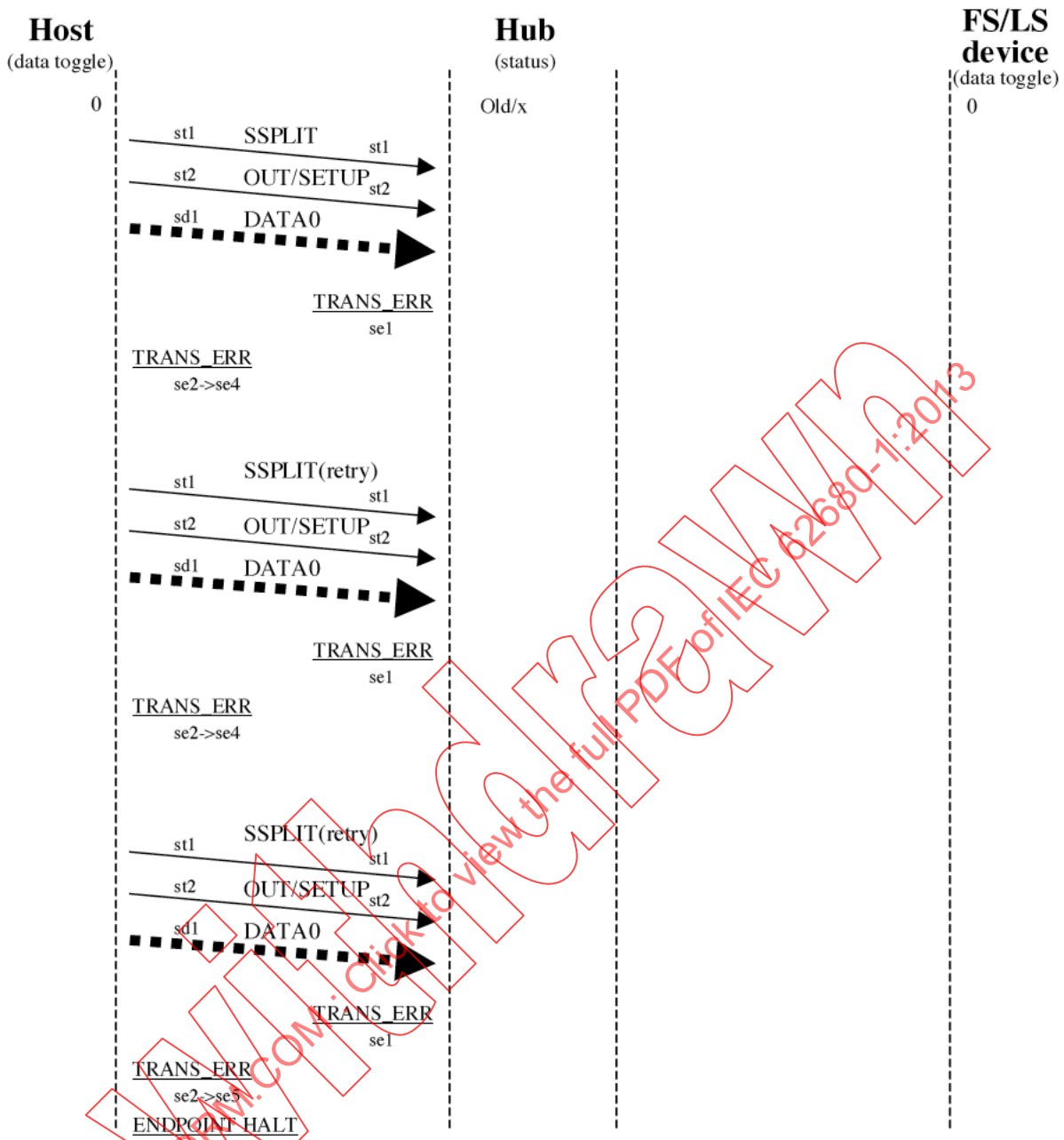


Figure A-3. Normal HS DATA0/1 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

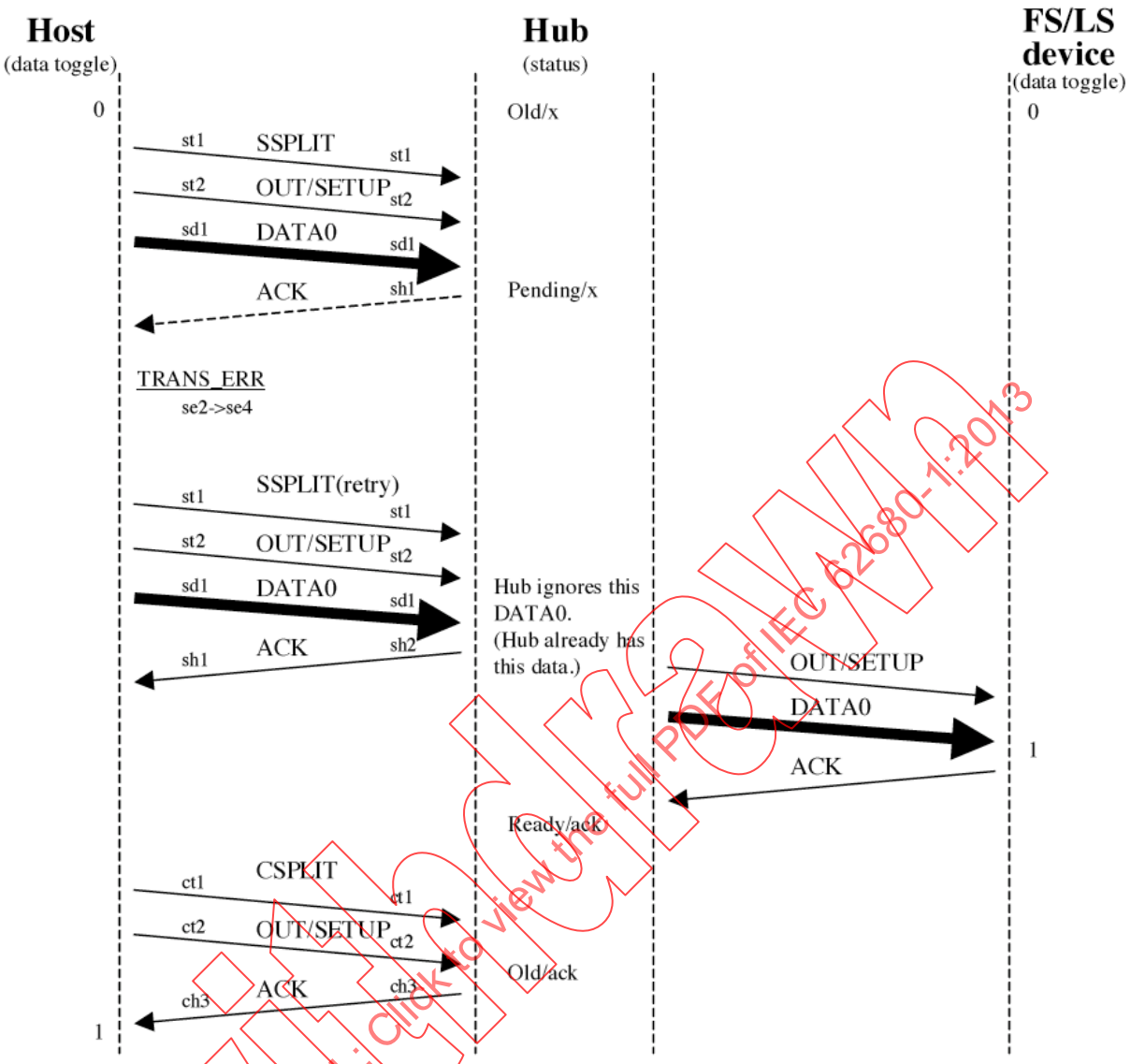


Figure A-4. Normal HS ACK(S) Smash(case 1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

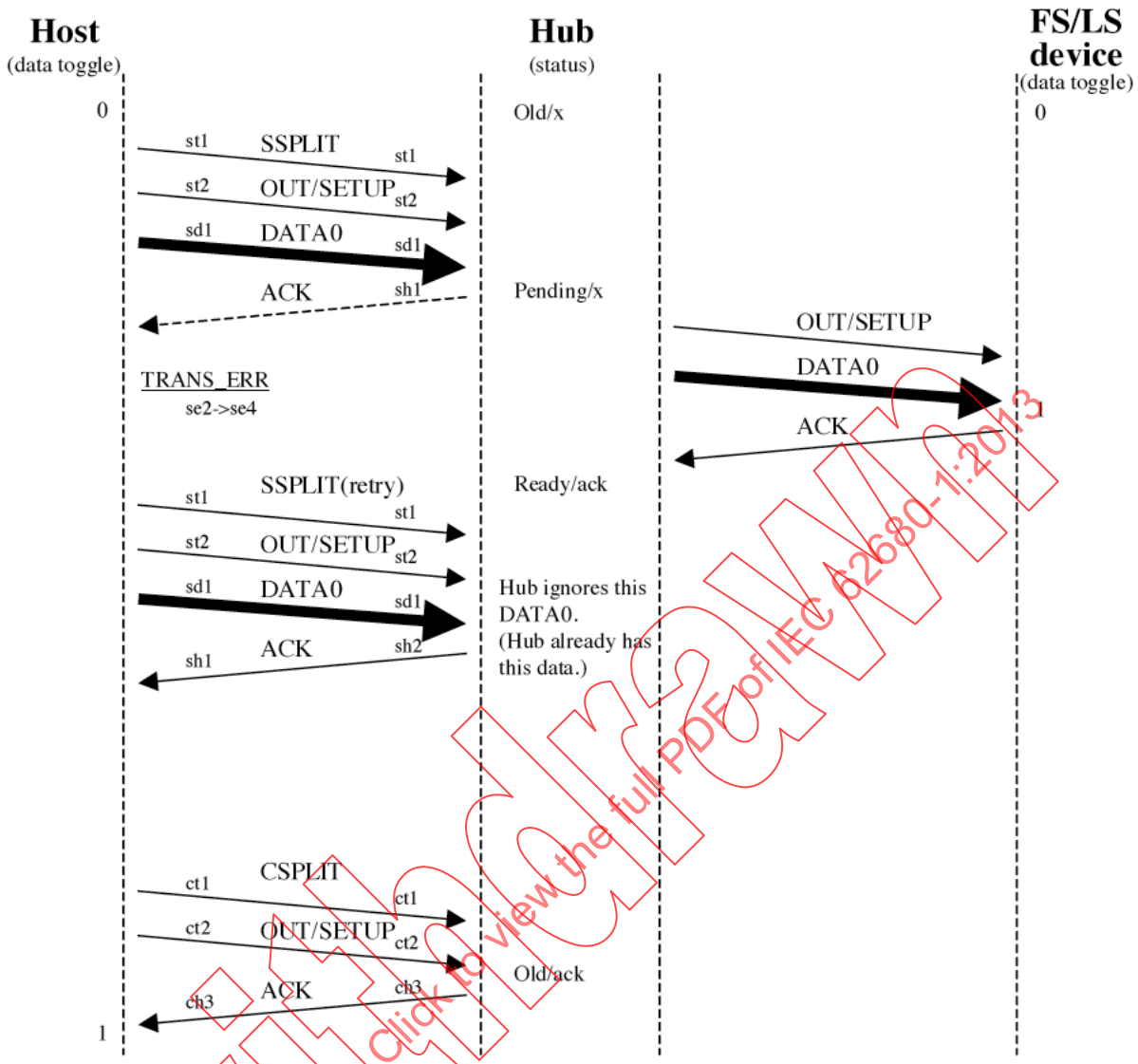


Figure A-5. Normal HS ACK(S) Smash(case 2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

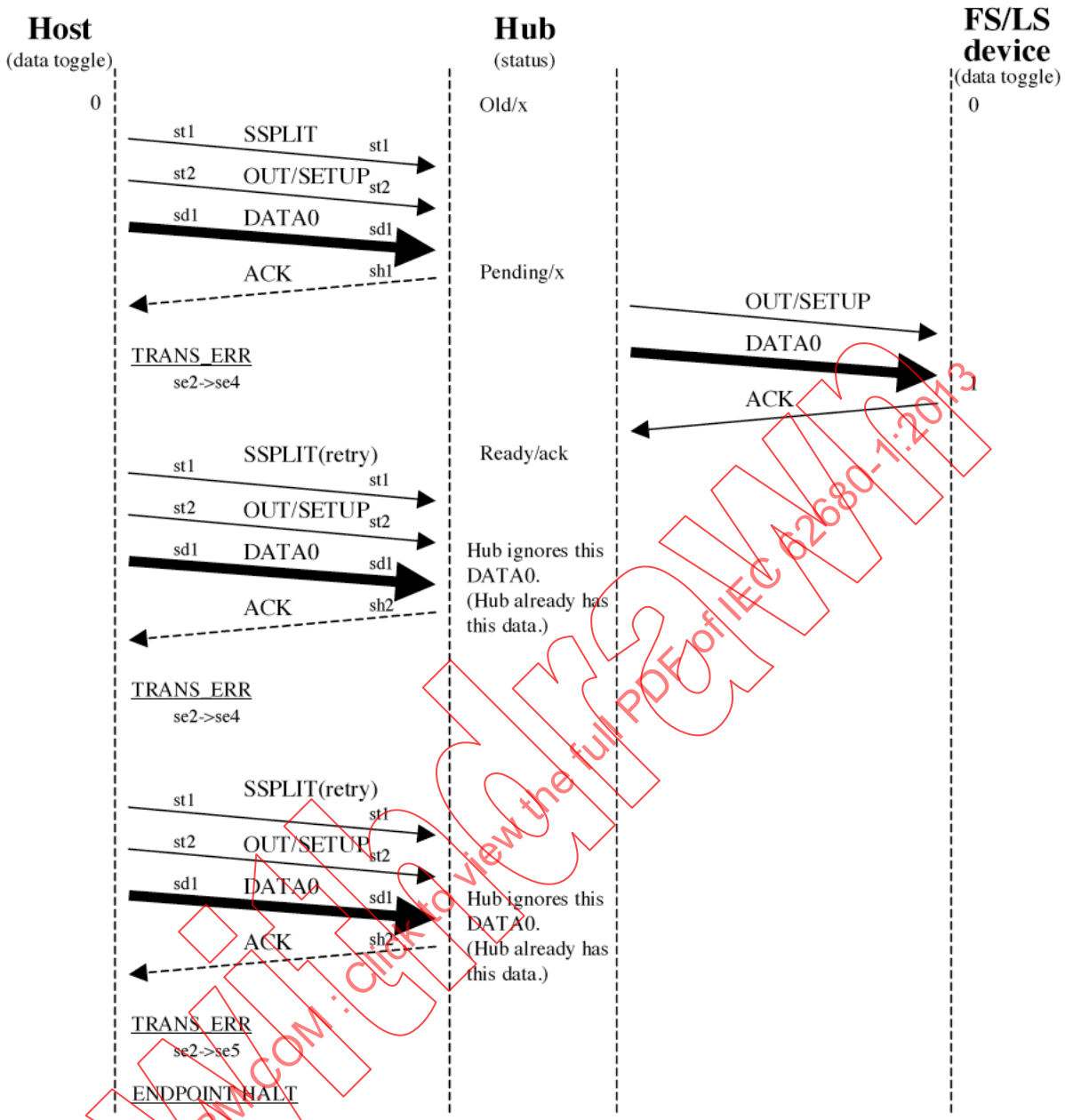


Figure A-6. Normal HS ACK(S) 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

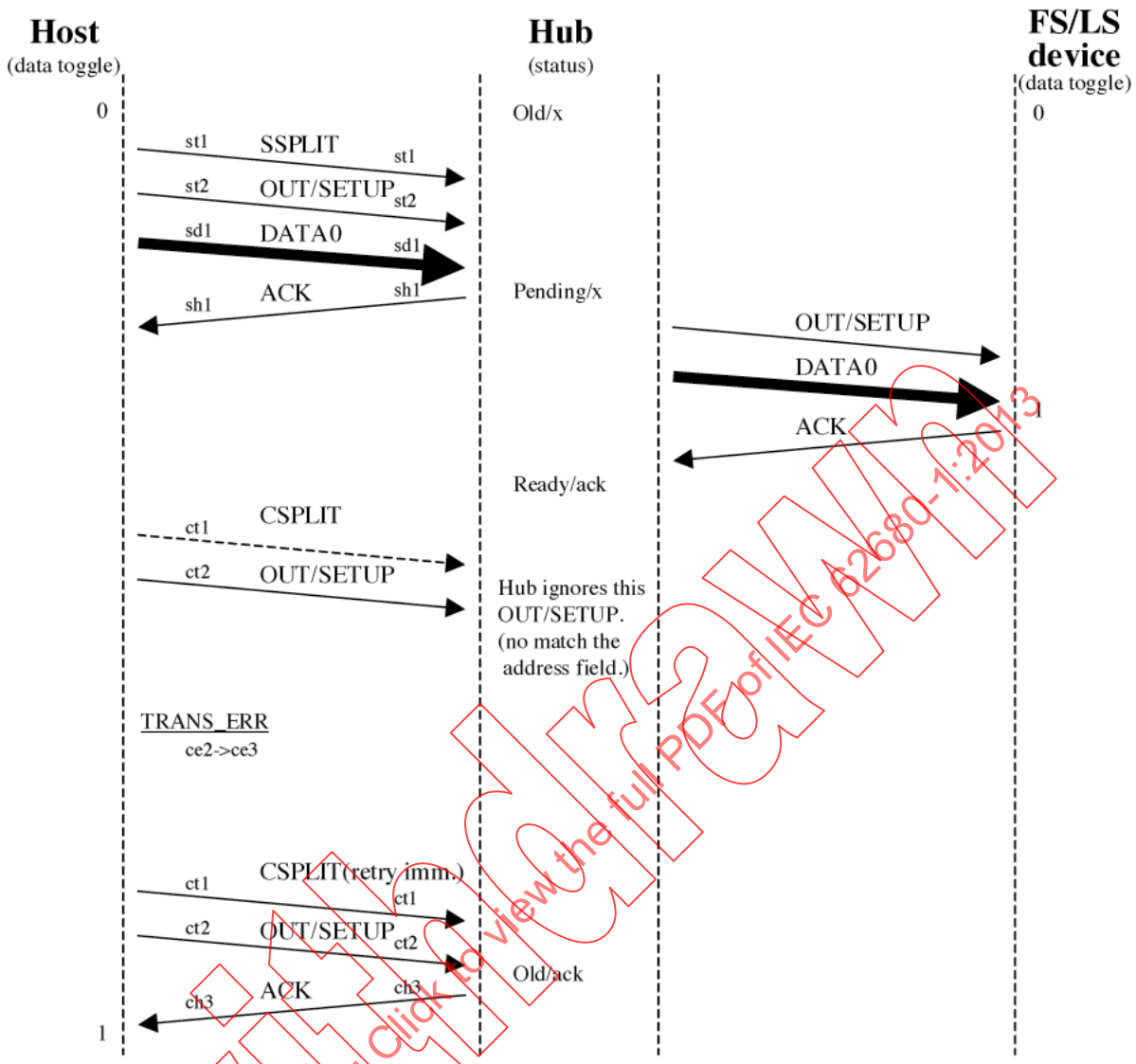


Figure A-7. Normal HS CSPLIT Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

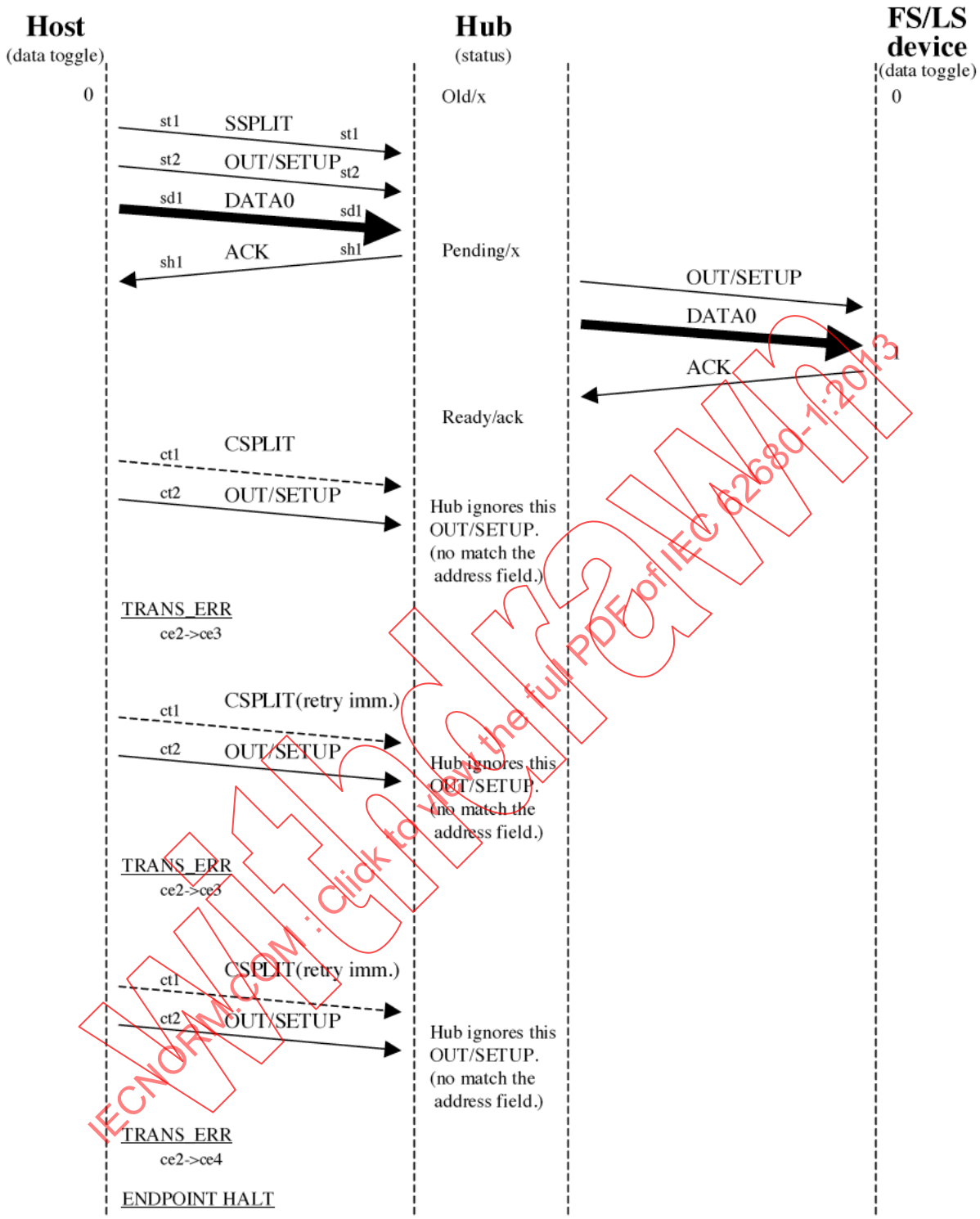


Figure A-8. Normal HS CSPLIT 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

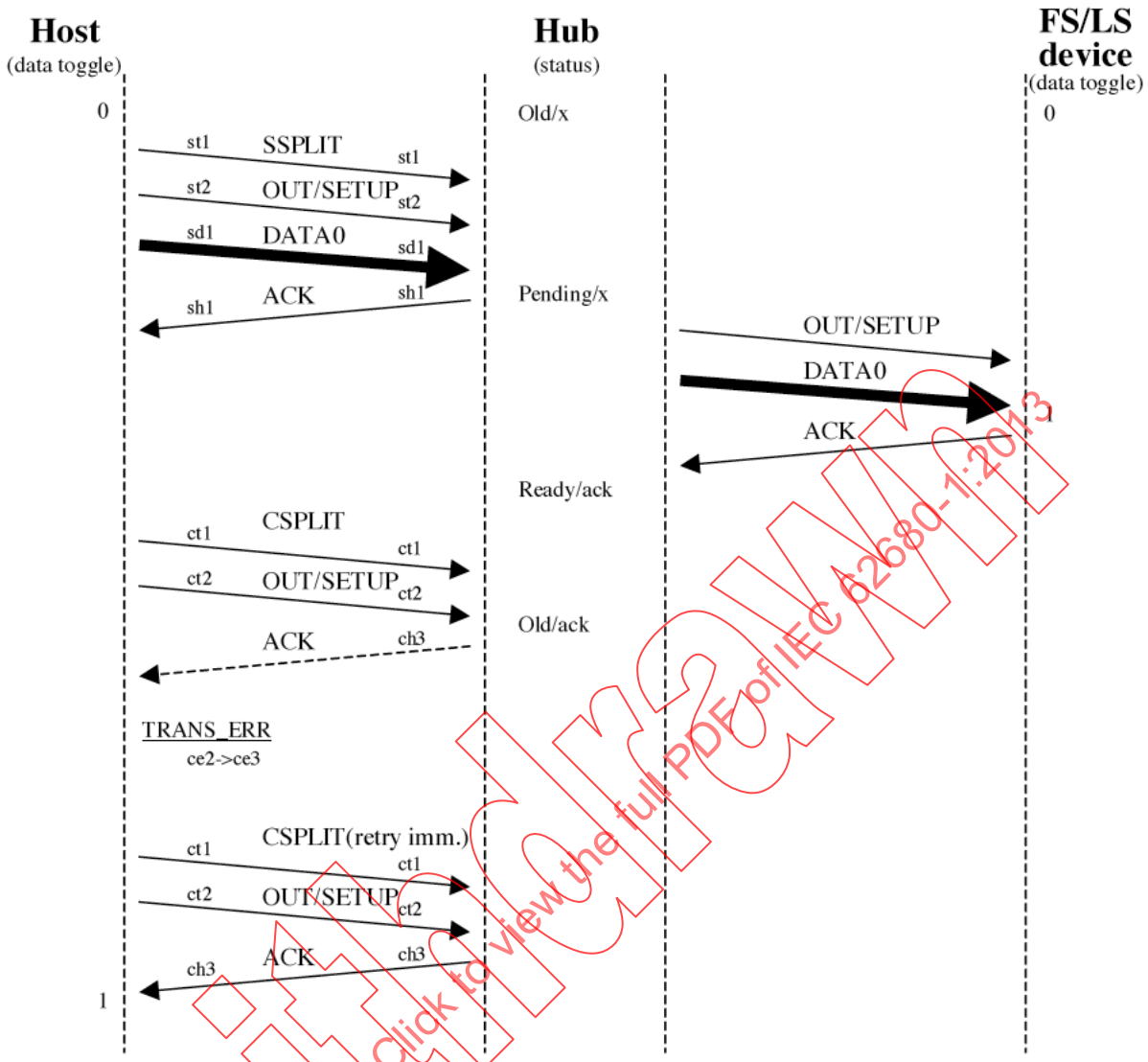


Figure A-9. Normal HS ACK(C) Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

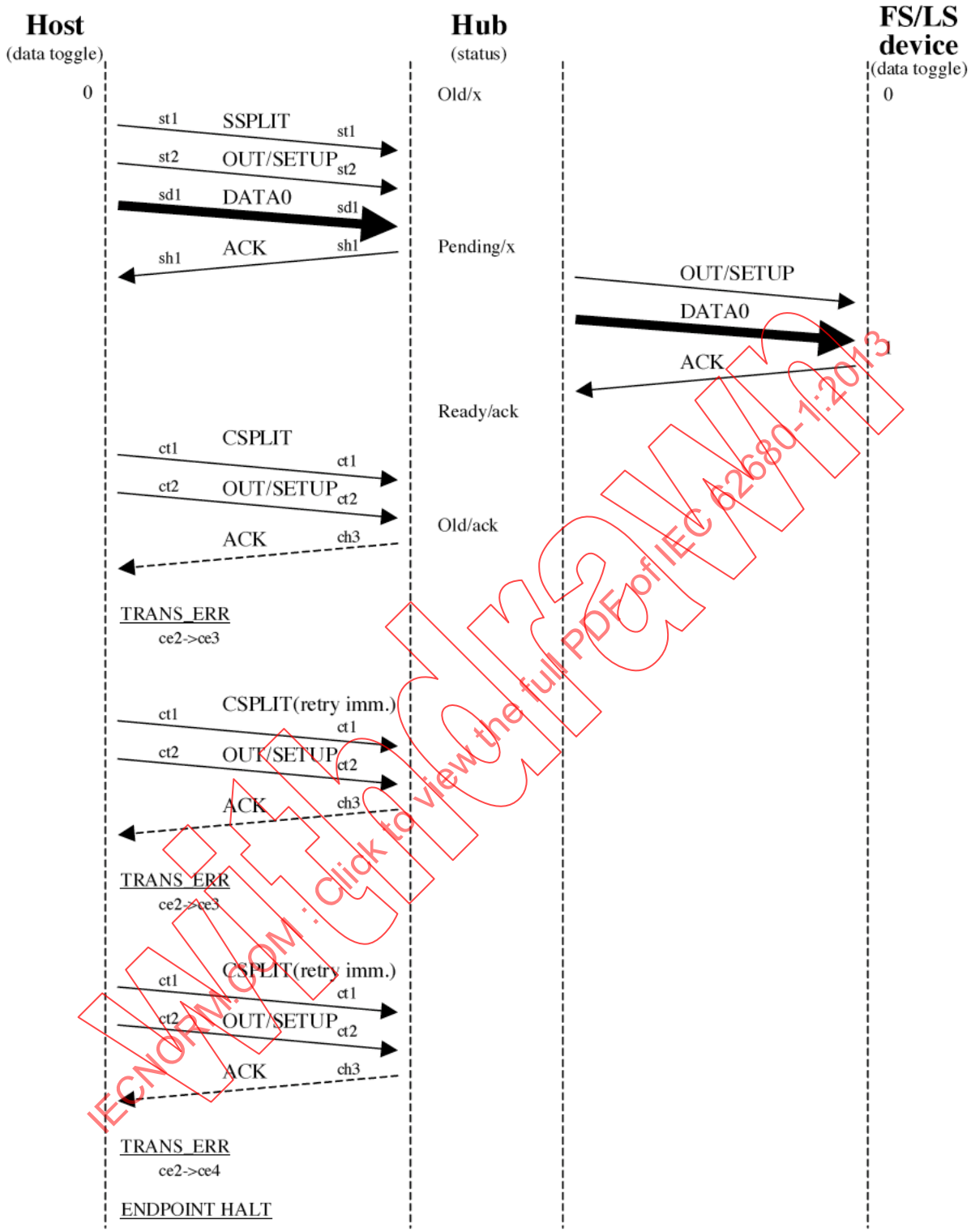


Figure A-10. Normal S ACK(C) 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

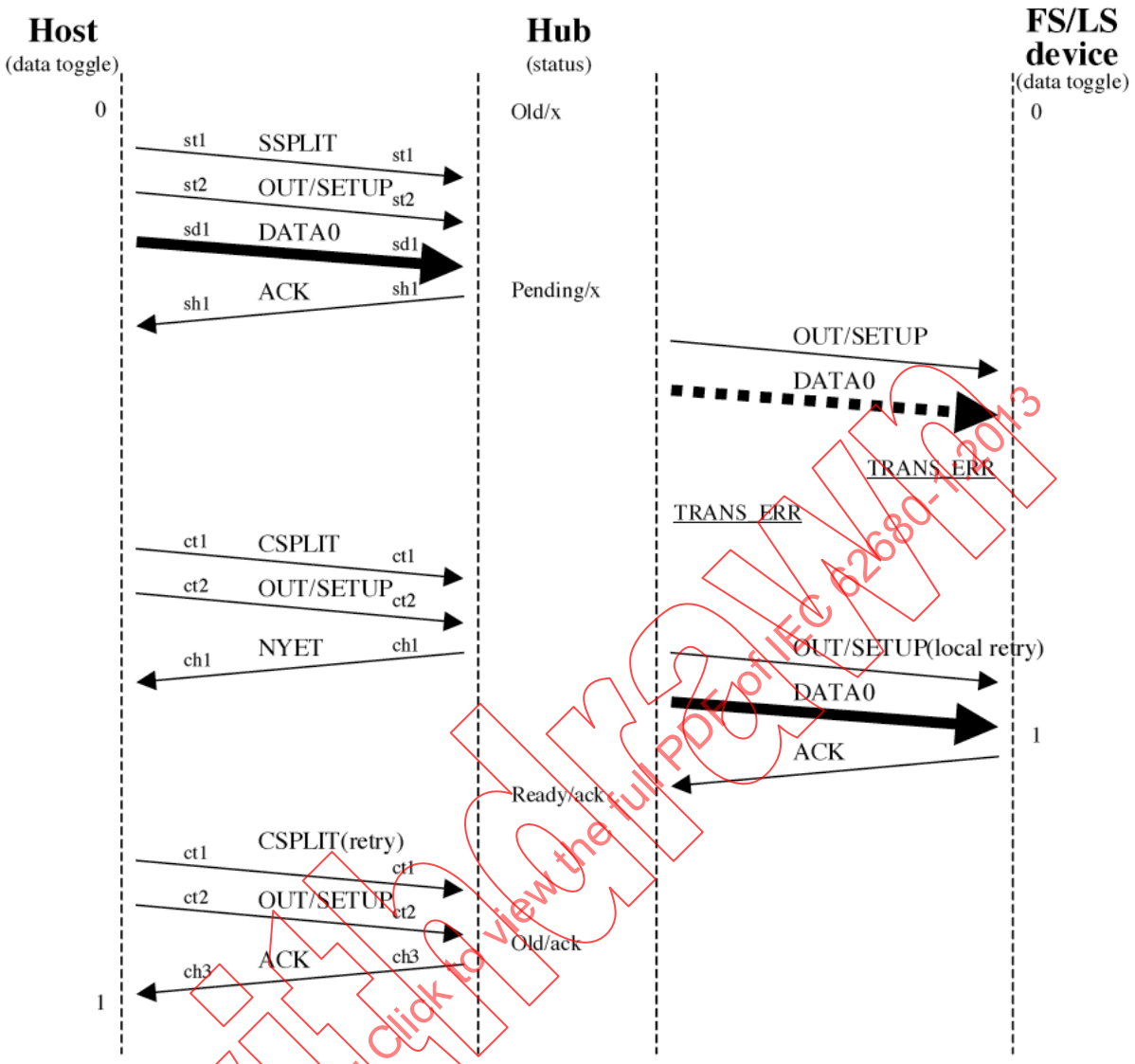


Figure A-11. Normal FS/LS DATA0/1 Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

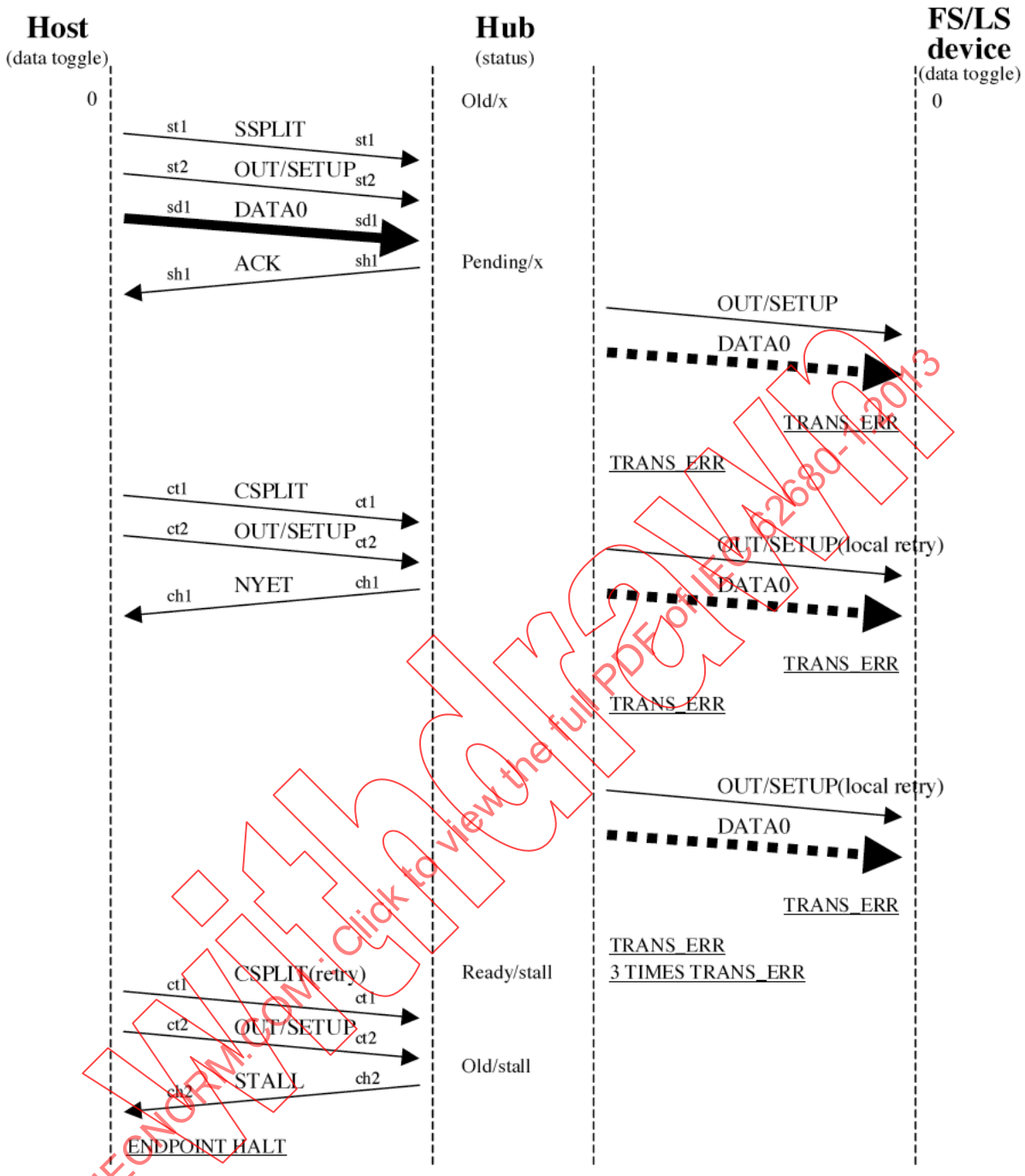


Figure A-12. Normal FS/LS DATA0/1 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

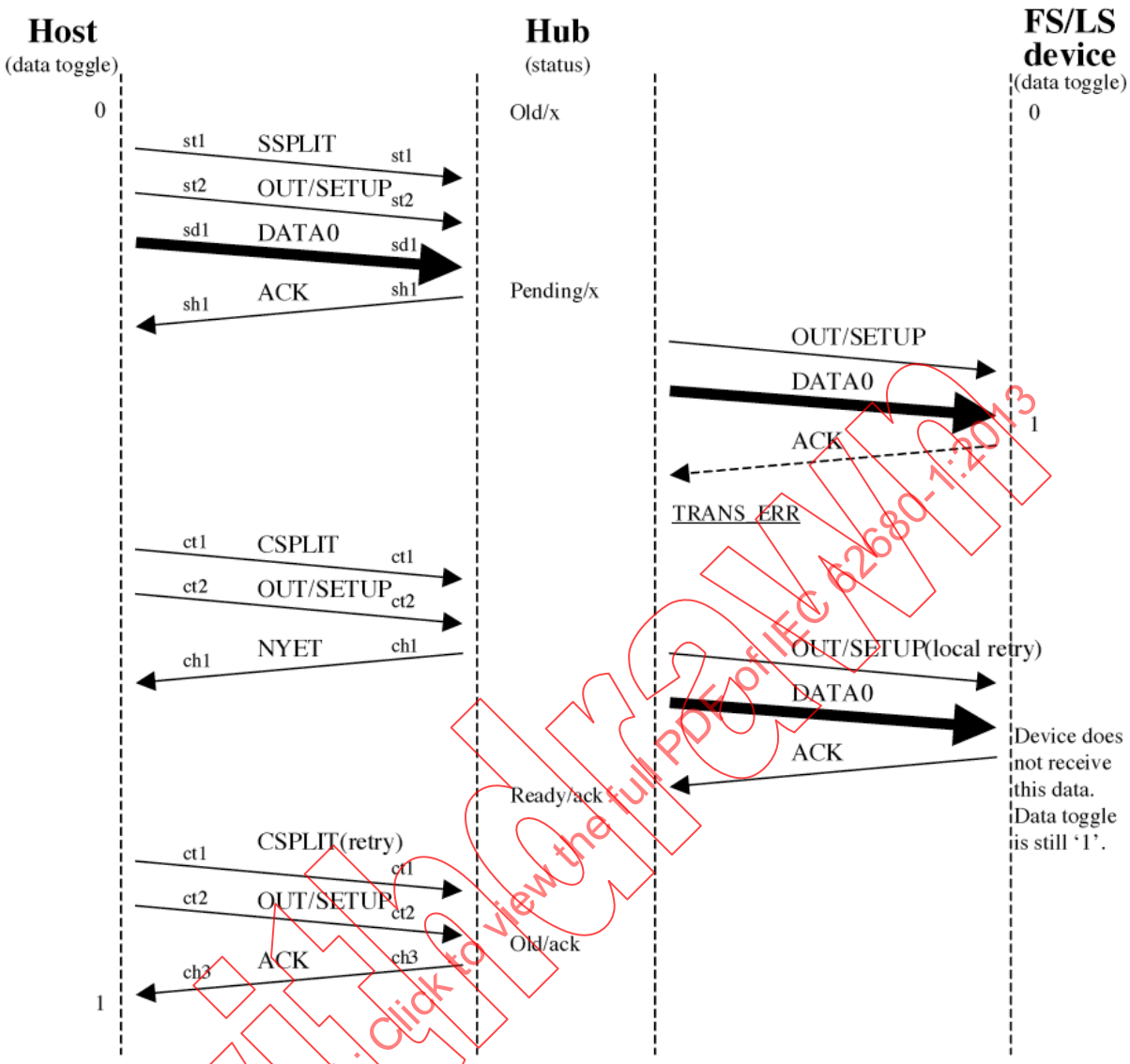


Figure A-13. Normal FS/LS ACK Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

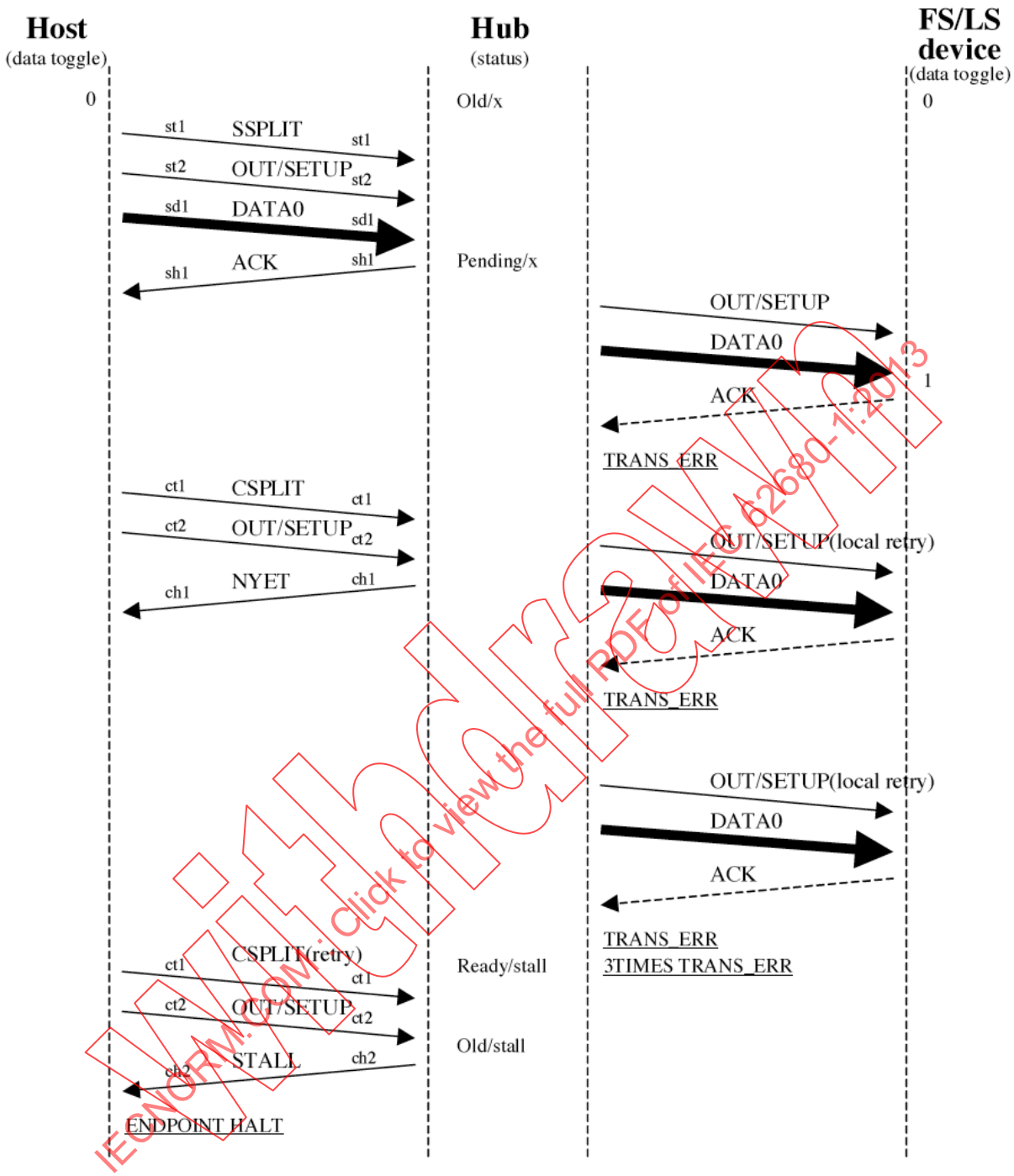


Figure A-14. Normal FS/LS ACK 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

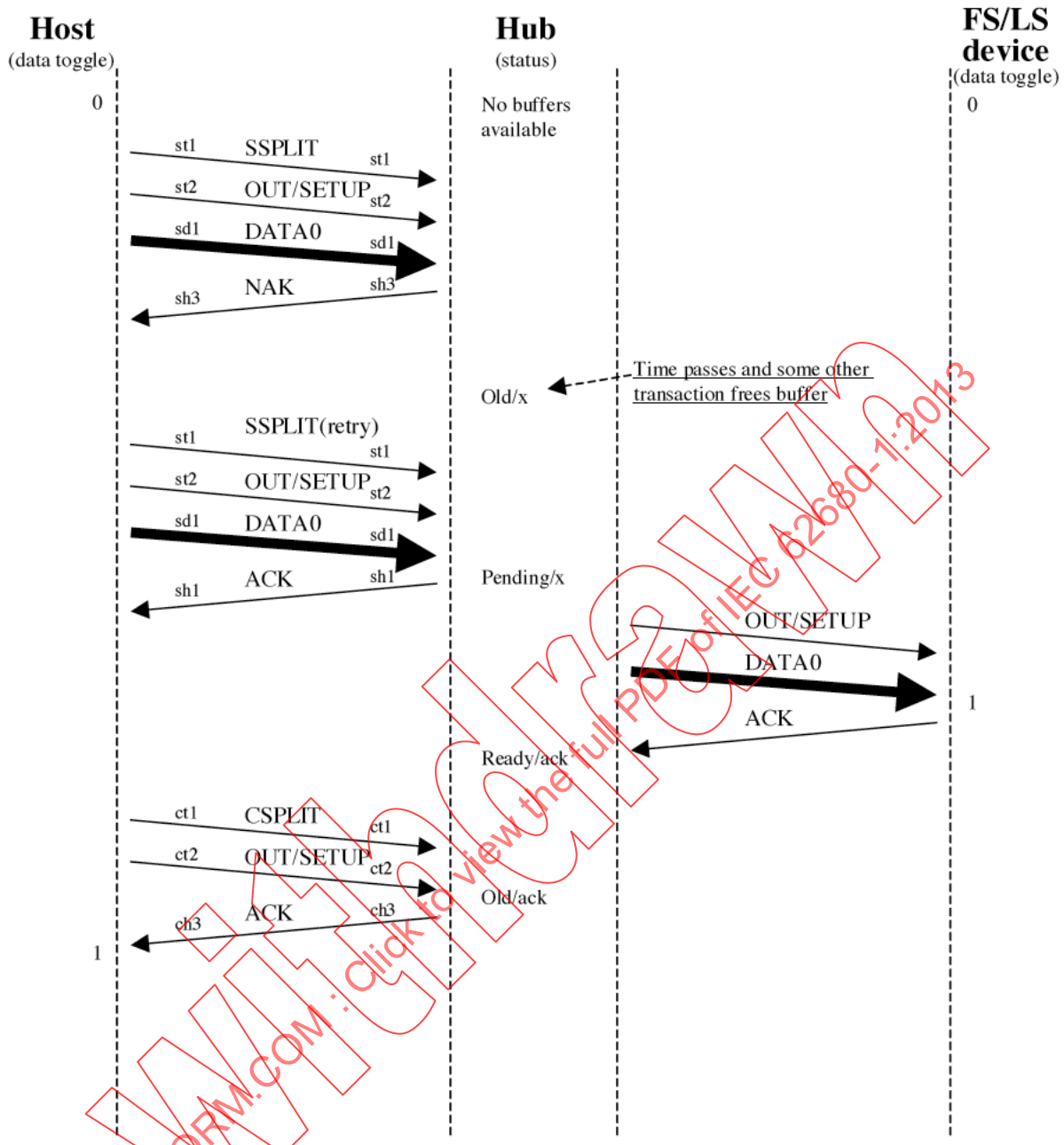


Figure A-15. No buffer Available No Smash (HS NAK(S))

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

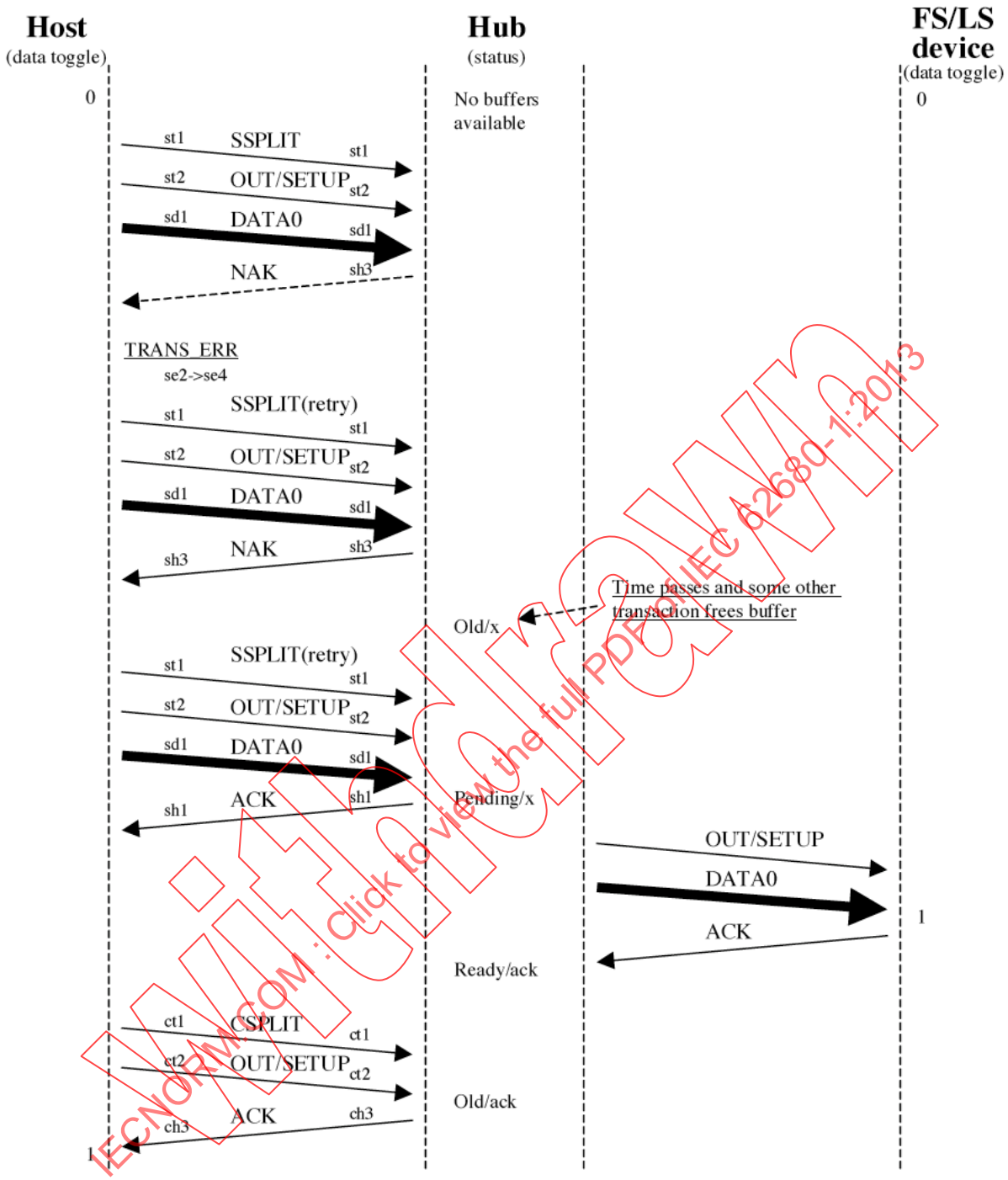


Figure A-16. No Buffer Available HS NAK(S) Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

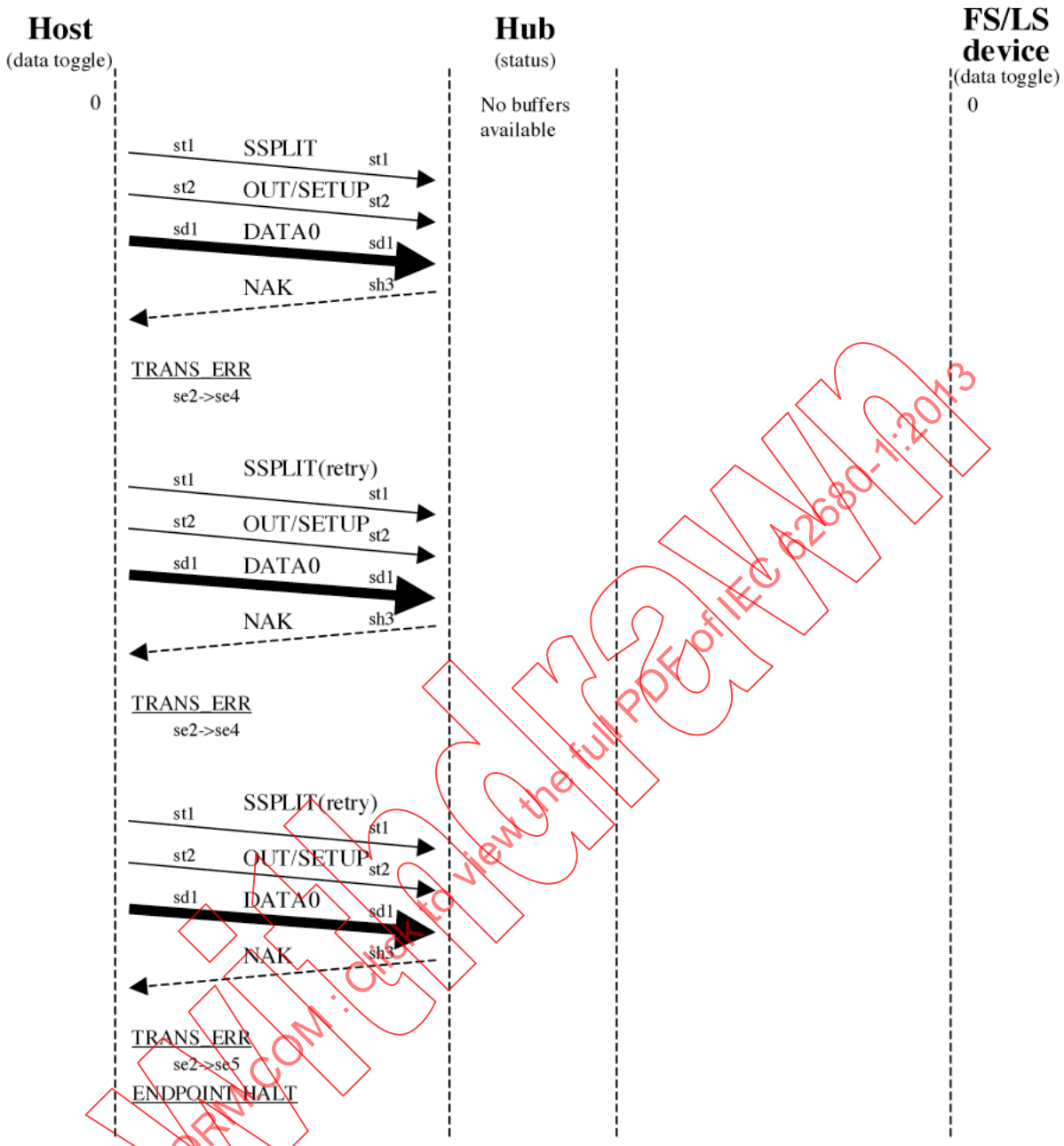


Figure A-17. No Buffer Available HS NAK(S) 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

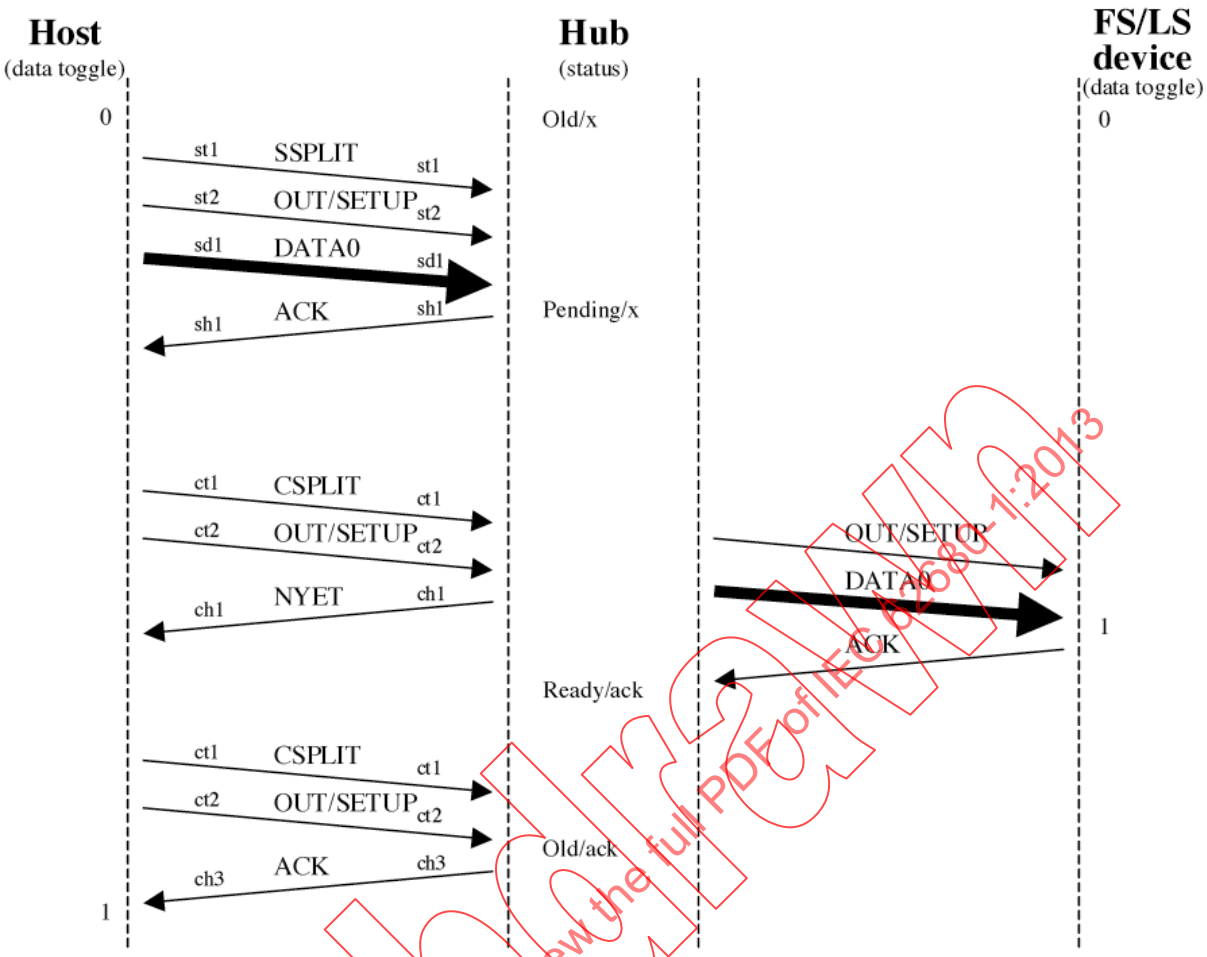


Figure A-18. CS Earlier No Smash (HS NYET)

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

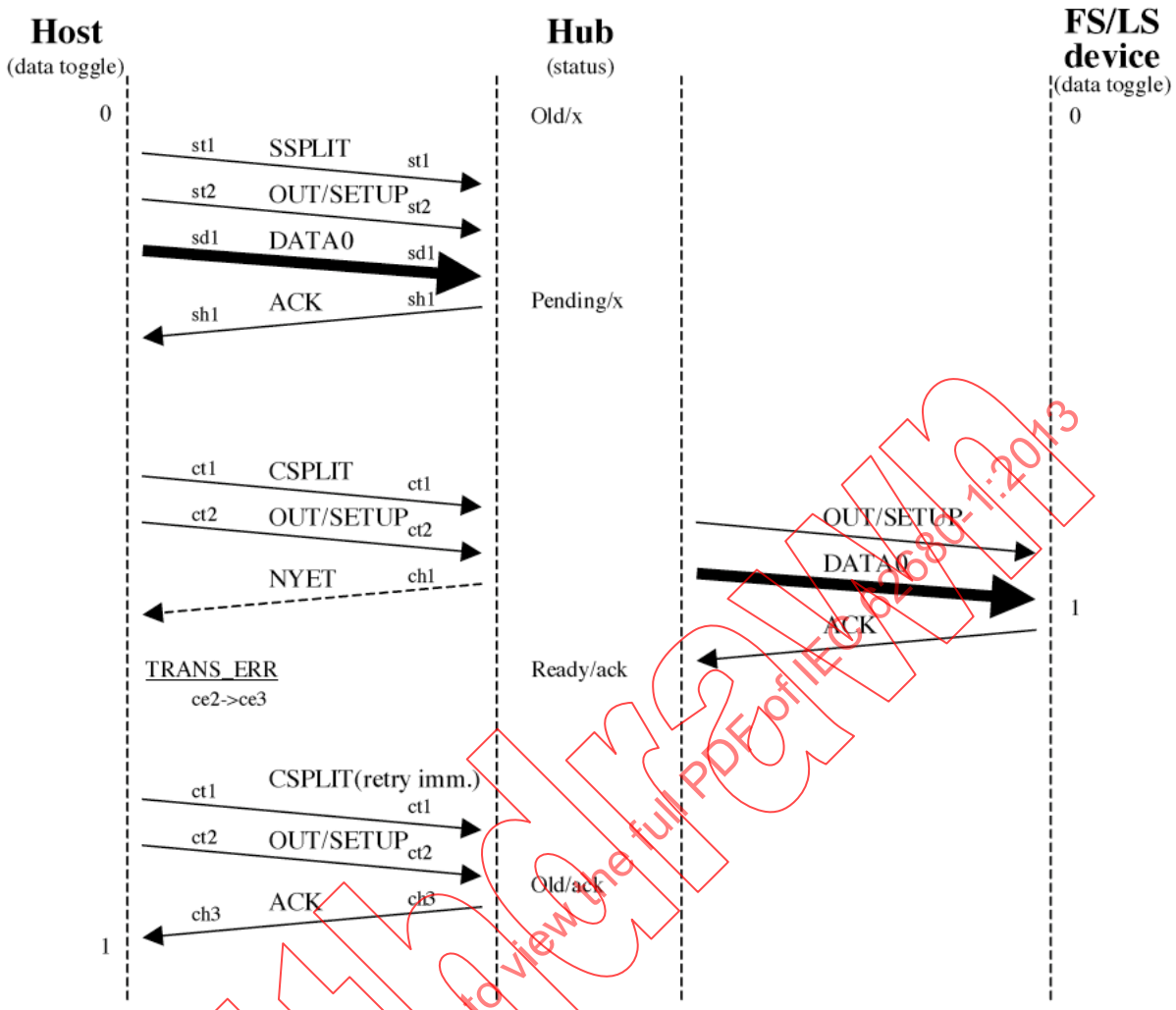


Figure A-19. CS Earlier HS NYET Smash(case 1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

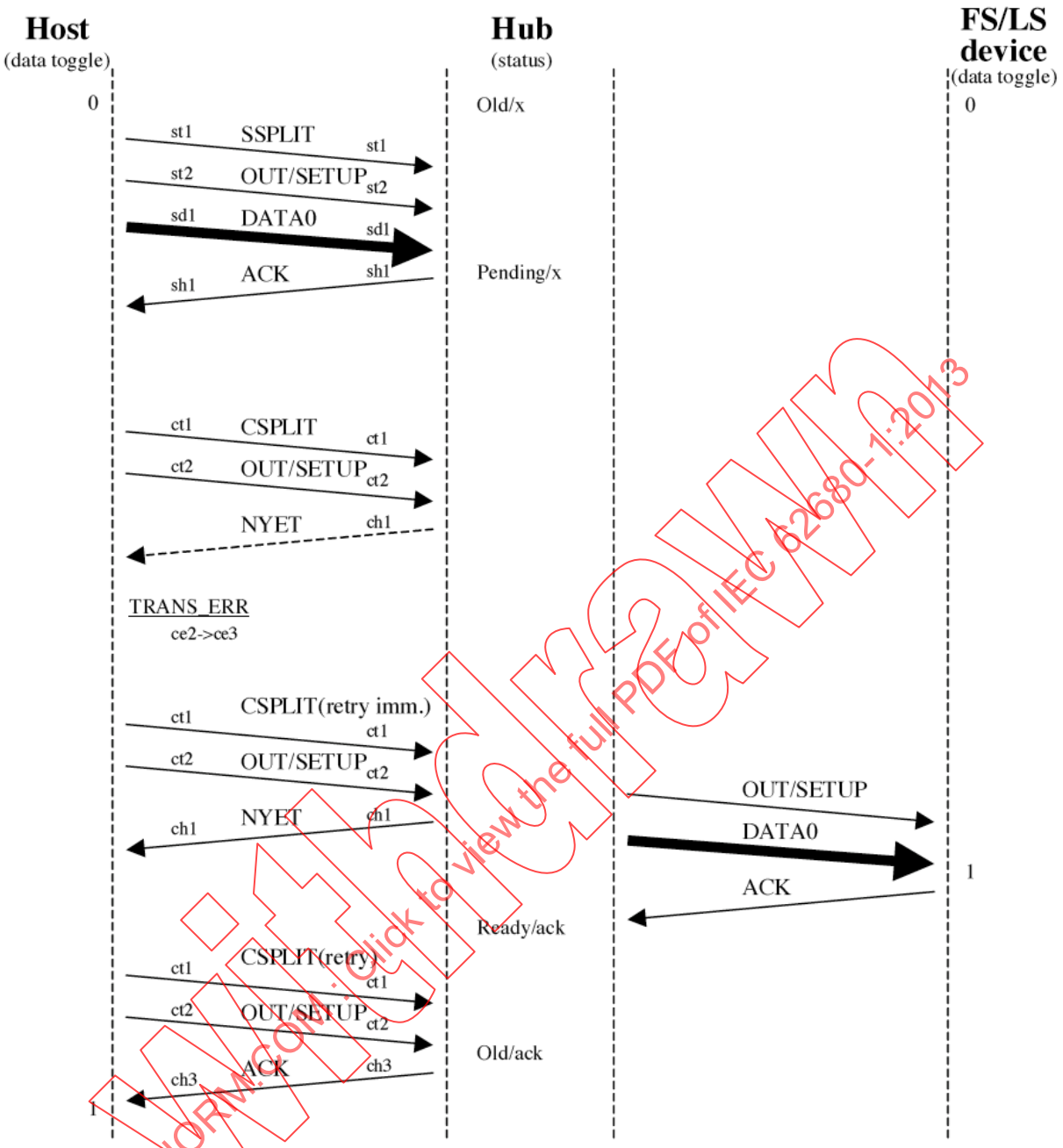


Figure A-20. CS Earlier HS NYET Smash(case 2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

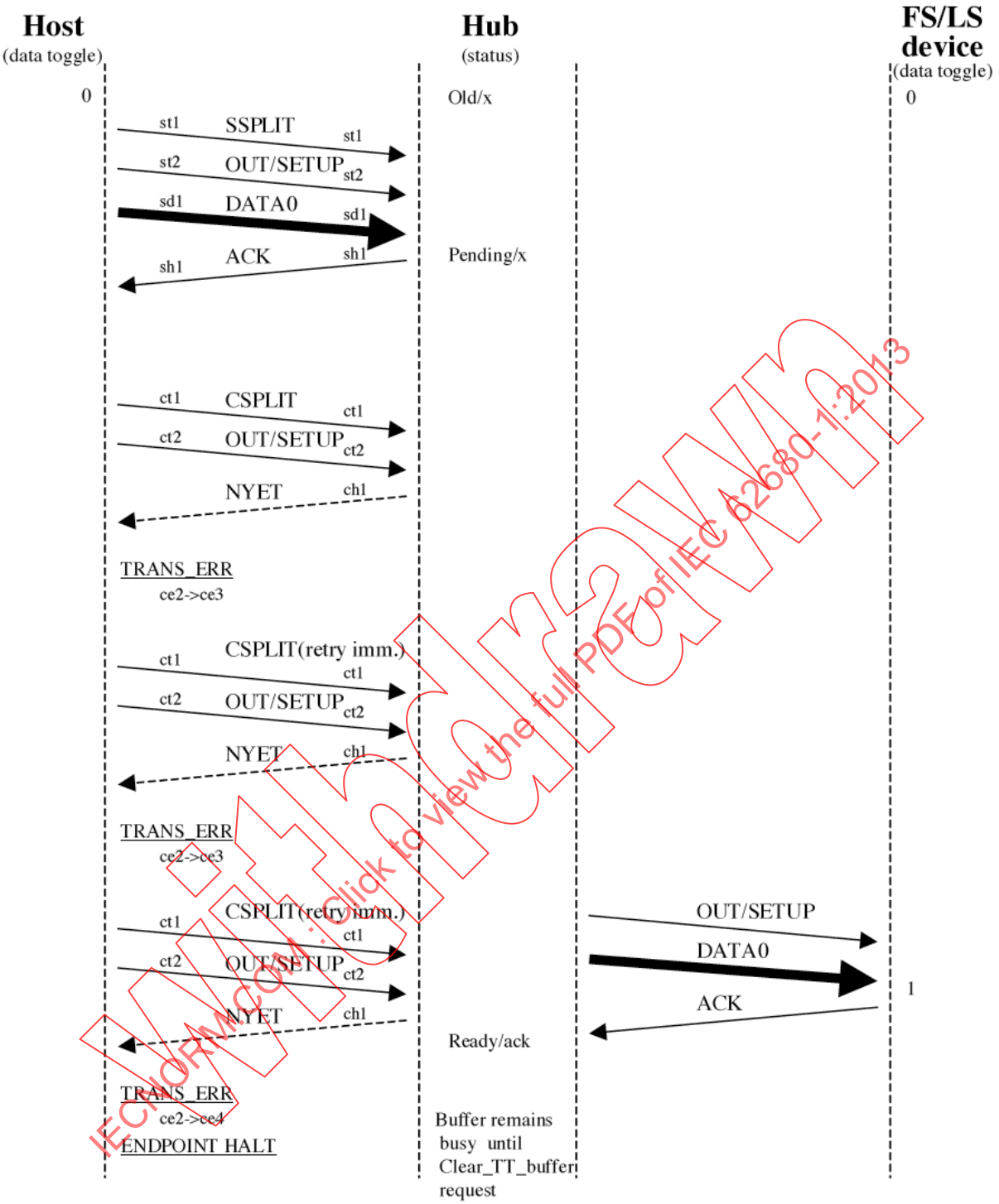


Figure A-21. CS Earlier HS NYET 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

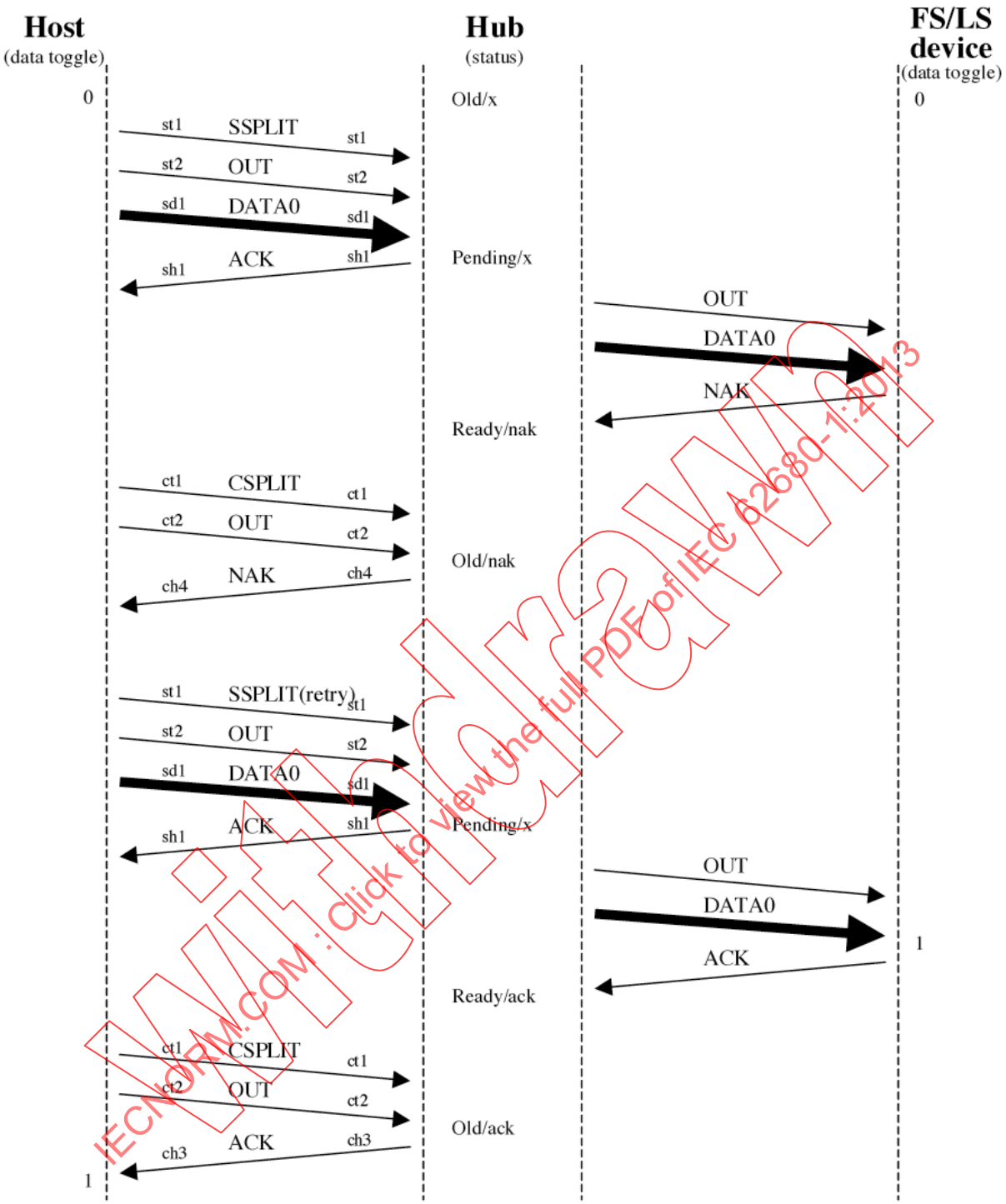


Figure A-22. Device Busy No Smash(FS/LS NAK)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

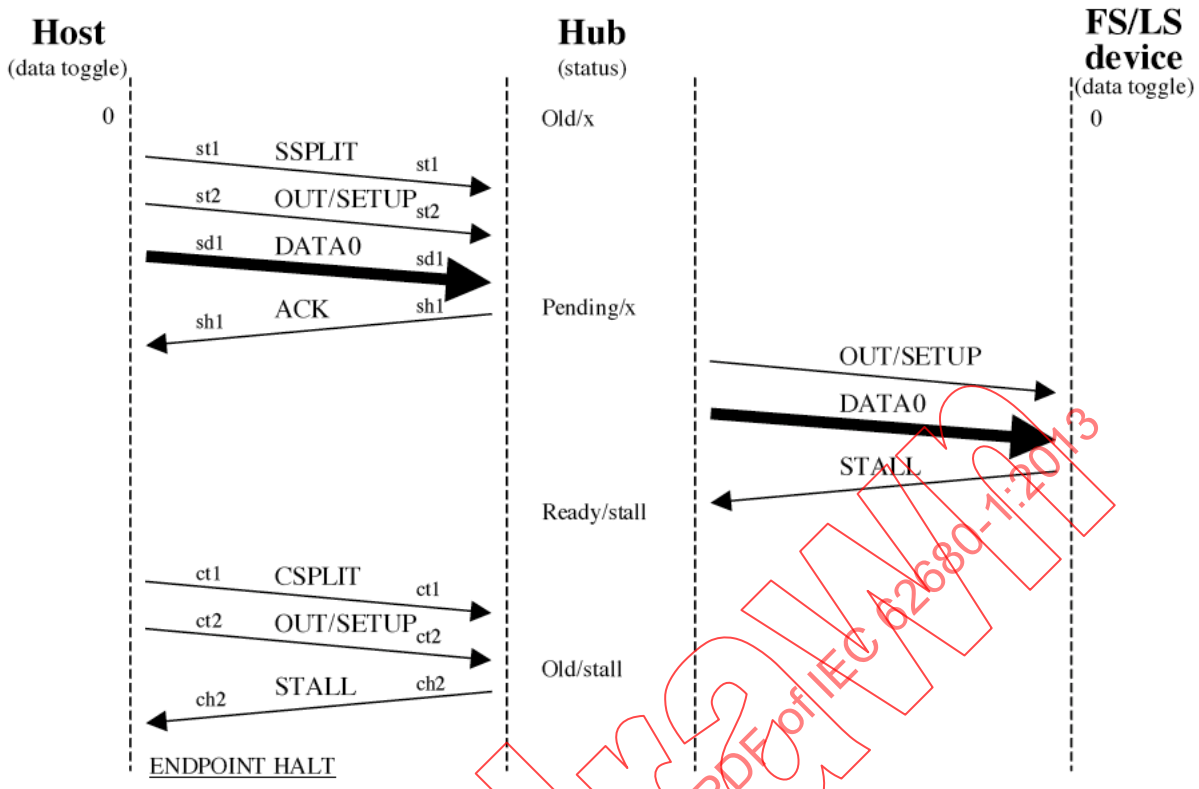


Figure A-23. Device Stall No Smash(FS/LS STALL)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A.2 Bulk/Control IN Transaction Examples

Legend:

(S): Start Split

(C): Complete Split

Summary of cases for bulk/control IN transaction

- Normal cases

Case	Reference Figure	Similar Figure
No smash	Figure A-24	
HS SSPLIT smash	Figure A-25	
HS SSPLIT 3 strikes smash	Figure A-26	
HS IN(S) smash		Figure A-25
HS IN(S) 3 strikes smash		Figure A-26
HS ACK(S) smash	Figure A-27 Figure A-28	
HS ACK(S) 3 strikes smash	Figure A-29	
HS CSPLIT smash	Figure A-30	
HS CSPLIT 3 strikes smash	Figure A-31	
HS IN(C) smash		Figure A-30
HS IN(C) 3 strikes smash		Figure A-31
HS DATA0/1 smash	Figure A-32	
HS DATA0/1 3 strikes smash	Figure A-33	
FS/LS IN smash	Figure A-34	
FS/LS IN 3 strikes smash	Figure A-35	
FS/LS DATA0/1 smash	Figure A-36	
FS/LS DATA0/1 3 strikes smash	Figure A-37	
FS/LS ACK smash	Figure A-38	
FS/LS ACK 3 strikes smash	No figure	

- No buffer(on hub) available cases

Case	Reference Figure	Similar Figure
No smash(HS NAK(S))	Figure A-39	
HS NAK(S) smash	Figure A-40	
HS NAK(S) 3 strikes smash	Figure A-41	

- CS(Complete-split transaction) earlier cases

Case	Reference Figure	Similar Figure
No smash(HS NYET)	Figure A-42	
HS NYET smash	Figure A-43 Figure A-44	
HS NYET 3 strikes smash	No figure	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Device busy cases

Case	Reference Figure	Similar Figure
No smash(HS NAK(C))	Figure A-45	
HS NAK(C) smash		Figure A-32
HS NAK(C) 3 strikes smash		Figure A-33
FS/LS NAK smash		Figure A-36
FS/LS NAK 3 strikes smash		Figure A-37

- Device stall cases

Case	Reference Figure	Similar Figure
No smash	Figure A-46	
HS STALL(C) smash		Figure A-32
HS STALL(C) 3 strikes smash		Figure A-33
FS/LS STALL smash		Figure A-36
FS/LS STALL 3 strikes smash		Figure A-37

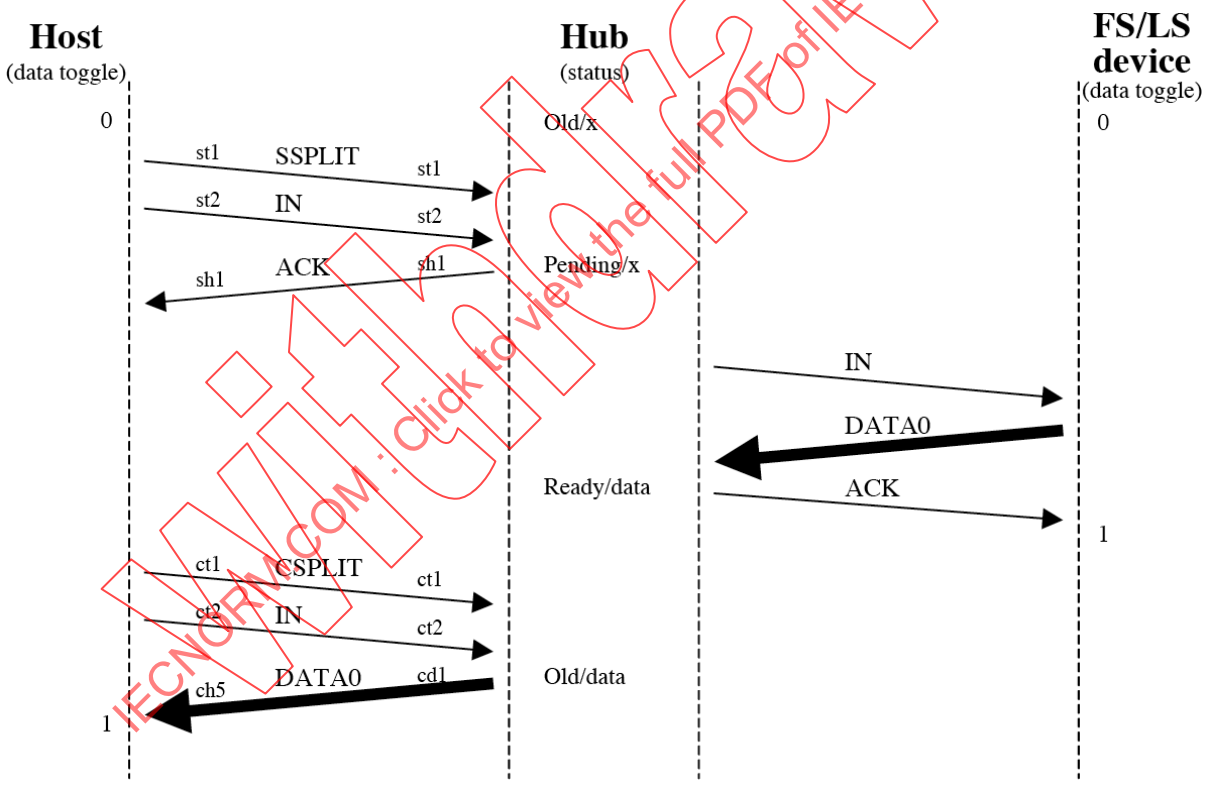


Figure A-24. Normal No Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

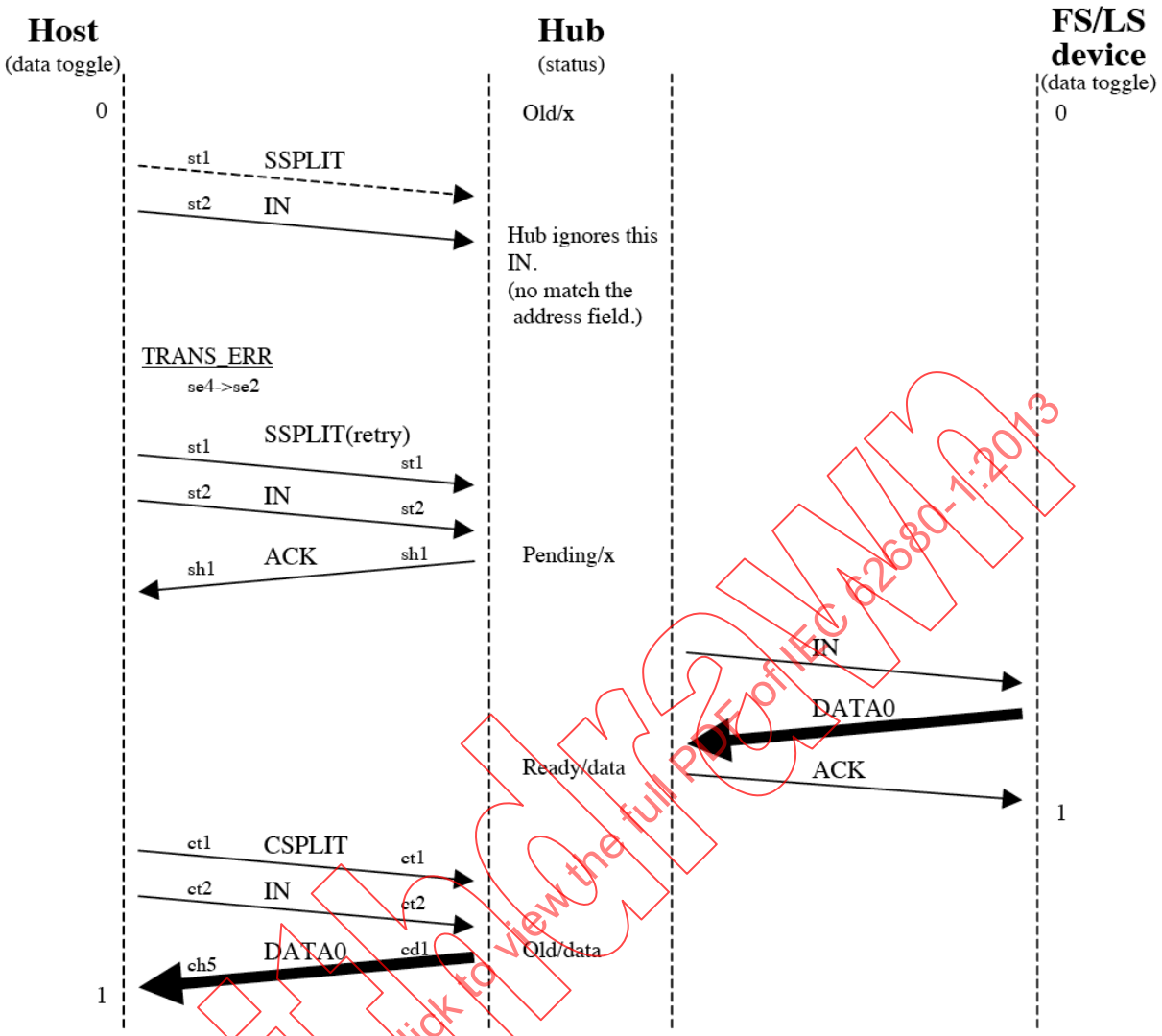


Figure A-25. Normal HS SSPLIT Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

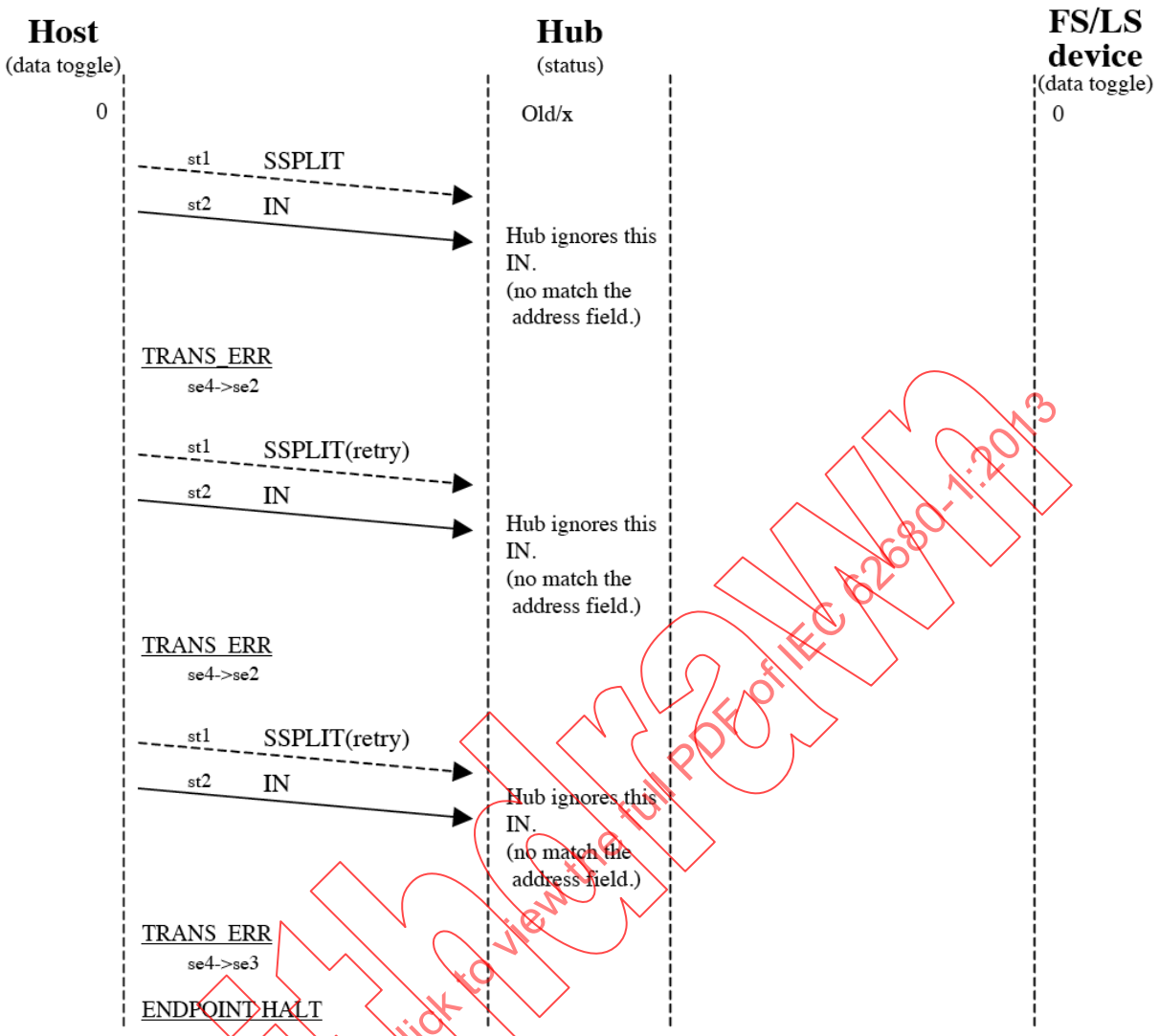


Figure A-26. Normal SSPLIT 3 Strikes Smash

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

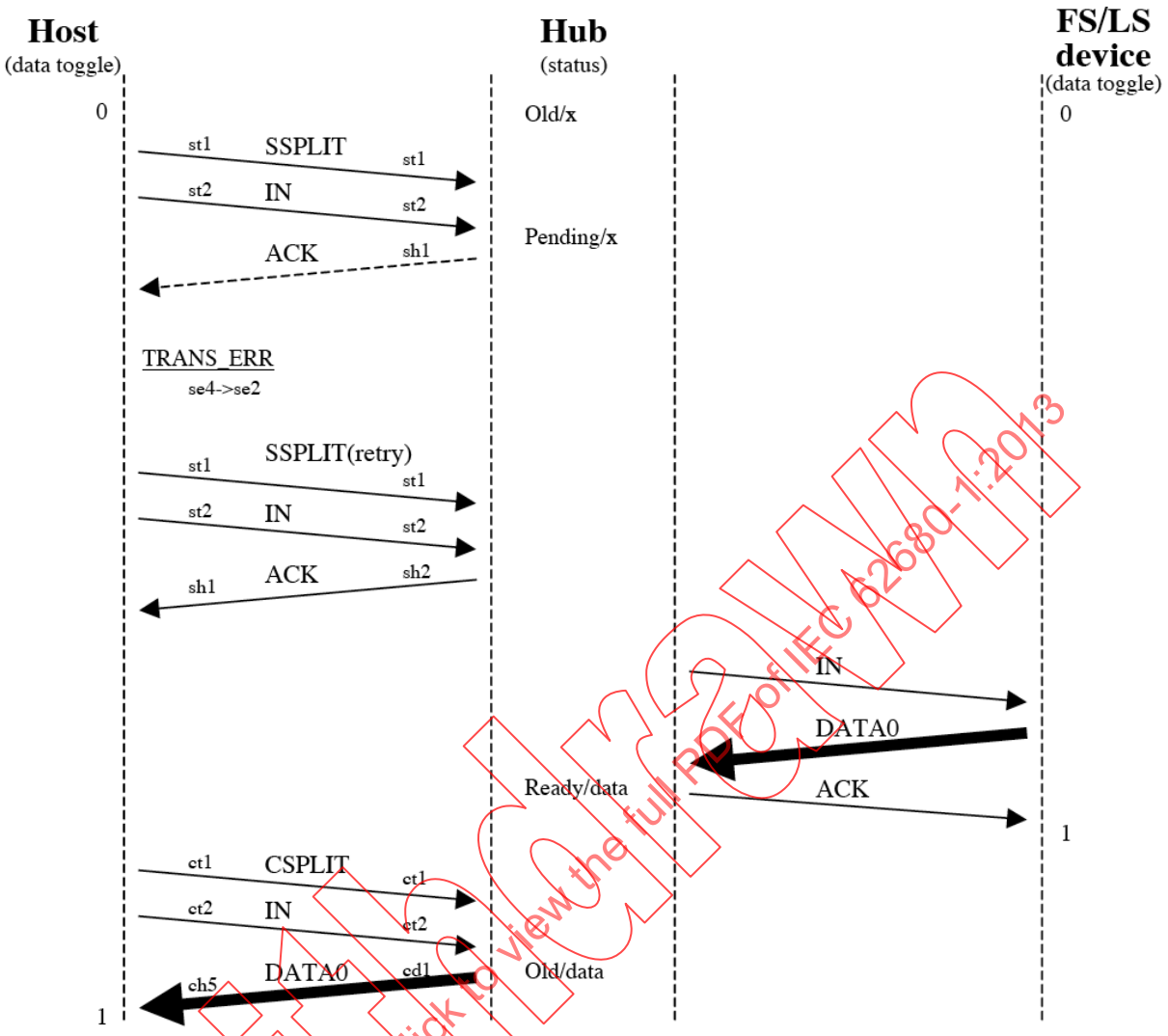


Figure A-27. Normal HS ACK(S) Smash(case 1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

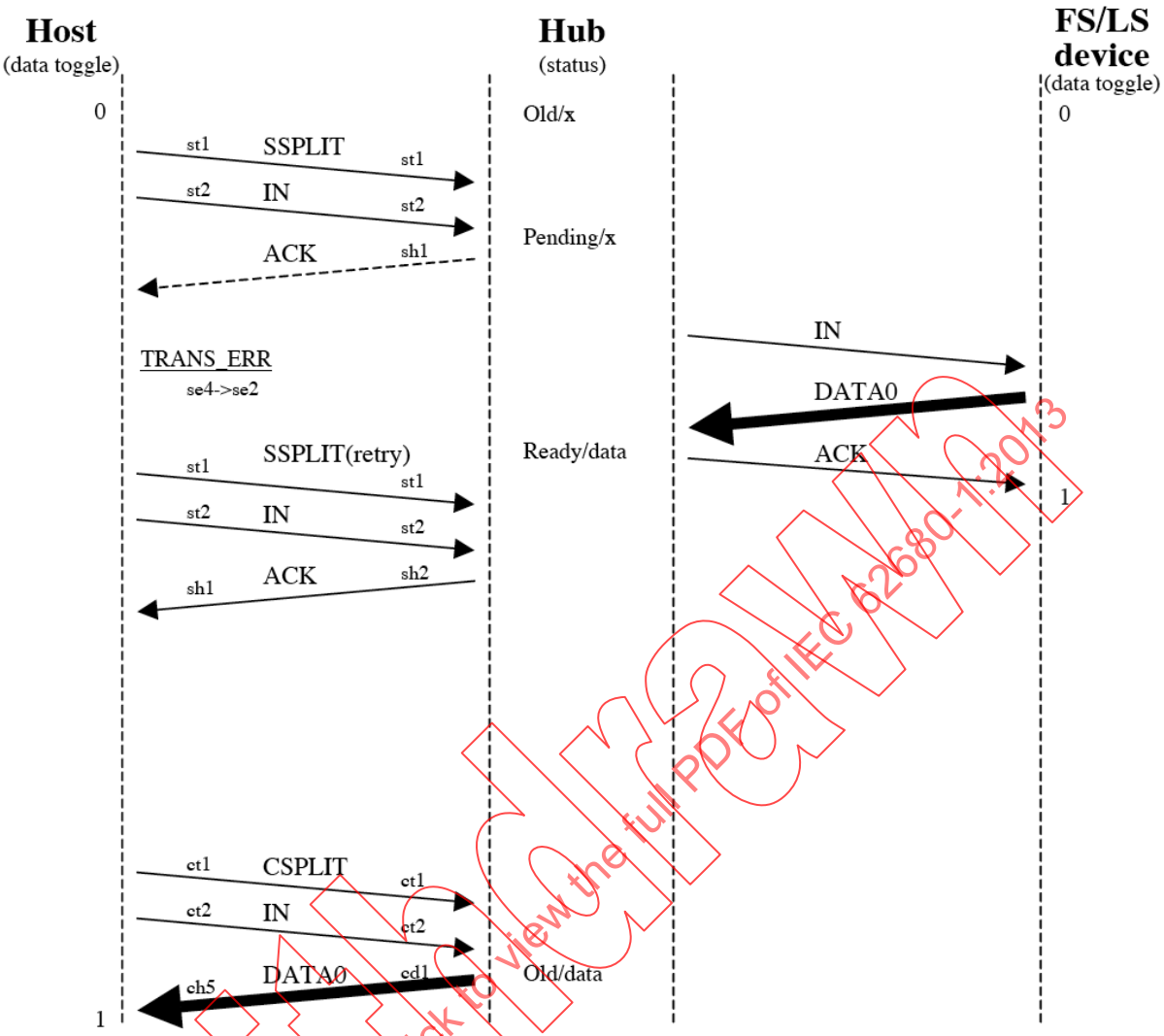


Figure A-28. Normal HS ACK(S) Smash(case 2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

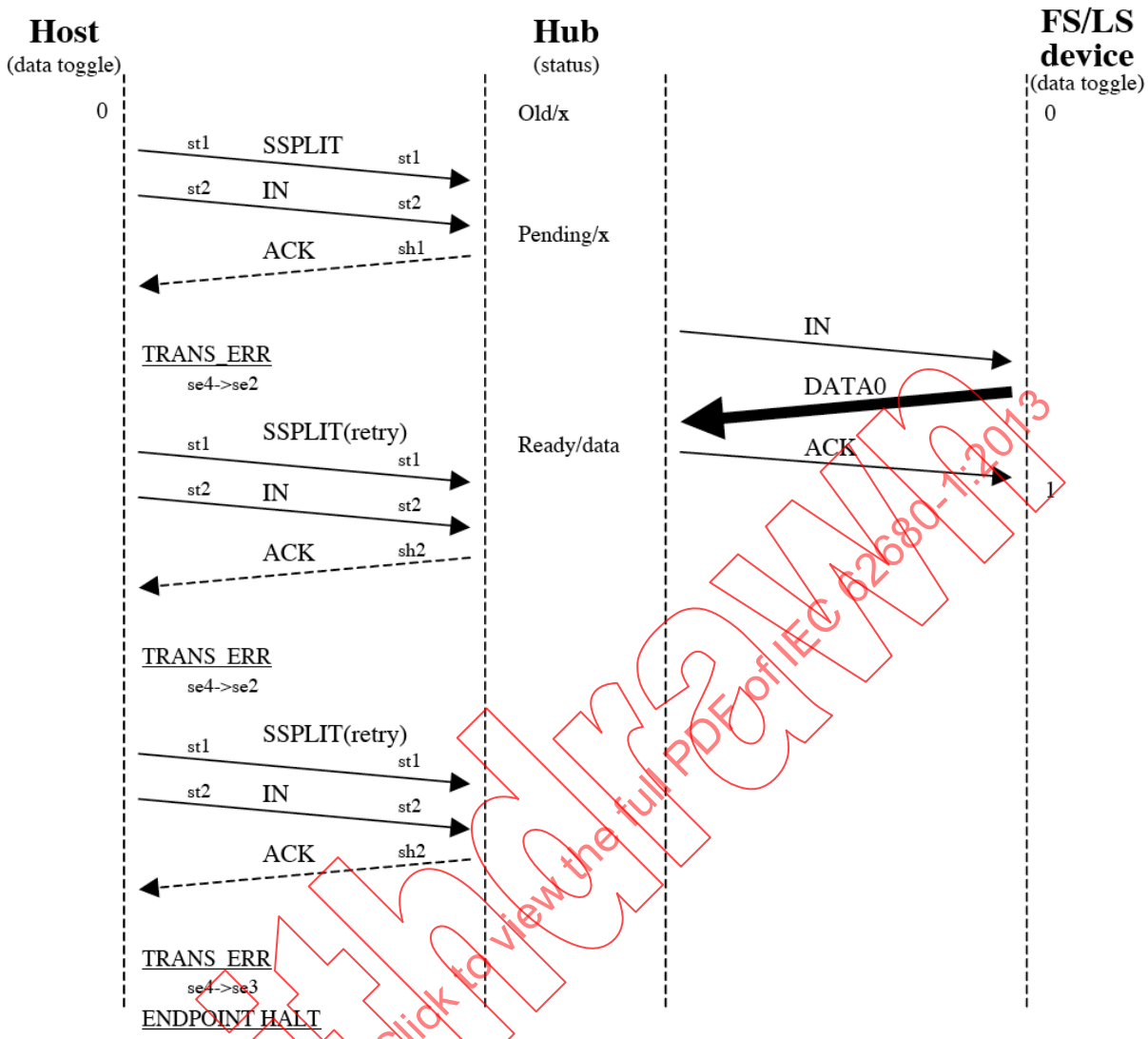


Figure A-29. Normal HS ACK(S) 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

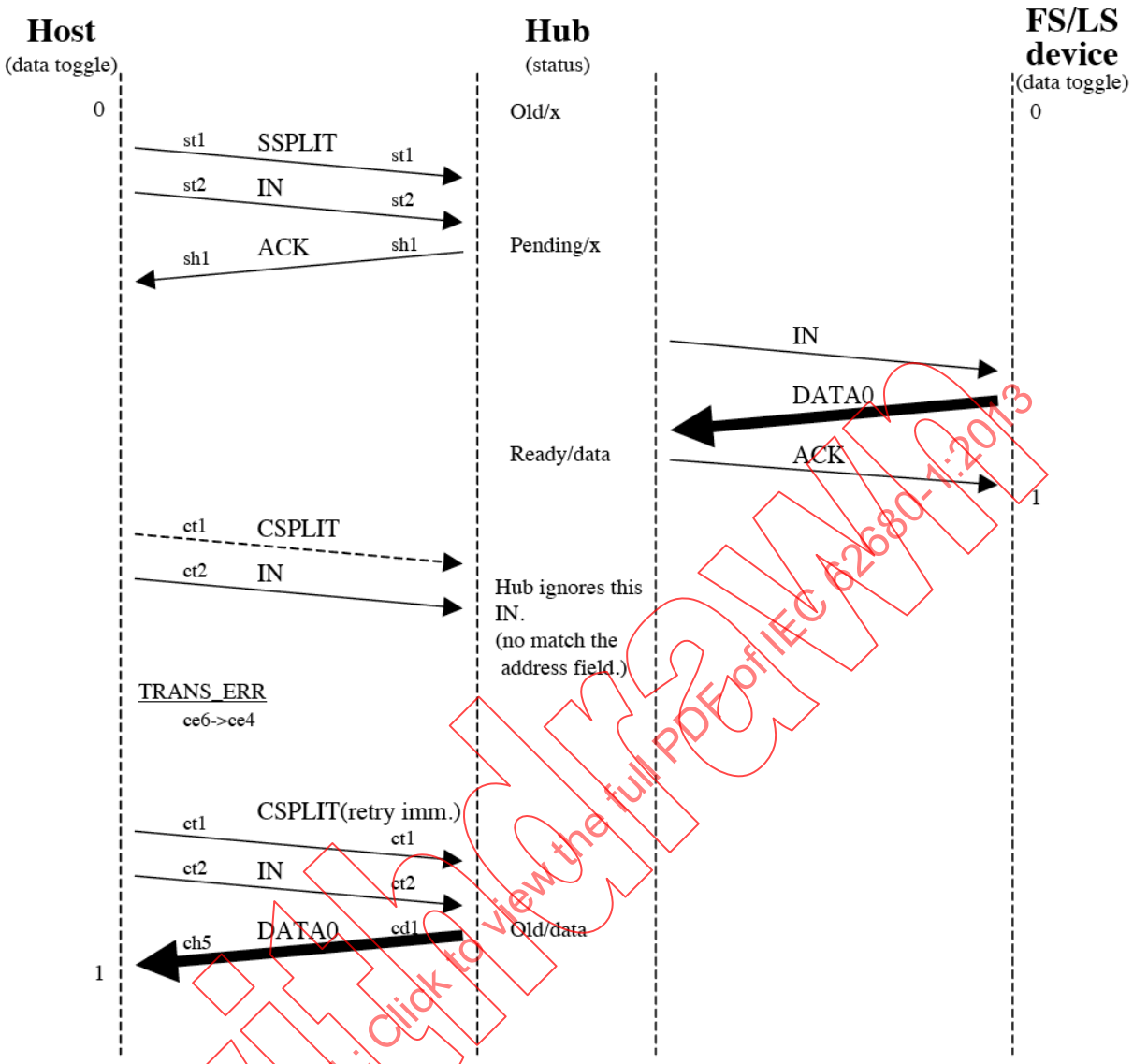


Figure A-30. Normal HS CSPLIT Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

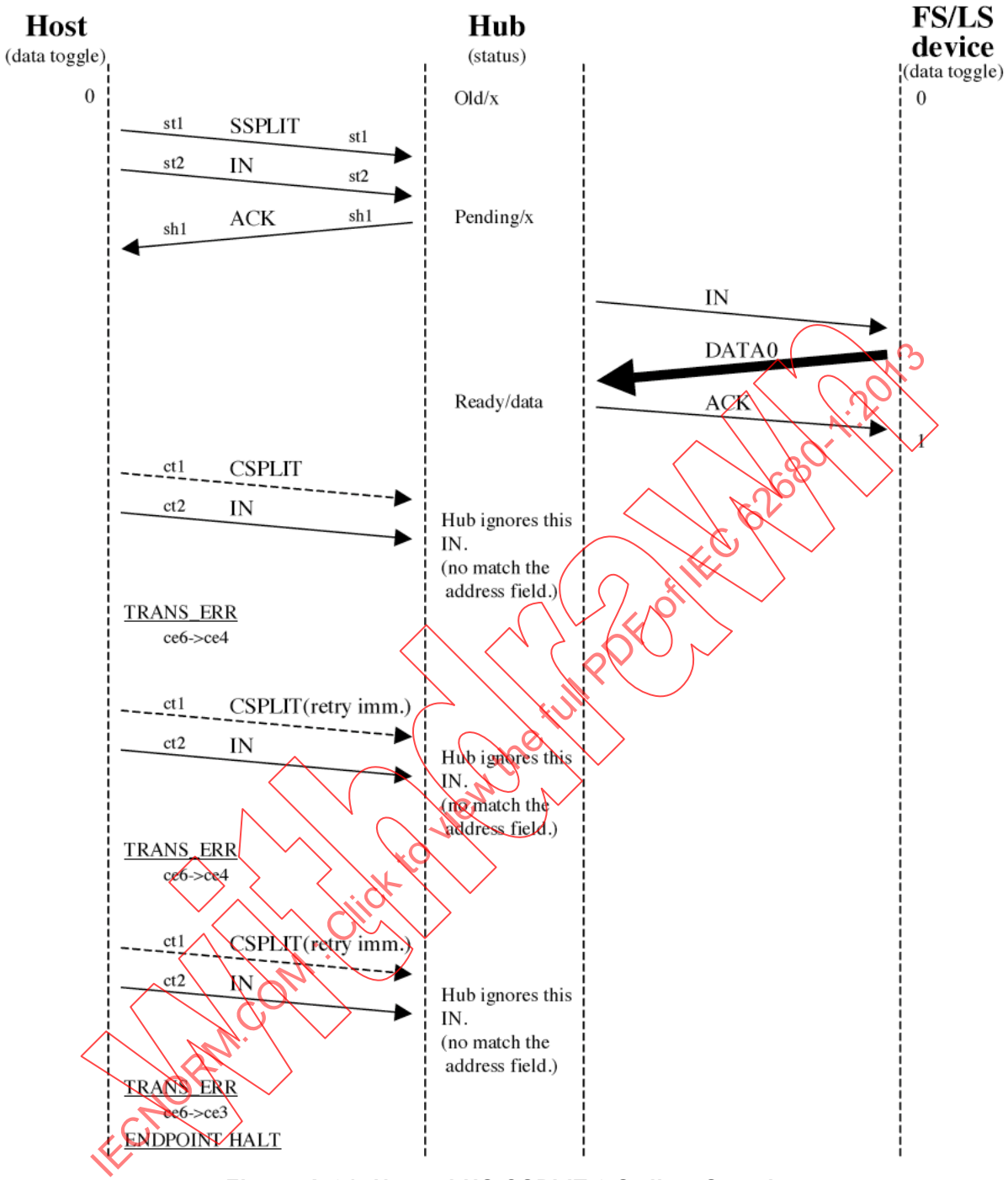


Figure A-31. Normal HS CSPLIT 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

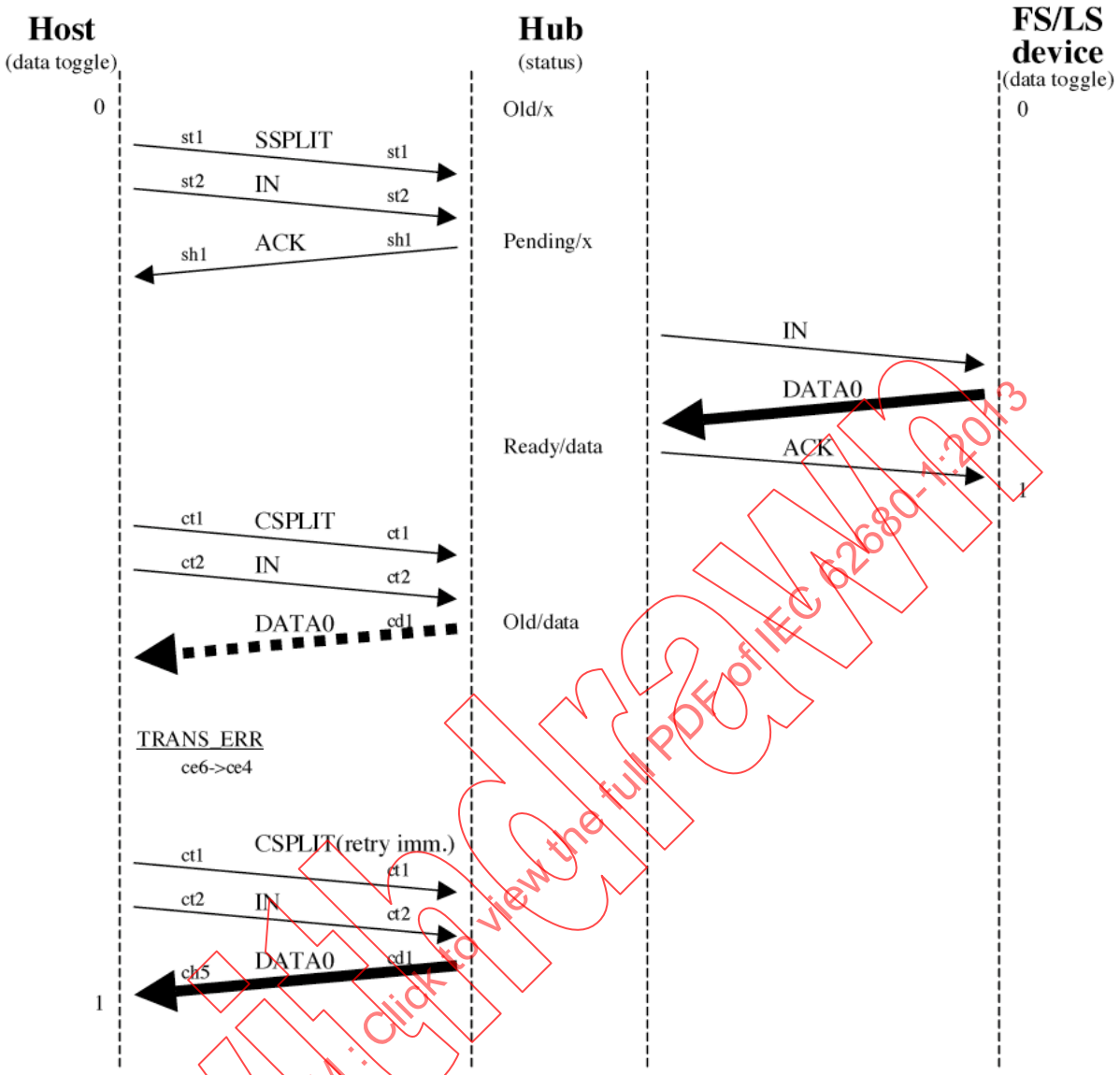


Figure A-32. Normal HS DATA0/1 Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

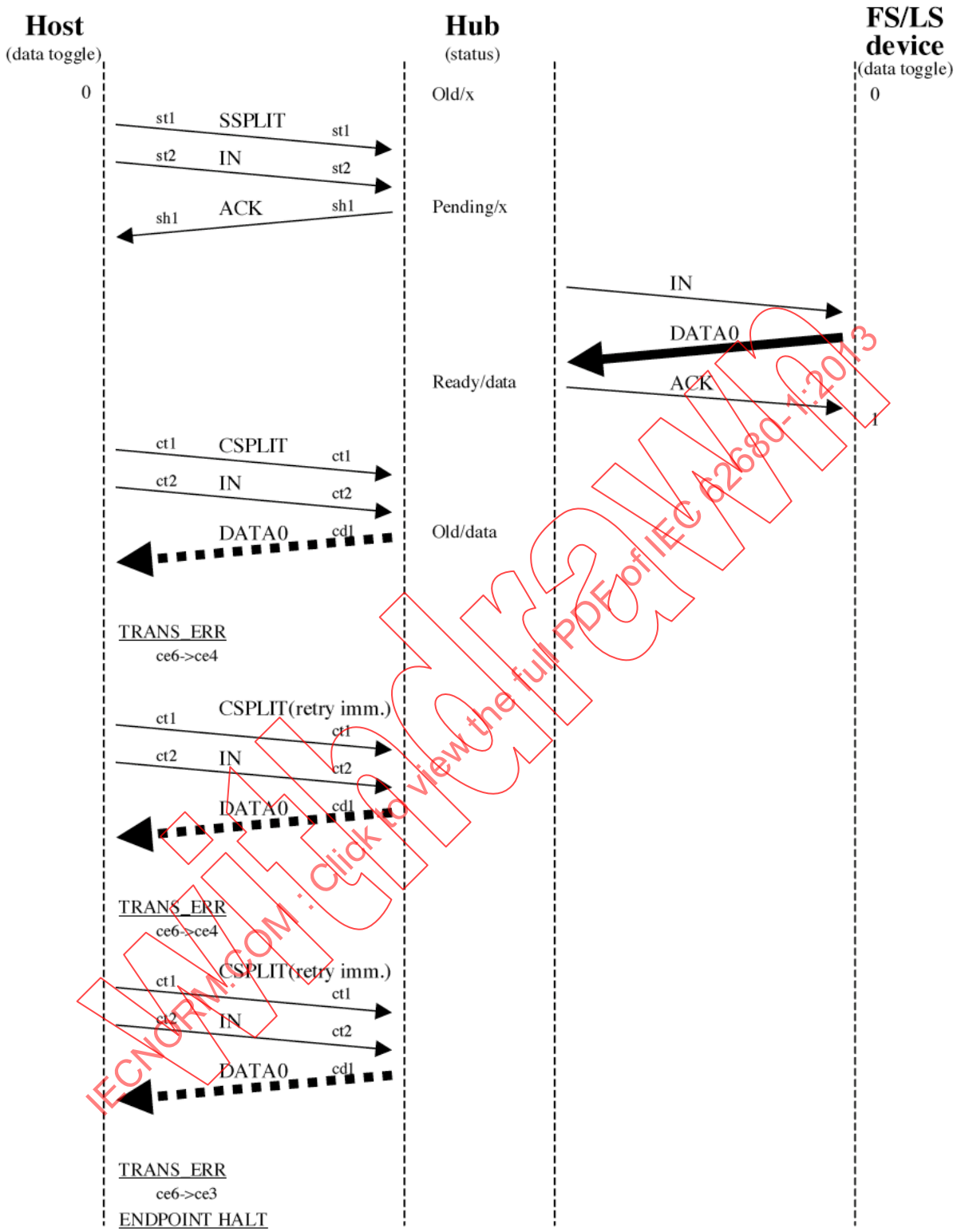


Figure A-33. Normal HS DATA0/1 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

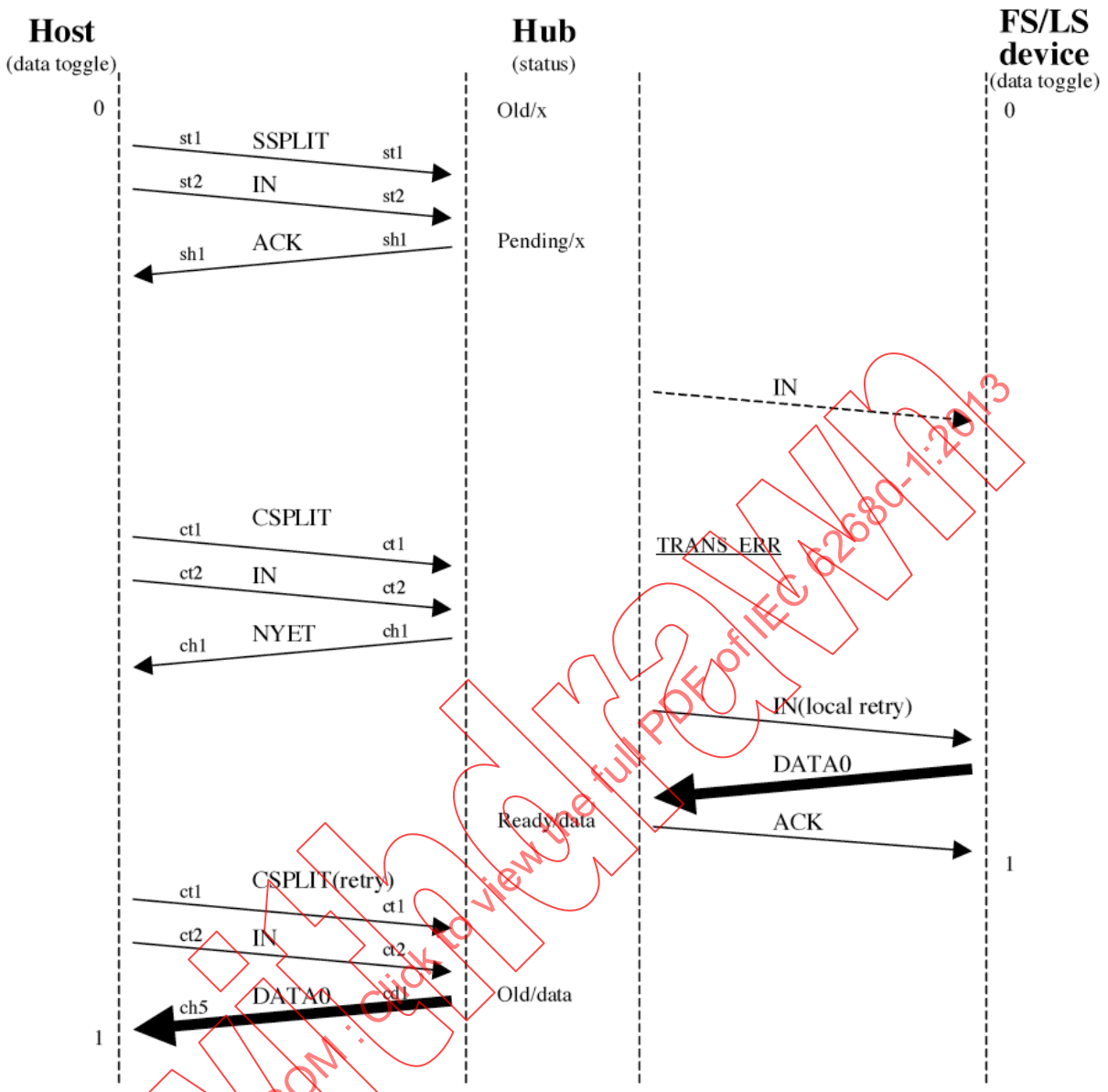


Figure A-34. Normal FS/LS IN Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

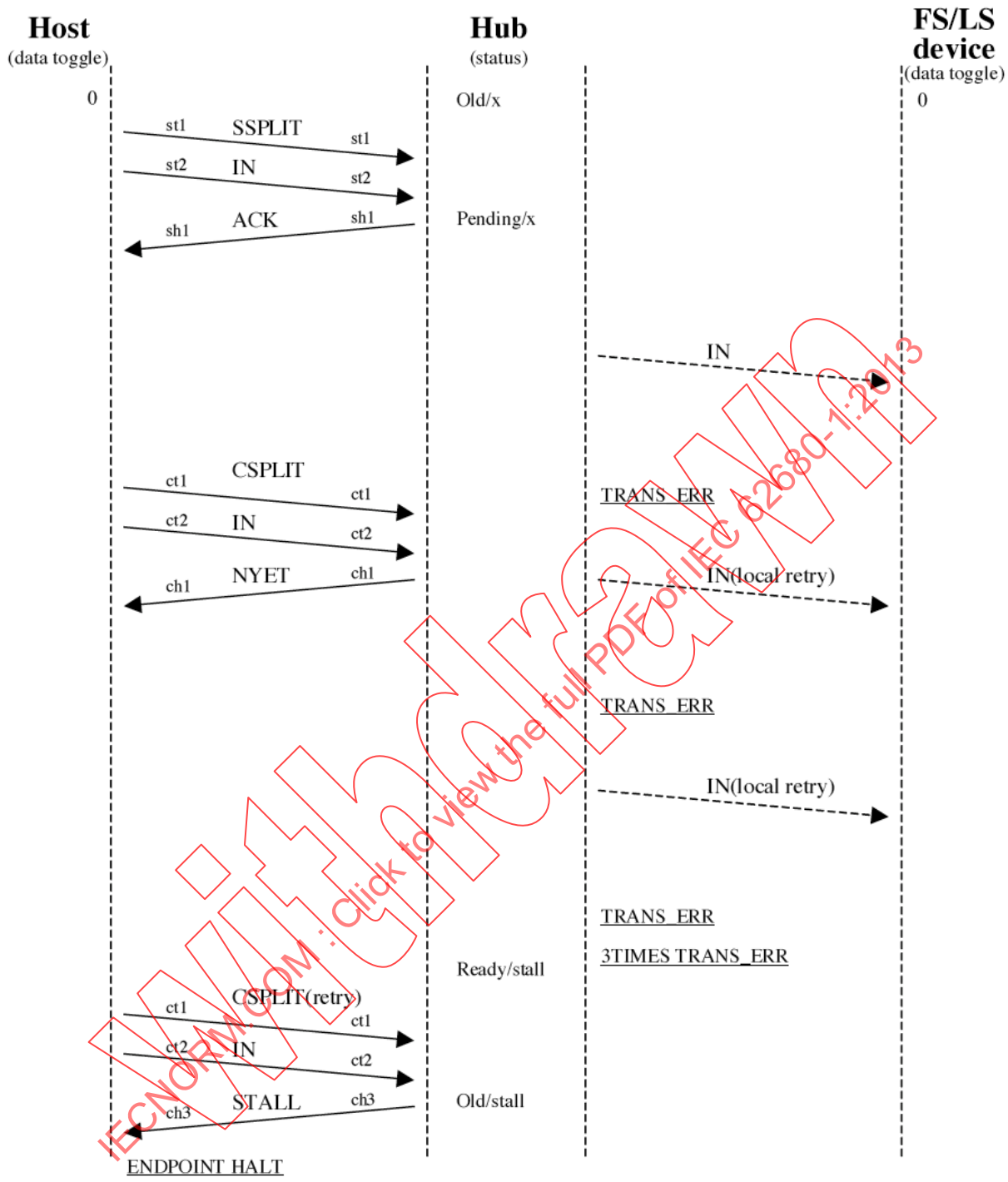


Figure A-35. Normal FS/LS IN 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

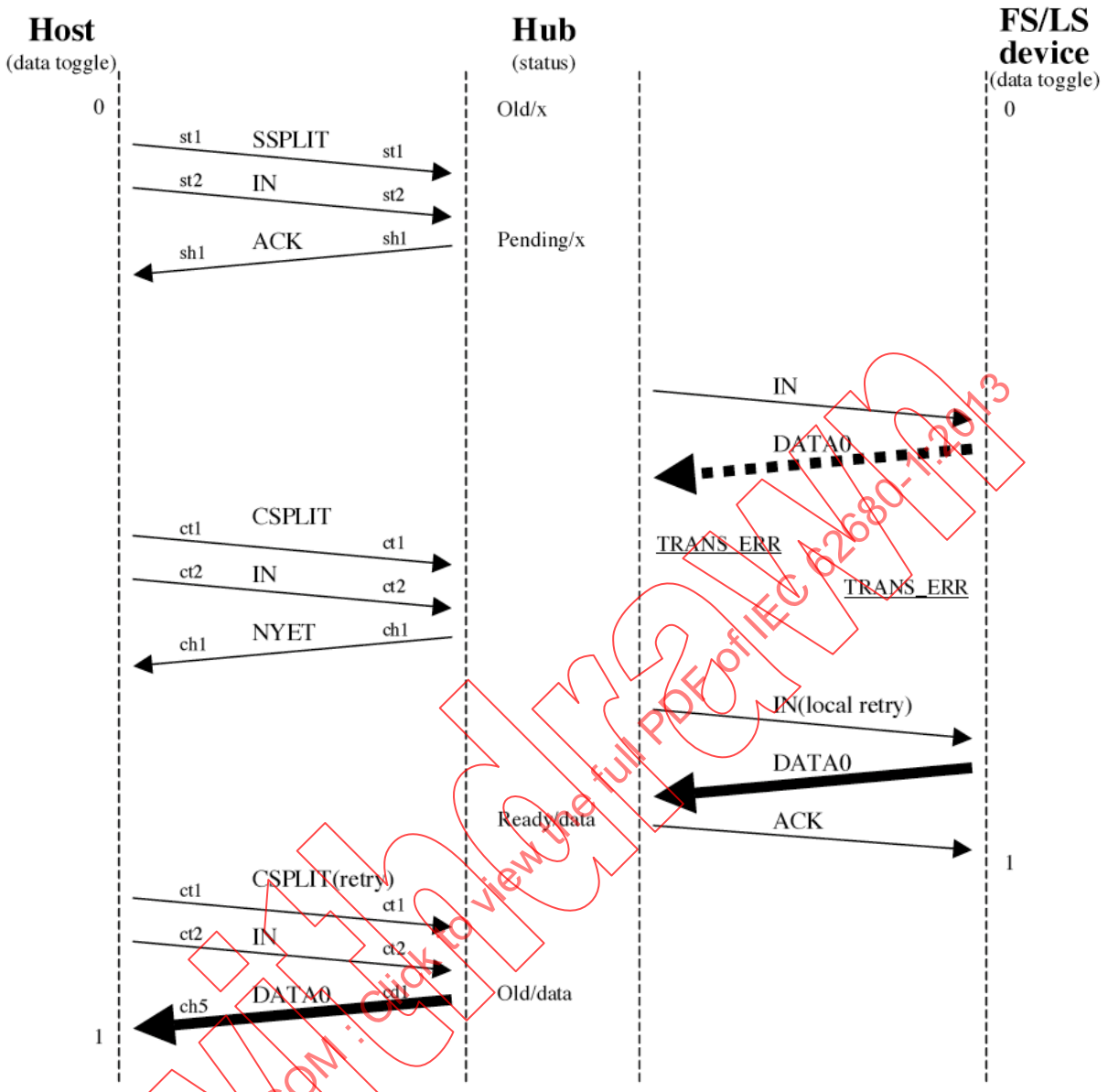


Figure A-36. Normal FS/LS DATA0/1 Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

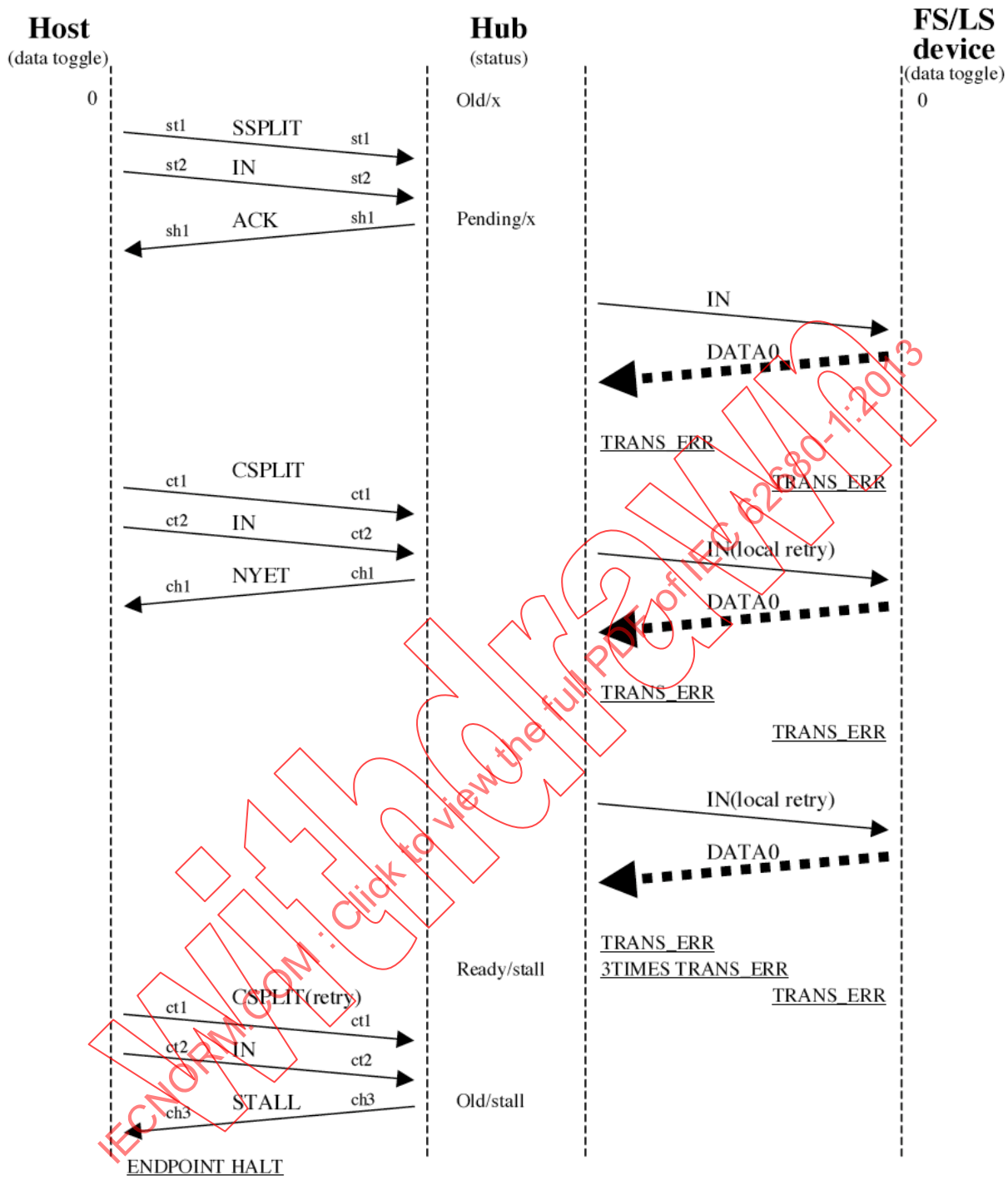


Figure A-37. Normal FS/LS DATA0/1 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

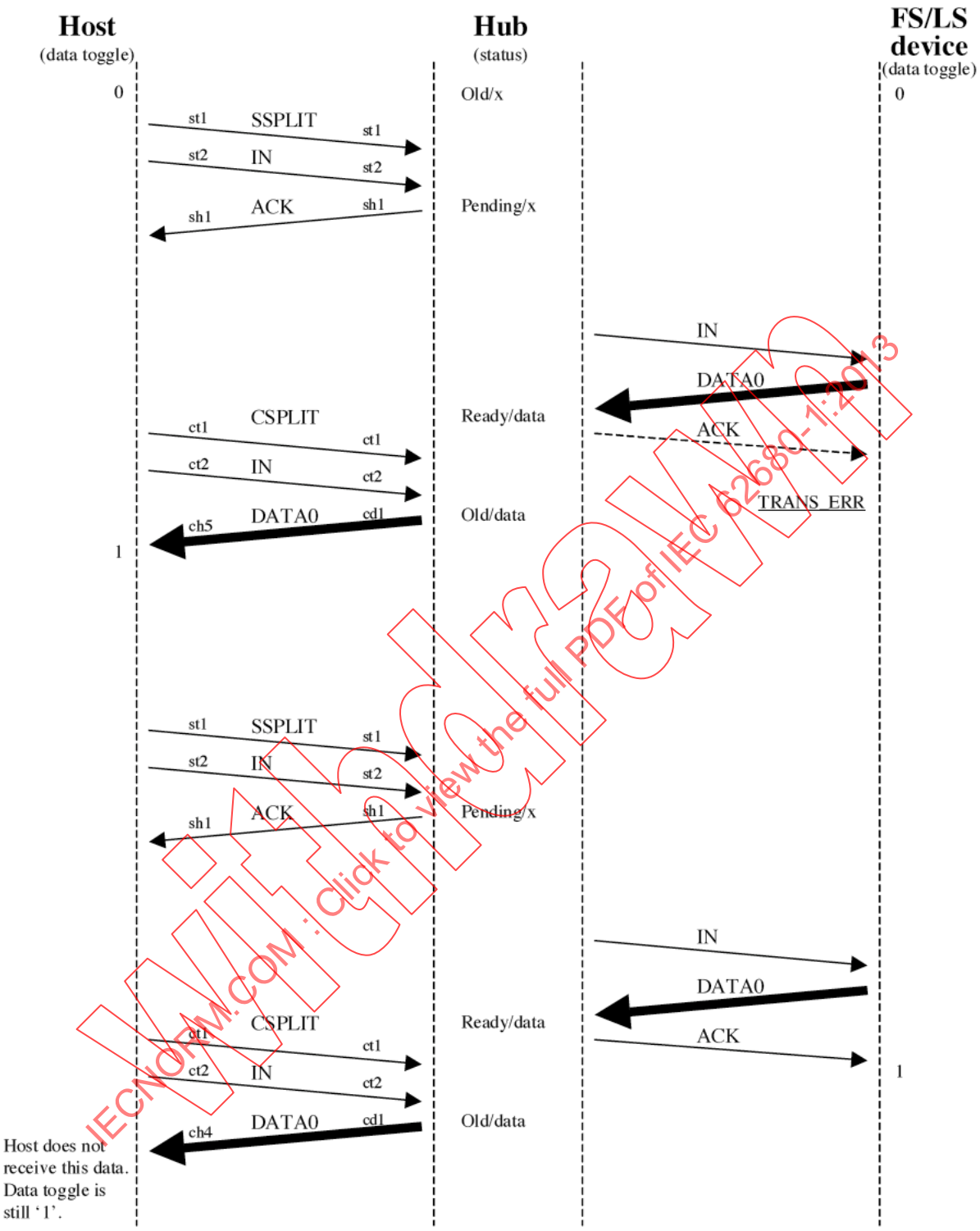


Figure A-38. Normal FS/LS ACK Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

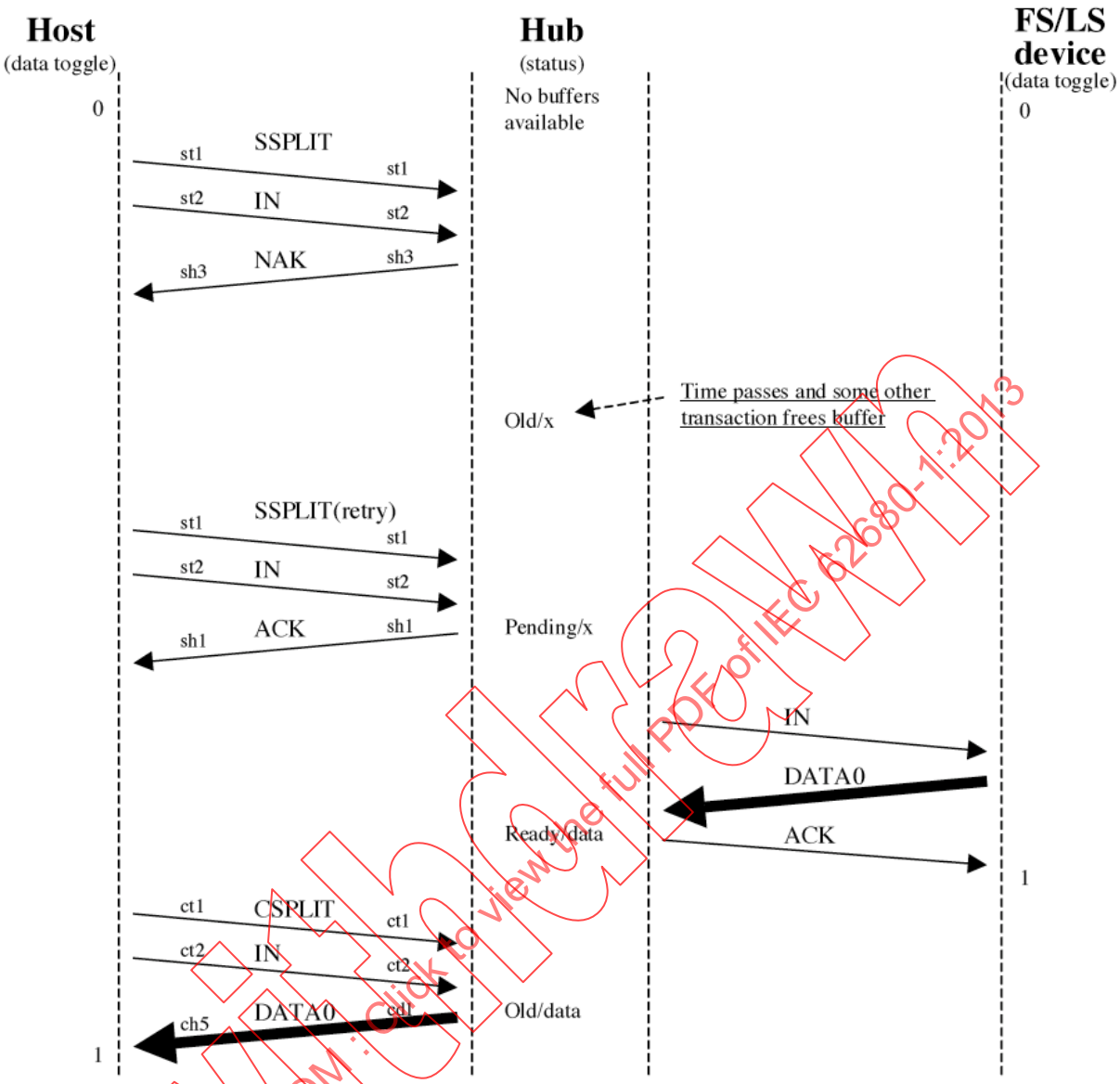


Figure A-39. No Buffer Available No Smash(HS NAK(S))

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

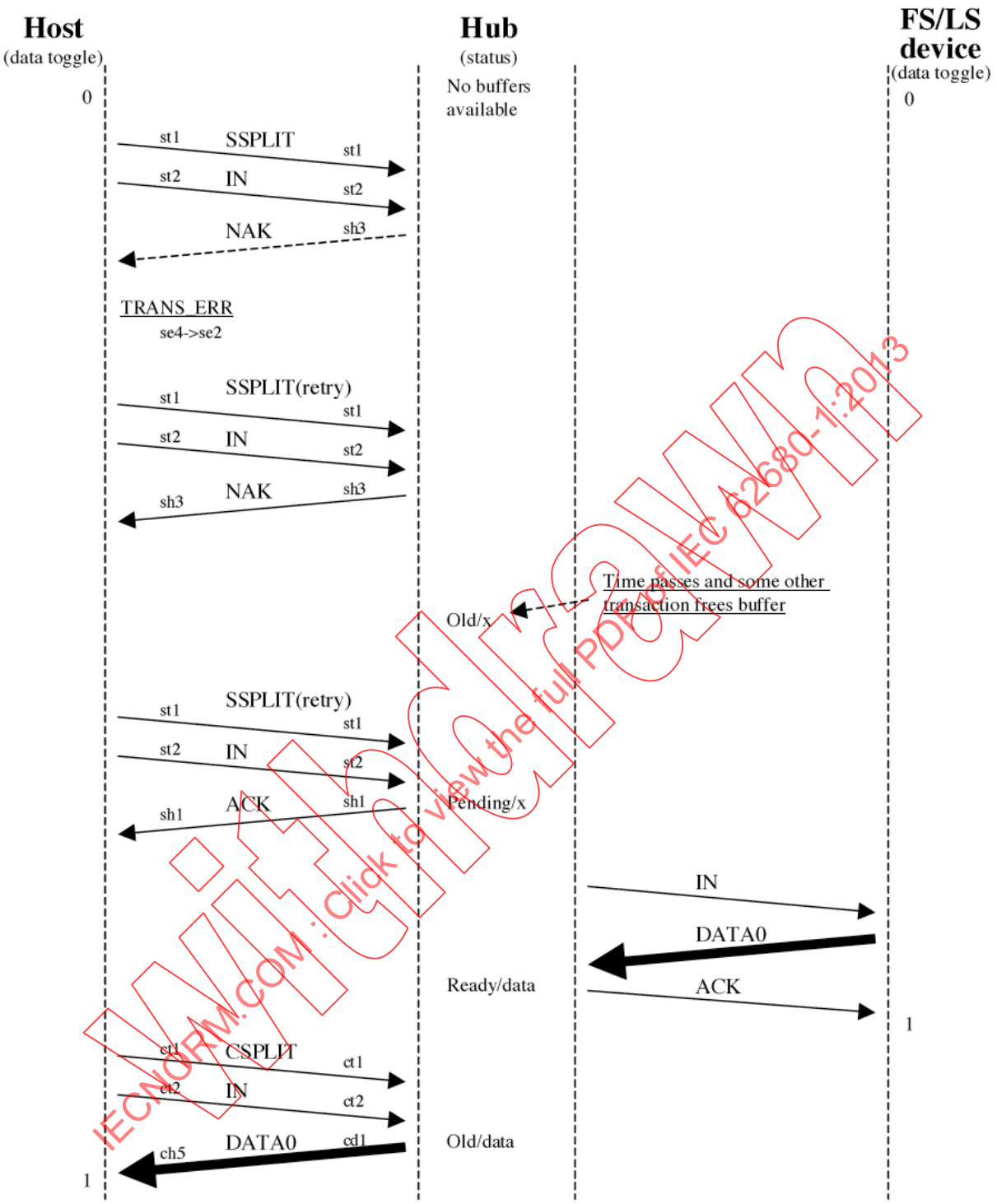


Figure A-40. No Buffer Available HS NAK(S) Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

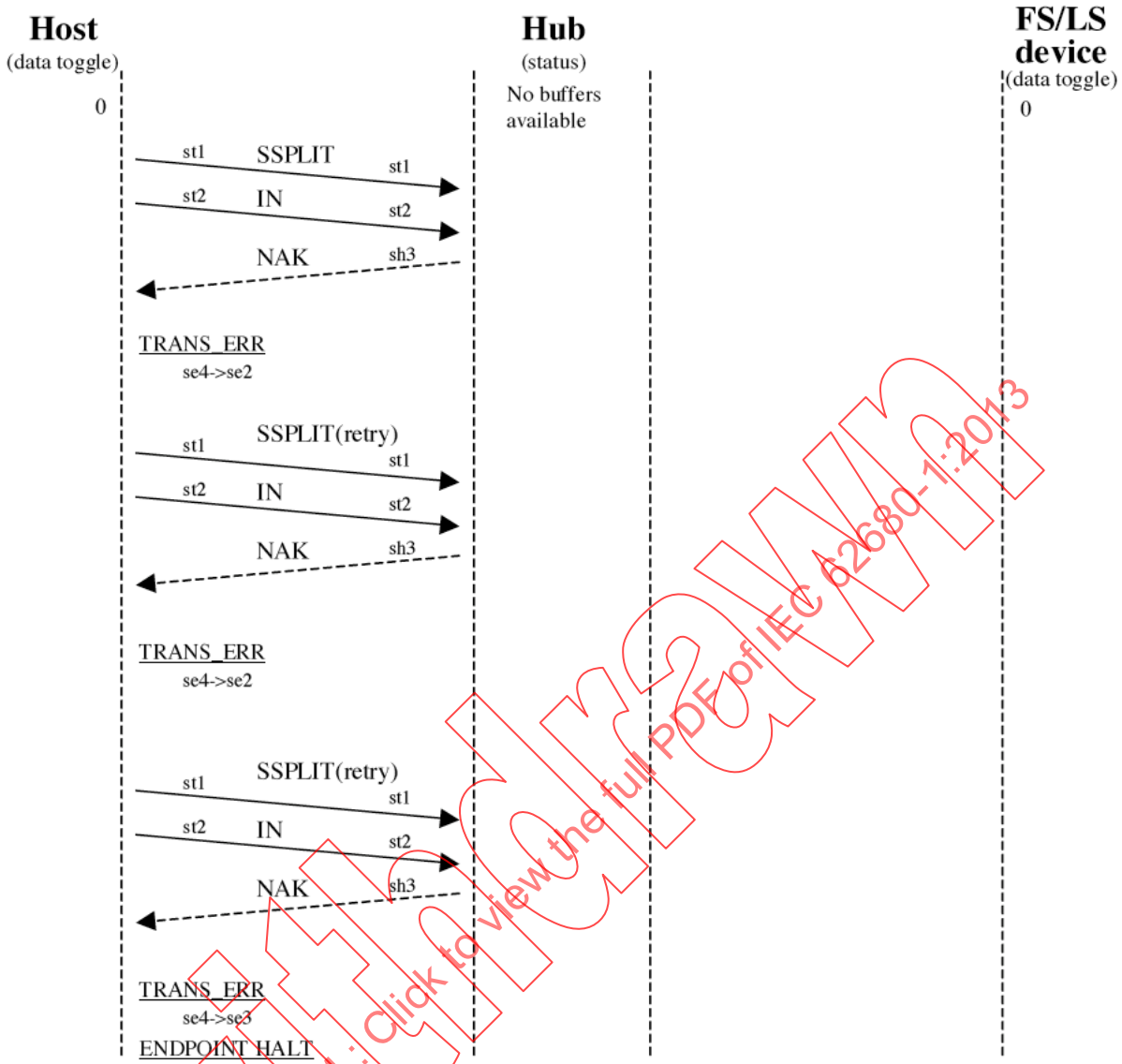


Figure A-41. No Buffer Available HS NAK(S) 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

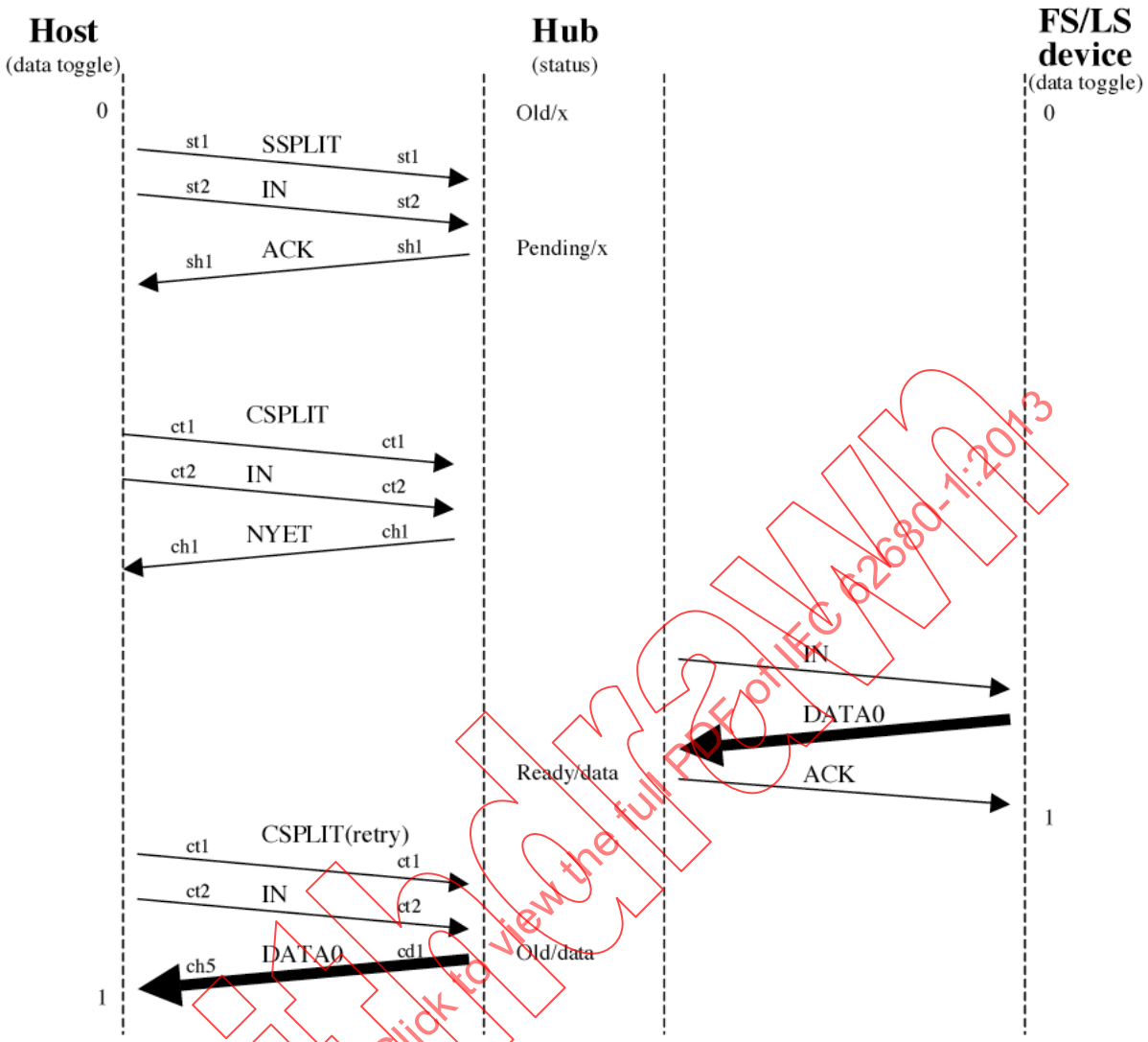


Figure A-42. CS Earlier No Smash(HS NYET)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

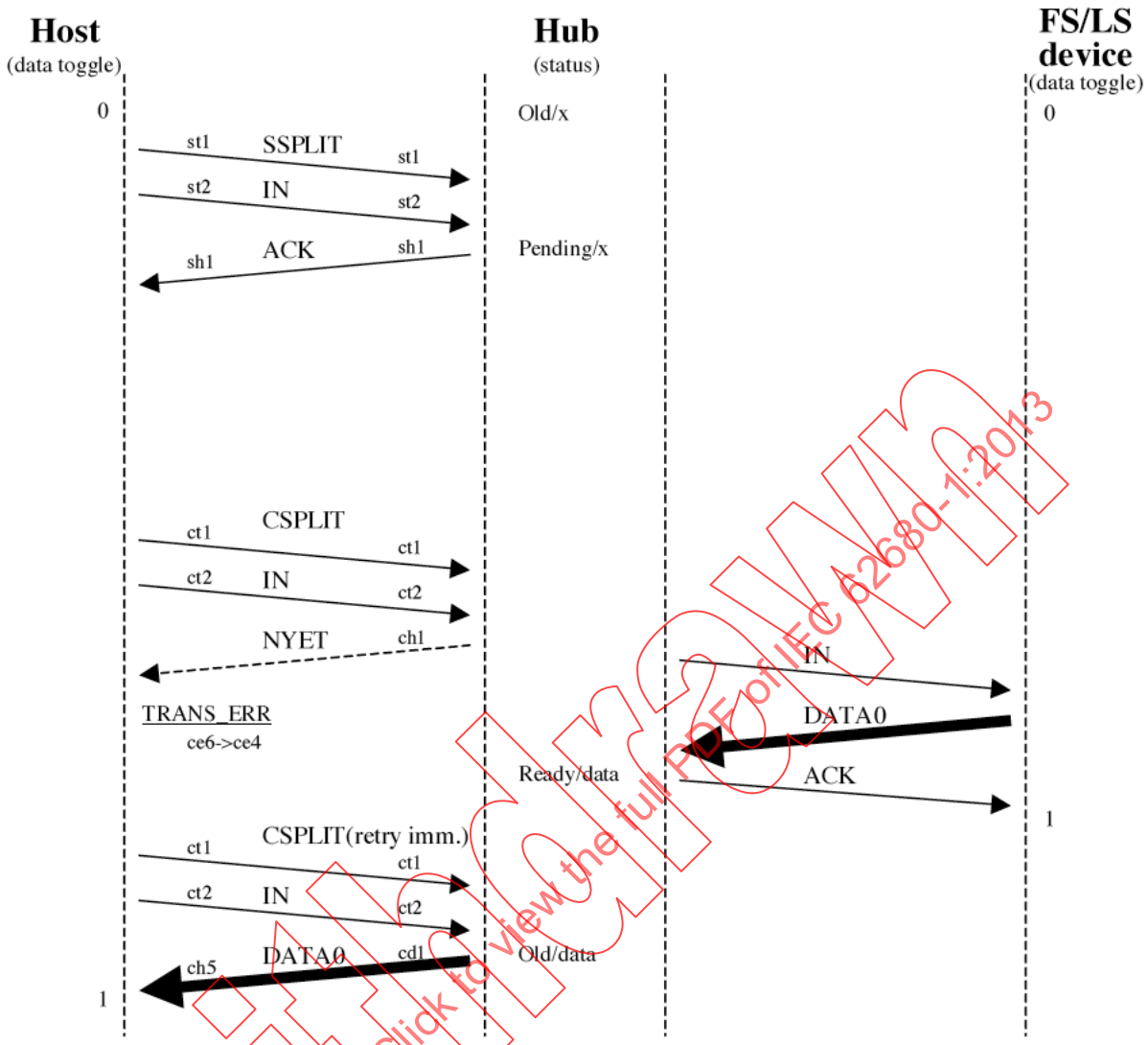


Figure A-43. CS Earlier HS NYET Smash(case 1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

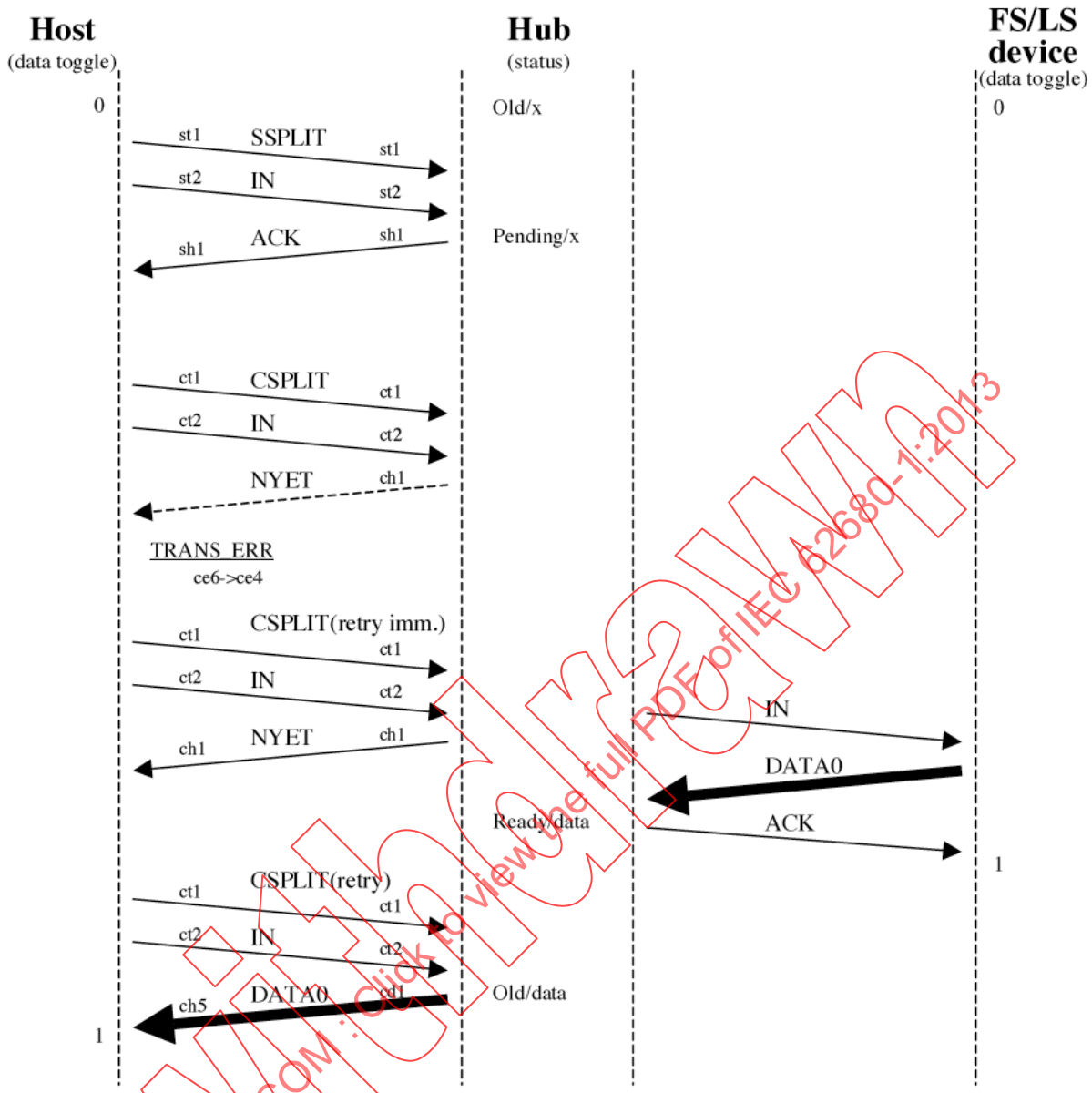


Figure A-44. CS Earlier HS NYET Smash(case 2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

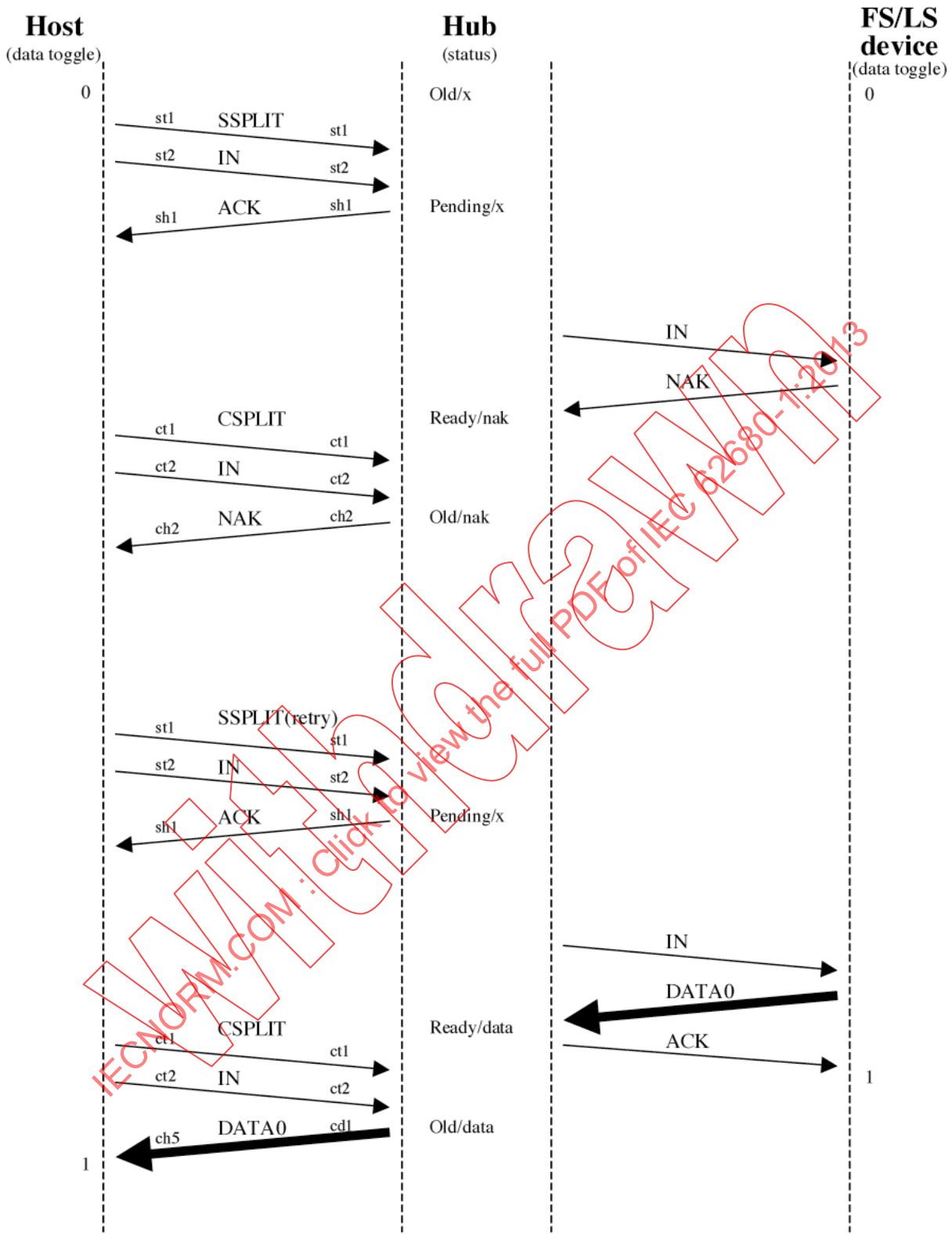


Figure A-45. Device Busy No Smash(FS/LS NAK)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

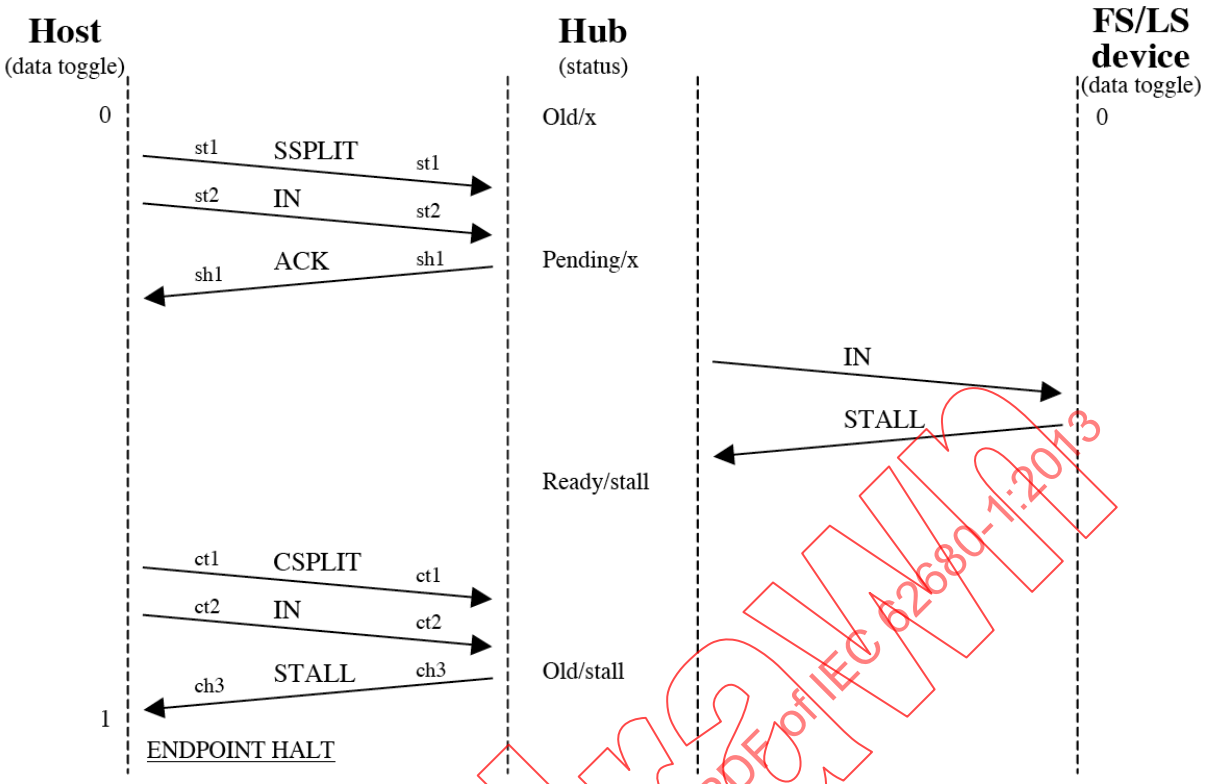


Figure A-46. Device Stall No Smash(FS/LS STALL)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A.3 Interrupt OUT Transaction Examples

Legend:

(S): Start Split

(C): Complete Split

Summary of cases for Interrupt OUT transaction

- Normal cases

Case	Reference Figure	Similar Figure
No smash (FS/LS handshake packet is done by M+1)	Figure A-47	
HS SSPLIT smash		Figure A-48
HS SSPLIT 3 strikes smash	No figure	
HS OUT(S) smash		Figure A-48
HS OUT(S) 3 strikes smash	No figure	
HS DATA0/1 smash	Figure A-48	
HS DATA0/1 3 strikes smash	No figure	
HS CSPLIT smash	Figure A-49	
HS CSPLIT 3 strikes smash	Figure A-50	
HS OUT(C) smash		Figure A-49
HS OUT(C) 3 strikes		Figure A-50
HS ACK(C) smash	Figure A-51	
HS ACK(C) 3 strikes smash	Figure A-52	
FS/LS OUT smash		Figure A-53
FS/LS OUT 3 strikes smash	No figure	
FS/LS DATA0/1 smash	Figure A-53	
FS/LS DATA0/1 3 strikes smash	No figure	
FS/LS ACK smash	Figure A-54	
FS/LS ACK 3 strikes smash	No figure	

- Searcing

Case	Reference Figure	Similar Figure
No smash	Figure A-55	

- CS(Complete-split transaction) earlier cases

Case	Reference Figure	Similar Figure
No smash (HS NYET and FS/LS handshake packet is done by M+2)	Figure A-56	
No smash(HS NYET and FS/LS handshake packet is done by M+3)	Figure A-57	
HS NYET smash	Figure A-58	
HS NYET 3 strikes smash	Figure A-59	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- Abort and Free cases

Case	Reference Figure	Similar Figure
No smash and abort (HS NYETand FS/LS transaction is continued at end of M+3)	Figure A-60	
No smash and free(HS NYETand FS/LS transaction is not started at end of M+3)	Figure A-61	

- FS/LS transaction error cases

Case	Reference Figure	Similar Figure
HS ERR smash		Figure A-51
HS ERR 3 strikes smash		Figure A-52

- Device busy cases

Case	Reference Figure	Similar Figure
No smash(HS NAK(C))	Figure A-62	
HS NAK(C) smash		Figure A-51
HS NAK(C) 3 strikes smash		Figure A-52
FS/LS NAK smash		Figure A-53
FS/LS NAK 3 strikes smash	No figure	

- Device stall cases

Case	Reference Figure	Similar Figure
No smash	Figure A-63	
HS STALL(C) smash		Figure A-51
HS STALL(C) 3 strikes smash		Figure A-52
FS/LS STALL smash		Figure A-53
FS/LS STALL 3 strikes smash	No figure	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

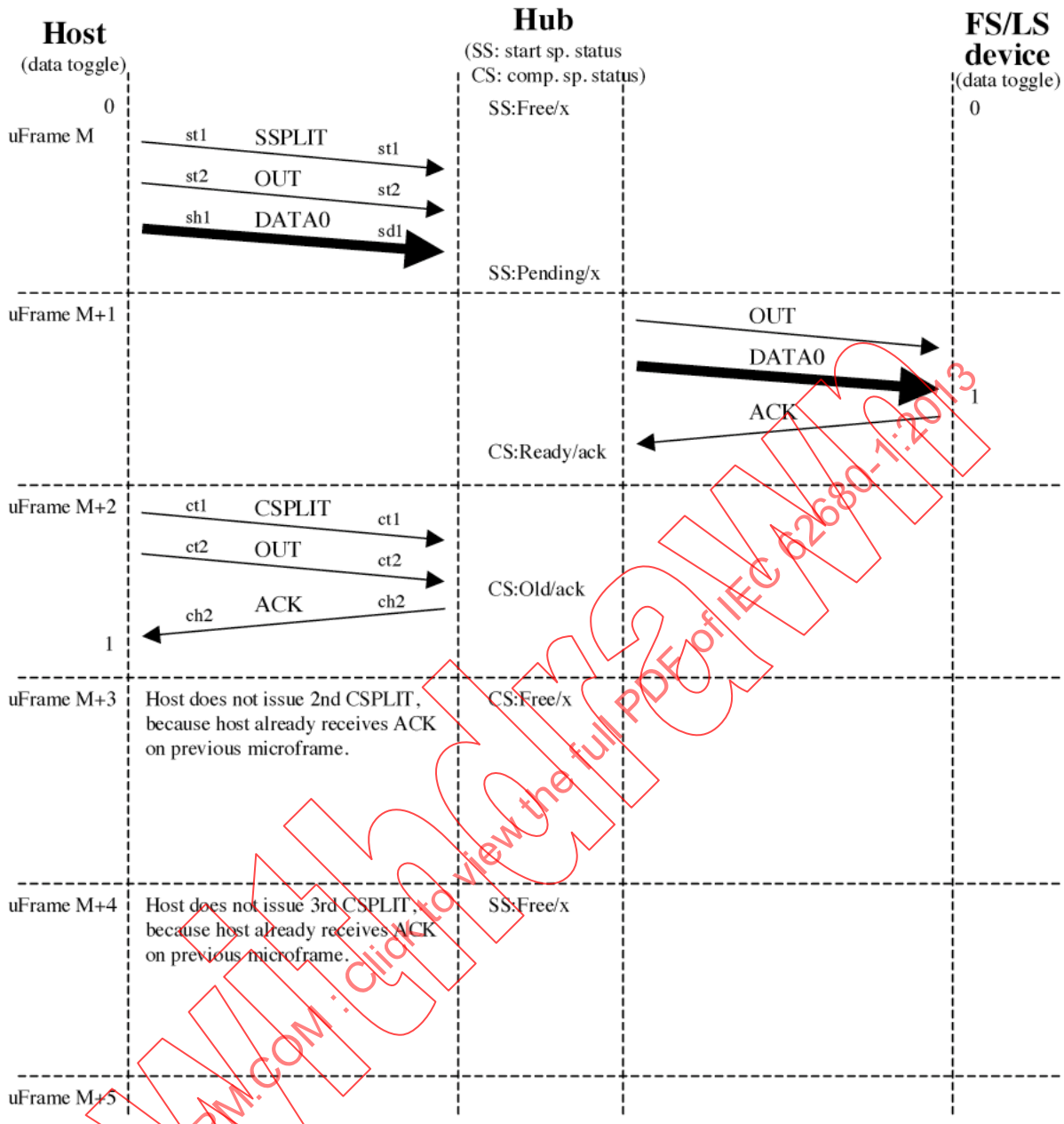


Figure A-47. Normal No Smash(FS/LS Handshake Packet is Done by M+1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

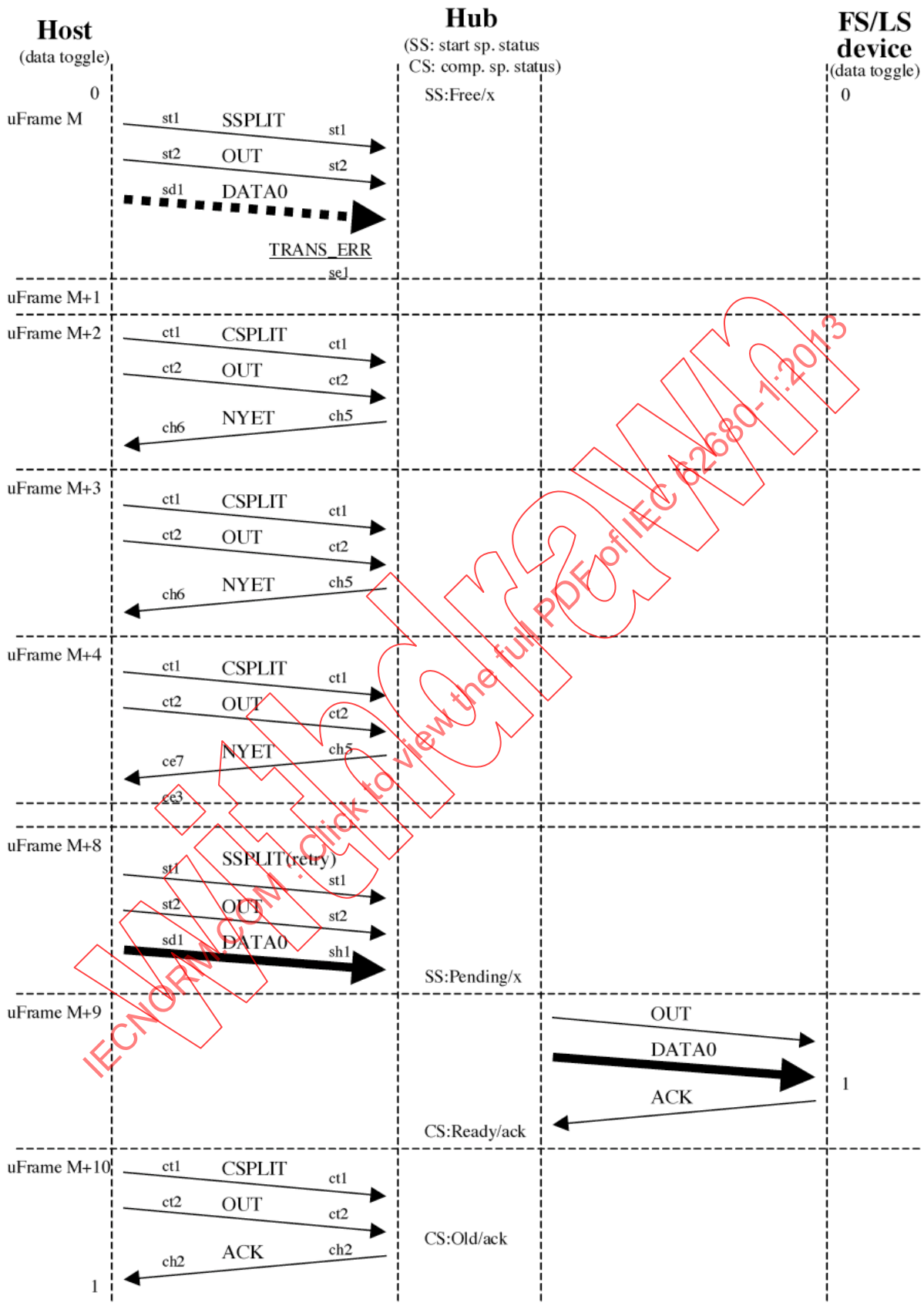


Figure A-48. Normal HS DATA0/1 Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

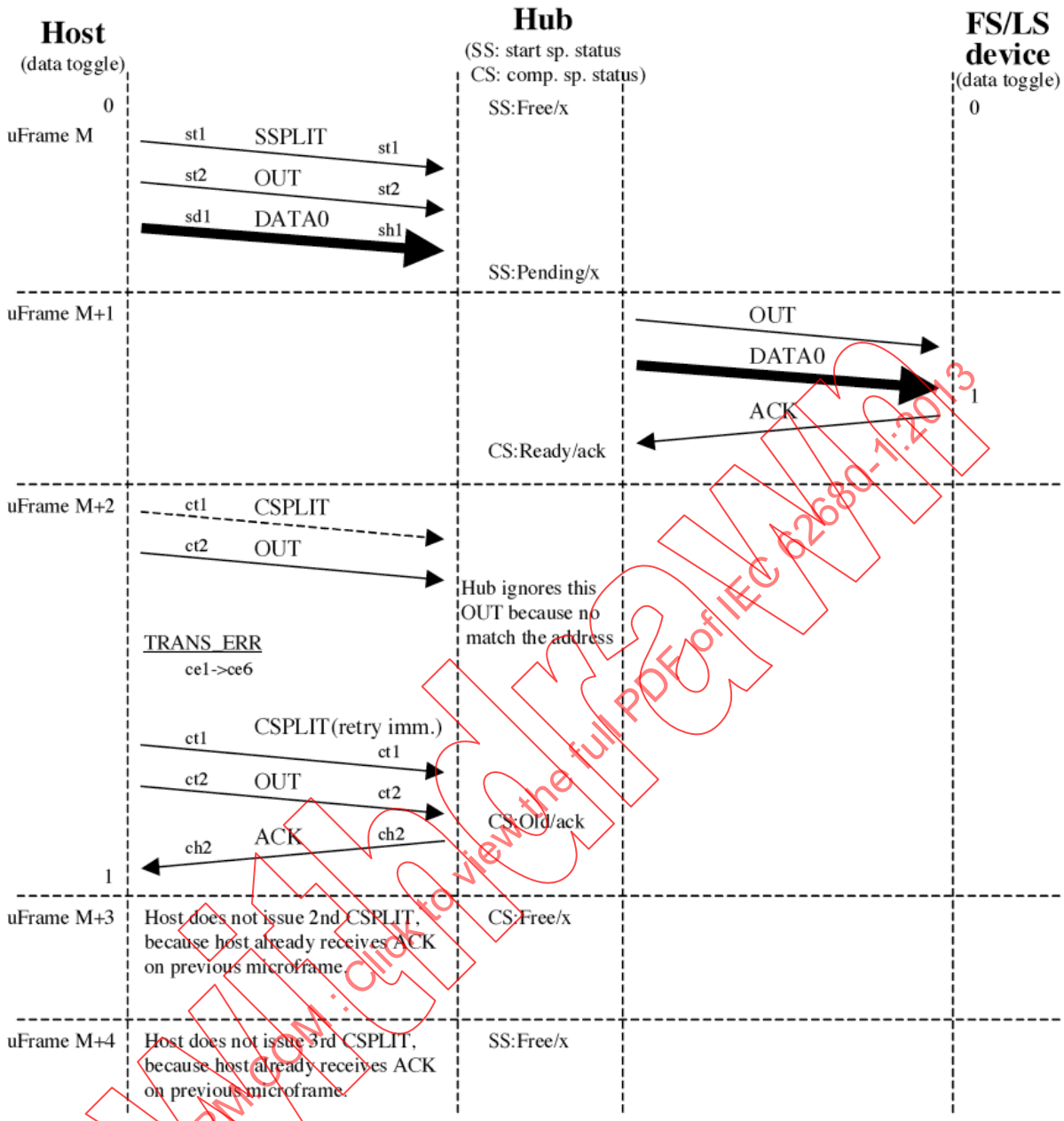


Figure A-49. Normal HS CSPLIT Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

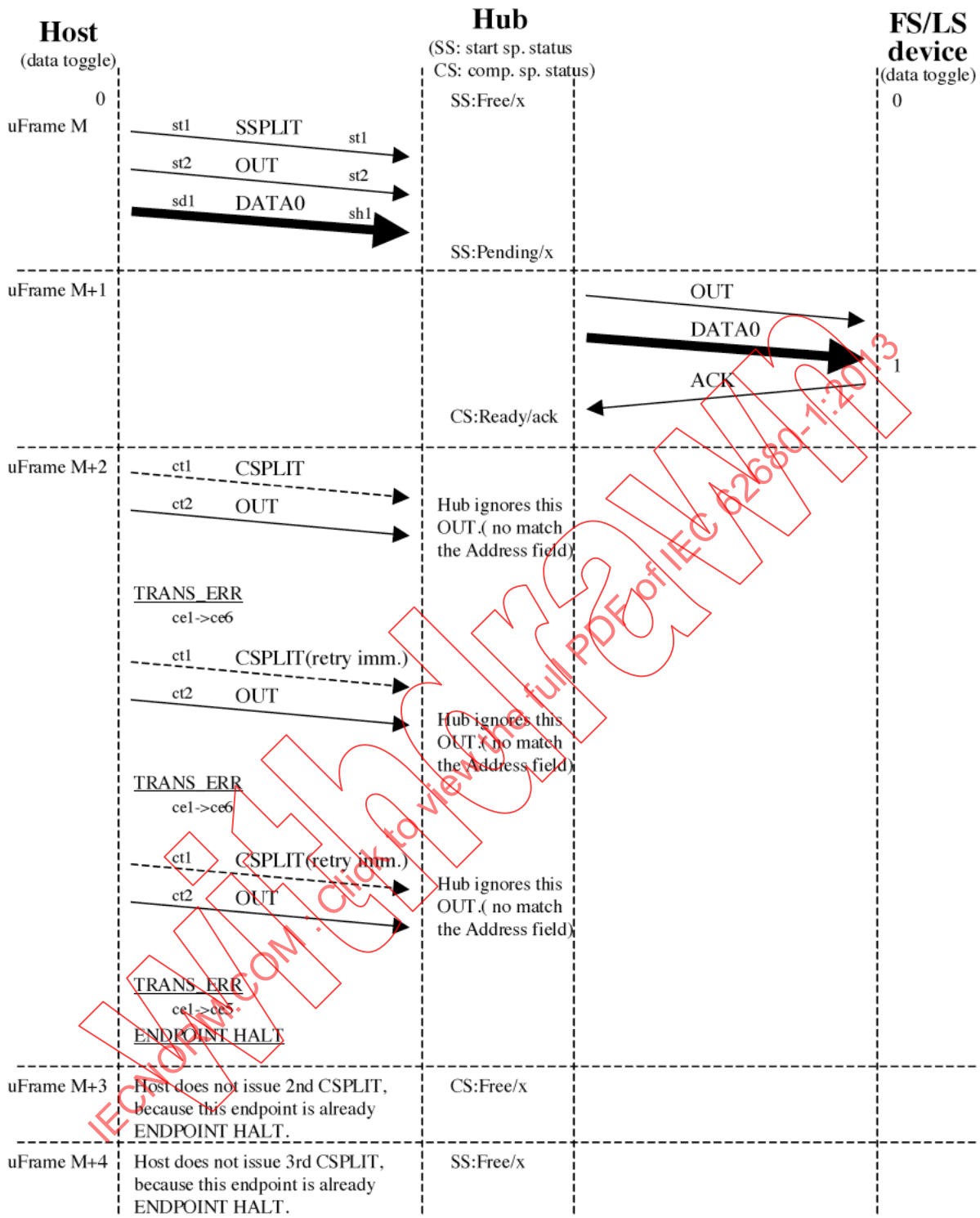


Figure A-50. Normal HS CSPLIT 3 Strikes Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

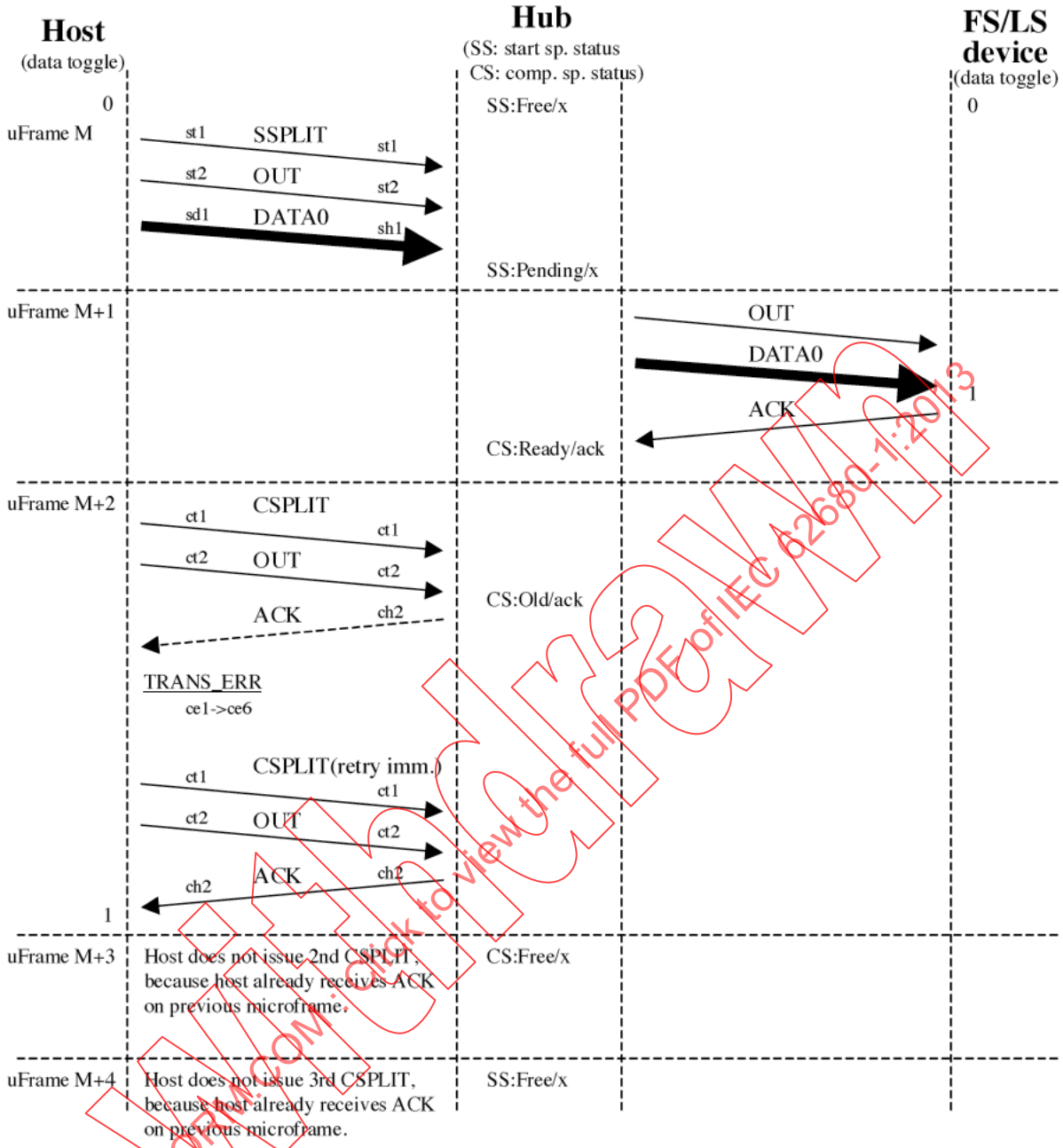
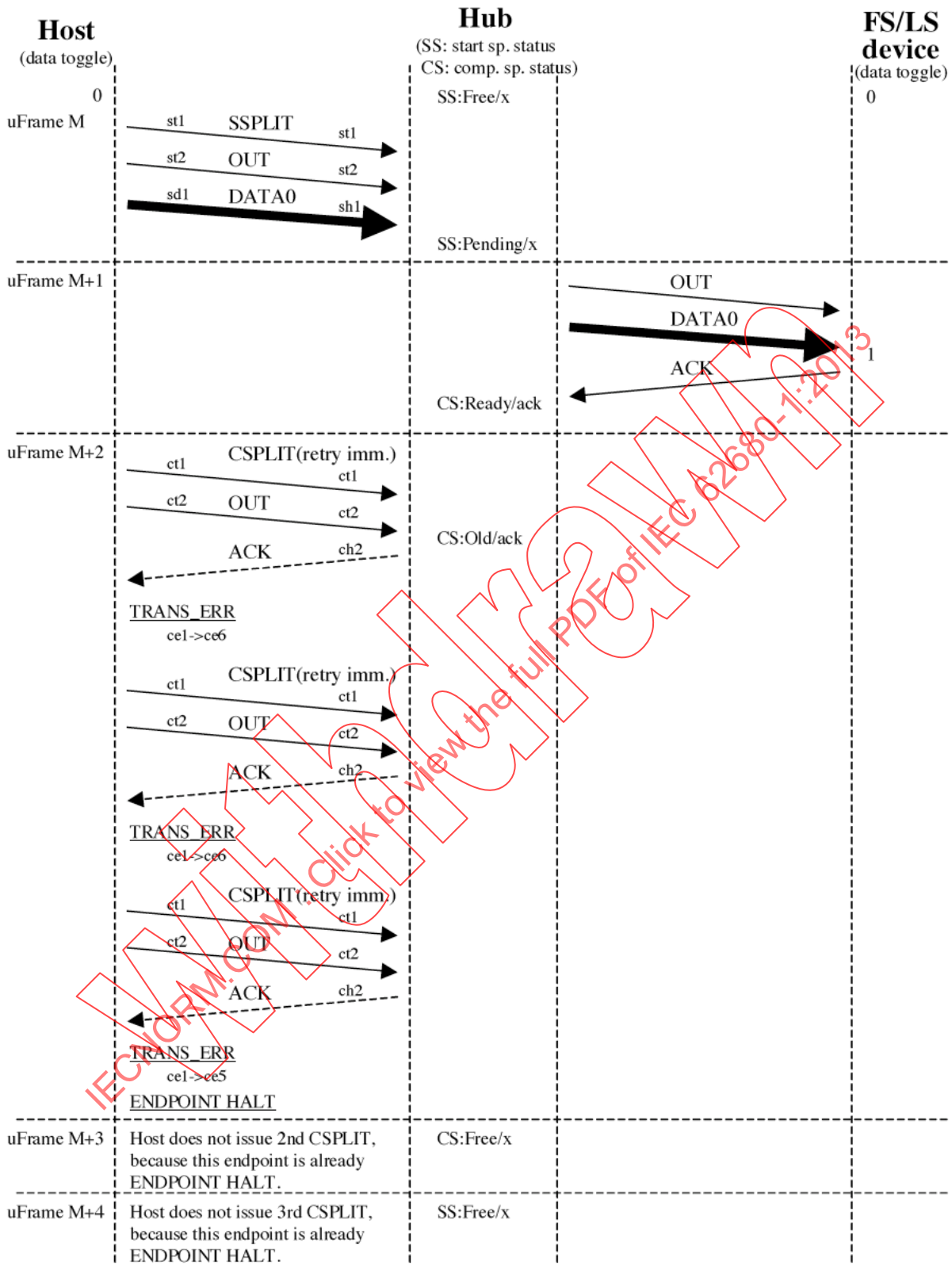


Figure A-51. Normal HS ACK(C) Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure A-52. Normal HS ACK(C) 3 Strikes Smash

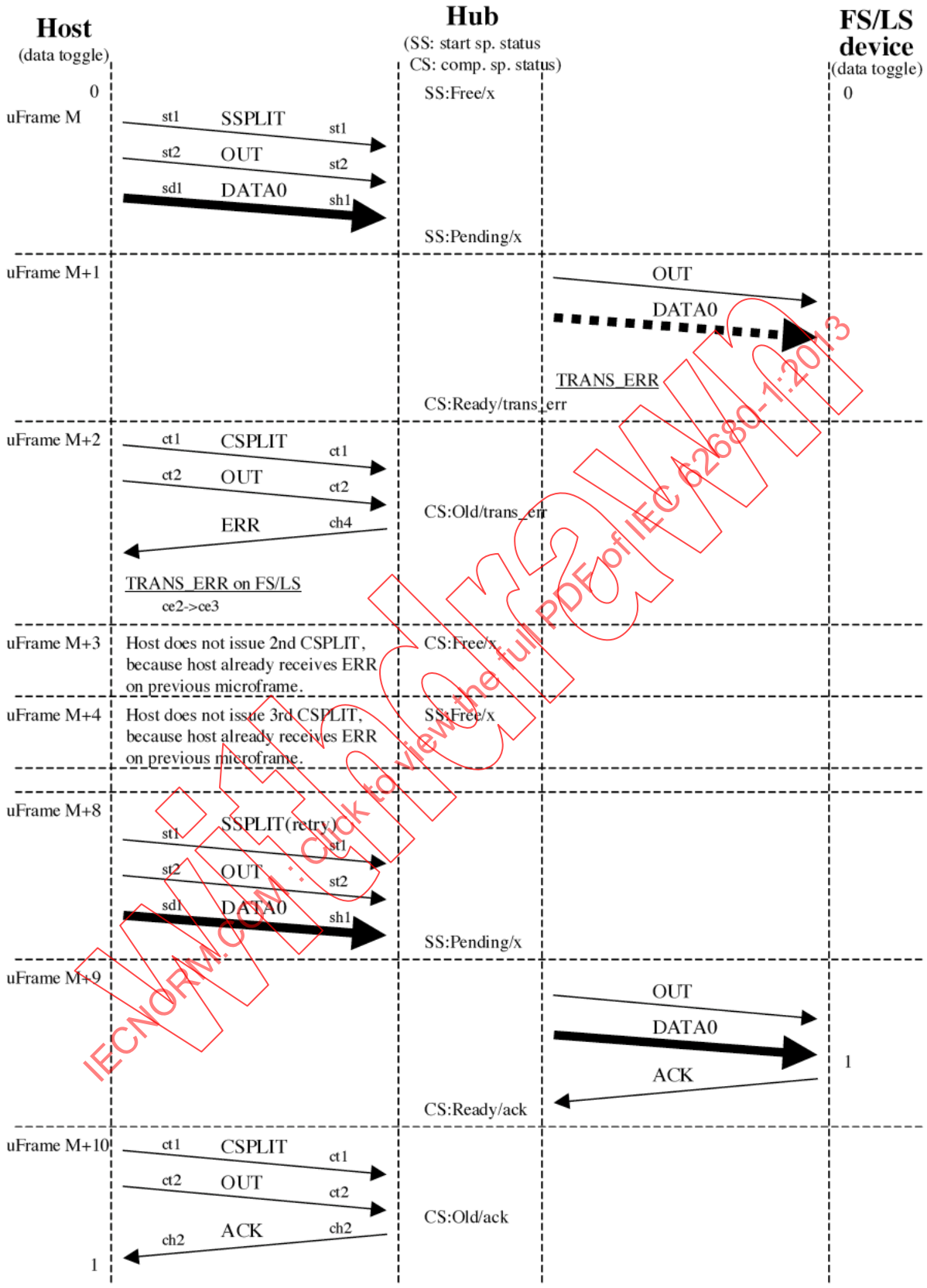
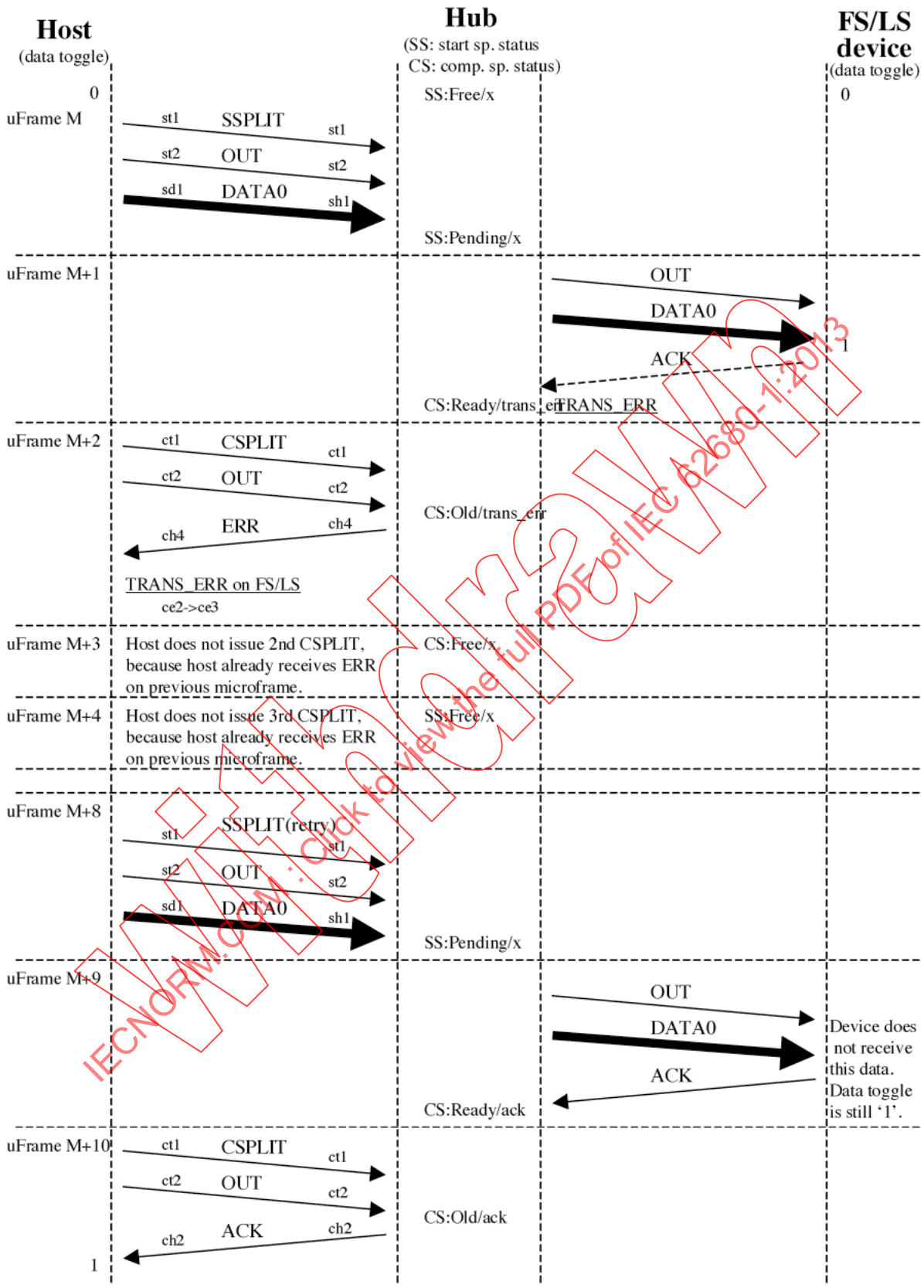


Figure A-53. Normal FS/LS DATA0/1 Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure A-54. Normal FS/LS ACK Smash

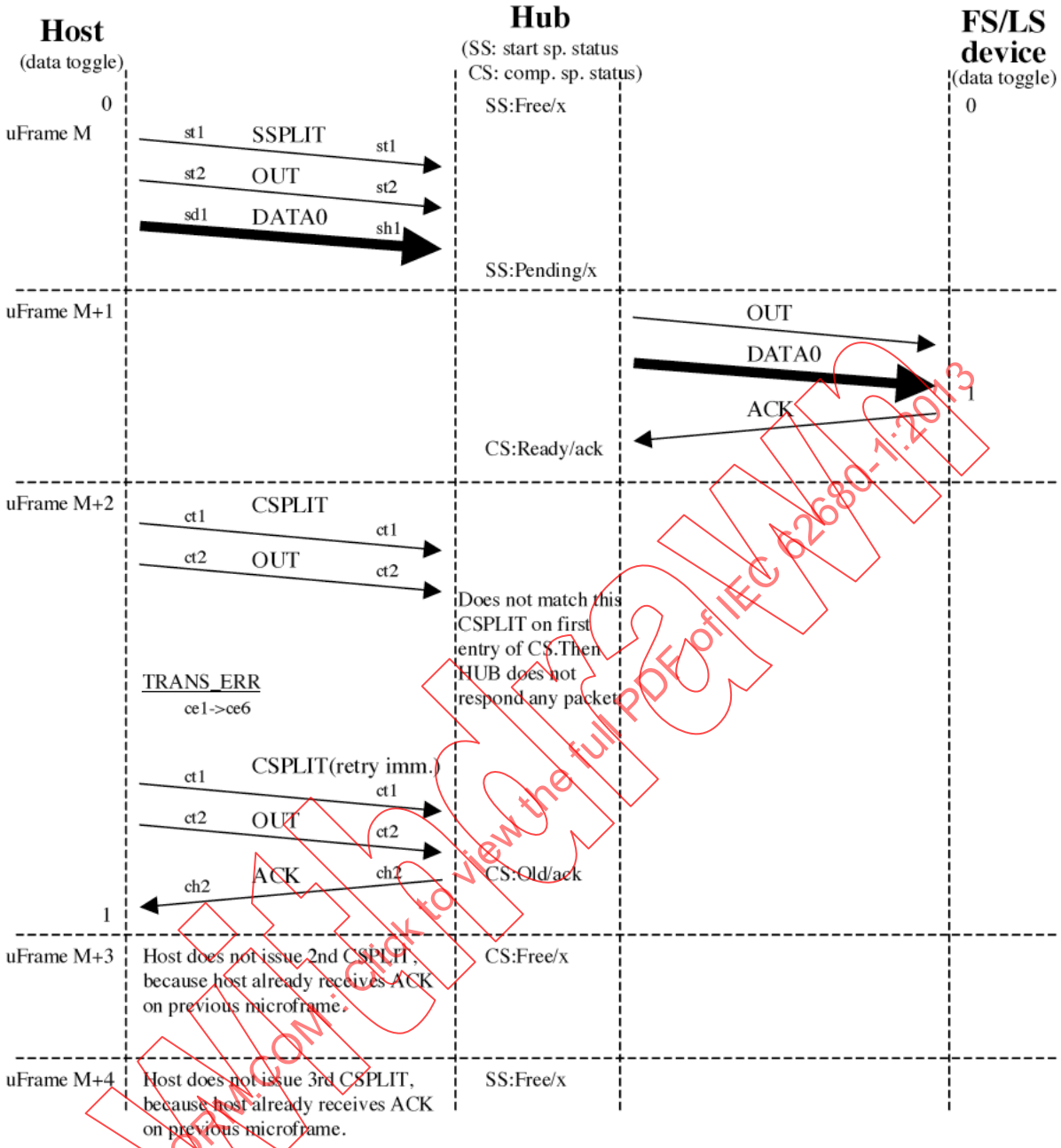


Figure A-55. Searching No Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

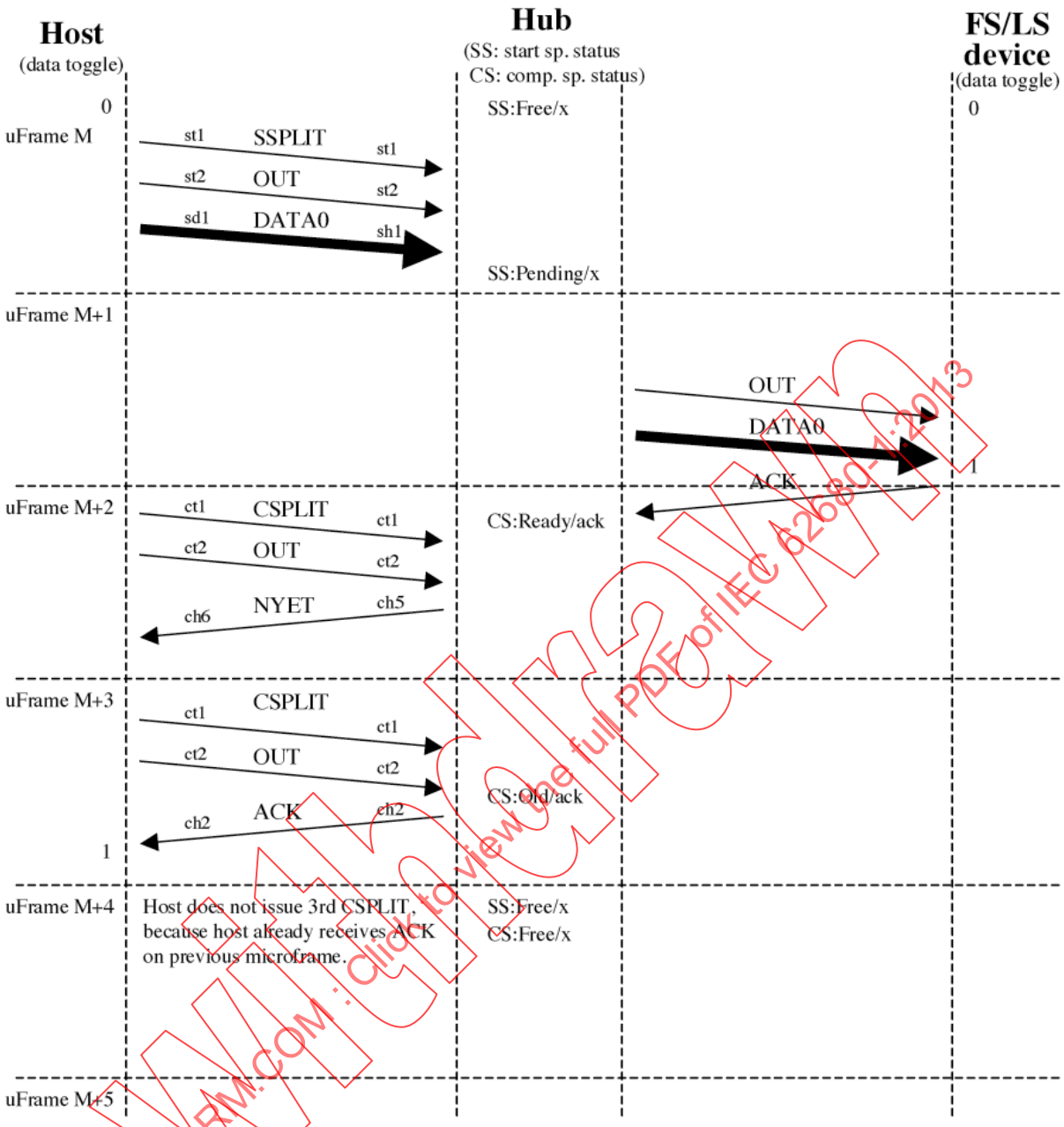


Figure A-56. CS Earlier No Smash(HS NYET and FS/LS Handshake Packet is Done by M+2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

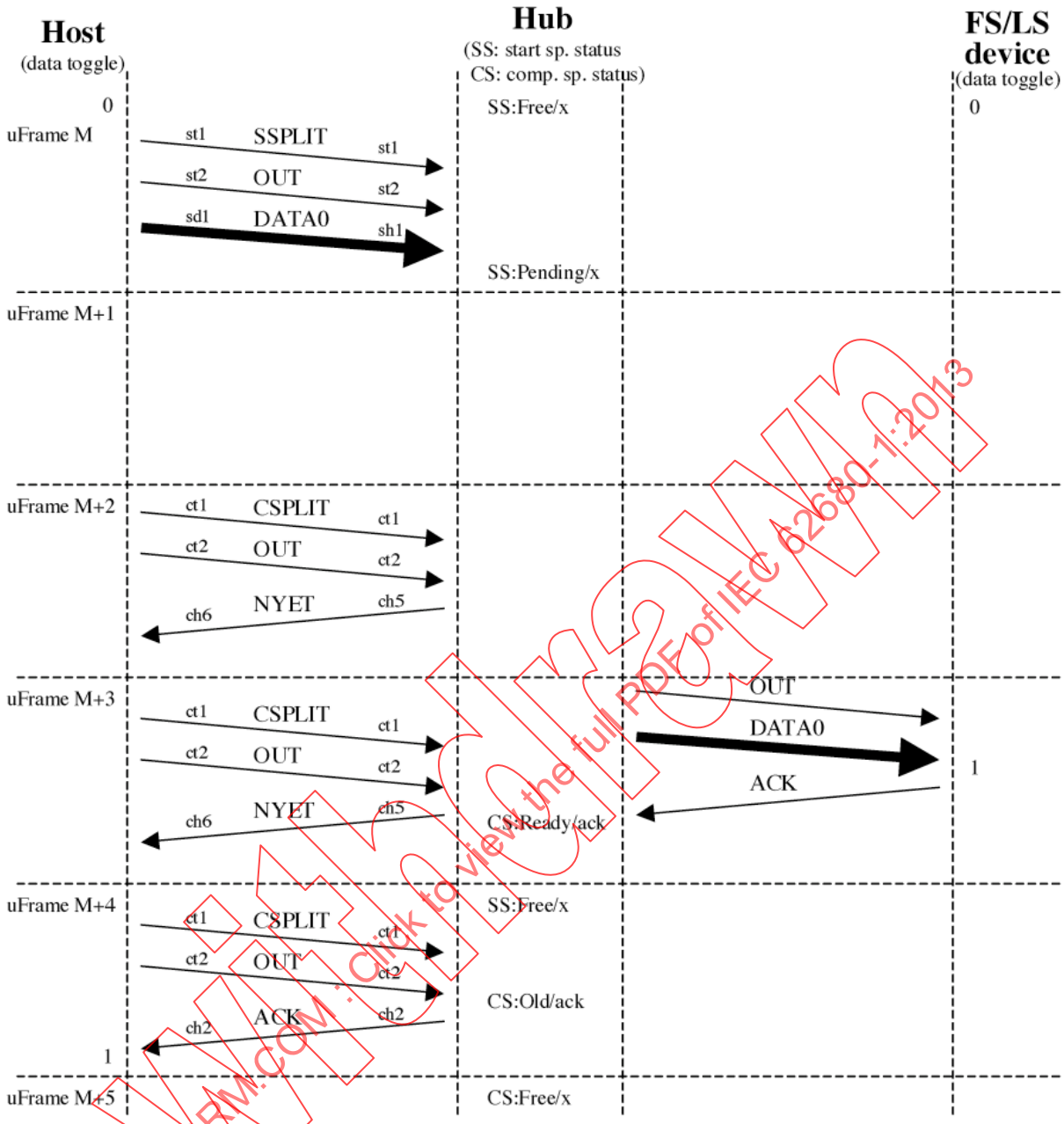


Figure A-57. CS Earlier No Smash(HS NYET and FS/LS Handshake Packet is Done by M+3)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

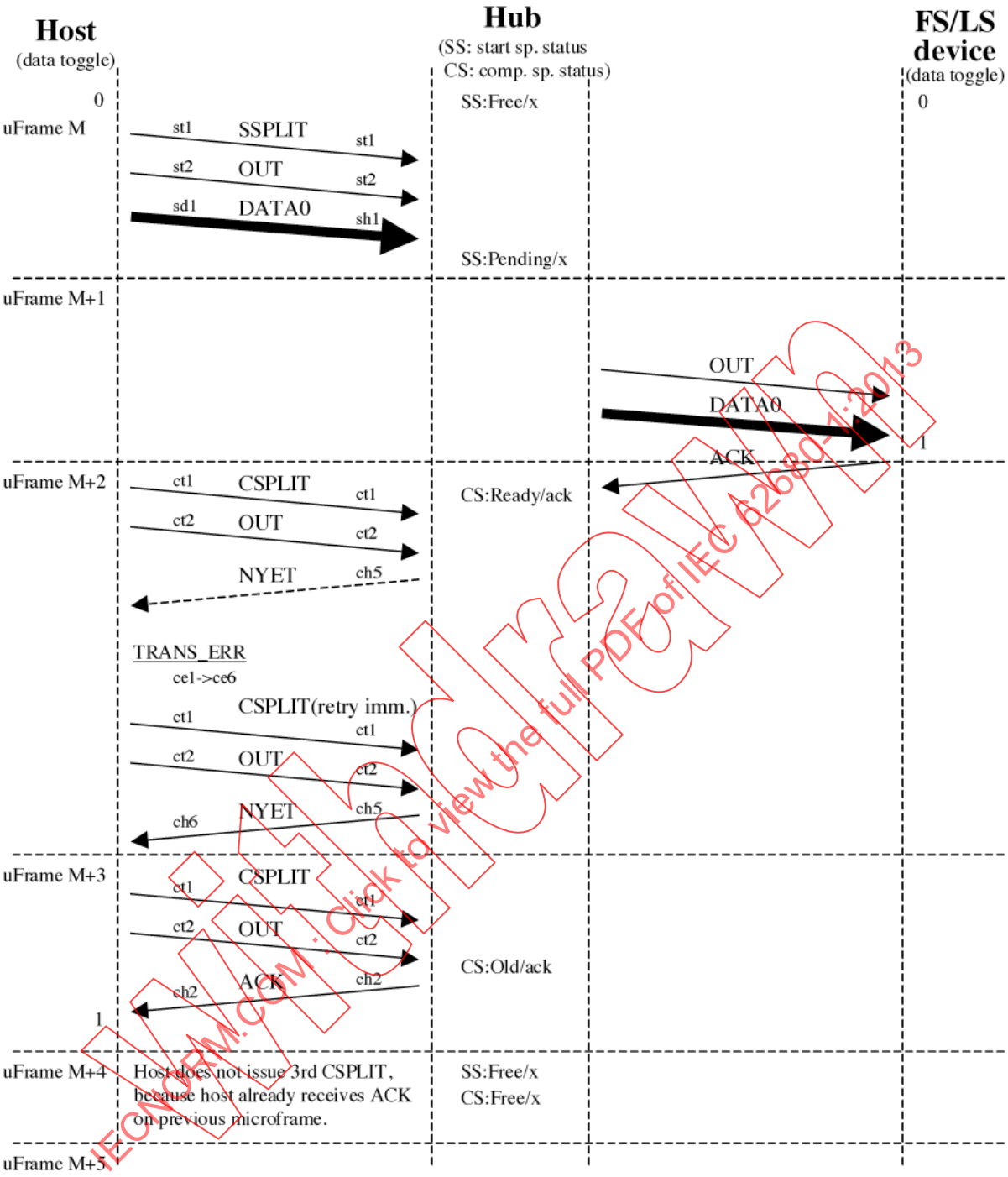
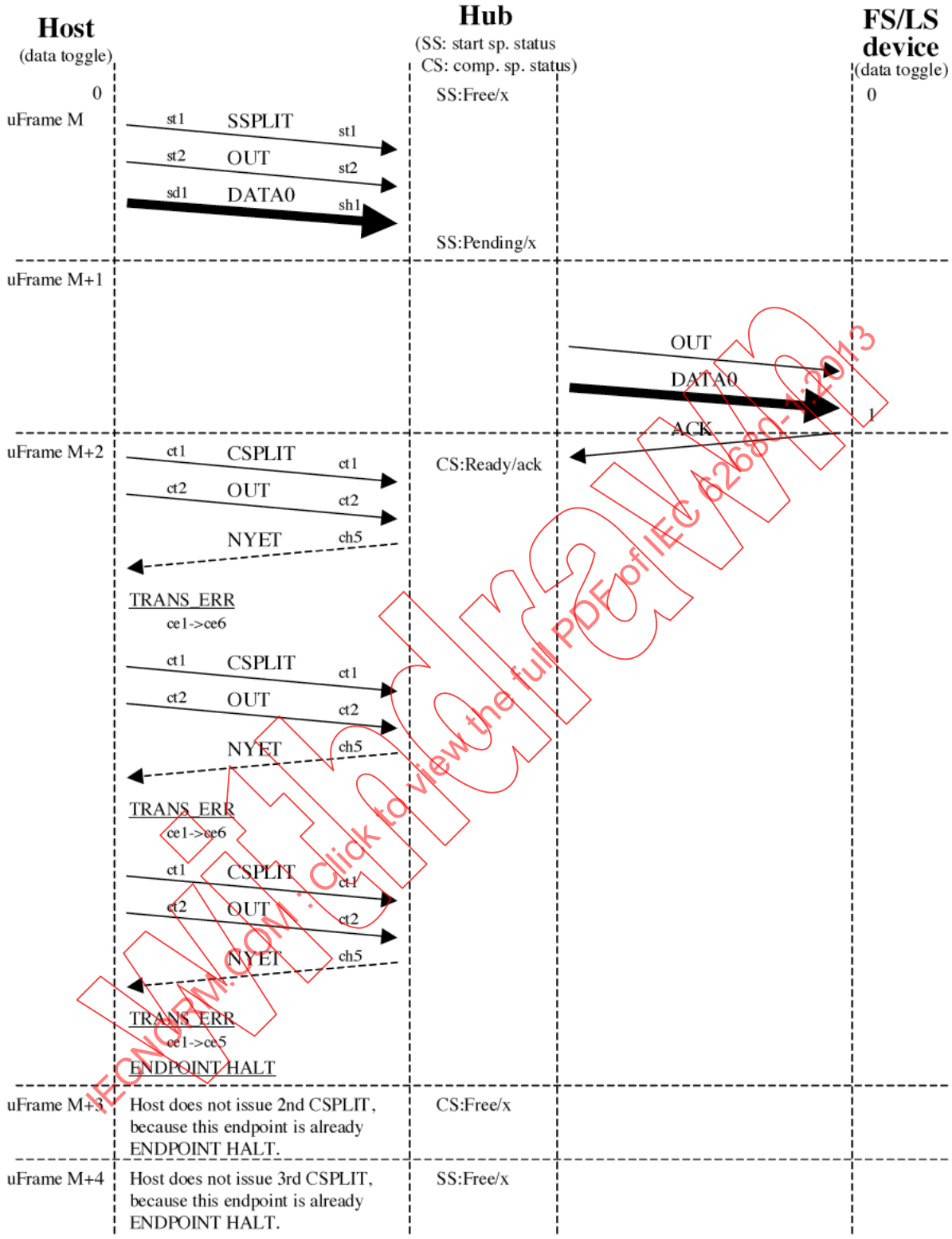


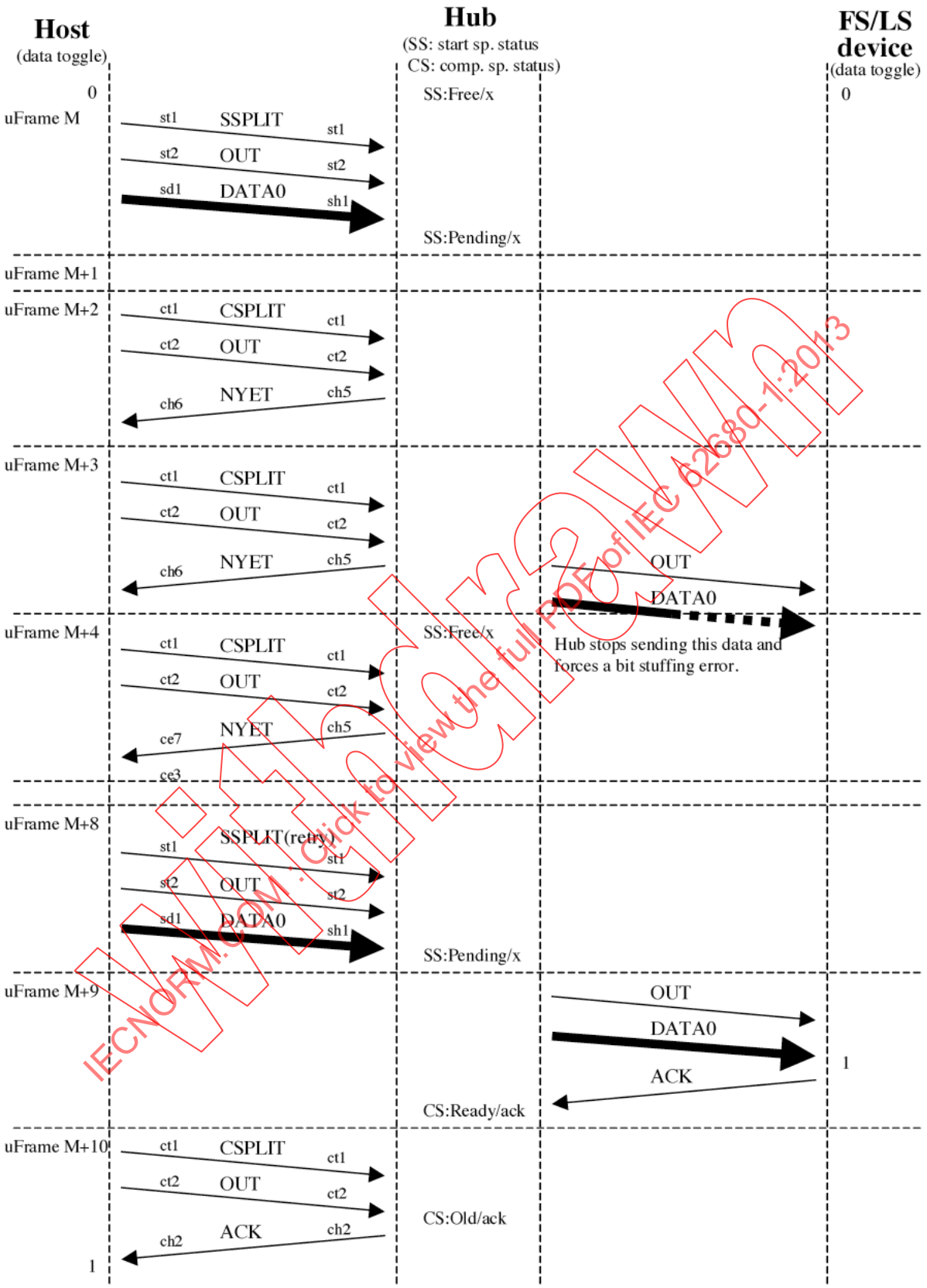
Figure A-58. CS Earlier HS NYET Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



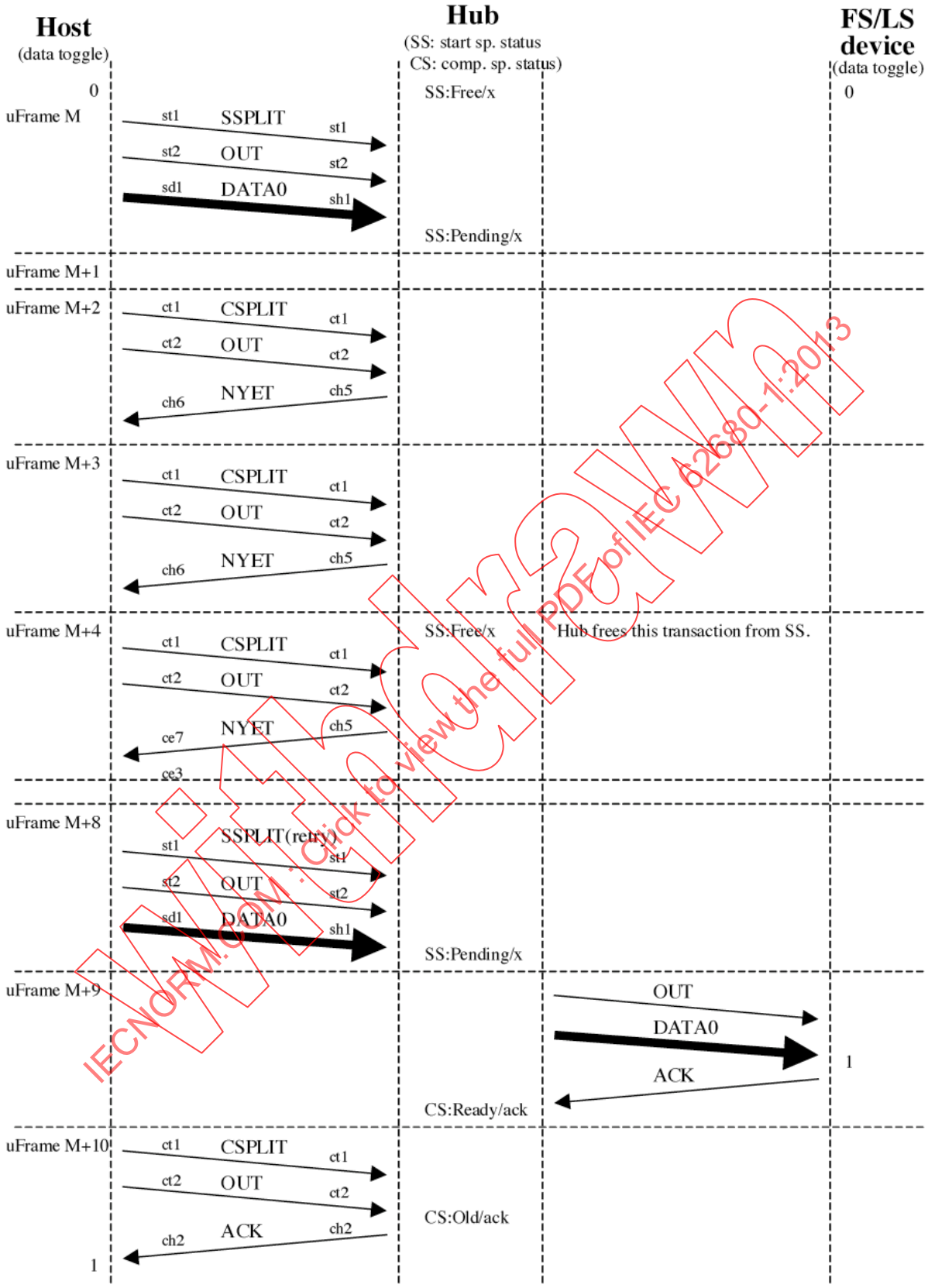
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure A-59. CS Earlier HS NYET 3 Strikes Smash



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure A-60. Abort and Free Abort(FS/LS Transaction is Continued at End of M+3)



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure A-61. Abort and Free Free(FS/LS Transaction is not Started at End of M+3)

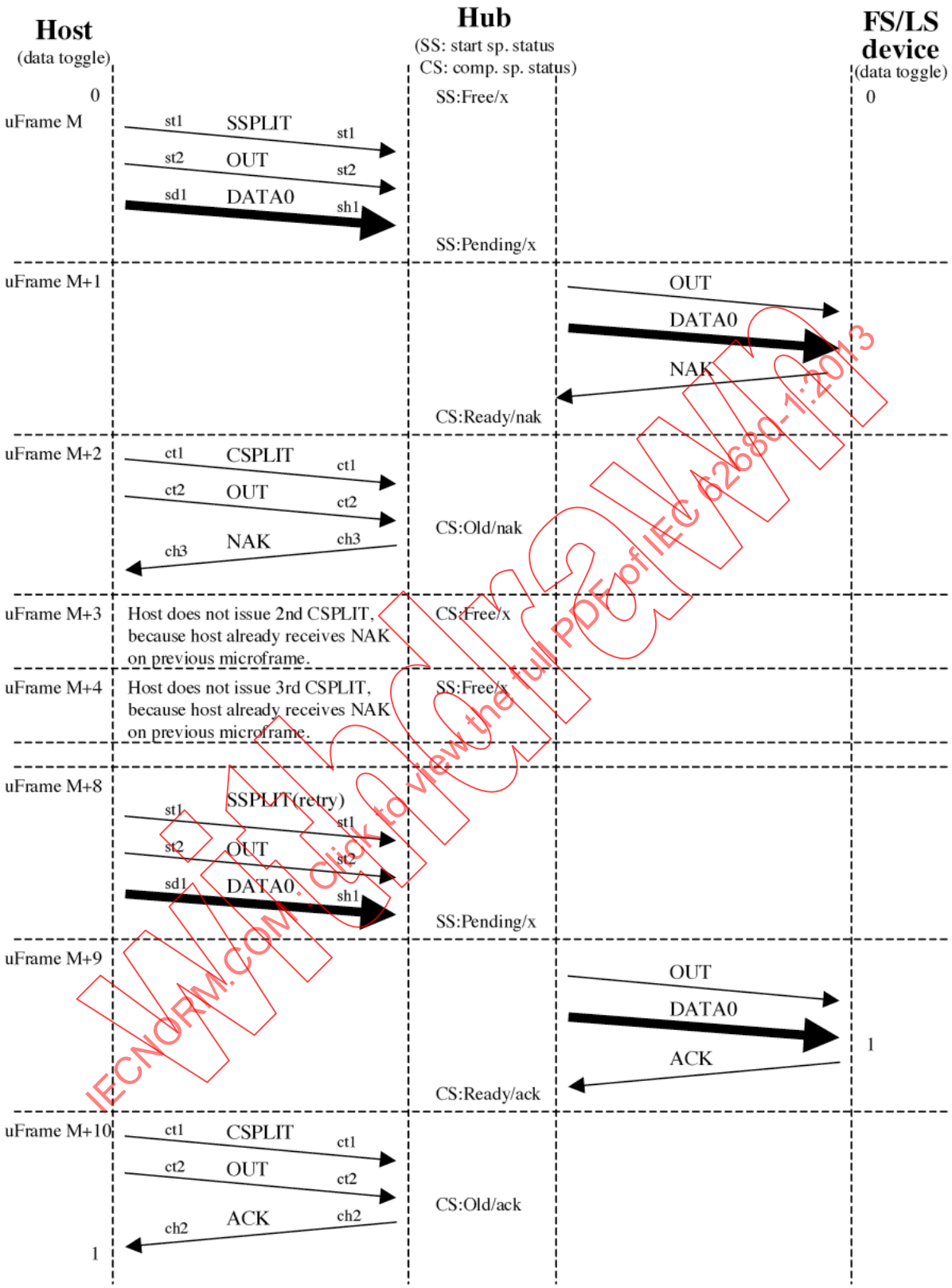


Figure A-62. Device Busy No Smash(FS/LS NAK)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

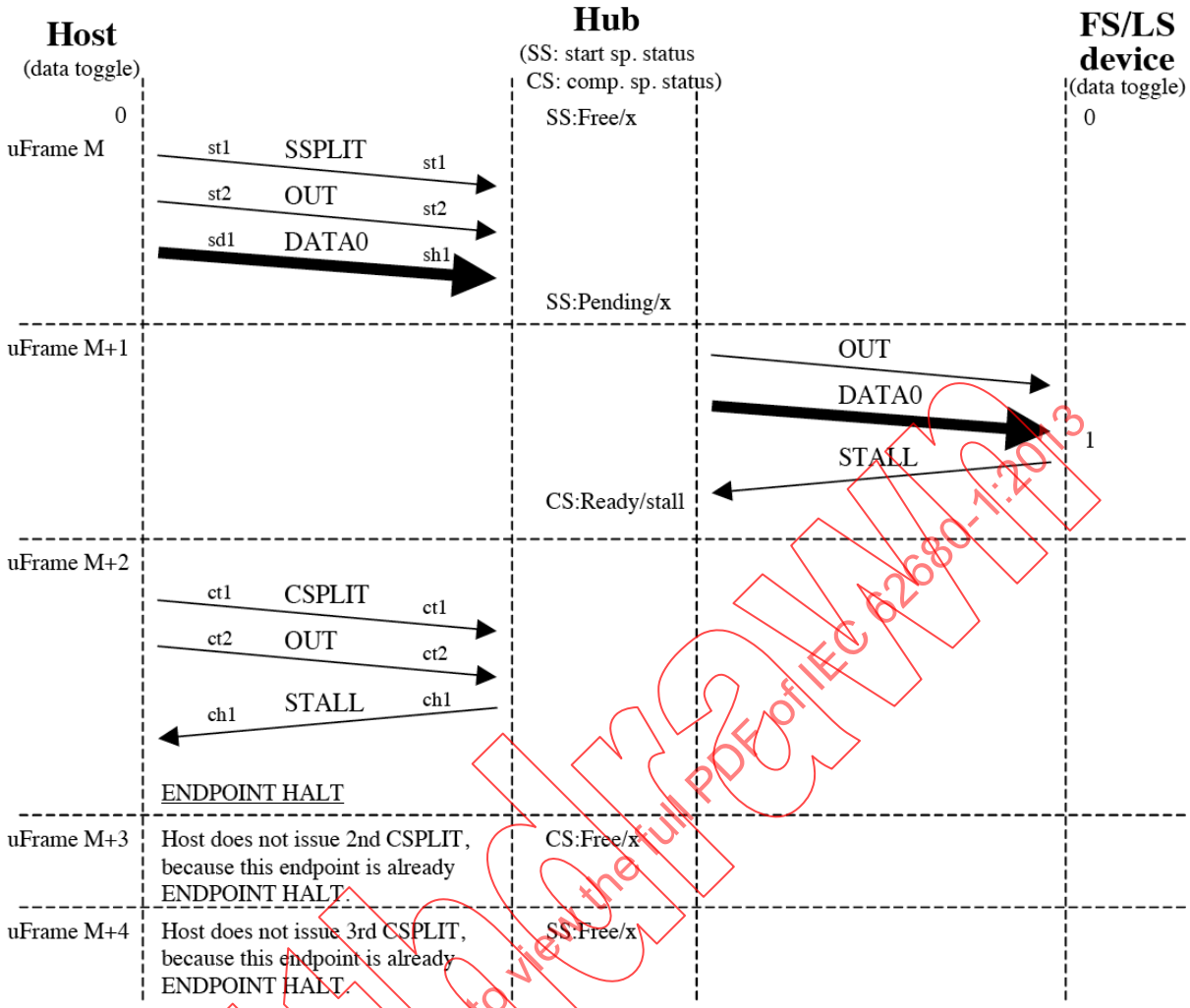


Figure A-63. Device Stall No Smash(FS/LS STALL)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A.4 Interrupt IN Transaction Examples

Legend:

(S): Start Split

(C): Complete Split

Summary of cases for Interrupt OUT transaction

- Normal cases

Case	Reference Figure	Similar Figure
No smash (FS/LS handshake packet is on M+1)	Figure A-64	
HS SSPLIT smash	Figure A-65	
HS SSPLIT 3 strikes smash	No figure	
HS IN(S) smash		Figure A-65
HS IN(S) 3 strikes smash	No figure	
HS CSPLIT smash	Figure A-66	
HS CSPLIT 3 strikes smash	Figure A-67	
HS IN(C) smash		Figure A-66
HS IN(C) 3 strikes		Figure A-67
HS DATA0/1 smash	Figure A-68	
HS DATA0/1 3 strikes smash	Figure A-69	
FS/LS IN smash	Figure A-68	
FS/LS IN 3 strikes smash	No figure	
FS/LS DATA0/1 smash	Figure A-71	
FS/LS DATA0/1 3 strikes smash	No figure	
FS/LS ACK smash	Figure A-72	
FS/LS ACK 3 strikes smash	No figure	

- Searcing

Case	Reference Figure	Similar Figure
No smash	Figure A-73	

- CS(Complete-split transaction) earlier cases

Case	Reference Figure	Similar Figure
No smash (HS MDATA and FS/LS transaction is on M+1 and M+2)	Figure A-74	
No smash(HS NYET and FS/LS transaction is on M+2)	Figure A-75	
No smash (HS NYET and MDATA and FS/LS transaction is on M+2 and M+3)	Figure A-76	
No smash (HS NYET and FS/LS transaction is on M+3)	Figure A-77	
HS NYET smash	Figure A-78	
HS NYET 3 strikes smash	Figure A-79	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- Abort and Free cases

Case	Reference Figure	Similar Figure
No smash and abort (HS NYET and FS/LS transaction is continued at end of M+3)	Figure A-80	
No smash and free(HS NYET and FS/LS transaction is not started at end of M+3)	Figure A-81	

- FS/LS transaction error cases

Case	Reference Figure	Similar Figure
HS ERR smash		Figure A-68
HS ERR 3 strikes smash		Figure A-69

- Device busy cases

Case	Reference Figure	Similar Figure
No smash(HS NAK(C))	Figure A-82	
HS NAK(C) smash		Figure A-68
HS NAK(C) 3 strikes smash		Figure A-69
FS/LS NAK smash		Figure A-71
FS/LS NAK 3 strikes smash	No figure	

- Device stall cases

Case	Reference Figure	Similar Figure
No smash	Figure A-83	
HS STALL(C) smash		Figure A-68
HS STALL(C) 3 strikes smash		Figure A-69
FS/LS STALL smash		Figure A-71
FS/LS STALL 3 strikes smash	No figure	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

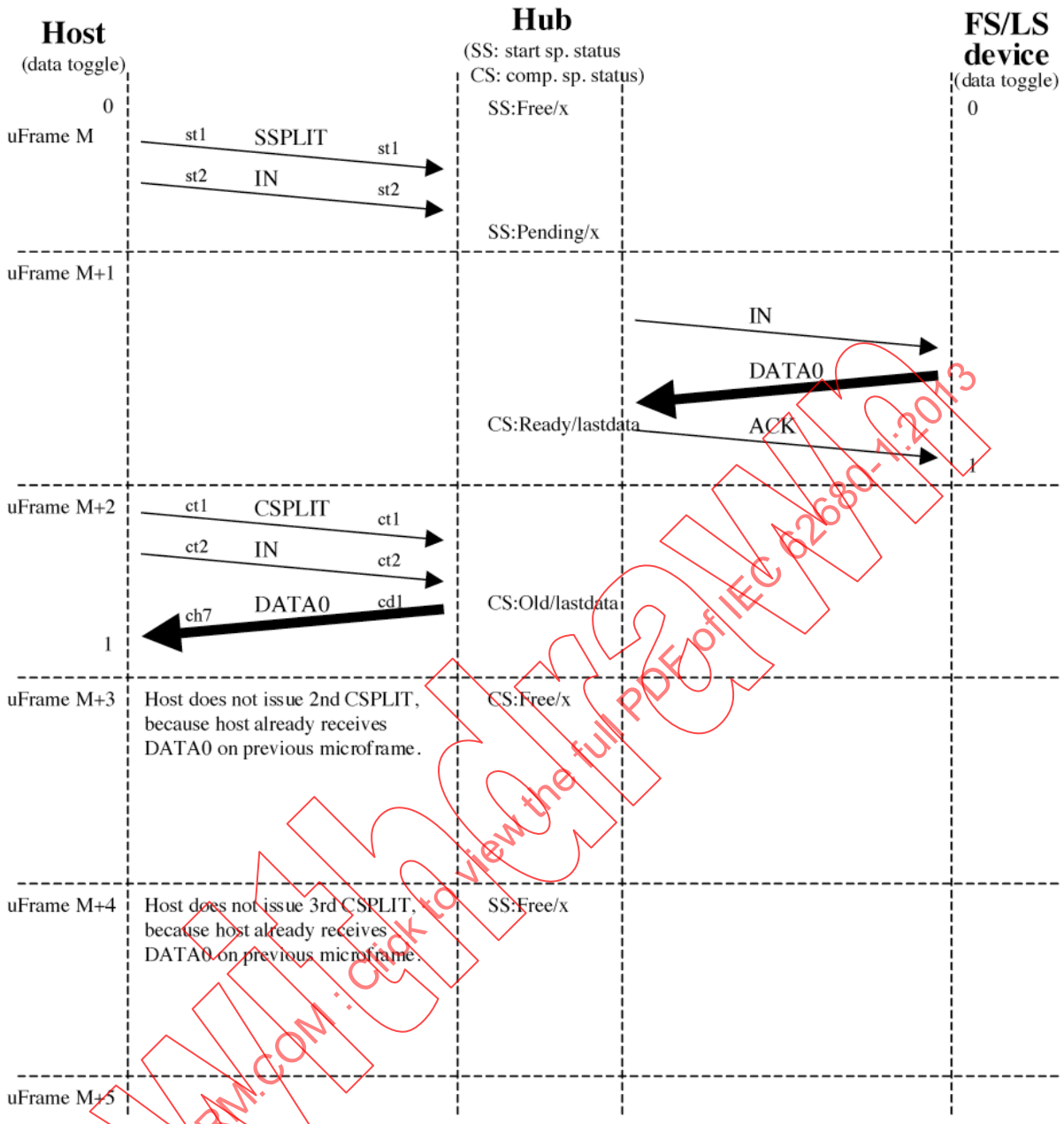
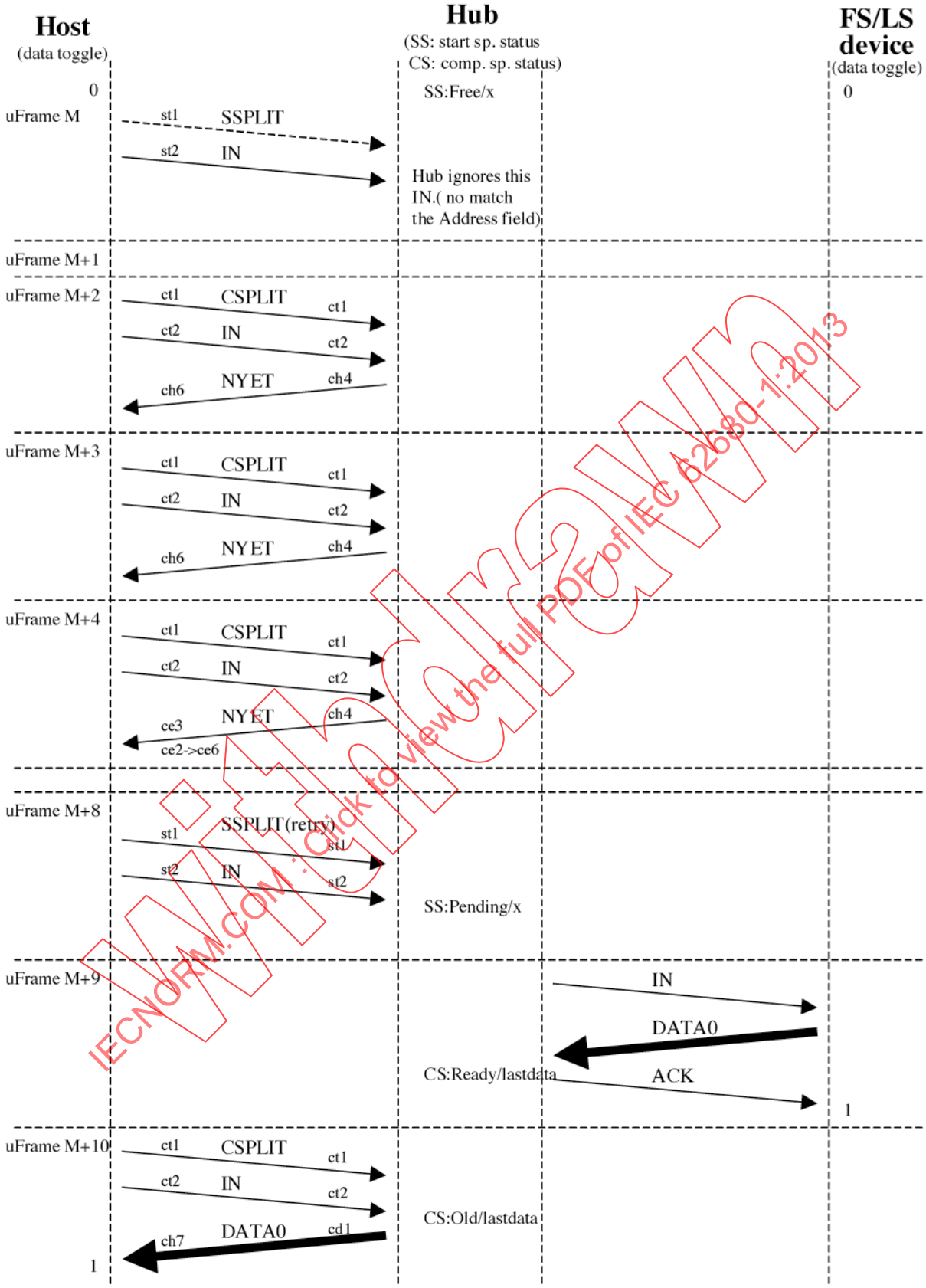


Figure A-64. Normal No Smash(FS/LS Data Packet is on M+1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure A-65. Normal HS SSPLIT Smash

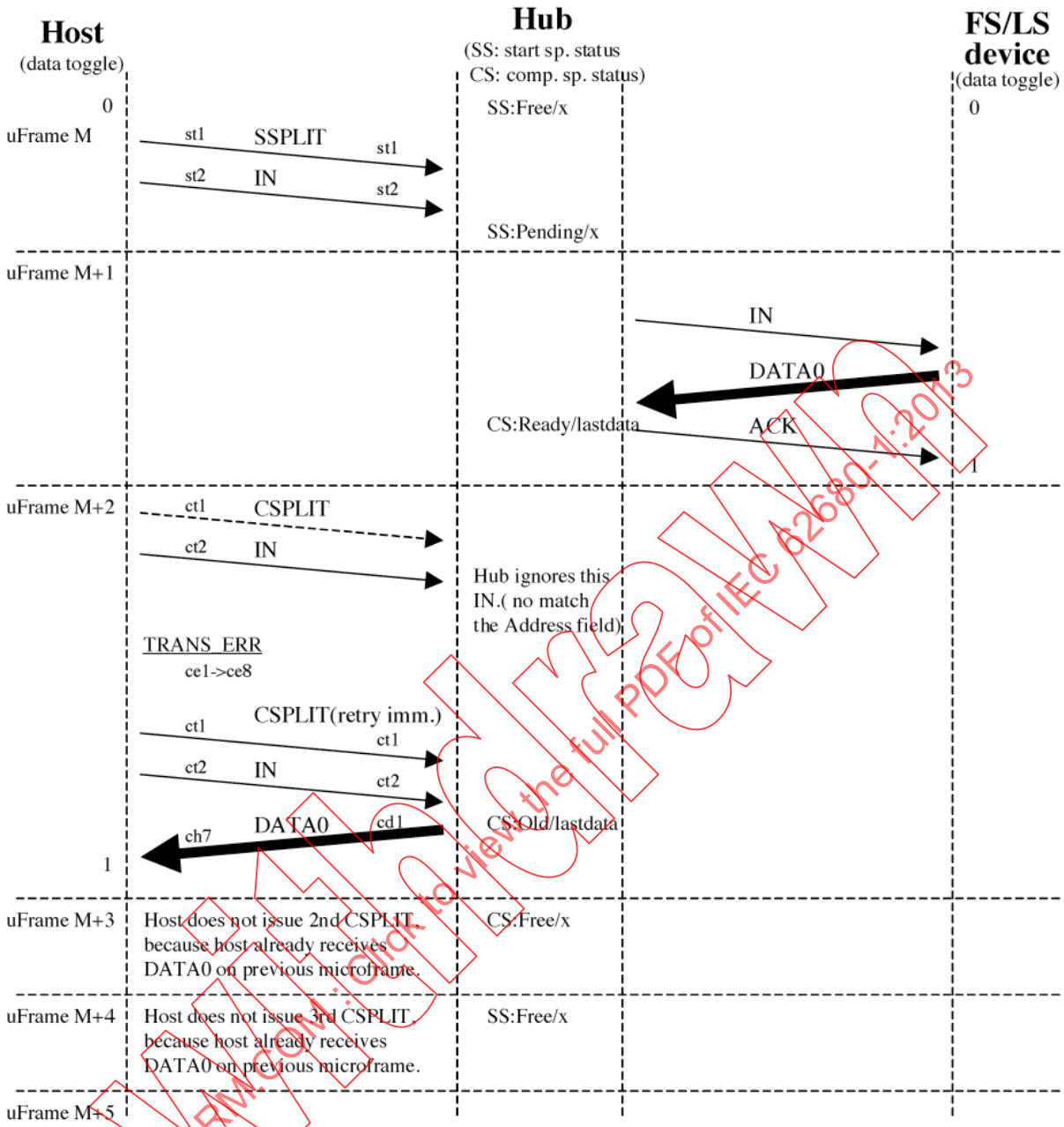
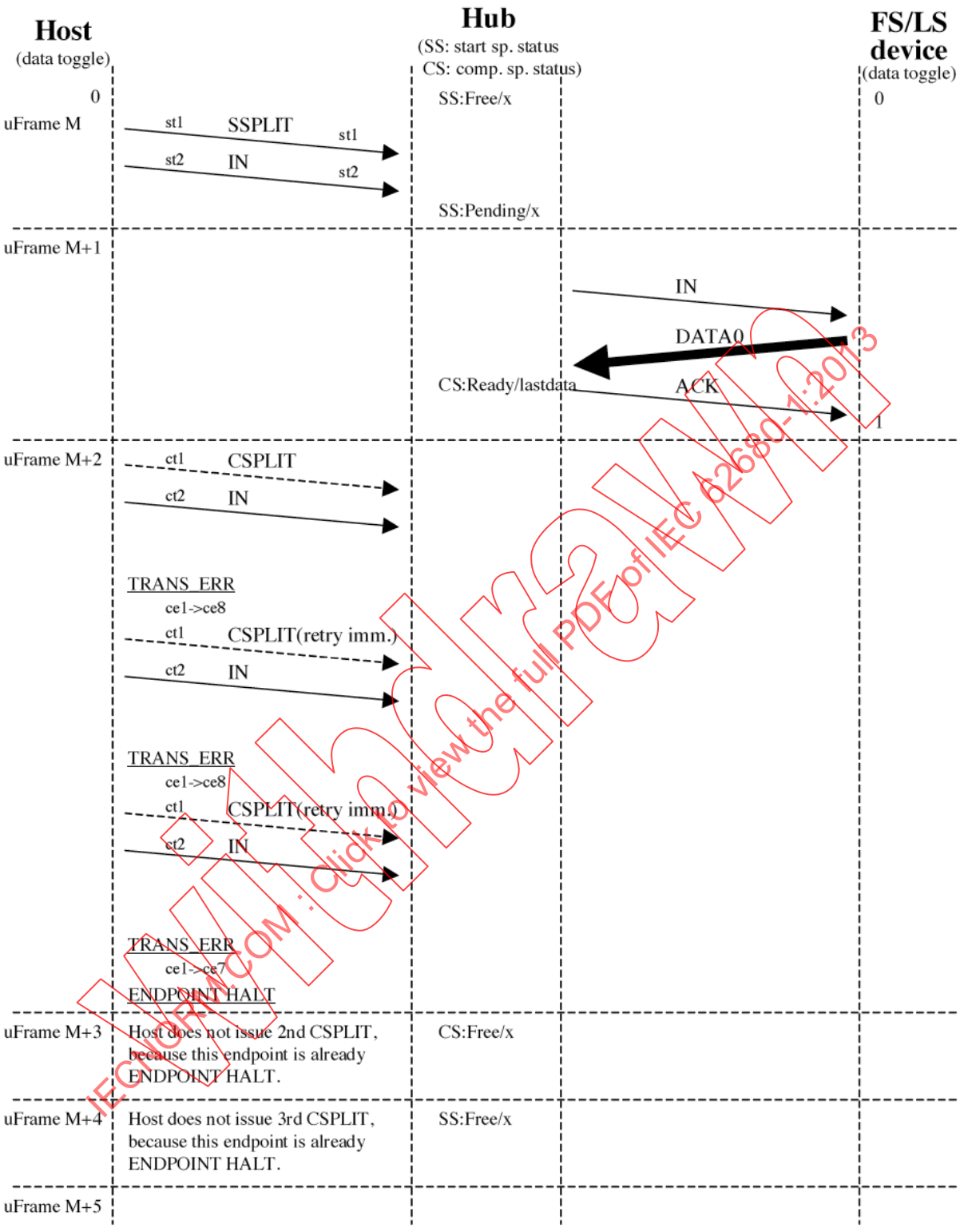


Figure A-66. Normal HS CSPLIT Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure A-67. Normal HS CSPLIT 3 Strikes Smash

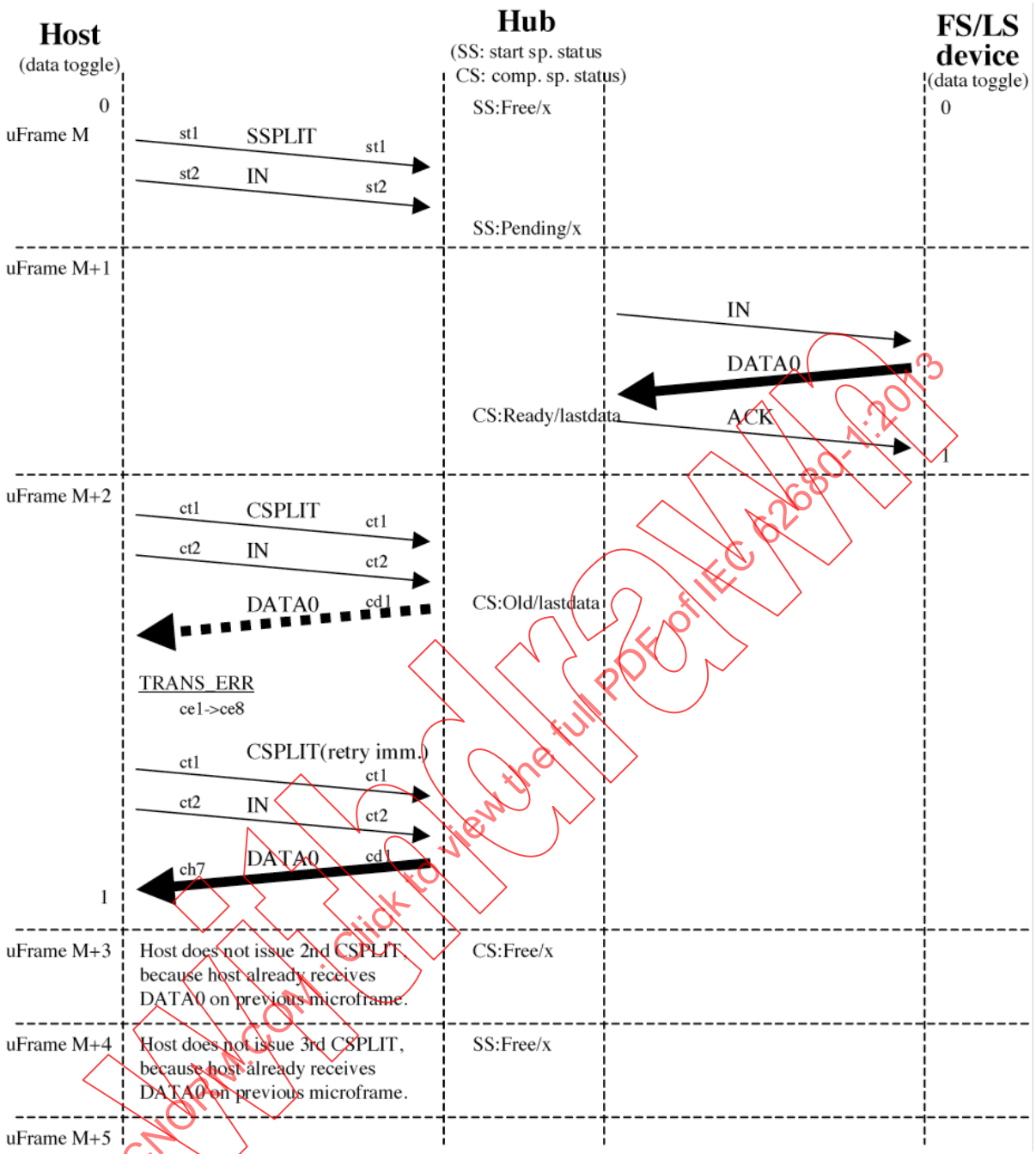
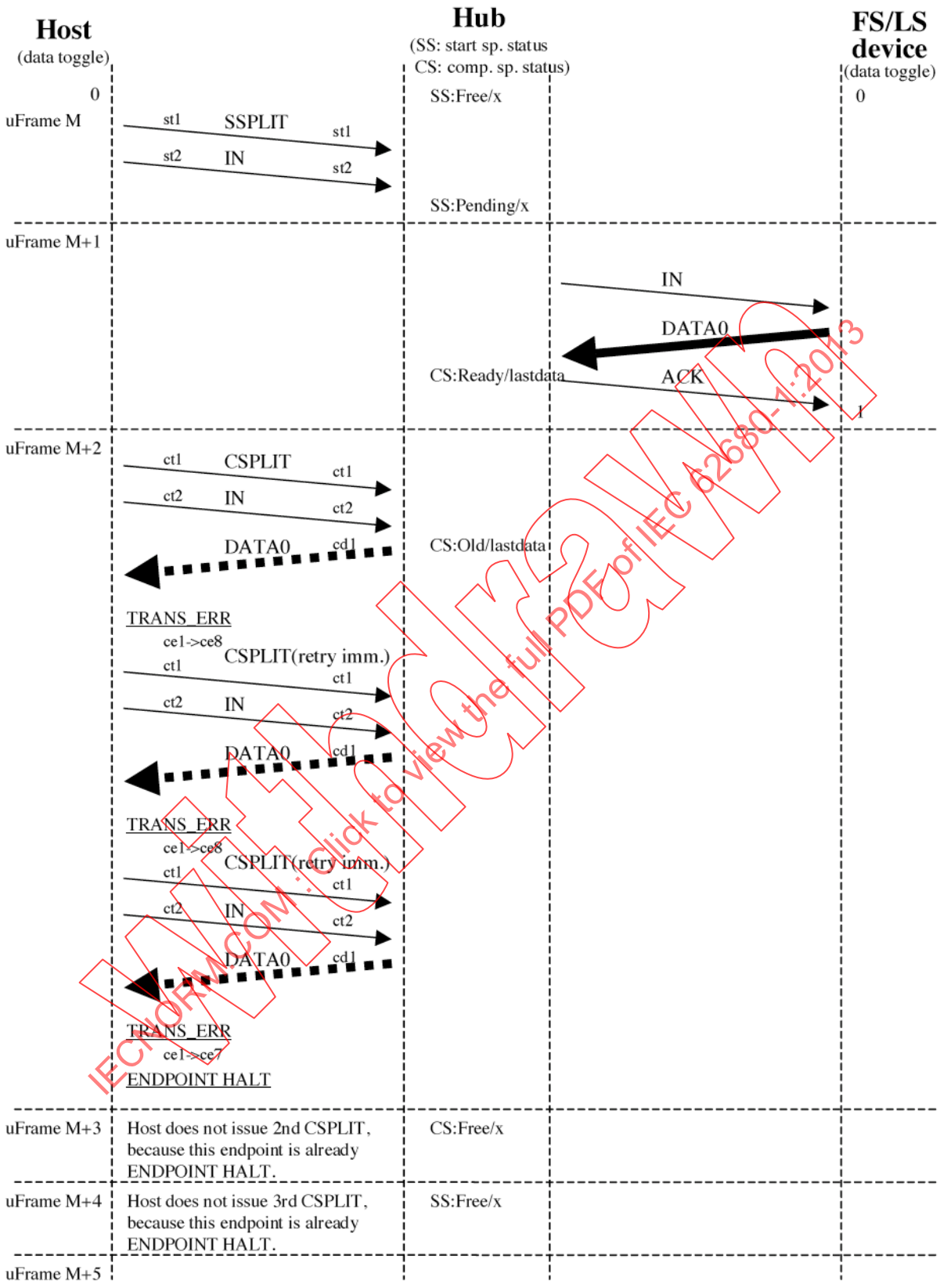


Figure A-68. Normal HS DATA0/1 Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure A-69. Normal HS DATA0/1 3 Strikes Smash

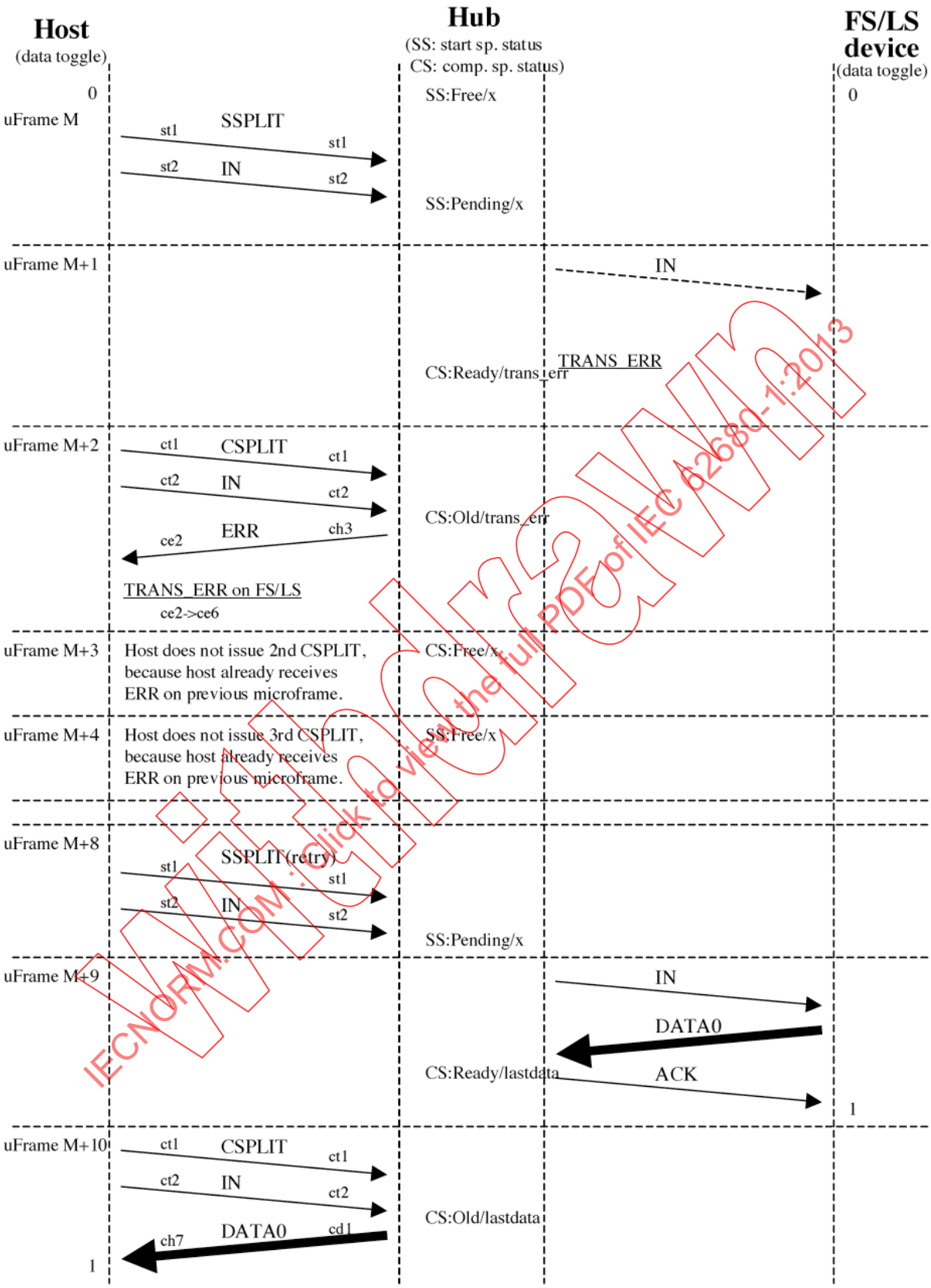


Figure A-70. Normal FS/LS IN Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

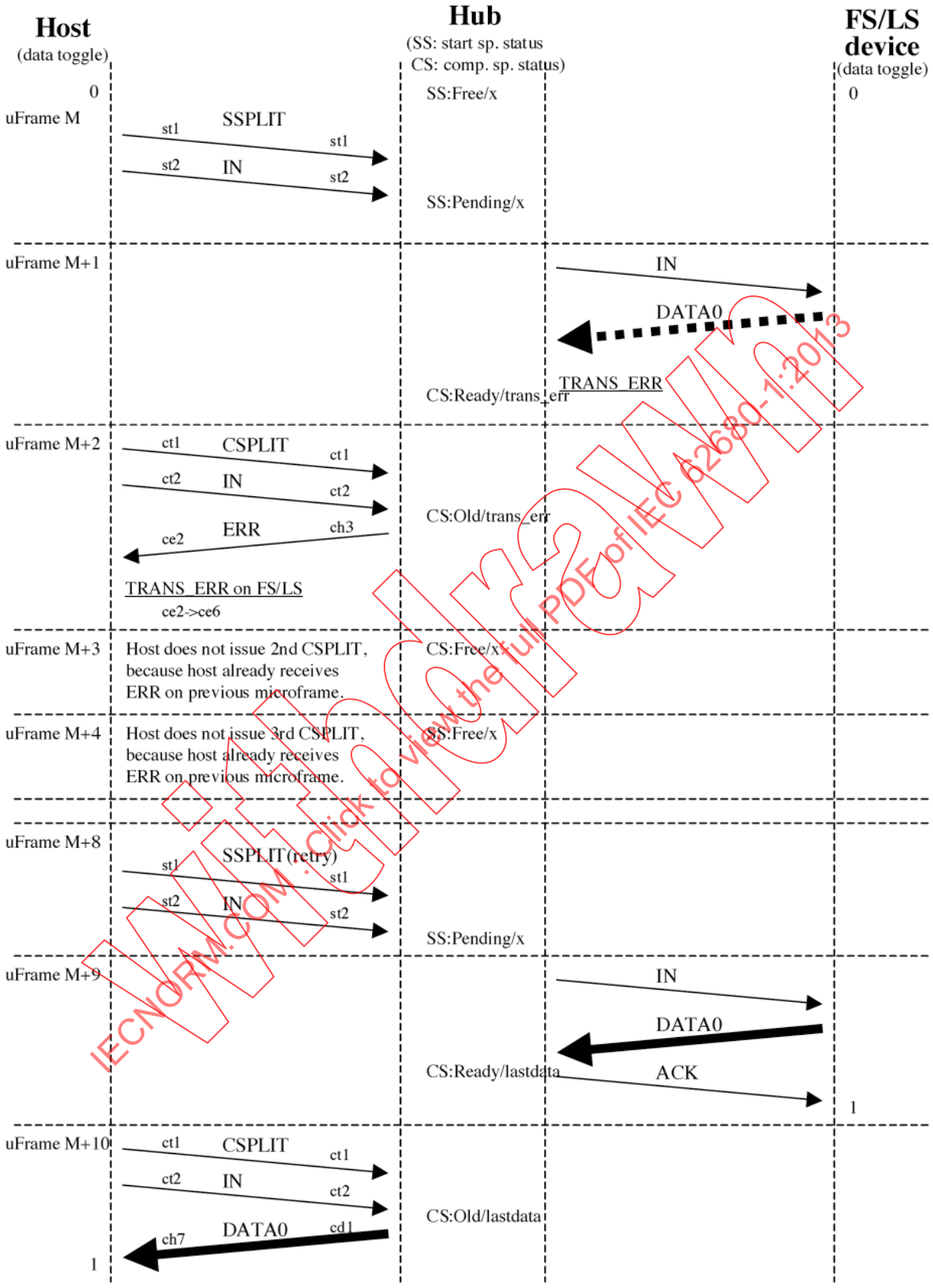


Figure A-71. Normal FS/LS DATA0/1 Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

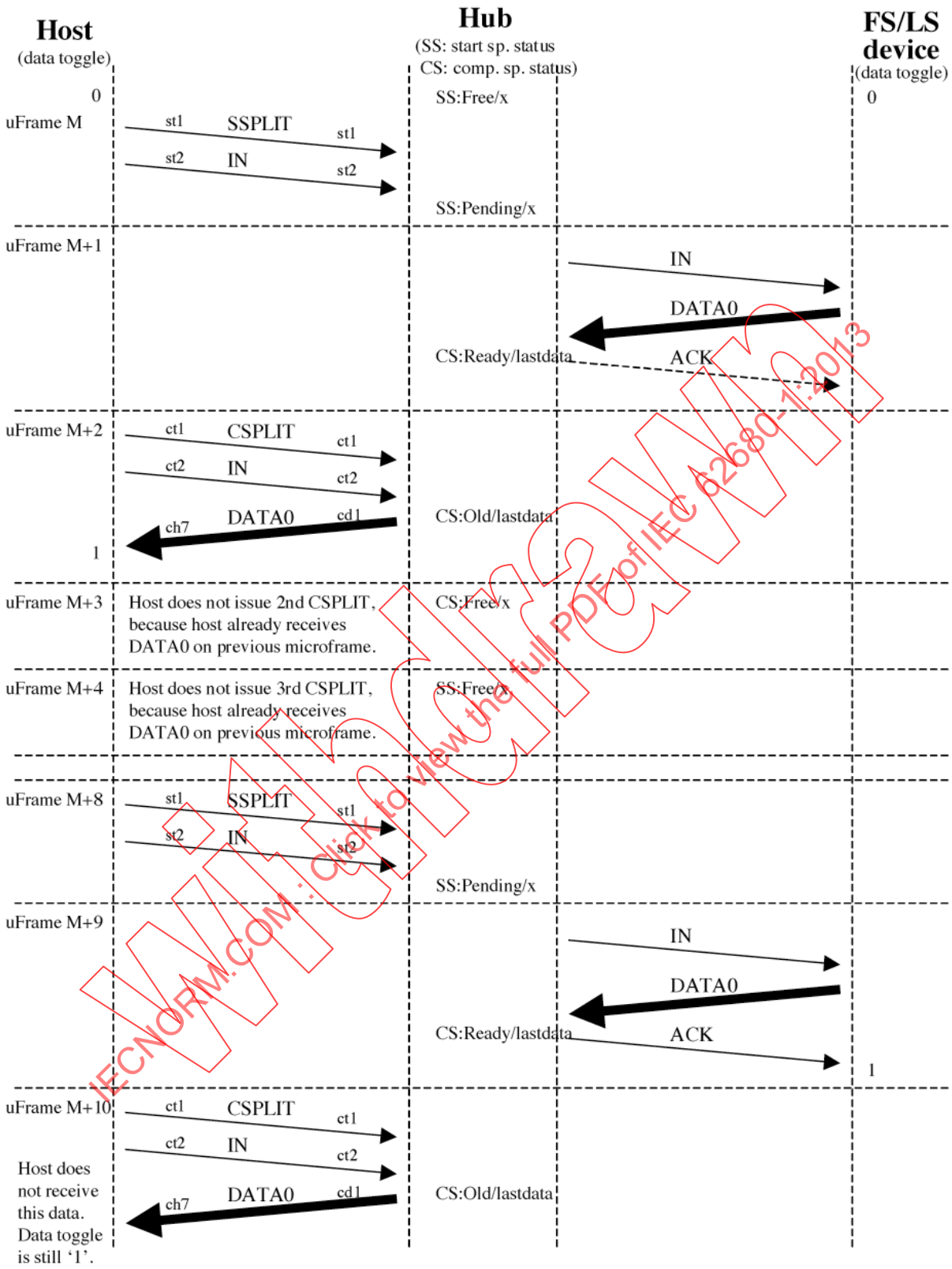


Figure A-72. Normal FS/LS ACK Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

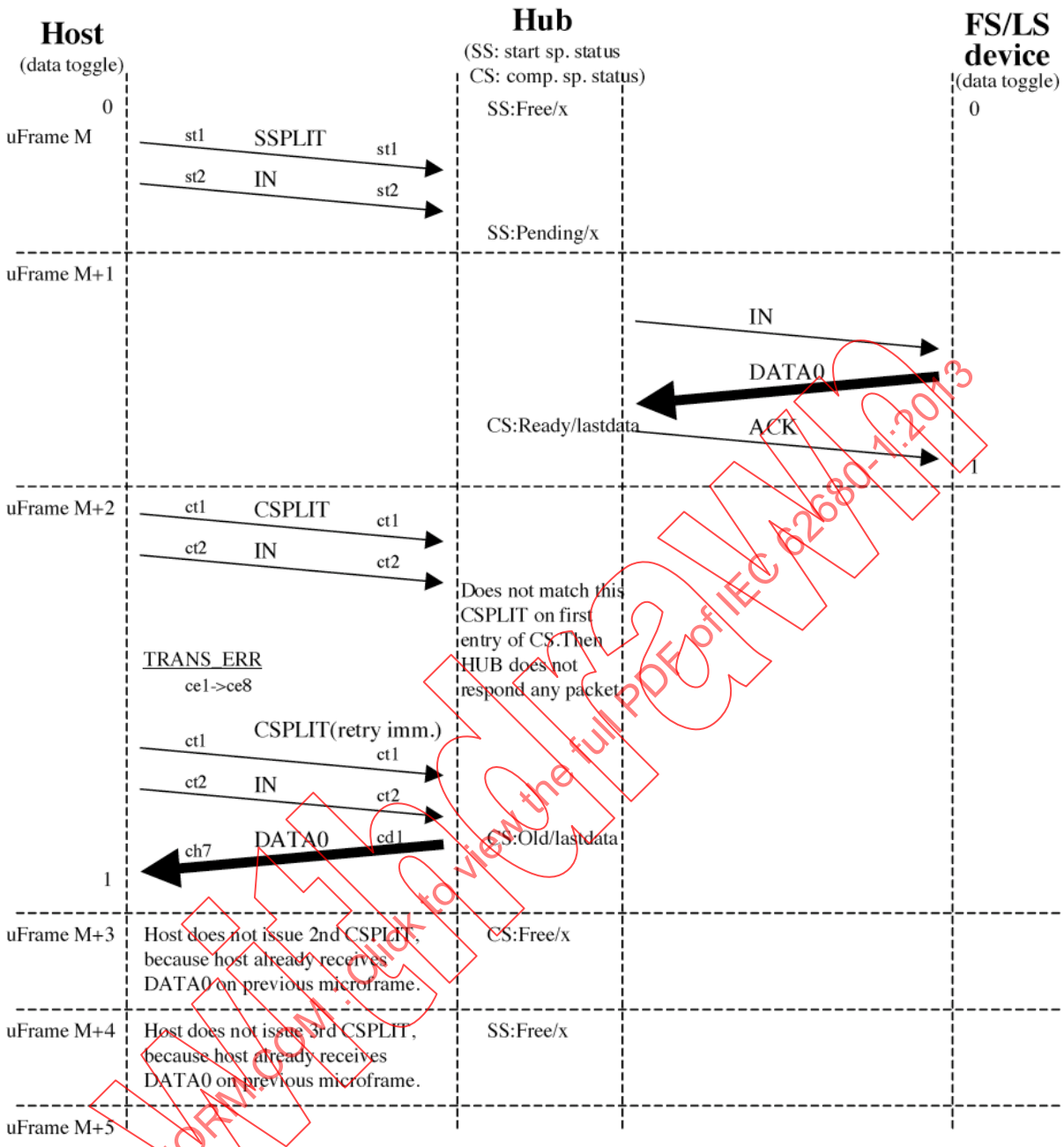


Figure A-73. Searching No Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

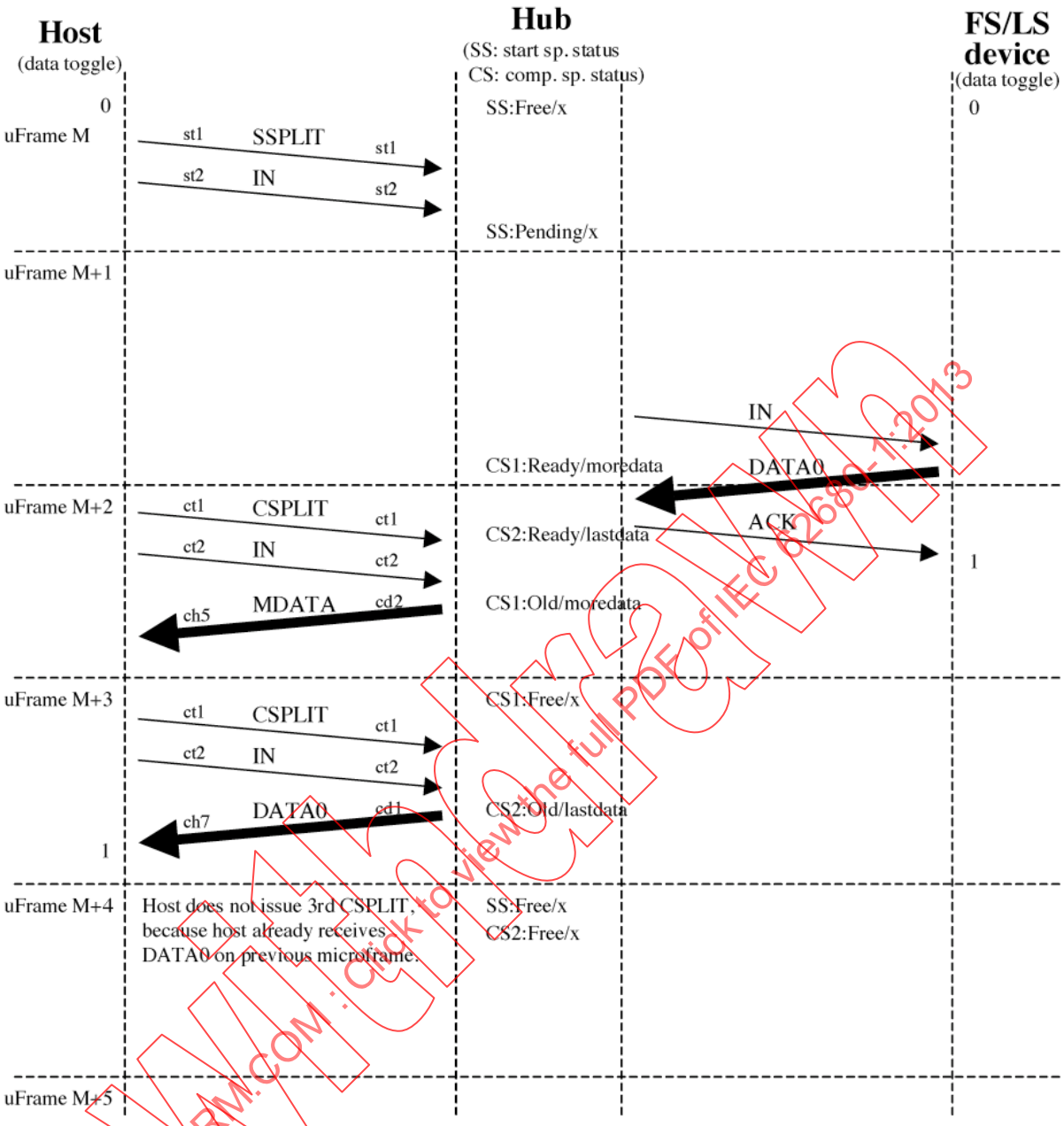


Figure A-74. CS Earlier No Smash(HS MDATA and FS/LS Data Packet is on M+1 and M+2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

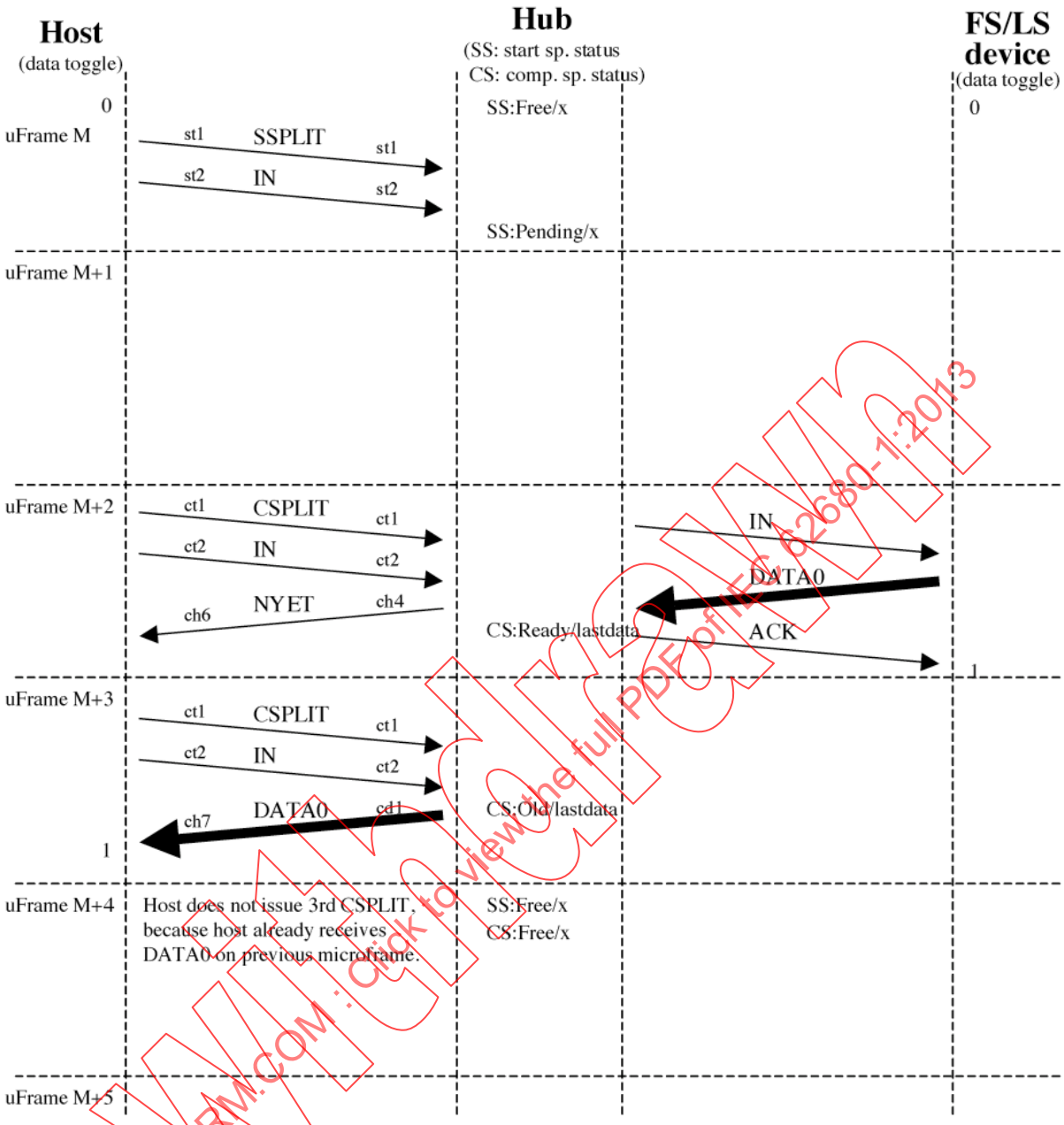


Figure A-75. CS Earlier No Smash(HS NYET and FS/LS Data Packet is on M+2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

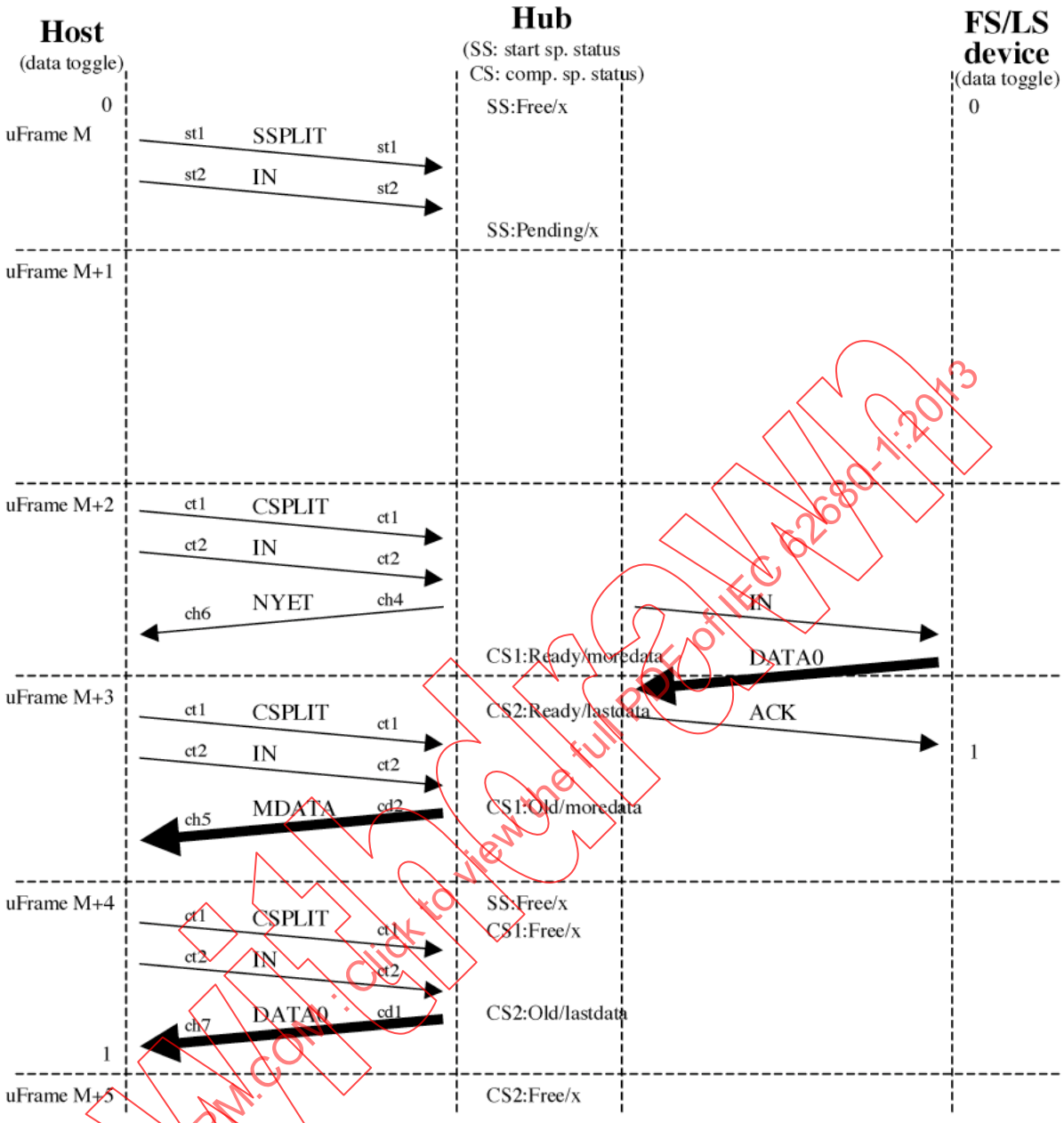


Figure A-76. CS Earlier No Smash(HS NYET and MDATA and FS/LS Data Packet is on M+2 and M+3)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

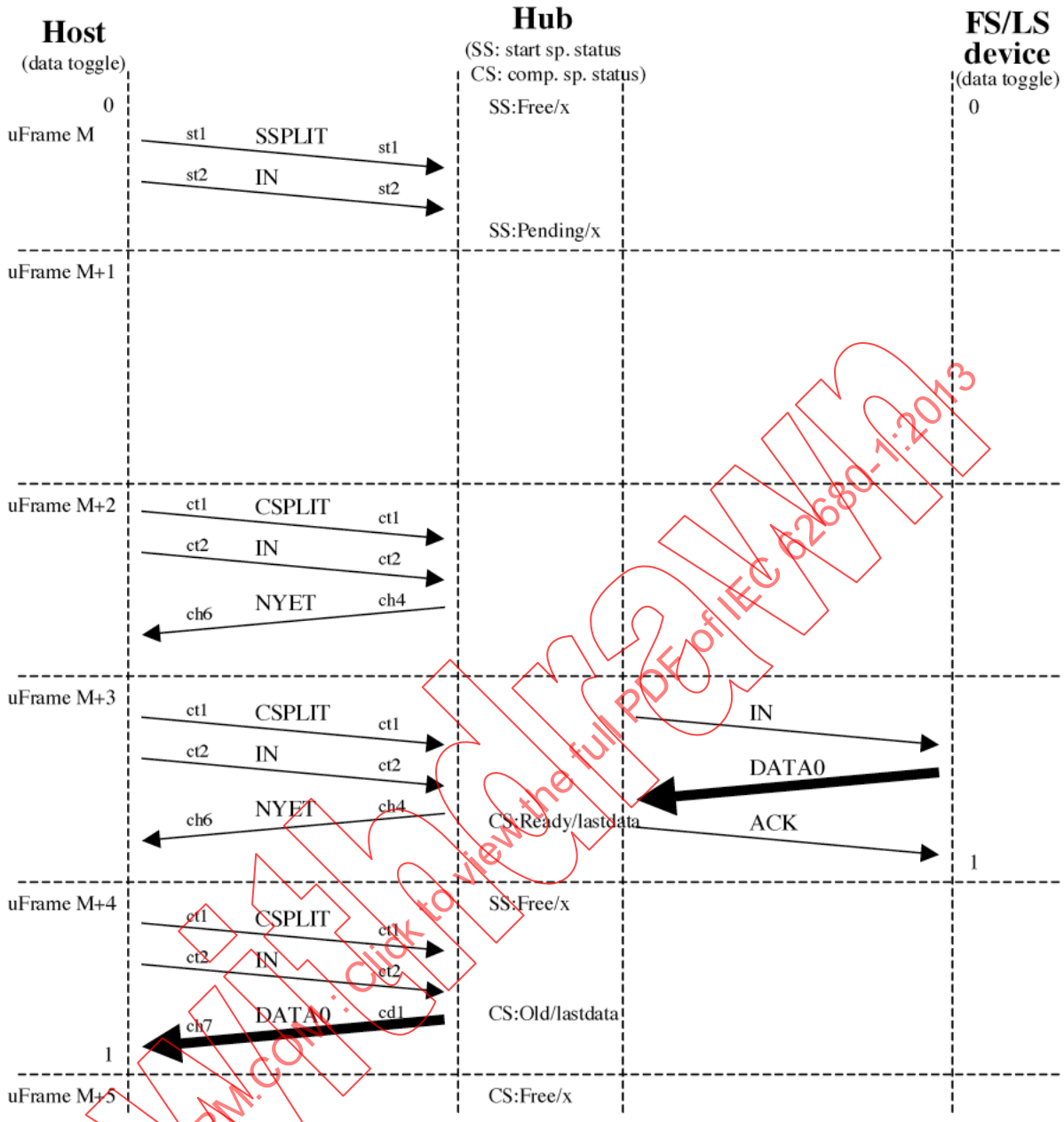


Figure A-77. CS Earlier No Smash(HS NYET and FS/LS Data Packet is on M+3)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

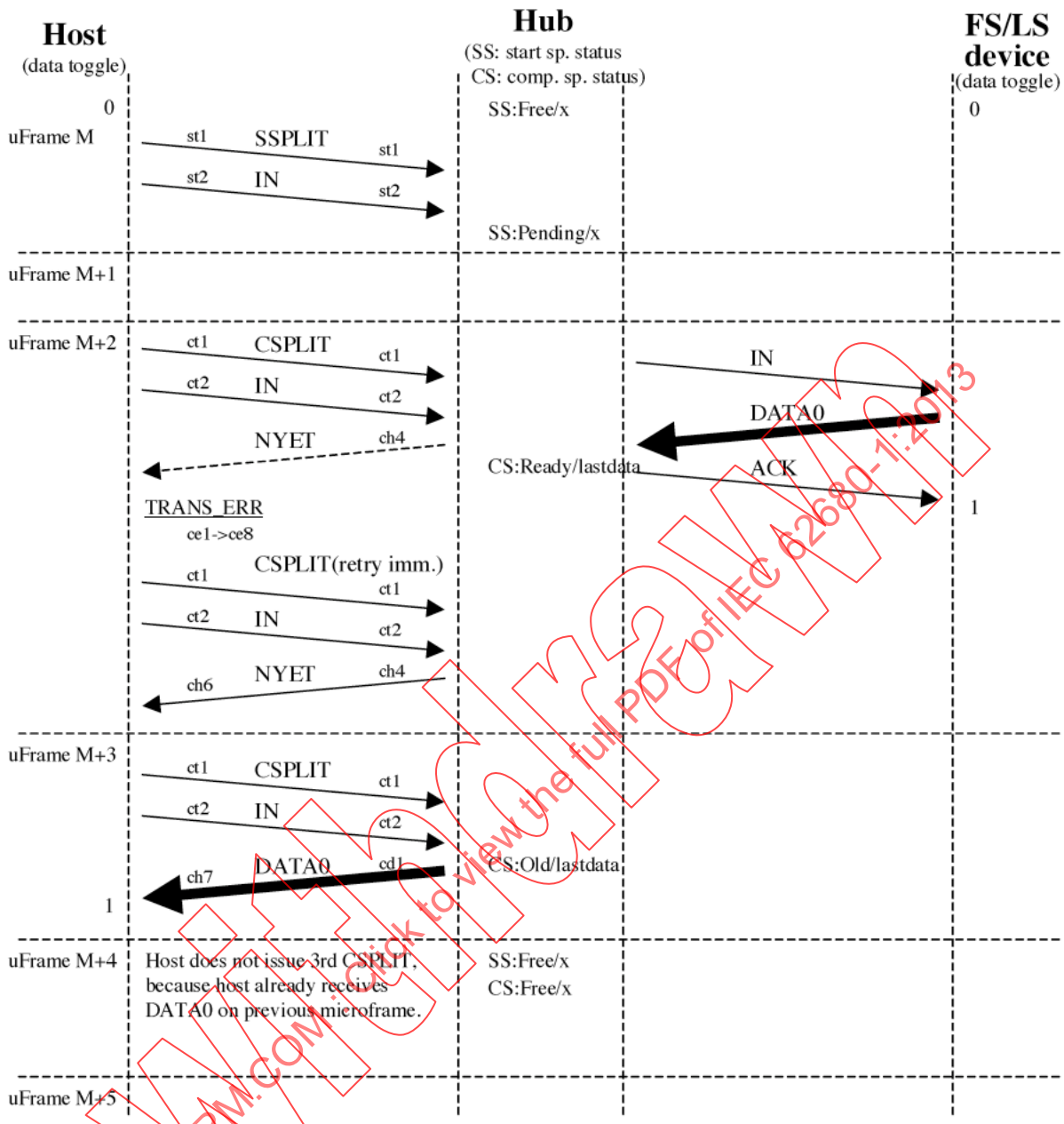
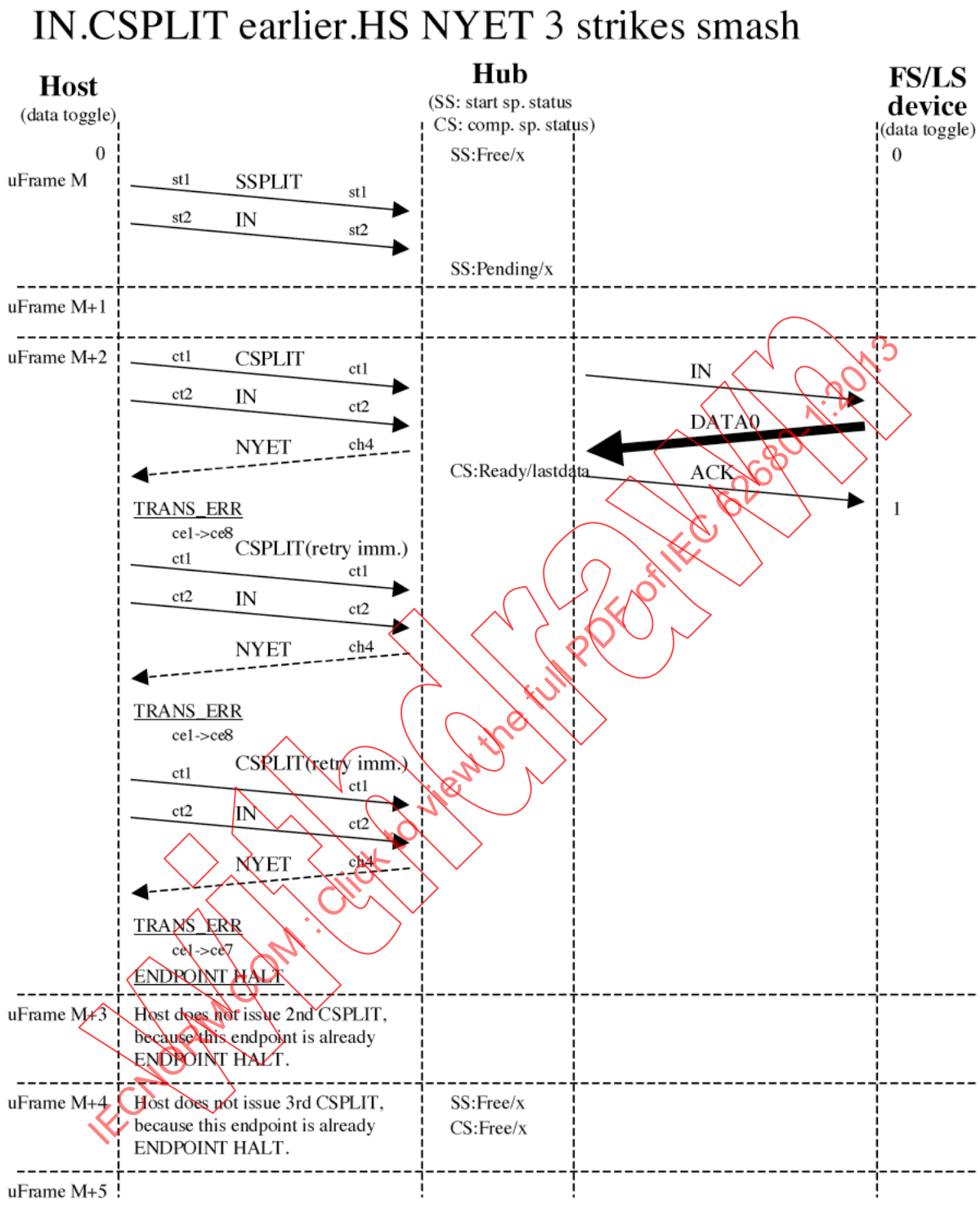


Figure A-78. CS Earlier HS NYET Smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure A-79. CS Earlier HS NYET 3 Strikes Smash

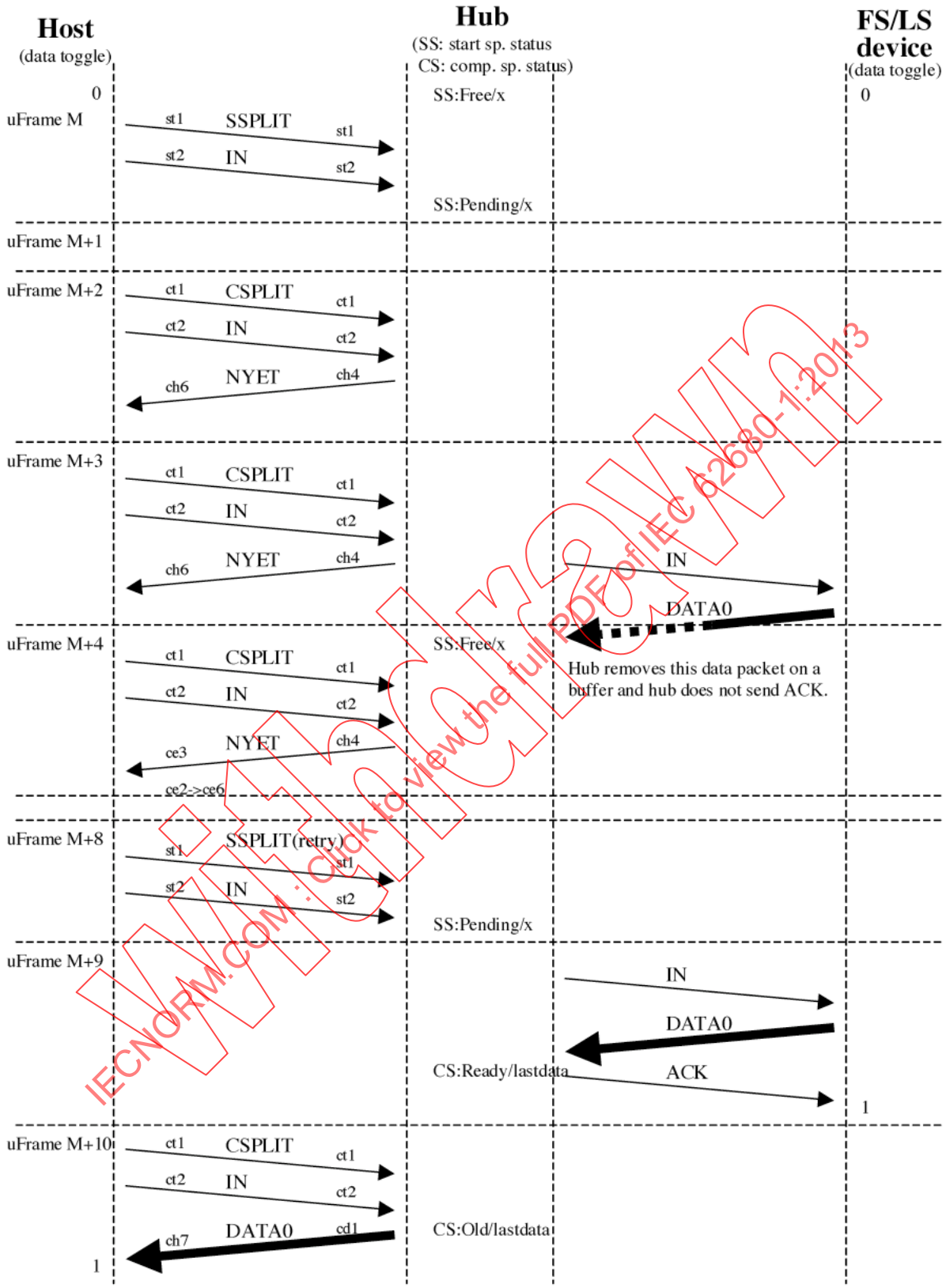


Figure A-80. Abort and Free Abort(HS NYET and FS/LS Transaction is Continued at End of M+3)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

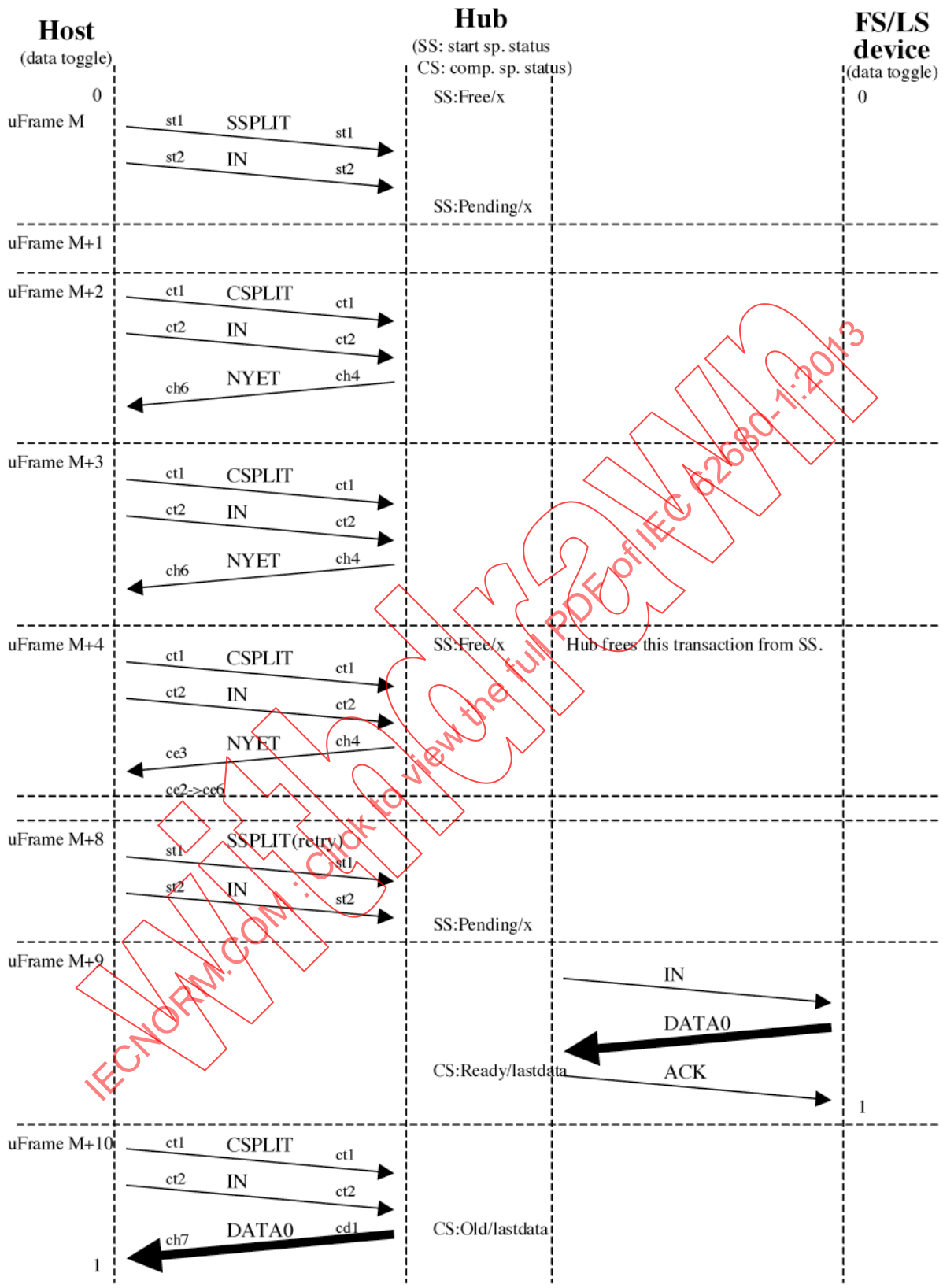


Figure A-81. Abort and Free Free(HS NYET and FS/LS Transaction is not Started at End of M+3)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

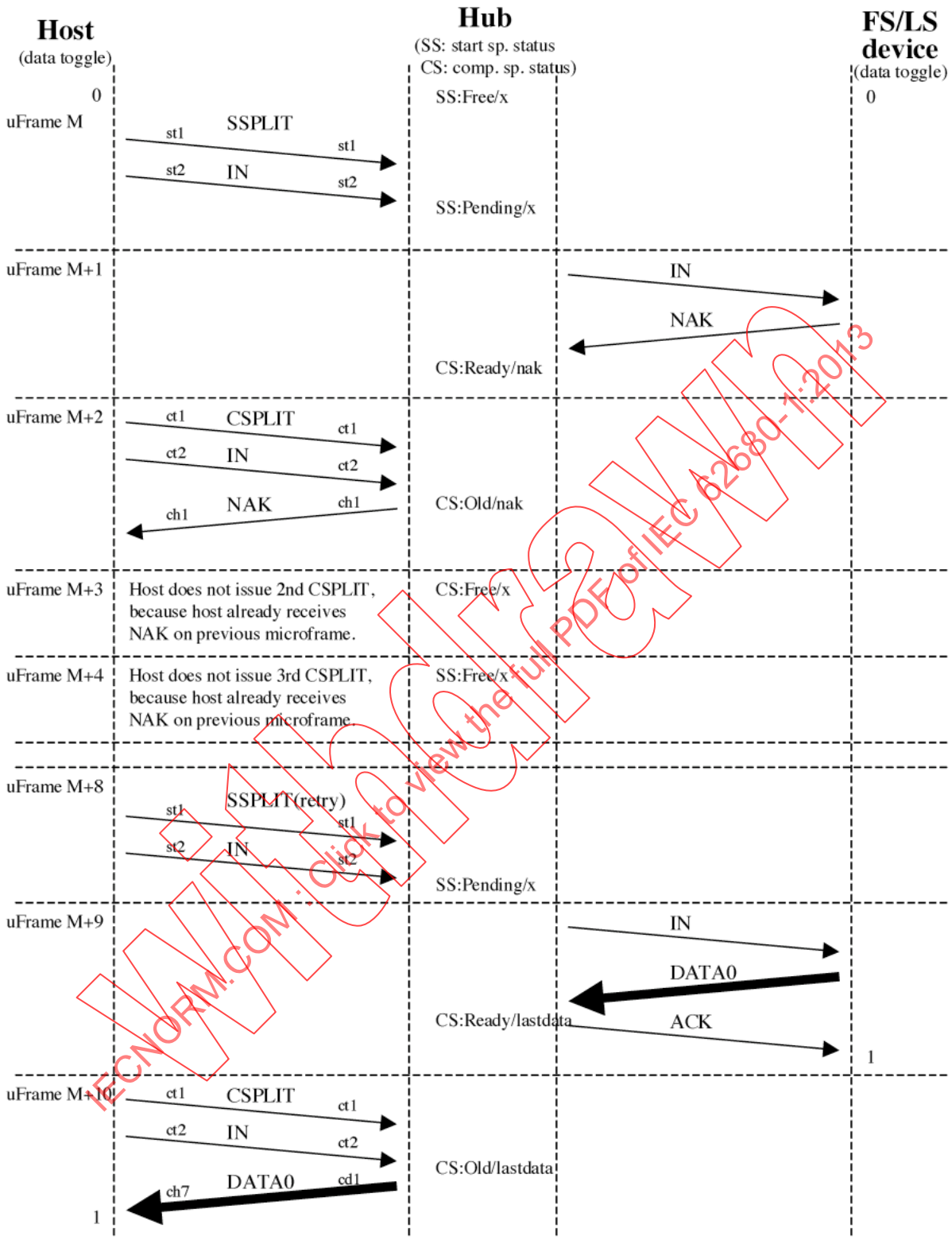


Figure A-82. Device Busy No Smash(FS/LS NAK)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

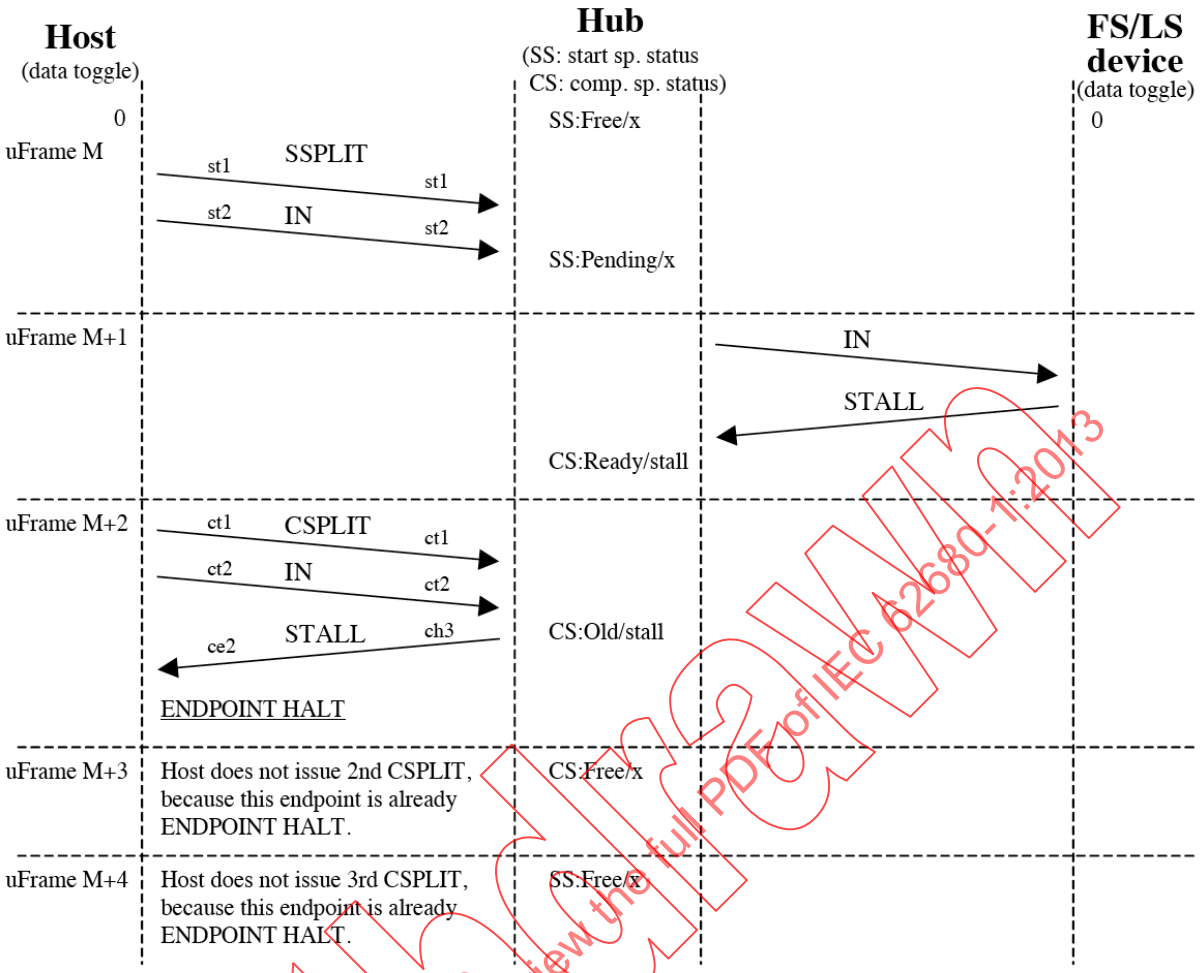


Figure A-83. Device Stall No Smash(FS/LS STALL)

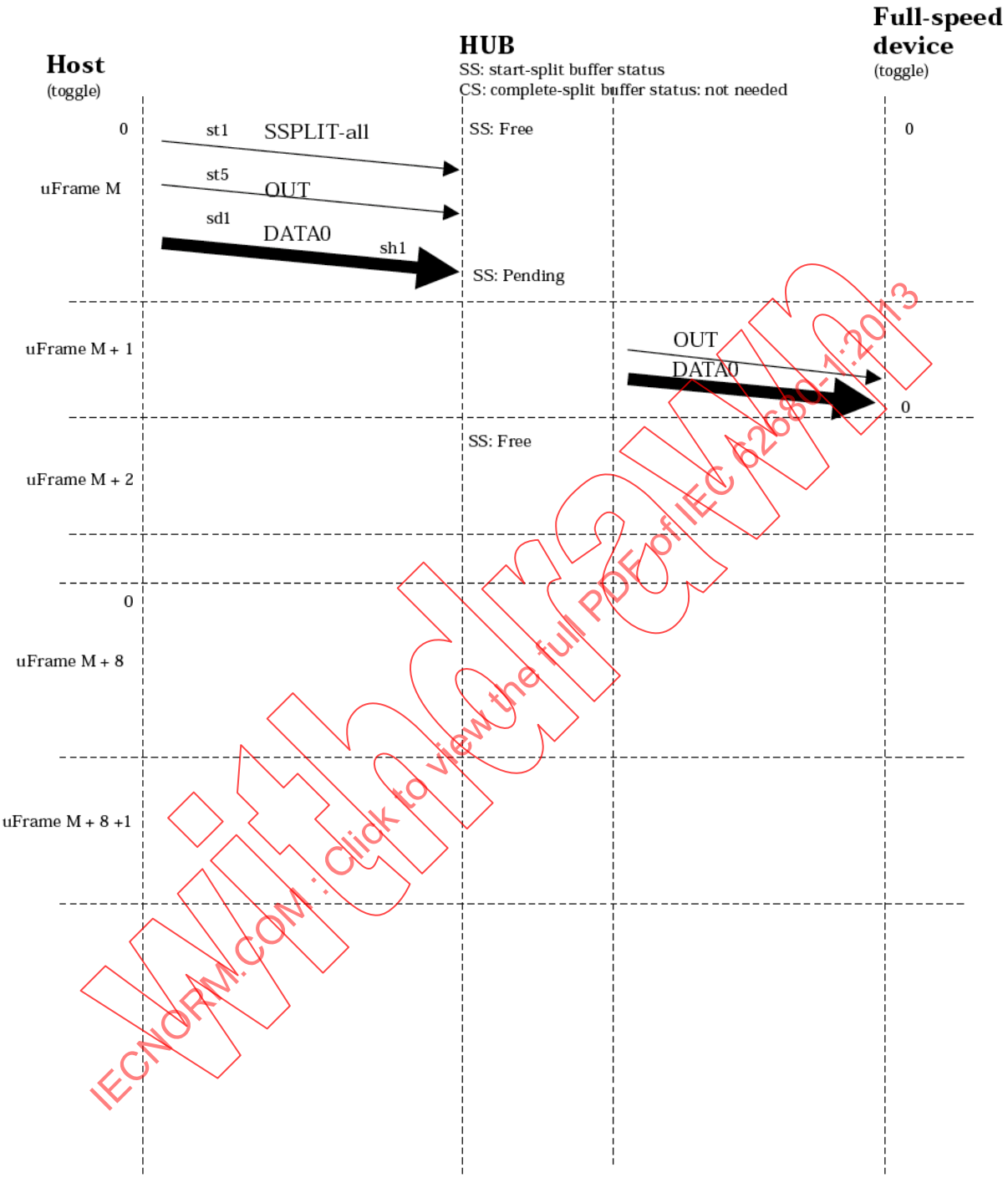
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A.5 Isochronous OUT Split-transaction Examples

Case	Reference Figure
Normal: small payload (<=188)	1
Normal: large payload (> 188)	2
HS SSPLIT-all corrupted, HS OUT corrupted	3
HS DATA0 corrupted (small payload)	4
HS SSPLIT-begin corrupted	5
HS OUT after the HS SSPLIT-begin is corrupted	6
HS DATA0 corrupted (large payload)	7
HS SSPLIT-mid or OUT or DATA0 corrupted	8

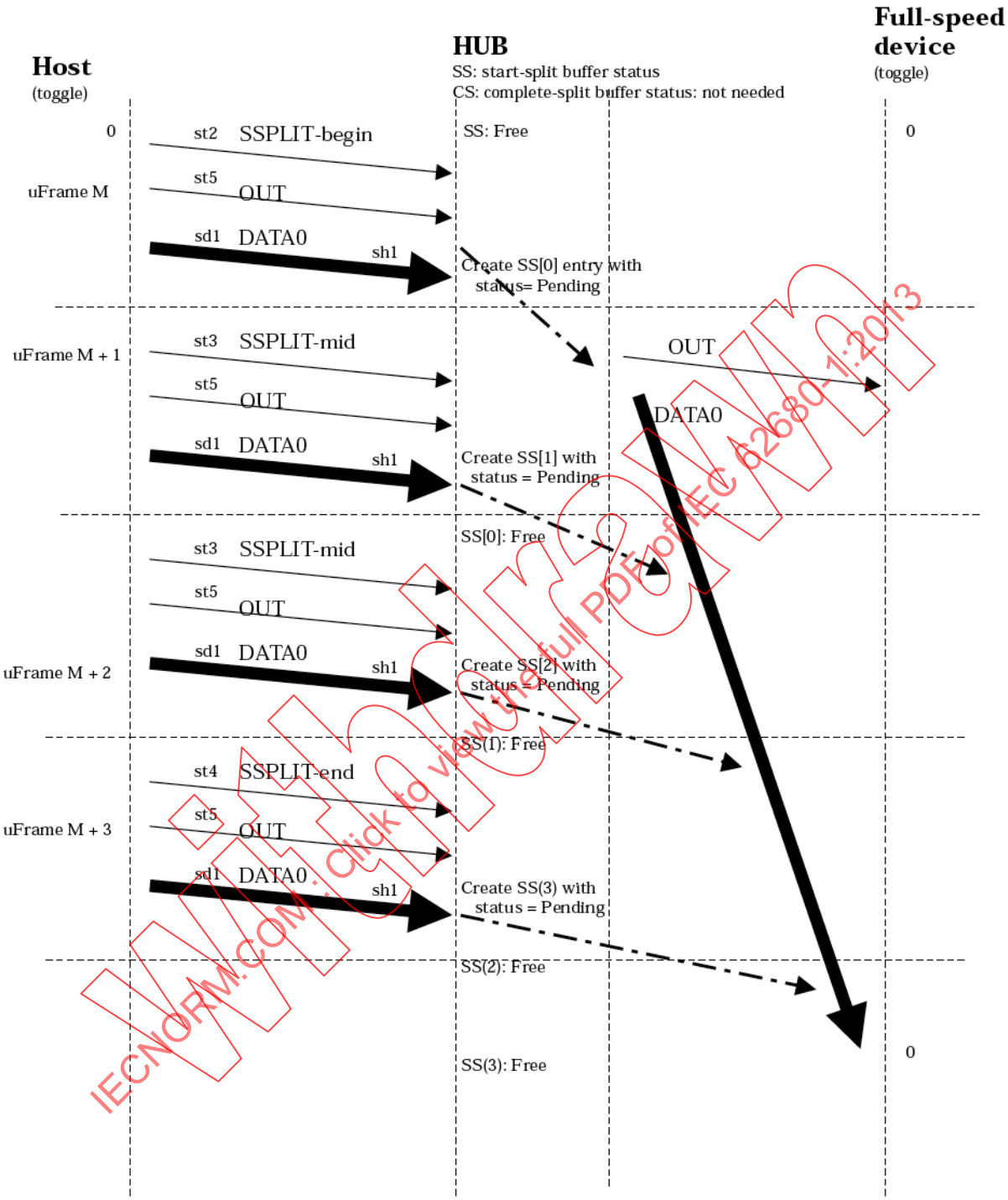
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

1) Normal. Payload <= 188 bytes:



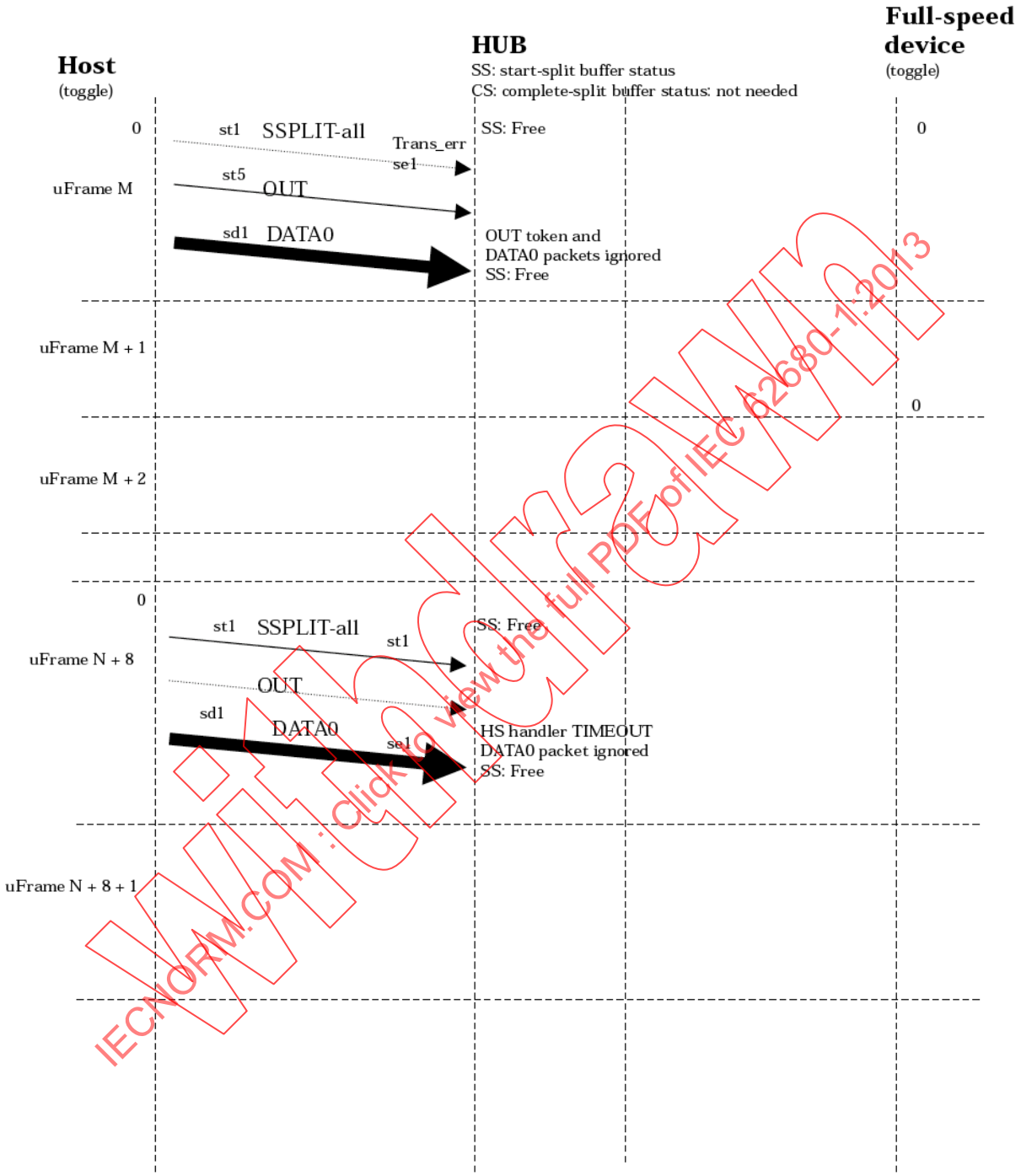
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

2) Normal. Payload > 188 Bytes



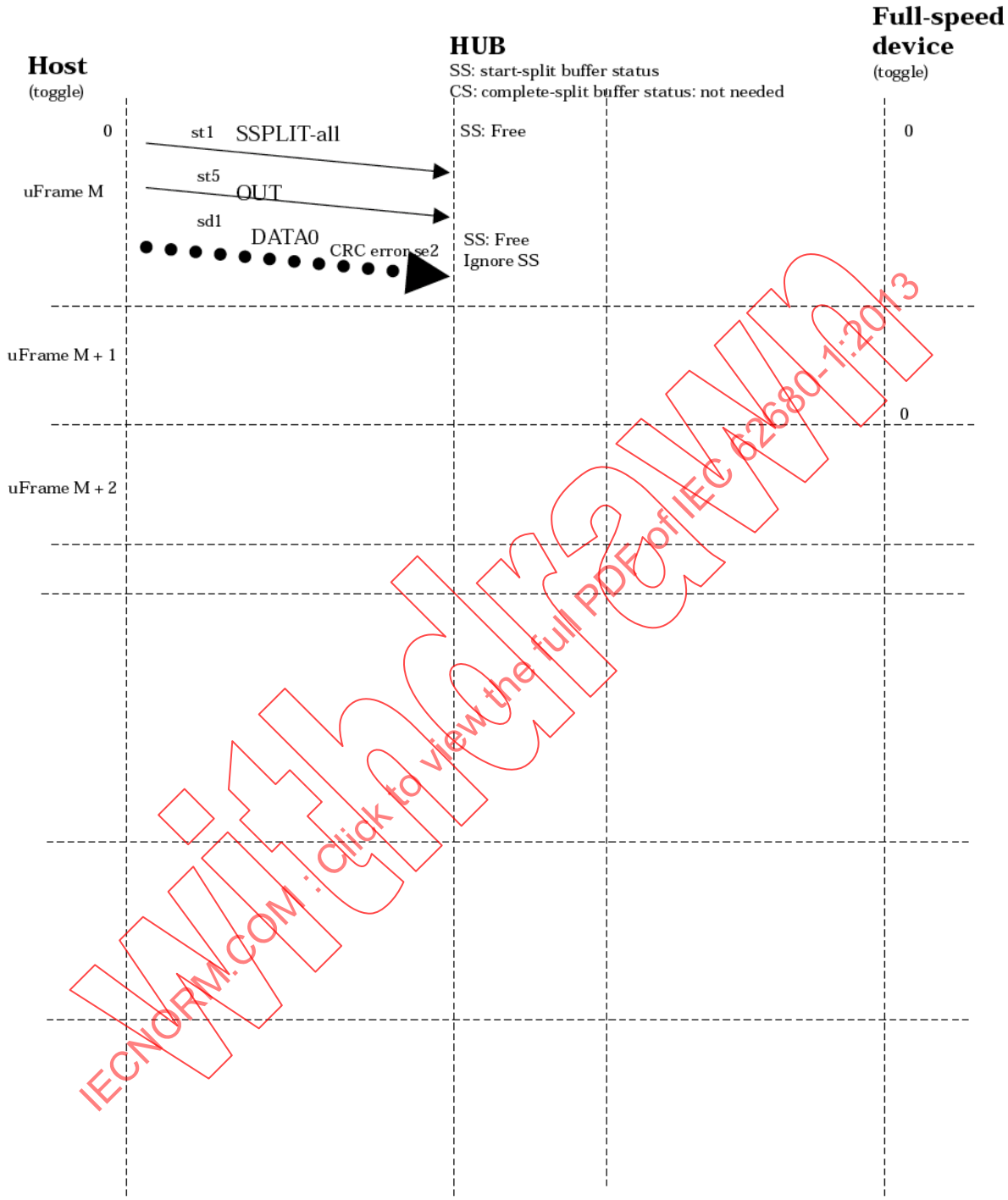
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

3) HS SSPLIT-all corrupted (missing or CRC error etc.)
HS OUT corrupted



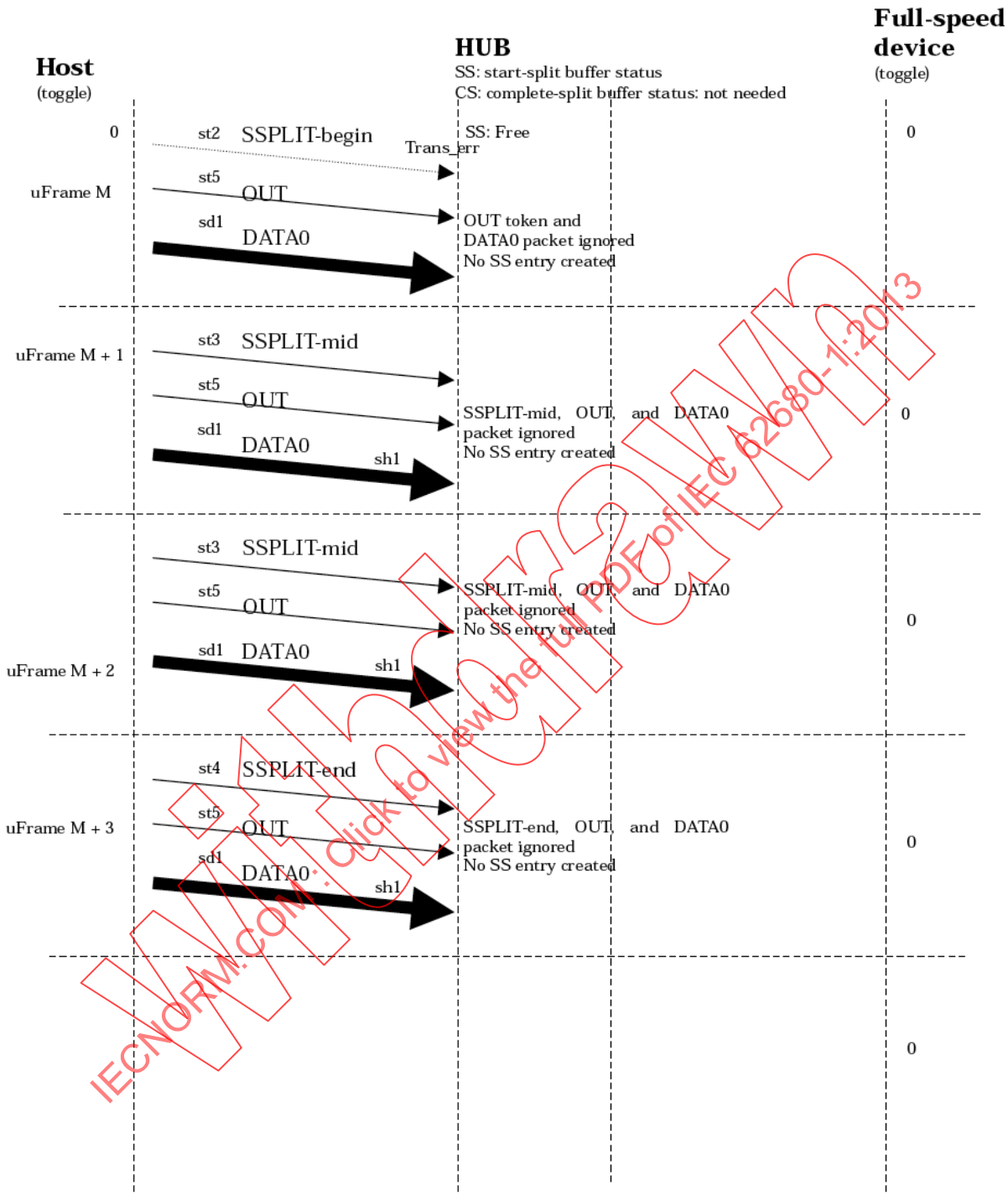
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4) HS DATA0 corrupted



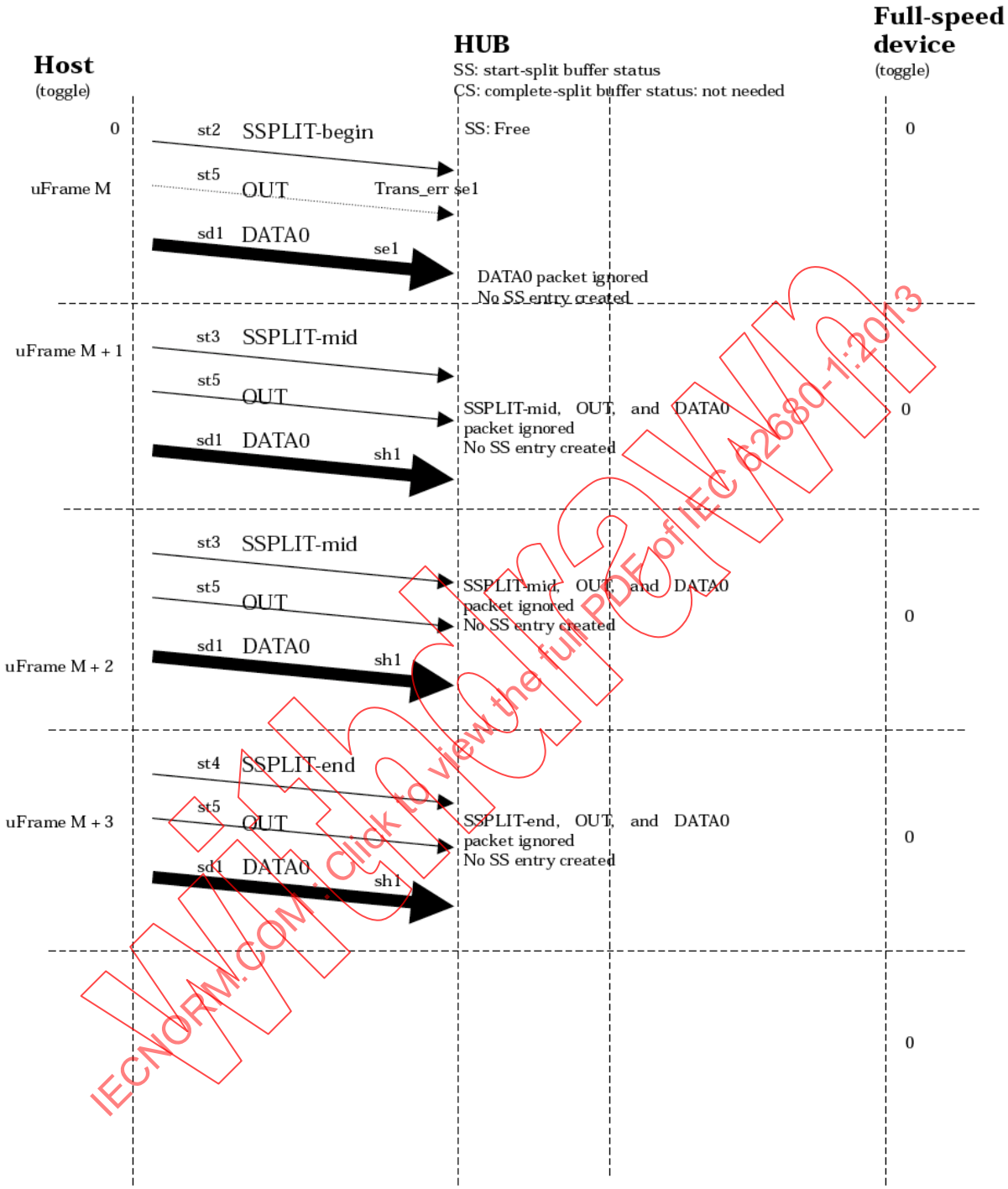
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5) HS SSPLIT-begin corrupted (missing or CRC error etc.)



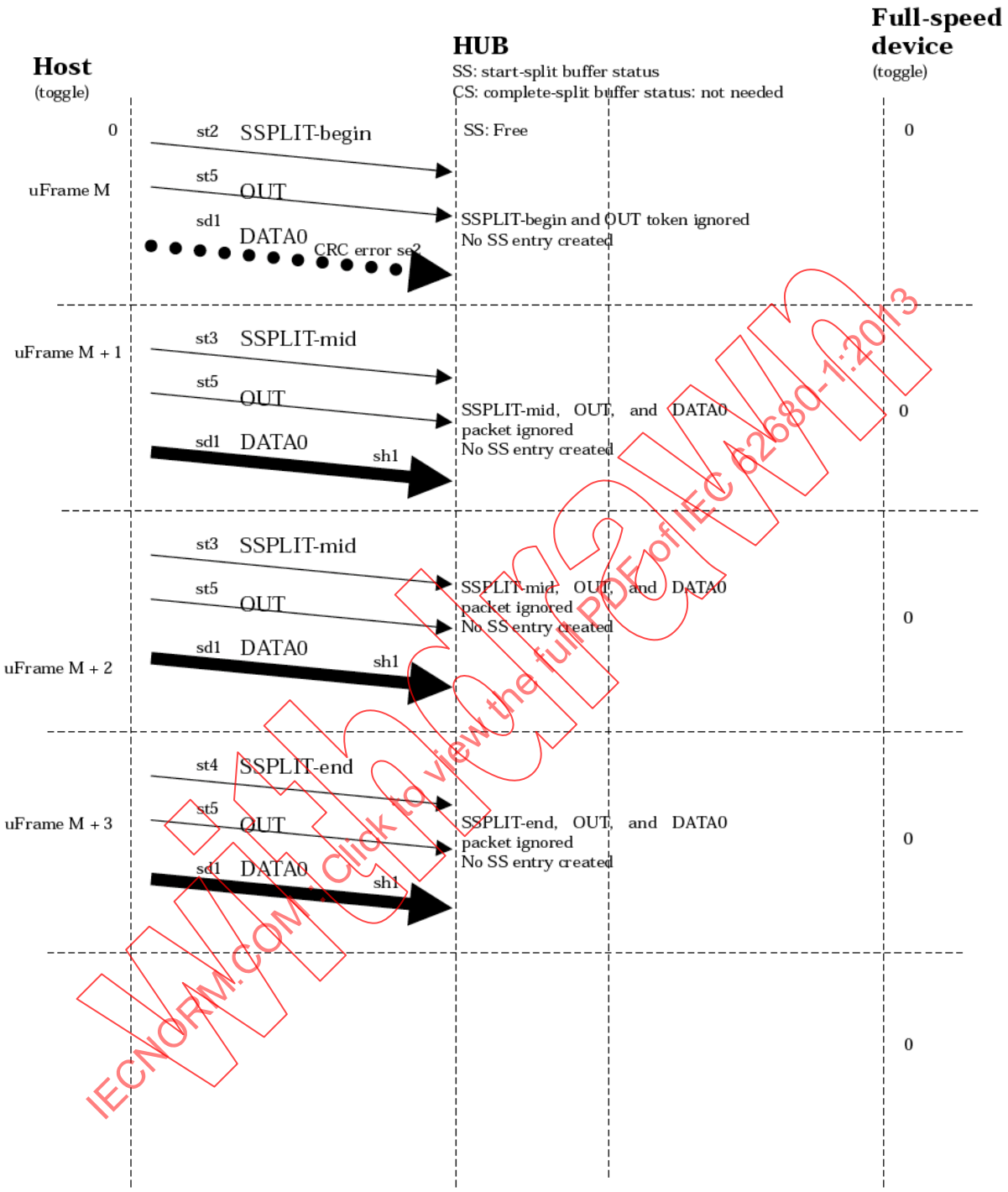
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

6) HS OUT after the HS SSPLIT-begin is corrupted



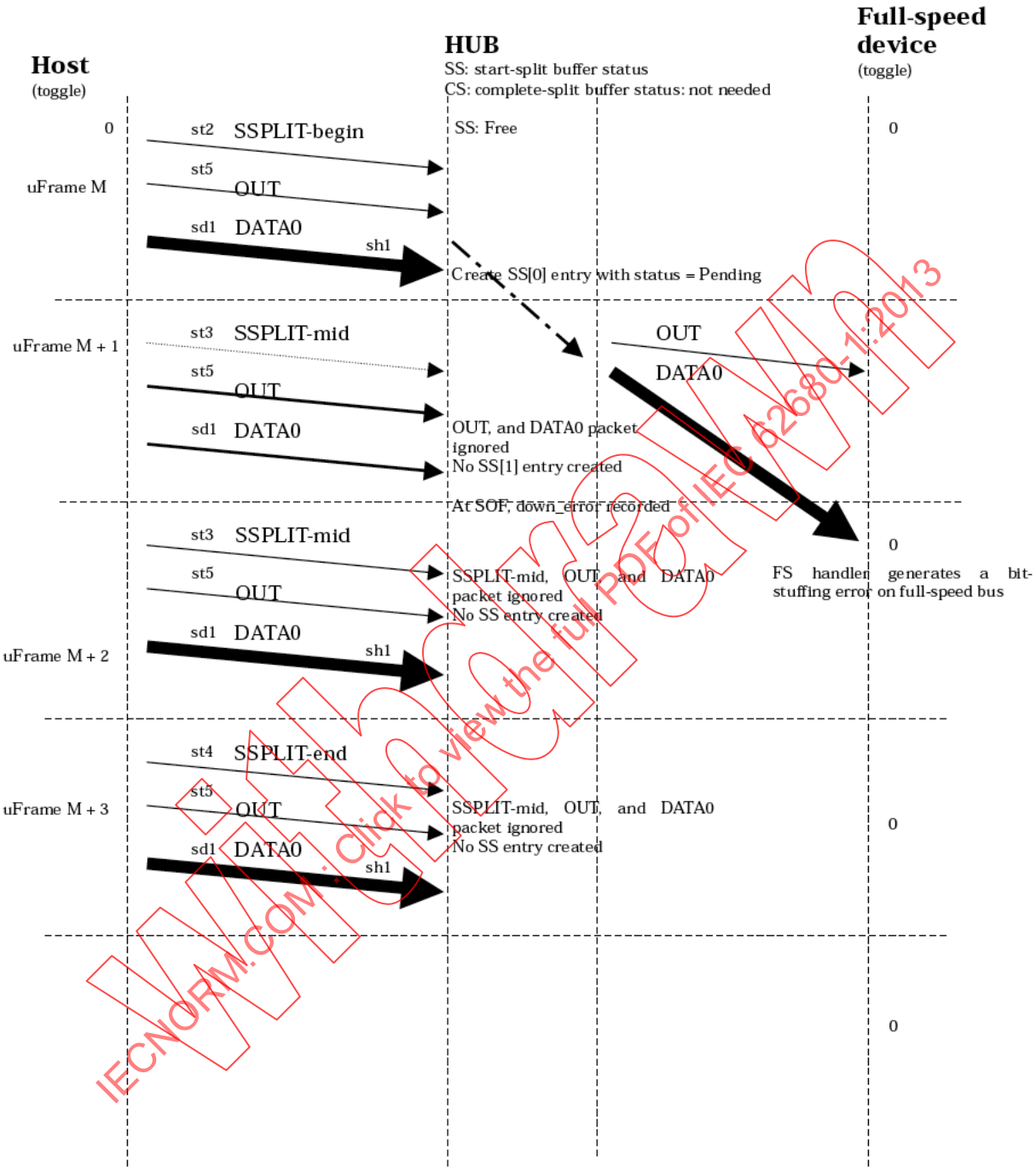
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7) HS DATA0 corrupted



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8) HS SSPLIT-mid or OUT token or DATA0 packet after it is corrupted



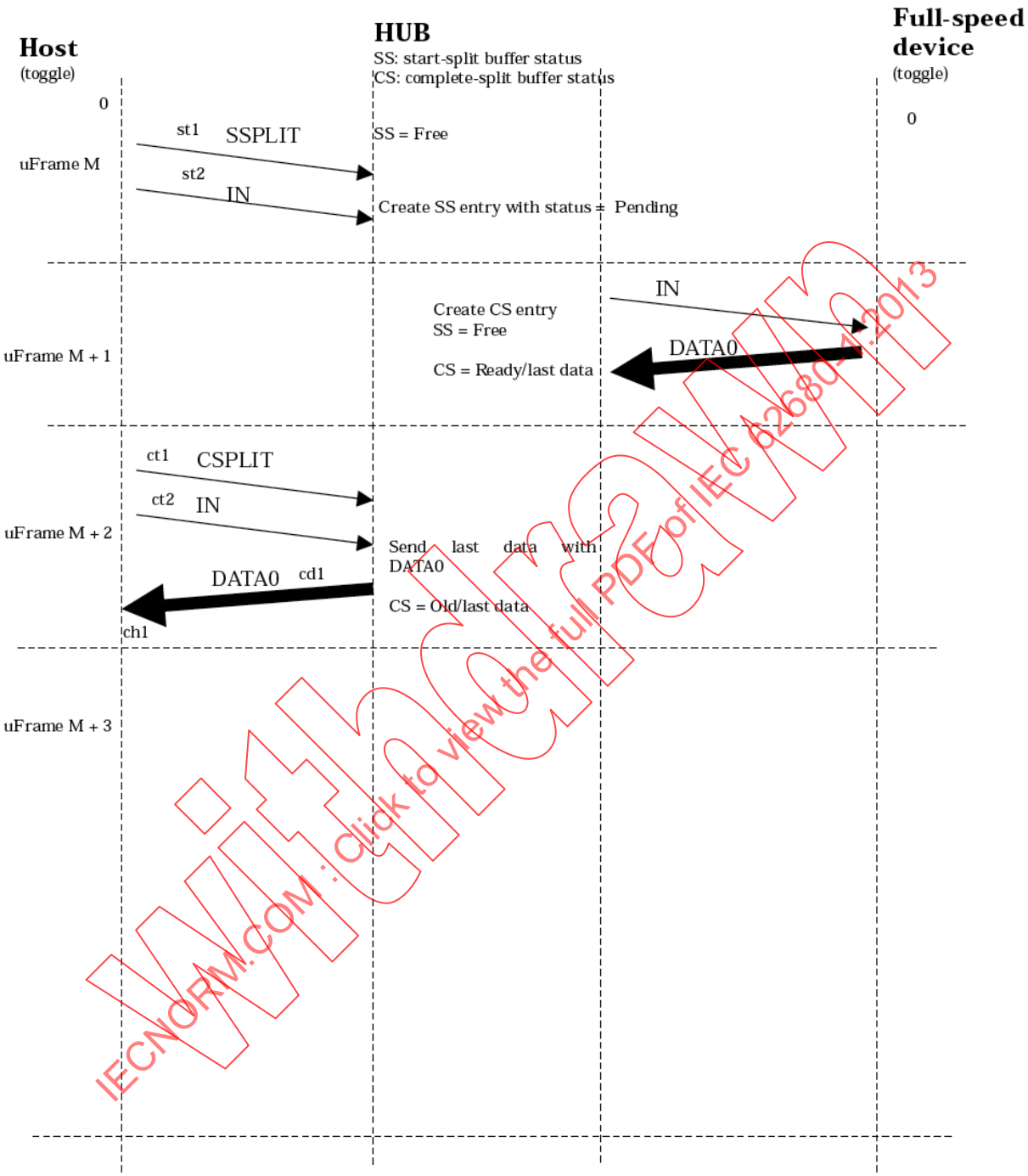
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A.6 Isochronous IN Split-transaction Examples

Case	Reference Figure
Normal: full-speed bus transaction does not cross microframe boundary	1
Normal: full-speed bus transaction crosses microframe boundary	2
HS SSPLIT corrupted	3
IN after HS SSPLIT corrupted	4
HS CSPLIT corrupted	5
Consecutive HS CSPLIT corrupted	6
HS IN corrupted	7
Consecutive HS IN corrupted	8
HS data corrupted (case 1)	9
HS data corrupted (case 2)	10
TT has more data than HS expects	11
HS CS too early (full-speed data not available yet)	12
Full-speed timeout or CRC error	13

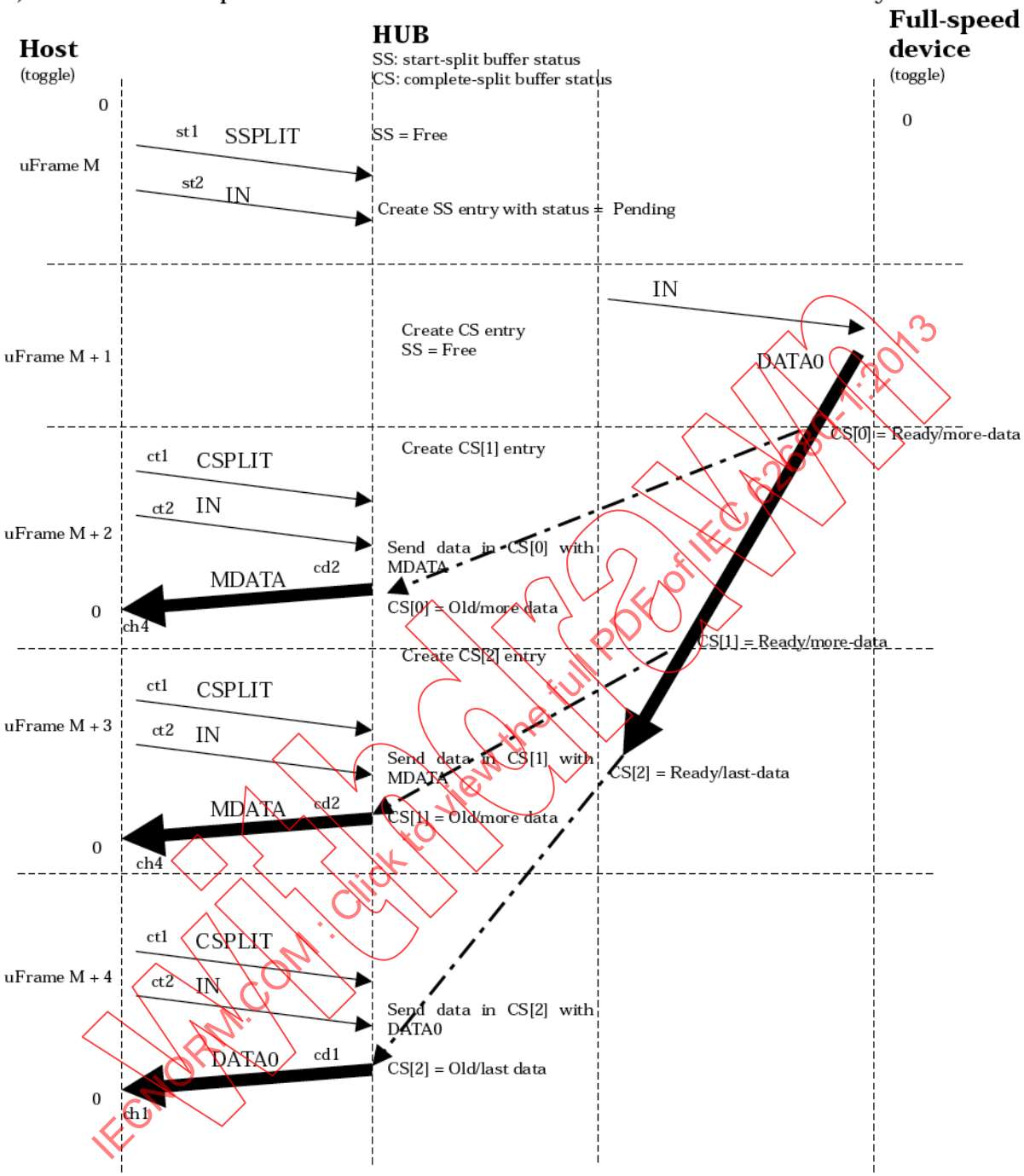
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

1) Normal: full-speed bus transaction does not cross microframe boundary



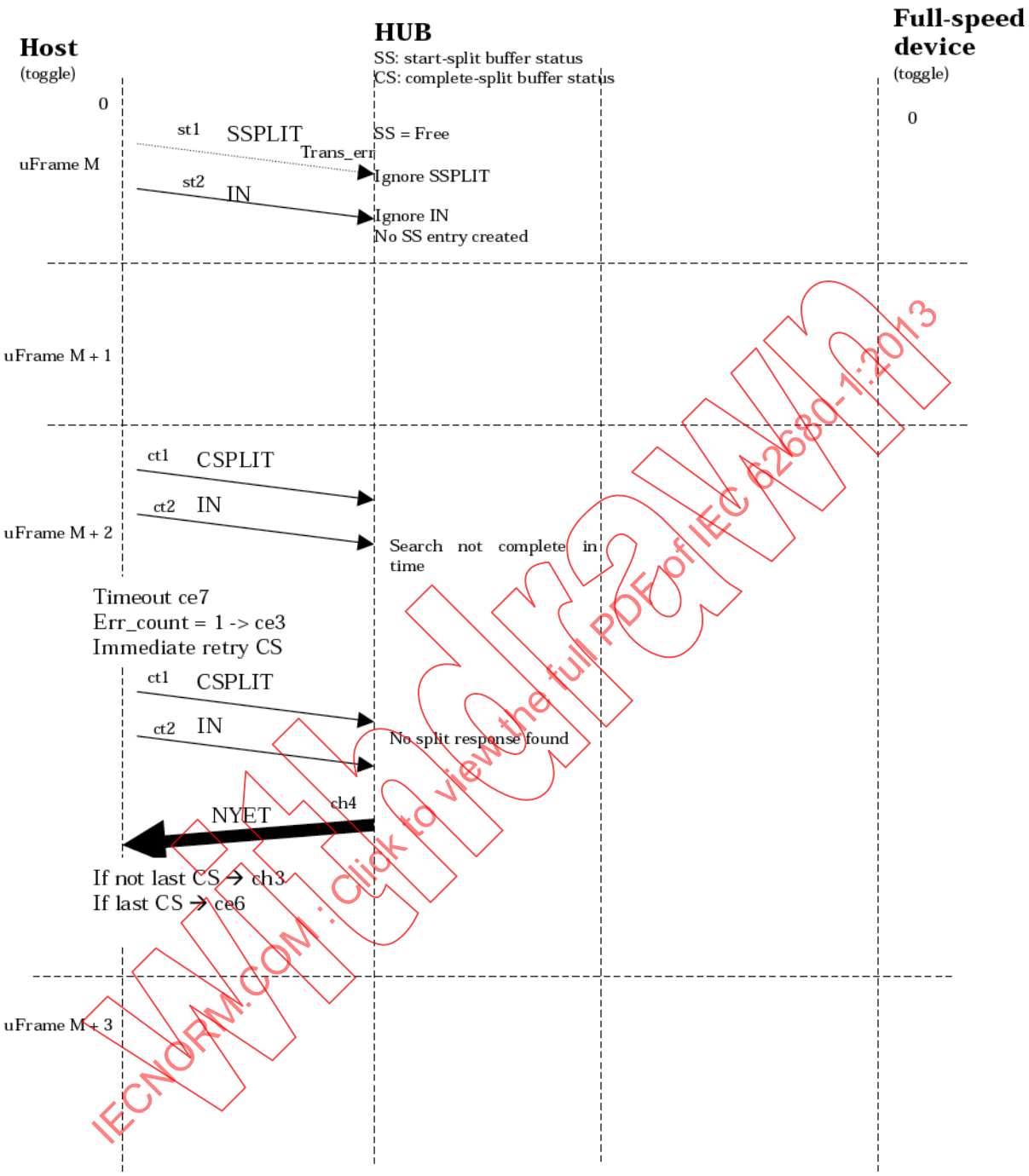
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

2) Normal: full-speed bus transaction crosses microframe boundary



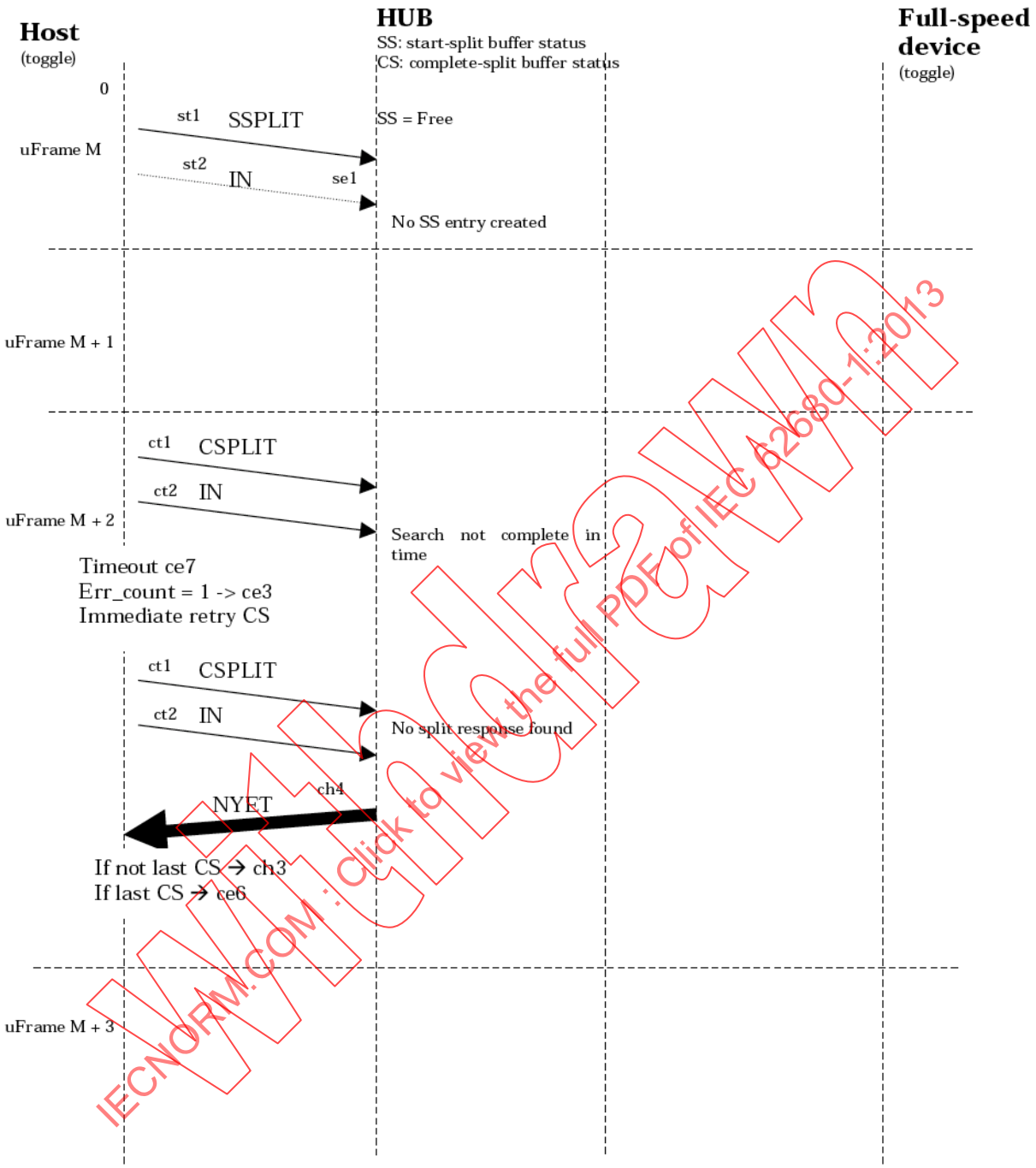
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

3) HS SSPLIT corrupted



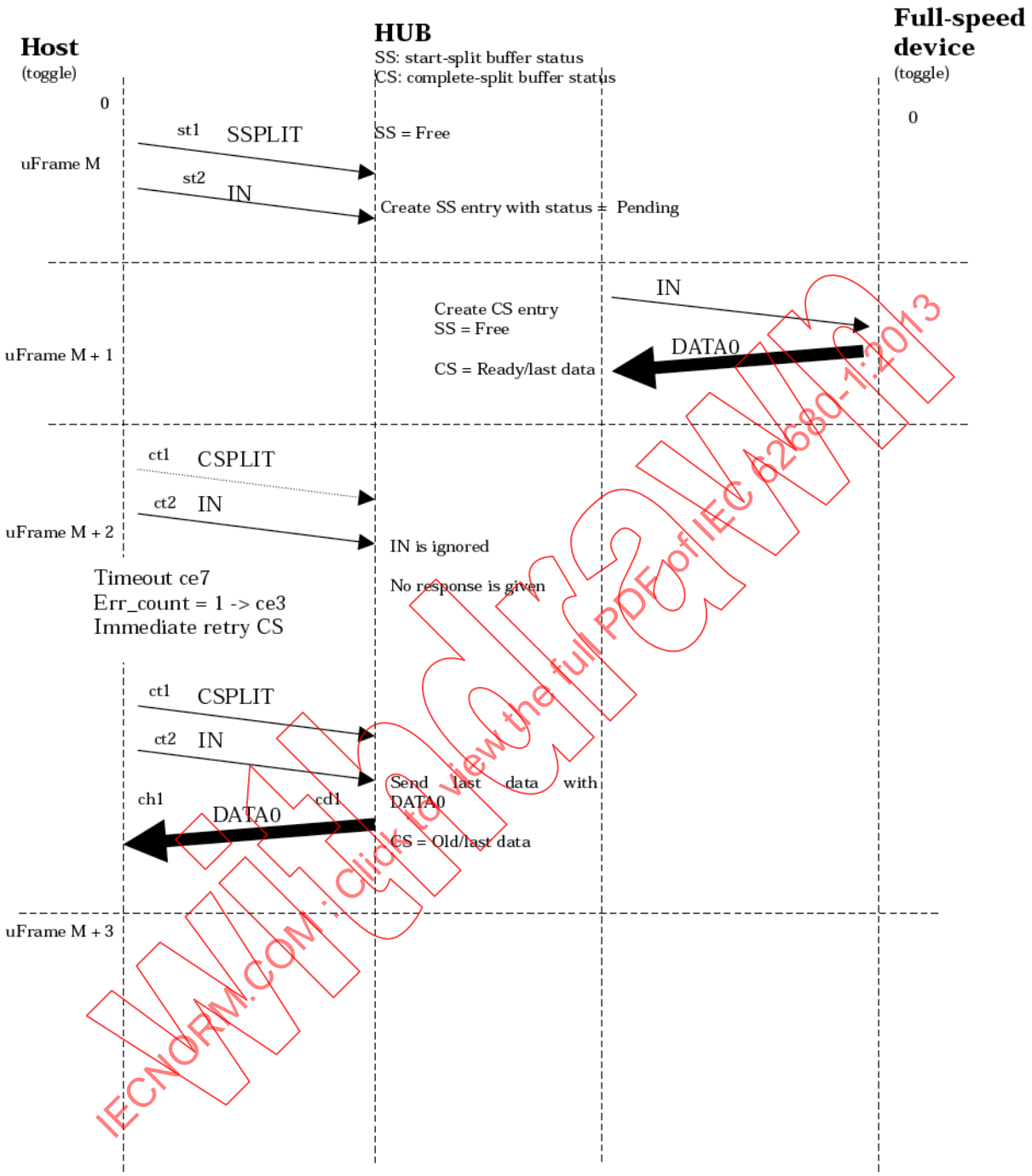
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4) IN after HS SSPLIT corrupted



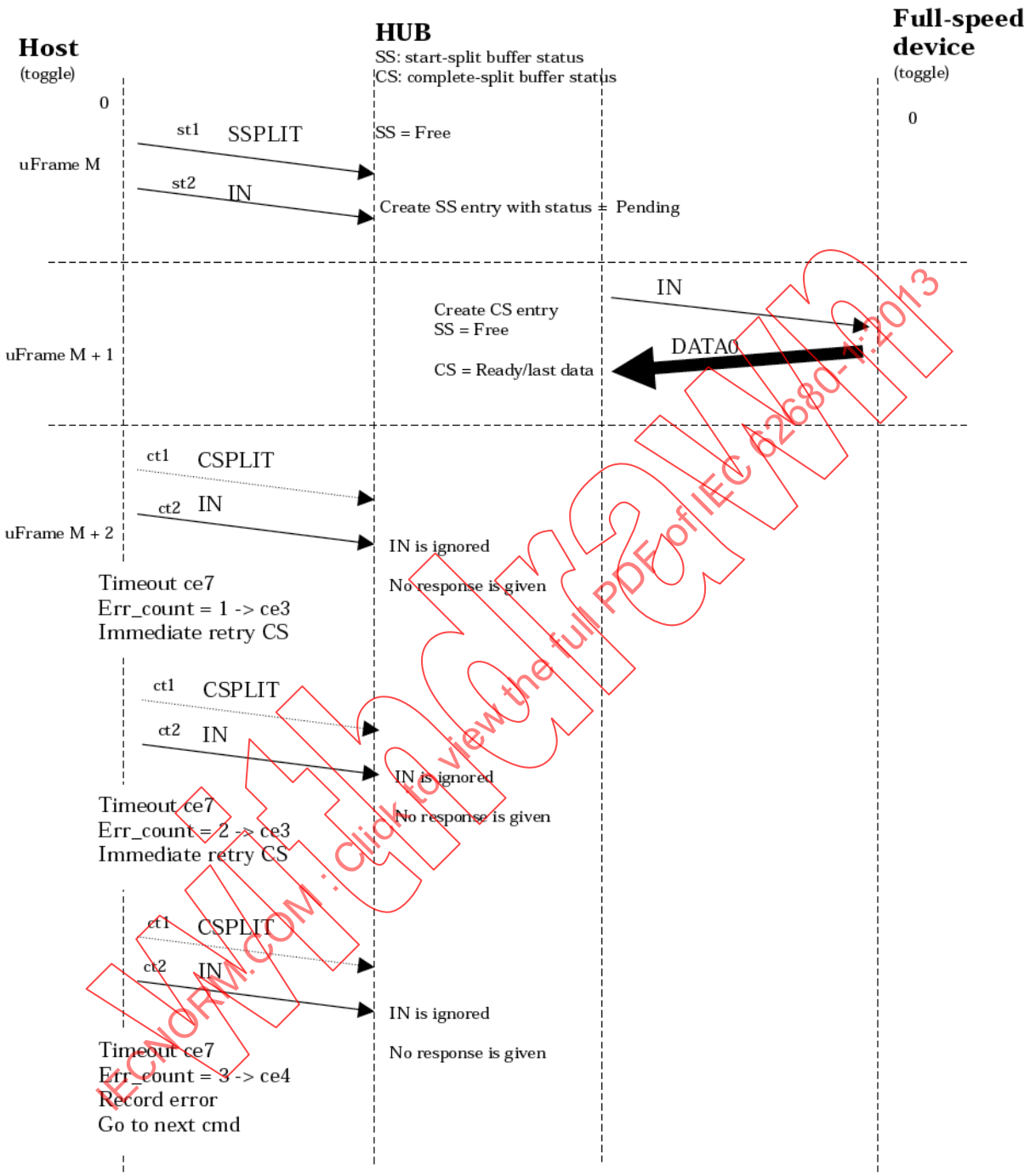
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5) HS CSPLIT corrupted



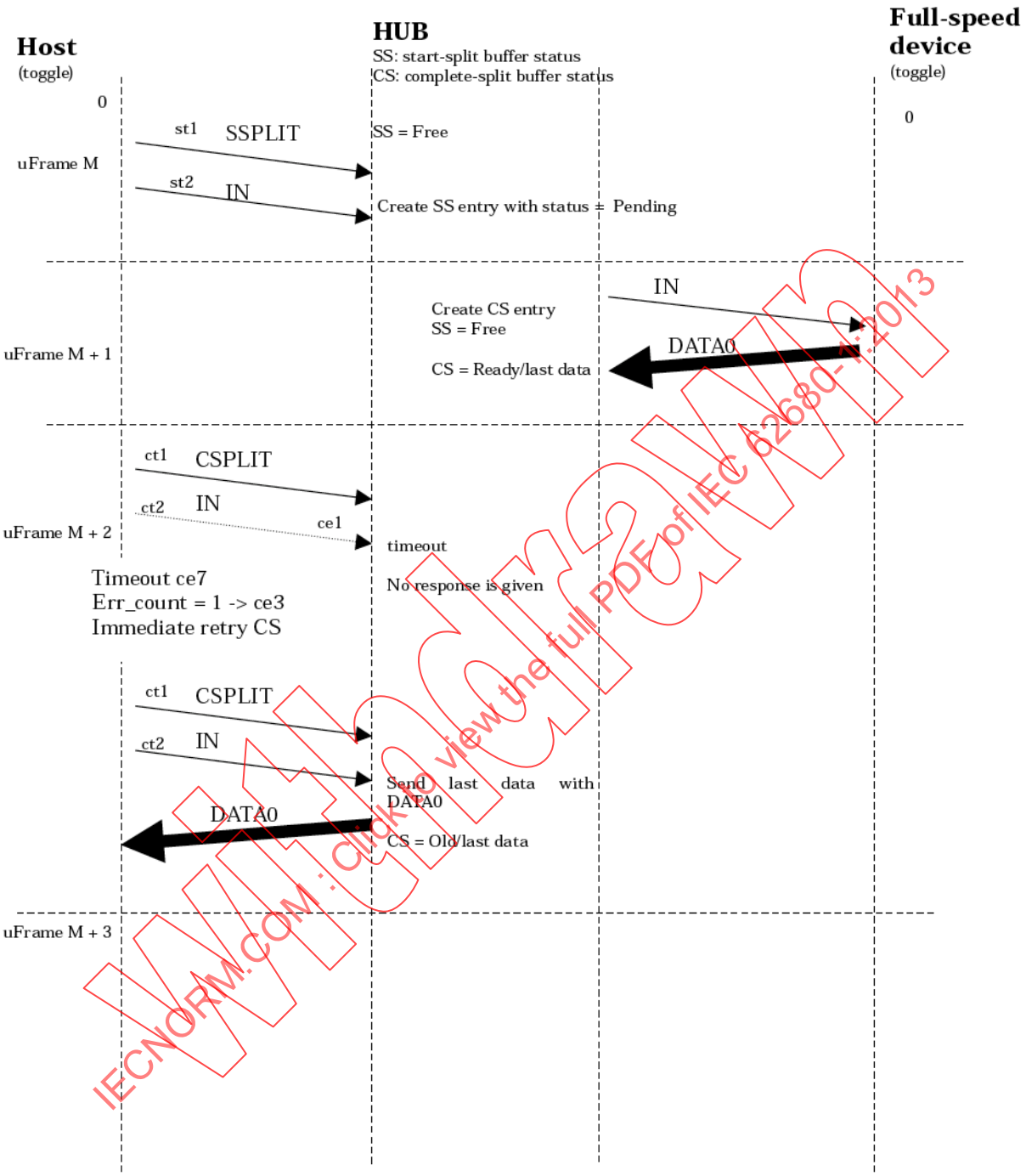
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

6) Consecutive HS CSPLIT corrupted



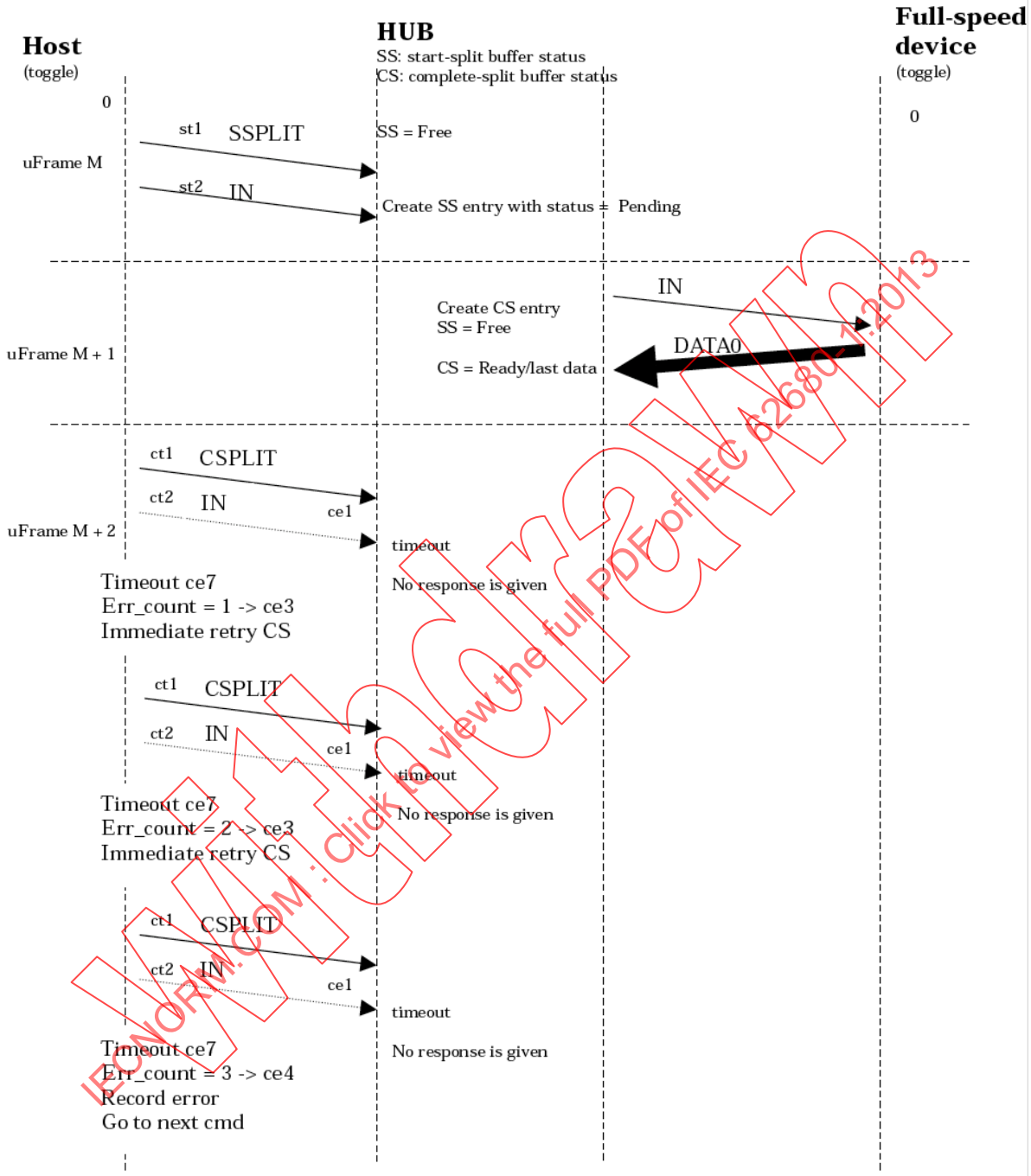
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7) HS IN corrupted



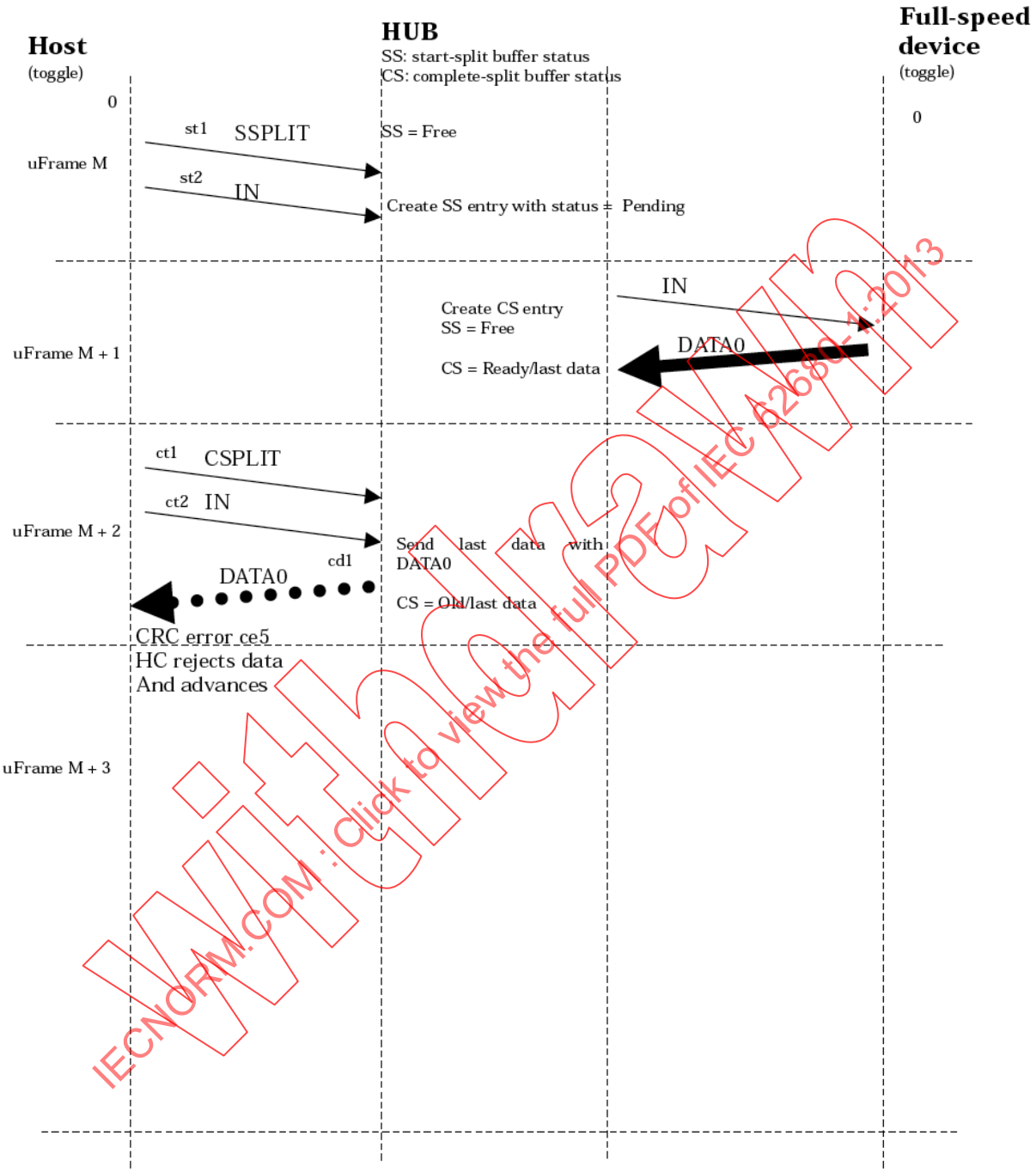
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8) Consecutive HS IN corrupted



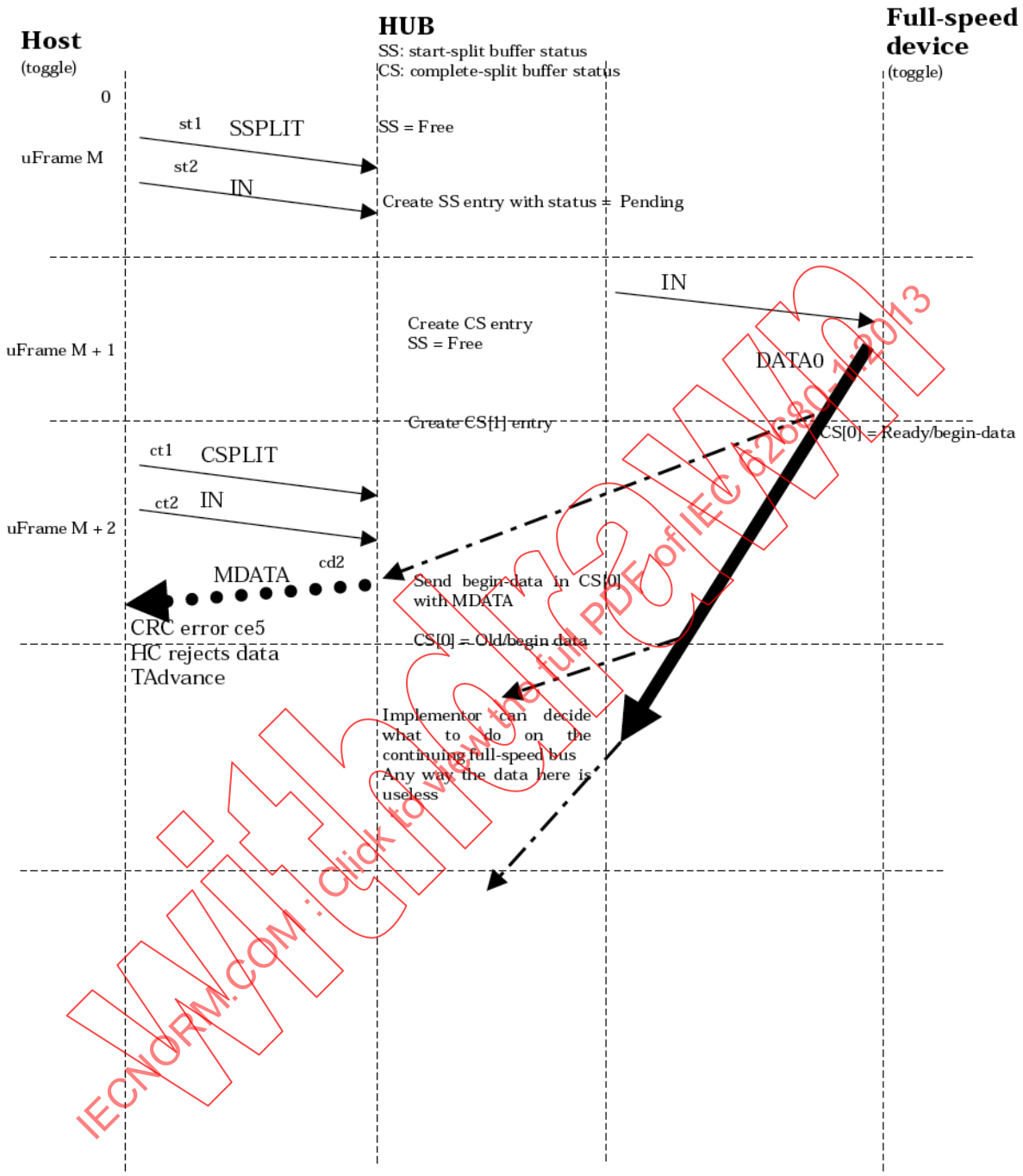
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9) HS data corrupted (case 1)



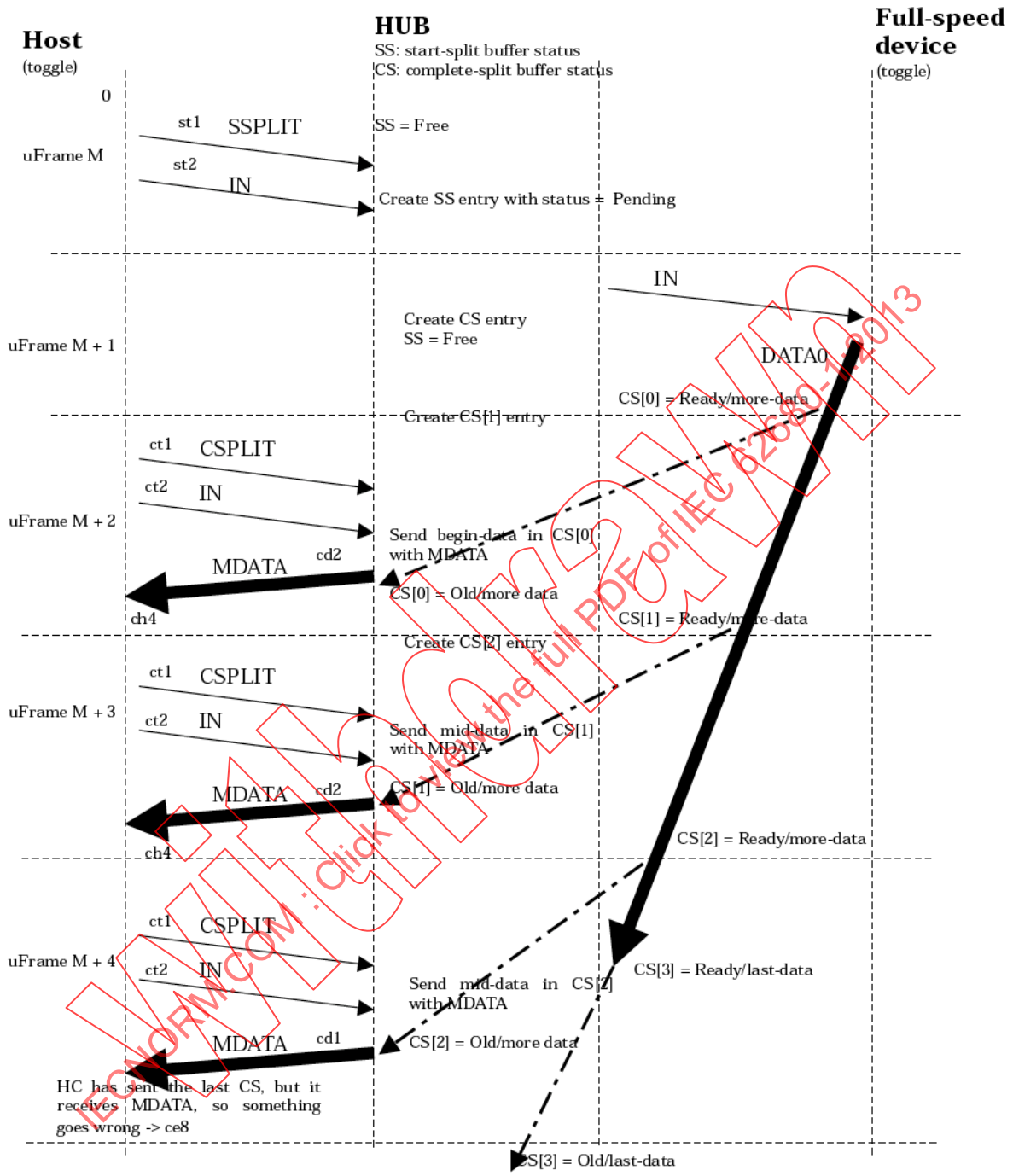
Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

10) HS data corrupted (case 2)



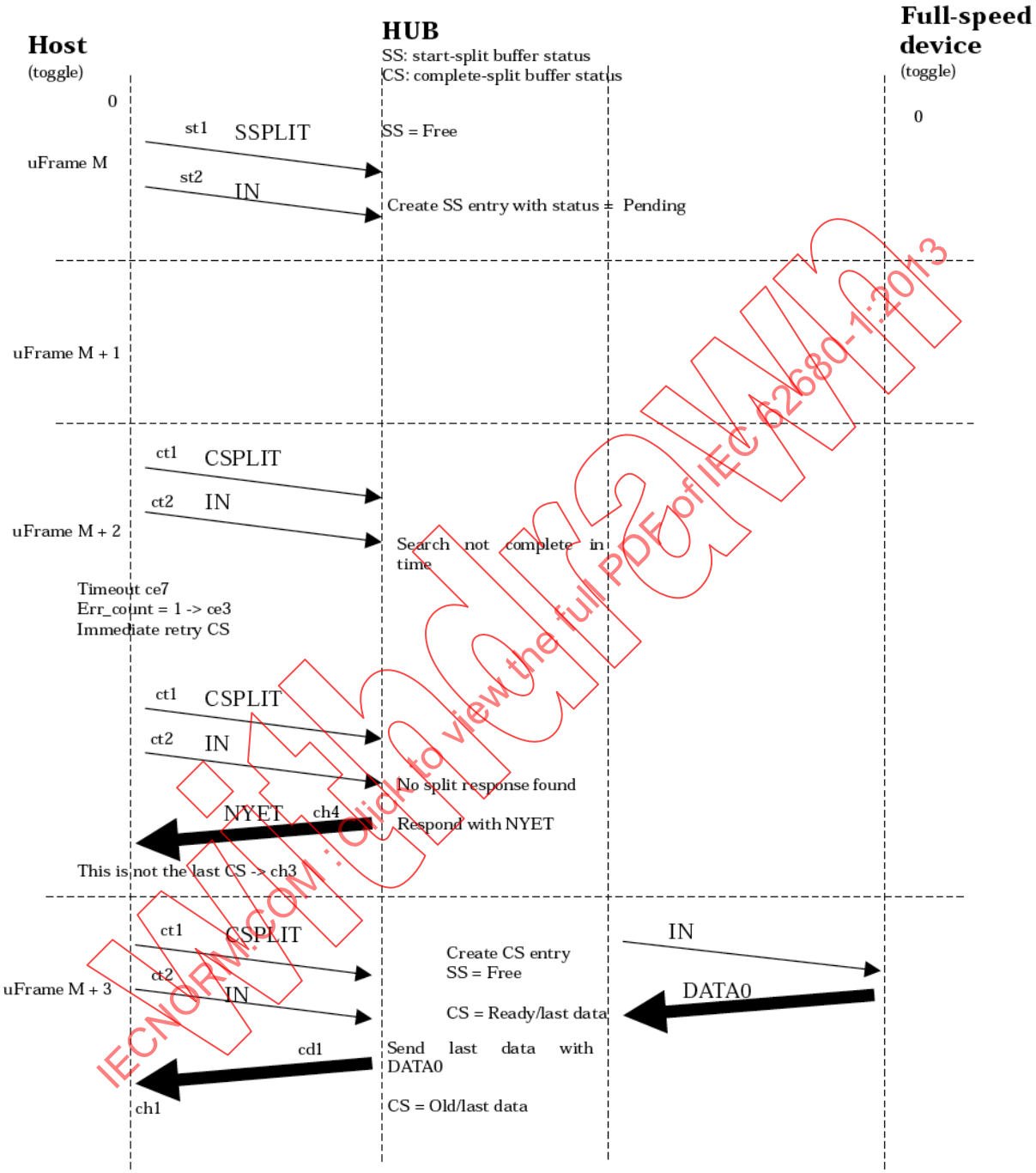
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11) TT has more data than HC expects



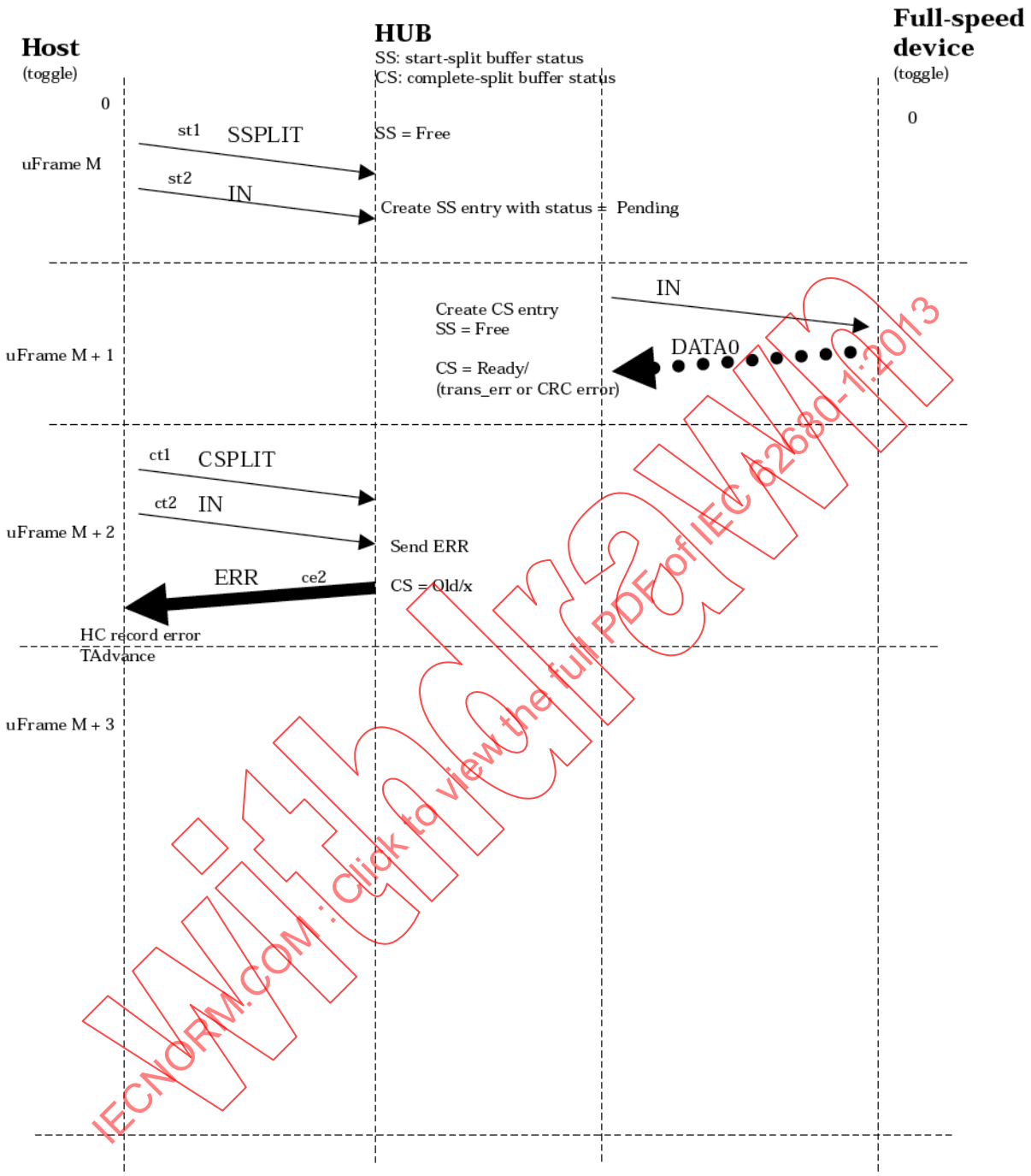
CONFORM.COM : Click to view the full PDF of IEC 62680-1:2013

12) HS CS too early (full-speed data not available yet)



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

13) Full-speed timeout or CRC error



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Appendix B Example Declarations for State Machines

This appendix contains example declarations used in the construction of the state machines in Chapters 8 and 11. These declarations may help in understanding some aspects of the state machines. There are three sets of declarations: global declarations, host controller specific declarations, and transaction translator declarations.

B.1 Global Declarations

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;

PACKAGE behav_package IS

    CONSTANT FIFO_DEPTH  : INTEGER := 3;      -- Size of bulk buffer.
                                              -- Determines how many outstanding
                                              -- Split transactions are allowed.

    CONSTANT ERROR_INJECT_DEPTH  : INTEGER := 16;     -- Size of Error Inject FIFO.

    TYPE ep_types IS (bulk, control, isochronous, interrupt); -- endpoint types

    TYPE directions IS (in_dir, out_dir);  -- data transfer directions

    TYPE pids IS (NAK, ACK, STALL,         -- possible packet PIDs
      tokenIN, tokenOUT, tokenSETUP,
      SOF, ping,
      MDATA,
      DATAx,                    -- represents both DATA0 and DATA1
      CSPLIT, SSPLIT,
      NYET, ERR,
      TRANS_ERR);               -- pseudo PIDs for error cases

    TYPE cmds IS (start_split, complete_split, nonsplit, SOF);       -- HC commands

    TYPE data_choices IS (alldata, begindata, enddata, middata);
      -- isochronous data part for an HC command

    TYPE HCresponses IS (       -- what HC should do next for this command
      do_start,                 -- do start-split transaction
      do_complete,              -- do complete-split transaction
      do_complete_immediate,
          -- do complete-split immediately before doing a different transaction
      do_halt,
          -- do endpoint halt processing for the endpoint of this command
      do_next_cmd,              -- do next command for this endpoint
                                -- advance data pointer appropriately
      do_same_cmd,              -- do same command over again
      do_comp_immed_now,
          -- do complete-split imediately within same microframe
      do_next_complete,
          -- do next complete-split in next microframe (periodic)
      do_next_ping,
      do_ping,
      do_out,
      do_idle                   -- Response not active - Used for Simulation
    );

    TYPE Devresponses IS (
      do_next_data,
      do_nothing
    );
```

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

```
TYPE waits IS (
  ITG,                    -- wait up to an inter packet (intra transaction) gap
                          --  for the next packet.
  none);                  -- wait forever for next packet

TYPE CRCs IS (bad, ok);

TYPE states IS (old, pending, ready, no_match, match_busy);
-- states of a buffer

TYPE results IS (          -- full/low speed transaction result in a buffer
        r_ack,
        r_nak,
        r_trans_err,
        r_stall,
        r_badcrc,
        r_lastdata,
        r_moredata,
        r_data);

TYPE epinfo_rec IS RECORD
        space_avail     : boolean;
        data_avail      : boolean;
        ep_type         : ep_types;
        ep_trouble      : boolean;
        toggle          : boolean;
        END RECORD;

TYPE epinfo_array IS ARRAY(1 DOWNTO 0) OF epinfo_rec;

TYPE device_rec IS RECORD
        ep              : epinfo_array;
        HS             : BOOLEAN;
        END RECORD;


TYPE match_rec IS RECORD  -- result of matching a high-speed complete-split
        state           : states;
        down_result     : results;
        END RECORD;

TYPE HS_bus_rec IS RECORD
-- partial high speed transaction state from a high speed bus
        ep_type         : ep_types;
        PID             : pids;
        dev_addr        : INTEGER RANGE 0 TO 127;
        endpt           : INTEGER RANGE 0 TO 15;
        CRC16           : CRCs;
        direction       : directions;
        x               : boolean;
        datapart        : data_choices;
        ready           : boolean;
        timeout         : boolean;
        END RECORD;

TYPE command_rec IS RECORD          -- command state that the HC must act upon
        ep_type         : ep_types;
        cmd             : cmds;
        setup           : boolean;        -- true is control setup
        ping            : boolean;
        HS              : boolean;
        dev_addr        : INTEGER RANGE 0 TO 127;
        endpt           : INTEGER RANGE 0 TO 15;
        CRC16           : CRCs;
        direction       : directions;
        datapart        : data_choices;
        toggle            : boolean;
        last              : boolean;
        END RECORD;
```

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

```
    TYPE bc_buf_status IS (OLD,NU,NOSPACE);     -- Responses from Compare_BC_buff.

    TYPE BC_buff_rec IS RECORD       -- (partial) state of a bulk/control buffer
           match           : match_rec;
           index           : INTEGER RANGE 0 TO (FIFO_DEPTH-1);
           status          : bc_buf_status;
           END RECORD;

    TYPE CS_buff_rec IS RECORD
      -- (partial) state of a periodic complete-split buffer
           match           : match_rec;
           store           : hs_bus_rec;
           END RECORD;

    TYPE SS_buff_rec IS RECORD
           saw_split: boolean;
           isochO:    boolean;  -- was the last transaction an isochronous OUT SS
           lastdata:  data_choices;
             -- if isochO is true, then what was the last data portion
           END RECORD;

    TYPE cam_rec IS RECORD       -- Information stored in the bulk/control Buffer.
           store           : hs_bus_rec;
           match           : match_rec;
           END RECORD;

    TYPE phases IS (SPLIT, TOKEN, DATA);     -- Error Inject phases.

    TYPE err_inject_rec IS RECORD            -- Error Injection FIFO record.
           phase           : phases;
           timeout         : boolean;
           crc             : CRCs;
           pid             : boolean;
           END RECORD;

    TYPE err_inject_type IS ARRAY((ERROR_INJECT_DEPTH - 1) DOWNTO 0)
      of err_inject_rec;

    TYPE cam_type IS ARRAY((FIFO_DEPTH - 1) DOWNTO 0) OF cam_rec;

    --returns true when there is a packet ready to receive from a bus

    FUNCTION Packet_ready(HS_bus_in: HS_bus_rec) RETURN boolean;

    -- wait until there is a packet ready on a bus
    PROCEDURE Wait_for_packet(HS_bus_in: HS_bus_rec; wait_type: waits);

    PROCEDURE RespondDev(dr: devresponses);

    PROCEDURE HC_Accept_data;

    PROCEDURE HC_Reject_data;

    PROCEDURE Dev_Accept_data;

    PROCEDURE Dev_Record_error;
END behav_package;
```

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

B.2 Host Controller Declarations

```
    shared VARIABLE ErrorCount      :  integer :=0;
    shared VARIABLE HC_response_v   : HCresponses;
    shared VARIABLE rd_ptr          : integer RANGE 0 TO (ERROR_INJECT_DEPTH-1) := 0;
    SIGNAL HSU2_ready               : boolean;
    SIGNAL HC_command_ready         : boolean := FALSE;
    SIGNAL HC_cmd                   : command_rec;
    SIGNAL HCresponse               : HCresponses;
    SIGNAL err_inject_fifo          : err_inject_type;
    SIGNAL wr_ptr                   : integer RANGE 0 TO (ERROR_INJECT_DEPTH-1) := 0;

    ------------------------------------------------------------------------------
    -- Issue a packet onto the HS bus.
    ------------------------------------------------------------------------------
    PROCEDURE Issue_packet(SIGNAL HS_bus_out : OUT HS_bus_rec;
                           pid        : pids) IS
    BEGIN

        HS_bus_out.ep_type  <= HC_cmd.ep_type;
        HS_bus_out.endpt    <= HC_cmd.endpt;
        HS_bus_out.dev_addr <= HC_cmd.dev_addr;
        HS_bus_out.direction <= HC_cmd.direction;
        HS_bus_out.datapart <= HC_cmd.datapart;

        HS_bus_out.x        <= HC_cmd.toggle;     -- ????

        -- Check for Error injection when FIFO is not empty.
        IF (wr_ptr /= rd_ptr) THEN

            -- Insert an error during SPLIT phase ?
            IF ((err_inject_fifo(rd_ptr).phase = SPLIT AND (pid = SSPLIT OR pid =
CSPLIT)) OR
            -- Insert an error during Token phase ?
                (err_inject_fifo(rd_ptr).phase = TOKEN AND
                  (pid = tokenIN OR pid = tokenOUT OR pid = tokenSETUP)) OR
            -- Insert an error during Data phase ?
                (err_inject_fifo(rd_ptr).phase = DATA AND (pid = MDATA OR pid = DATAx)))
THEN

                HS_bus_out.crc16   <= err_inject_fifo(rd_ptr).crc;
                HS_bus_out.timeout <= err_inject_fifo(rd_ptr).timeout;
                IF (err_inject_fifo(rd_ptr).pid) THEN
                    HS_bus_out.pid     <= TRANS_ERR;
                ELSE
                    HS_bus_out.pid     <= pid;
                END IF;

                -- Update read pointer.
                IF (rd_ptr = (ERROR_INJECT_DEPTH-1)) THEN
                    rd_ptr := 0;
                ELSE
                    rd_ptr := rd_ptr + 1;
                END IF;
            ELSE
                -- Otherwise issue packet with no errors.
                HS_bus_out.crc16    <= ok;
                HS_bus_out.timeout  <= FALSE;
                HS_bus_out.pid      <= pid;

            END IF;

        -- Otherwise issue packet with no errors.
        ELSE
            HS_bus_out.crc16    <= ok;
            HS_bus_out.timeout  <= FALSE;
            HS_bus_out.pid      <= pid;
        END IF;
        HS_bus_out.ready    <= TRUE;
        HS_bus_out.ready    <= FALSE after 500 ps;
```

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

```
END Issue_packet;

-------------------------------------------------------------------------------
-- Get next command for HC to execute.
-- NOT USED FOR THIS IMPLEMENTATION !!!
-------------------------------------------------------------------------------
PROCEDURE HC_Get_next_command IS
BEGIN
END;

-------------------------------------------------------------------------------
-- Tells HC what happened to this command.
-------------------------------------------------------------------------------
PROCEDURE RespondHC (HCresponse : HCresponses) IS
BEGIN
    HC_response_v := HCresponse;
END;

-------------------------------------------------------------------------------
-- Update command status for the next time the command will be executed by HC.
-- NOT USED FOR THIS IMPLEMENTATION !!!
-------------------------------------------------------------------------------
PROCEDURE Update_command (SIGNAL HCdone : OUT boolean) IS
BEGIN
    HCdone <= TRUE;
END;

-------------------------------------------------------------------------------
-- Increment "3 strikes" error count for endpoint transaction.
-------------------------------------------------------------------------------
PROCEDURE IncError IS
BEGIN
    ErrorCount := ErrorCount + 1;
END;

-------------------------------------------------------------------------------
-- Record Error for current command.
-- NOT USED FOR THIS IMPLEMENTATION !!!
-------------------------------------------------------------------------------
PROCEDURE Record_error IS
BEGIN
    ErrorCount := 0;

END;
```

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

B.3 Transaction Translator Declarations

```
shared VARIABLE cam          : cam_type;         -- TT buffer.
shared VARIABLE BC_buff      : BC_buff_rec;
shared VARIABLE CS_Buff      : CS_buff_rec;
shared VARIABLE rd_ptr       : integer RANGE 0 TO (ERROR_INJECT_DEPTH-1) := 0;
shared VARIABLE derror_v     : boolean;
shared VARIABLE ss_avail_v   : boolean;
shared VARIABLE periodic     : boolean := FALSE;
shared VARIABLE error_time   :time := 1000000000 ns;
SIGNAL split                 : HS_bus_rec;       -- Stored Shared Split Token
SIGNAL token                 : HS_bus_rec;       -- Stored Token
SIGNAL SS_Buff               : SS_buff_rec;
SIGNAL CS_Buff_sig           : CS_buff_rec;
SIGNAL mem                   : cam_rec;
SIGNAL memwrite              : boolean;
SIGNAL err_inject_fifo       : err_inject_type;
SIGNAL wr_ptr                : integer RANGE 0 TO (ERROR_INJECT_DEPTH-1) := 0;
SIGNAL derror                : boolean;
SIGNAL ss_avail              : boolean;

-------------------------------------------------------------------------------
-- Is_no_space - Returns true when there is no space in the Bulk/Control buffers
--               for the current start-split.
-------------------------------------------------------------------------------
function Is_no_space(BC_buff: BC_buff_rec) return boolean is
  variable result:boolean:=FALSE;
begin
  IF (BC_buff.status = NOSPACE) THEN
      result := TRUE;
  END IF;
  return result;
end Is_no_space;

-------------------------------------------------------------------------------
-- Is_new_SS - Returns true when the current high speed start-split is new.
-------------------------------------------------------------------------------
function Is_new_SS(BC_buff: BC_buff_rec) return boolean is
  variable result:boolean:=FALSE;
begin
  IF (BC_buff.status = NU) THEN
      result := TRUE;
  END IF;
  return result;
end Is_new_SS;

-------------------------------------------------------------------------------
-- IS_old_SS - Returns true when the current high speed start-split is a retry.
-------------------------------------------------------------------------------
function Is_old_SS(BC_buff: BC_buff_rec) return boolean is
  variable result:boolean:=FALSE;
begin
  IF (BC_buff.status = OLD) THEN
      result := TRUE;
  END IF;
  return result;
end Is_old_SS;

-------------------------------------------------------------------------------
-- Issue_packet - Issue a packet onto the HS bus.
-------------------------------------------------------------------------------
procedure Issue_packet(signal HS_bus_out : out HS_bus_rec;
                               pid       : pids) IS
begin

  -- Setup HS packet based on whether its periodic or bulk.
    IF (periodic = TRUE) THEN
        HS_bus_out.ep_type      <= CS_Buff.store.ep_type;
        HS_bus_out.endpt        <= CS_Buff.store.endpt;
        HS_bus_out.dev_addr     <= CS_Buff.store.dev_addr;
```

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

```
        HS_bus_out.direction    <= CS_Buff.store.direction;
        HS_bus_out.datapart     <= CS_Buff.store.datapart;
        HS_bus_out.x            <= CS_Buff.store.x;     -- ????
    ELSE
        HS_bus_out.ep_type      <= cam(BC_buff.index).store.ep_type;
        HS_bus_out.endpt        <= cam(BC_buff.index).store.endpt;
        HS_bus_out.dev_addr     <= cam(BC_buff.index).store.dev_addr;
        HS_bus_out.direction    <= cam(BC_buff.index).store.direction;
        HS_bus_out.datapart     <= cam(BC_buff.index).store.datapart;
        HS_bus_out.x            <= cam(BC_buff.index).store.x;     -- ????

        -- Update bulk/control with state information which may have been updated
        -- by the complete-split state machines.
        cam(BC_buff.index).match.state := BC_buff.match.state;
    END IF;


    -- Check for Error injection when FIFO is not empty.
    IF (wr_ptr /= rd_ptr) THEN

        HS_bus_out.crc16   <= err_inject_fifo(rd_ptr).crc;
        HS_bus_out.timeout <= err_inject_fifo(rd_ptr).timeout;
        IF (err_inject_fifo(rd_ptr).pid) THEN
            HS_bus_out.pid     <= TRANS_ERR;
        ELSE
            HS_bus_out.pid      <= pid;
        END IF;

        --IF (now > error_time ) THEN
            -- Update read pointer.
            IF (rd_ptr = (ERROR_INJECT_DEPTH-1)) THEN
               rd_ptr := 0;
            ELSE
                rd_ptr := (rd_ptr + 1);
            END IF;
        --END IF;

        error_time := now;

    -- Otherwise issue packet with no errors.
    ELSE
        HS_bus_out.crc16    <= ok;
        HS_bus_out.timeout  <= FALSE;
        HS_bus_out.pid      <= pid;
    END IF;

    HS_bus_out.ready        <= TRUE;
    HS_bus_out.ready        <= FALSE after 500 ps;


end Issue_packet;

-- returns true when wrong combination of split start and last isoch out transaction
FUNCTION Bad_IsochOut (SS_Buff : SS_Buff_rec;
                       split : HS_bus_rec) RETURN boolean IS
  VARIABLE result:boolean:=FALSE;
BEGIN

  result := ((split.datapart = enddata OR split.datapart = middata) AND
             NOT(SS_Buff.lastdata = begindata OR SS_Buff.lastdata = middata)) OR
            ((split.datapart = begindata OR split.datapart = alldata) AND
             SS_Buff.isochO) OR
            ((split.datapart = middata OR split.datapart = enddata) AND NOT
             SS_Buff.isochO);

  RETURN result;
END Bad_IsochOut;

--------------------------------------------------------------------------------
-- Save - Save the Packet for use later.
--------------------------------------------------------------------------------
```

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

```
procedure Save(hs_bus_in : IN HS_bus_rec;
               SIGNAL hs_bus_out: OUT HS_bus_rec) IS
begin
    hs_bus_out <= hs_bus_in;
end Save;

--------------------------------------------------------------------------------
--  Compare_BC_buff - This procedure is used to look at the BC buffer to determine
--                    whether the packet should be stored. Compare_BC_buff will
--                    initialize BC_buff with the buffer location information.
--------------------------------------------------------------------------------
procedure Compare_BC_buff IS
  variable match:boolean:=FALSE;
begin

   -- Assume nospace and intialize index to 0.
   BC_buff.status := NOSPACE;
   BC_buff.index  := 0;

   FOR i IN 0 to FIFO_DEPTH-1 LOOP
       IF NOT match THEN
           -- Re-use buffer with same Device Address/End point.
           IF (token.endpt = cam(i).store.endpt AND
                token.dev_addr = cam(i).store.dev_addr AND
               ((token.direction = cam(i).store.direction AND
                 split.ep_type /= CONTROL) OR
                split.ep_type = CONTROL)) THEN

             -- If The buffer is already pending/ready this must be a retry.
             IF (cam(i).match.state = READY OR cam(i).match.state = PENDING) THEN
                 BC_buff.status := OLD;
             ELSE
                 BC_buff.status := NU;
             END IF;
             BC_buff.index := i;
             match := TRUE;

           -- Otherwise use the buffer if it's old.
           ELSIF (cam(i).match.state = OLD) THEN
             BC_buff.status := NU;
             BC_buff.index := i;
           END IF;
       END IF;
   END LOOP;

   BC_buff.match.state := cam(BC_buff.index).match.state;

end Compare_BC_buff;

--------------------------------------------------------------------------------
-- Accept_data - Store start-split into bulk/control buffer. Index is setup
--               in a previous call to Compare_BC_buff.
--------------------------------------------------------------------------------
procedure Accept_data IS
begin
    cam(BC_buff.index).store := token;
    cam(BC_buff.index).match.state := PENDING;
    BC_buff.match.state := PENDING;
end Accept_data;

--------------------------------------------------------------------------------
-- Match_split_state - This procedure finds the BC buffer location which matches
--                     the current complete-split.
--------------------------------------------------------------------------------
procedure Match_split_state IS
  variable match:boolean:=FALSE;
begin
   BC_buff.match.state  := NO_MATCH;
   BC_buff.index  := 0;

   FOR i IN 0 to FIFO_DEPTH-1 LOOP
```

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

```
        IF NOT match THEN
            -- Is this the buffer used for the start-split
            --  corresponding to this complete-split?
            -- If it is... store information into BC_buff and
            -- indicate match was found.

            IF (token.endpt = cam(i).store.endpt AND
                token.dev_addr = cam(i).store.dev_addr AND
                token.direction = cam(i).store.direction) THEN

              BC_buff.match.state       := cam(i).match.state;
              BC_buff.match.down_result := cam(i).match.down_result;
              BC_buff.index := i;
              match := TRUE;
            END IF;
        END IF;
    END LOOP;

    periodic := FALSE;                -- Setup Issue Packet.

end Match_split_state;

-------------------------------------------------------------------------------
-- Record an error in the SS pipeline for fowarding on the downstream bus.
-------------------------------------------------------------------------------
PROCEDURE Down_error IS
BEGIN
    derror_v := TRUE;
END Down_error;

-------------------------------------------------------------------------------
--
-------------------------------------------------------------------------------
procedure Data_into_SS_pipe IS
begin
    CS_Buff.match.state := MATCH_BUSY;
    ss_avail_v := TRUE;
end Data_into_SS_pipe;

-------------------------------------------------------------------------------
--
-------------------------------------------------------------------------------
procedure Fast_match IS
begin

   periodic := TRUE;                -- Setup Issue Packet.
end Fast_match;
```

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Appendix C Reset Protocol State Diagrams

This appendix presents state diagrams that provide implementation examples for the reset protocol as described in Section 7.1.7.5. These state diagrams should be considered as an example to guide implementers; the description of the reset protocol and the high-speed reset handshake in Section 7.1.7.5 is the complete required behavior. By necessity, state diagrams incorporate some implementation dependent parts that, although describing the reset protocol correctly, can also be implemented in a different way yielding similar behavior.

Any timer used in these state diagrams should have a resolution that allows it to always keep to the allowed time frame. For instance, if a timer times out between a time T_{TIMER}(min) and T_{TIMER}(max), the timer should have a minimal resolution of at least 1 clocktick in the range of T_{TIMER}. In a number of places, a time T_{TIMER} is mentioned in a state diagram; while in the tables in Section 7.3, a range is given for this time. In that case, the time represents a chosen value in the range such that it is at least 1 clocktick of the associated timer away from the upper boundary of that range. Under these conditions, a state in the state diagrams will never miss a branch because the associated timer overstepped the time-out condition.

In the state diagrams in this appendix, a timer can be either Run, Started, or Cleared. If a timer is Run, it will update itself every clocktick. If a timer is Cleared, it is stopped and its contents are reset to zero. A timer that is Started is first cleared and then immediately run. Stopping of a timer is never done explicitly in the state diagrams.

C.1 Downstream Facing Port State Diagram

This section describes the reset protocol state diagram for the downstream facing port.

The state diagram shown in Figure C-1 shows all the necessary and required behavior of a downstream facing port in case of a reset. As this is the initiating party in the reset protocol, the hub enters the Resetting state through a request from the host (the SetPortFeature(PORT_RESET) command). The downstream facing port then drives an SE0 to initiate the reset and at the same time starts a timer T0 to time the whole reset procedure.

If the attached device is low-speed, then the only way that reset ends is when the timer T0 times out (T_{DRST}) and the bus returns to idle. Whether a device is low-speed is determined prior to entering the Resetting state in the status bit PORT_LOW_SPEED. This is described in more detail in Section 11.8.2. When reset has completed, the hub enters the low-speed Enabled state.

If the attached device is full-speed and not high-speed capable, it will end reset when timer T0 expires (T_{DRST}) and the hub has not detected a valid upstream chirp (continuous Chirp K). It will then enter the full-speed enabled state.

Last, if the attached device is high-speed capable, it will send back an upstream chirp some time after the SE0 has been asserted on the bus. The actual time before the upstream chirp starts depends on whether the attached device was suspended or awake at the time the reset started. The loop between the blocks with “Clear timer T1” and “Run timer T1” represents the 2.5 µs (T_{FILT}) filtering the reset protocol asks for.

Note: The timer T1 is required to be reset after an interruption of 16 high-speed bit-times of the continuous Chirp K that makes up the upstream chirp. It may be reset by any shorter interruption.

If the filtering of the upstream chirp takes too much time, the downstream facing port may not be able to finish its downstream chirp in time to be able to end the reset procedure in time. Therefore, when timer T0 reaches beyond the time T_{UCHEND} (time to detect an upstream

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

chirp), the hub is put in a wait state, which it leaves after the timer has timed out the complete reset protocol (T_{DRST}). It will then enter the full-speed enabled state.

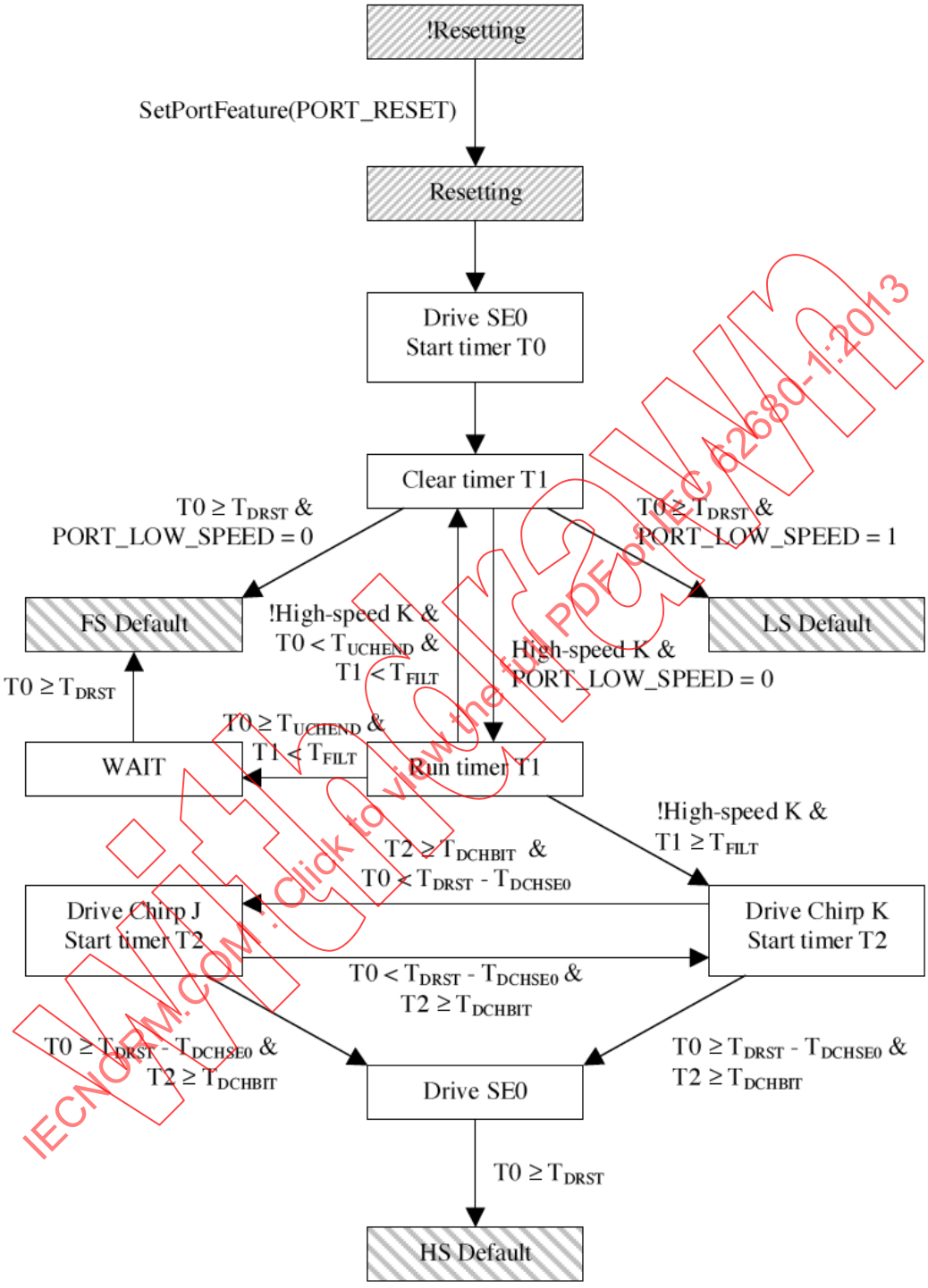


Figure C-1. Downstream Facing Port Reset Protocol State Diagram

When the downstream-facing port has successfully detected an upstream chirp, it will start transmitting the downstream chirp as soon as it has seen the bus leave the Chirp K state. This end of the upstream chirp will return the bus to the SE0 state. So immediately (actually within 100 µs (T_{WTDCH}) after the end of the upstream chirp according to Section 7.1.7.5), the hub drives a Chirp K for 40 to 60 µs (T_{DCHBIT}), then a Chirp J for 40 to 60 µs, then a Chirp K,

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

etc. It continues with this alternating sequence until timer T0 has come within 100 to 500 µs (T_{DCHSE0}) of the end of reset (T_{DRST}). When this time is reached, the downstream-facing port finishes the 40 to 60 µs of continuous signaling it was busy with when the timer T0 exceeds the value of $T_{DRST} - T_{DCHSE0}$ before driving SE0 until the end of reset.

C.2 Upstream Facing Port State Diagram

This section describes the reset protocol state diagrams for the upstream facing port. The state diagram for the upstream facing port is more complicated than the diagram for the downstream facing port as the device can be in any possible state when it receives a reset signal. Therefore, the state diagram has been split into two parts:

- The reset detection state diagram which describes the way a device reacts to reset signaling on its upstream facing port (see Figure C-2)
- The reset handshake state diagram that explains how a high-speed capable device performs a handshake procedure with the hub upstream to communicate each others high-speed capabilities and have both enter a high-speed state at the end of reset (see Figure C-3)

Therefore, all of these states must be covered in the diagram. Also, the fact that for a high-speed capable device a suspend is initially indistinguishable from a reset requires that the state diagram for the upstream facing port addresses the suspend procedure as well.

At the start of the reset, we can be any possible state, but we can collect them into three groups, where each group is handled differently, but all states in the same group handle reset in the same way. The states are as follows:

- Suspended
- Powered, FS Default, FS Address, and FS Configured
- HS Default, HS Address, and HS Configured

These groups of states correspond to an identical list of possibilities as described in Section 7.1.7.5 under item 3 of the reset protocol.

C.2.1 Reset From Suspended State

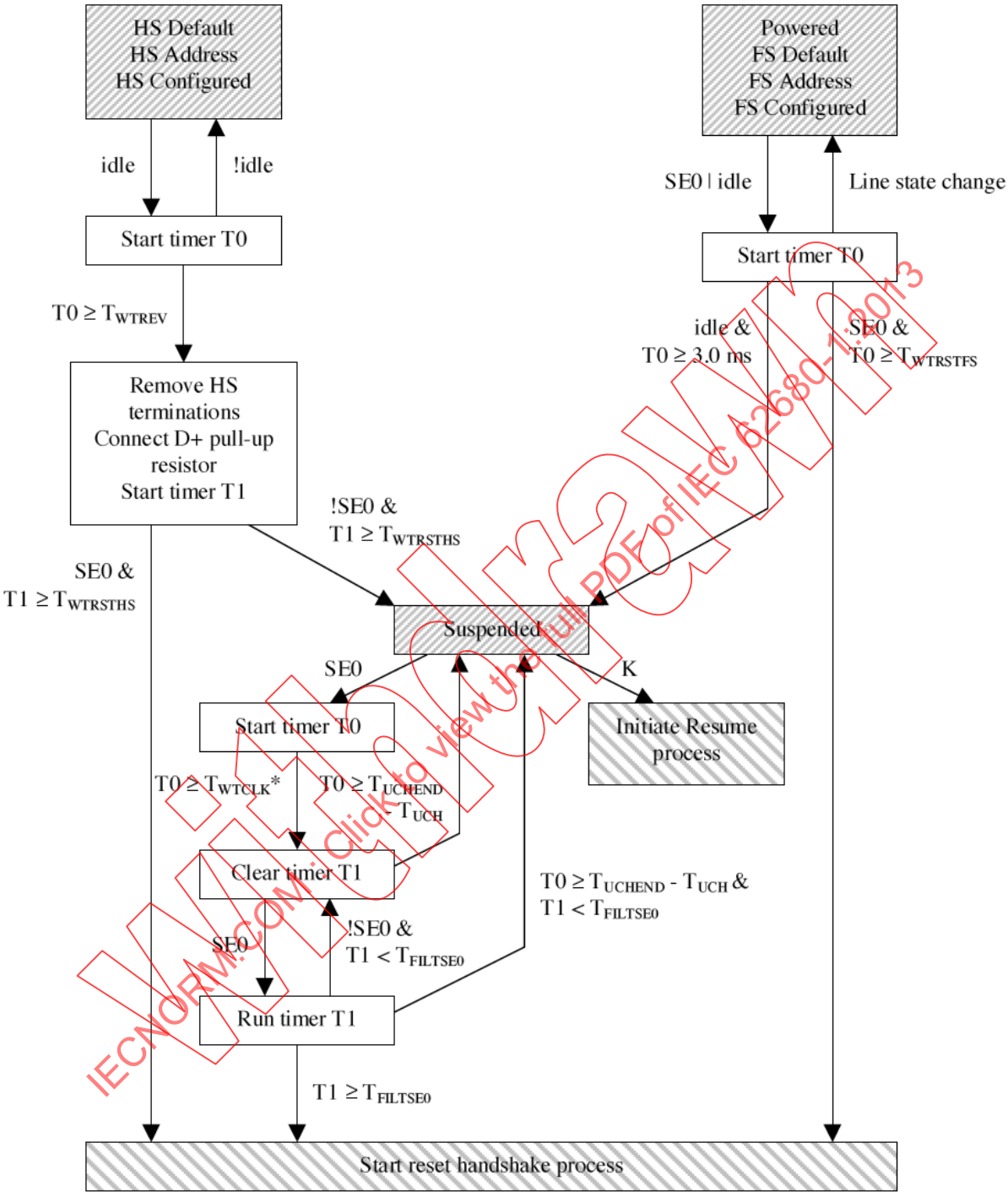
As can be seen from Figure C-2, the device wakes up from the Suspended state as soon as it sees a K or an SE0 on the bus. A J would be indistinguishable from idle on the bus that a suspended device sees normally. On seeing a K, the device will initiate a resume process. For the details of this process, see Section 7.1.7.7. On seeing an SE0, the device could enter the reset handshake procedure, so it starts timer T0.

The actual reset handshake is only started after seeing a continuous assertion of SE0 for at least 2.5 µs ($T_{FILTSE0}$). The loop between the blocks with “Clear timer T1” and “Run timer T1” represents this filtering. If the device has not detected a continuous SE0 before timer T0 exceeds the value of $T_{UCHEND} - T_{UCH}$, the device goes back into the Suspended state.

A device coming from suspend most probably had its high-speed clock stopped to meet the power requirements for a suspended device (see Section 7.2.3). Therefore, it may take some time to let the clock settle to a level of operation where it is able to perform the reset detection and handshake with enough precision. In the state diagram, a time symbol T_{WTCLK} is used to have the device wait for a stable clock. This symbol is not part of the USB 2.0 specification and does not appear in Chapter 7. It is an implementation specific detail of the reset detection state diagram for the upstream facing port, where it is marked with a asterisk (*). T_{WTCLK} should have a value somewhere between 0 and 5.0 ms. This allows at least 1.0 ms time to detect the continuous SE0.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

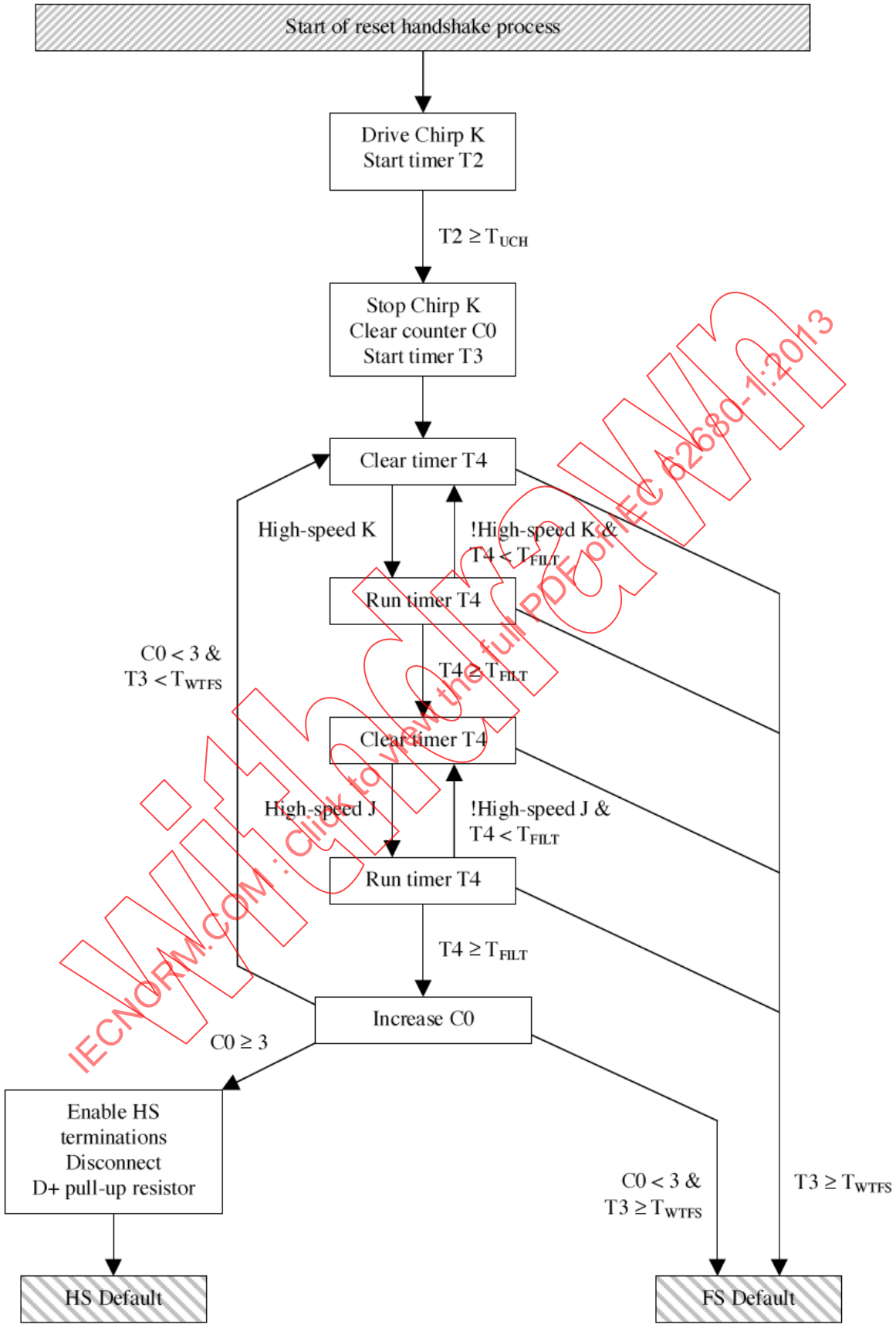
If the device has seen an SE0 signal on the bus for at least $T_{FILTSE0}$, then it can safely assume to have detected a reset and can start the reset handshake.



(*) **Note:** T_{WTCLK} is a symbol that is only used in this state diagram. It is not part of the USB 2.0 specification and does not appear in Chapter 7. It is an implementation specific detail of this state diagram. See Section C.2.1 for a detailed description.

Figure C-2. Upstream Facing Port Reset Detection State Diagram

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure C-3. Upstream Facing Port Reset Handshake State Diagram

C.2.2 Reset From Full-speed Non-suspended State

Timer T0 is started when seeing an SE0 or idle state from a full-speed Non-suspended state.

If a J (idle) is detected and the timer T0 exceeds the value of 3.0 ms while no change has been detected in the state of the bus, the device is suspended.

If an SE0 is detected and the timer T0 times out the value of $T_{WTRSTFS}$ (between 2.5 µs minimum and 3.0 ms maximum) while no change has been detected in the SE0 state, the device can start the reset handshake. On any line state change, the device aborts the detection of reset or suspend from upstream and returns to its previous state.

C.2.3 Reset From High-speed Non-suspended State

Timer T0 is started when seeing a high-speed idle on the bus from a high-speed Non-suspended state. If anything else than idle is detected on the bus, the device aborts detection of a reset and returns to its previous state. When timer T0 exceeds the value of T_{WTREV} (between 3.0 ms minimally and 3.125 ms maximally), the device reverts to full-speed by switching off its high-speed terminations and connecting the D+ pull-up resistor to the D+ line.

The reset protocol allows some time for debouncing and settling of the lines in the new state ($T_{WTRSTHS}$). After this time, the line should be sampled to see whether the device should be suspended (on detecting a full-speed idle) or reset (on detecting SE0).

If an idle was detected, the device should suspend; if an SE0 was detected, the device can start the reset handshake.

If something other than an idle or an SE0, in other words, a K, was detected, the device will also enter the suspended state. However, on seeing the K, the device will immediately resume, effectively returning to the high-speed state.

C.2.4 Reset Handshake

At this point, the behavior of devices has become independent of the initial state they were in when the reset started. The reset handshake is started by the device, when it sends an upstream chirp that is at least 1.0 ms long and stops before the timer T0 hits the 7.0 ms mark. Note: This is the same timer T0 that was started in the reset detection state diagram in Figure C-2.

A choice of implementation is available here. The one presented in the state diagram in Figure C-3 is where a timer T2 is started when the Chirp K is asserted to time the minimum required duration of the upstream chirp. The Chirp K is stopped when timer T2 exceeds the value of T_{UCH}. Another approach would be to wait until the timer T0 exceeds the value of T_{UCHEND}, before ending the upstream chirp. Both conform to the requirements of the reset protocol in Section 7.1.7.5, and the choice may depend on the particular application.

As soon as the upstream chirp has ended, the device starts listening for the downstream chirp. In order to detect at least a K-J-K-J-K-J pattern, it first starts looking for a continuously asserted Chirp K. The method employed in this state diagram is counting the number of K-J transitions. Here K and J are actually Chirp K and Chirp J, respectively, asserted continuously for at least 2.5 µs (T_{FILT}).

Continuous assertion is determined by the loop between the “Clear timer T4” and “Run timer T4”. This is similar to the method used in the downstream facing port state diagram in Figure C-1 to detect the upstream chirp. After this, a continuous Chirp J is detected in the same manner, most likely, even using the same hardware. Now we have detected one K-J transition. Until we have detected three K-J transitions in the same way, we will not revert to high-speed.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

The whole procedure of detecting the downstream chirp is timed by timer T3 which requires the device to perform the detection of the K-J-K-J-K-J for at least 1.0 ms, but at most 2.5 ms. If the device is unable to detect a sufficient number of K-J transitions before the timer T3 times out at T_{WTFS}, the device enters the full-speed default state. Reset ends when the bus state changes from SE0 to idle. The time T_{WTFS} is given a wide range to allow sufficient leverage for a device which has awoke from suspend to use its (possible not yet stable) clock to time this duration reliably.

Reversion to high-speed when the device has detected the K-J-K-J-K-J pattern is accomplished by enabling the high-speed terminations and disconnecting the pull-up resistor from the D+-line. According to Section 7.1.7.5, you may wait up to 500 µs before actually reverting to high-speed, but in this state diagram, this reversion is done immediately after detection of three K-J transitions. After this switching of terminations and pull-up, the device enters the high-speed Default state. The end of reset is signified by the first packet that is received, most likely an SOF packet.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Index

0th microframe, 9.4.11, 11.14.2.3, 11.18.3, 11.22.2

"3 strikes and you're out" mechanism, 11.17.1

4X over-sampling state machine DPLLs, 7.1.15.1

A

abnormal termination sequences, 11.3.3

aborting/retiring transfers

aborting control transfers, 5.5.5
- after loss of synchronization, 11.22.2
- client role in, 10.5.2.2
- conditions for, 5.3.2
- message pipes and, 5.3.2.2
- packet size and, 5.5.3
- Transaction Translator's role, 11.18.6, 11.18.6.1
- USBDI role, 10.5.3.2.1

access frequency of control pipes, 5.5.4

Acknowledge packet. *See* ACKs

ACKs, 8.3.1 *Table 8-1*
- in bulk transfers, 8.5.2, 11.17.1
- in control transfers, 8.5.3, 8.5.3.1, 11.17.1
- corrupted ACK handshake, 8.5.3.3, 8.6.4
- in data toggle, 8.6, 8.6.1, 8.6.2
- defined, 2.0 *glossary*
- function response to OUT transactions, 8.4.6.3
- host response to IN transactions, 8.4.6.2
- overview, 8.4.5
- PING flow control and OUT transactions, 8.5.1, 8.5.1.1
- Ready/ACK status, 11.15
- in request processing, 9.2.6

AC loading specifications, 7.1.6.2

A connectors. *See* Series "A" and "B" connectors

AC stress evaluative setup, 7.1.1

actions in state machines, 8.5, 11.15

active devices, defined, 2.0 *glossary*

active pipes, 10.5.2.2

adaptive endpoints
- connection requirements, 5.12.4.4
- feedback for isochronous transfers, 5.12.4.2
- overview, 5.12.4.1.3

adding devices. *See* dynamic insertion and removal

Address device state
- bus enumeration process, 9.1.2
- overview, 9.1.1.4
- standard device requests, 9.4.1 to 9.4.11
- visible device state table, 9.1.1 *Table 9-1*

addresses
- Address device state, 9.1.1.4, 9.1.1 *Table 9-1,* 9.1.2, 9.4.1 to 9.4.11
- aliasing, 8.3.2
- assignment

after dynamic insertion or removal, 4.6.3
bus enumeration, 2.0 *glossary,* 4.6.3, 9.1.2
device initialization, 10.5.1.1
operations overview, 9.2.2
re-enumerating sub-trees, 10.5.4.5
staged power switching in functions and, 7.2.1.4
time limits for completing, 9.2.6.3
USB System Software role, 4.9
- endpoint addresses, 5.3.1, 9.6.6
- SetAddress() request, 9.4.6

address fields
- address field (ADDR), 8.3.2.1, 8.3.5.1, 8.4.1, 8.4.2.2
- endpoint field (ENDP), 8.3.2.2, 8.3.5.1, 8.4.1
- Hub address field, 8.4.2.2
- packet address fields, 8.3.2 to 8.3.2.2

ADDR field
- overview, 8.3.2.1
- token CRCs, 8.3.5.1
- in token packets, 8.4.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Adopters Agreement, 1.4
advancing pipeline pseudocode, 11.18.7
aging, data-rate inaccuracies and, 7.1.11
aliasing addresses, 8.3.2
"all" encoding, 11.18.4
allocating bit times in handshake packets, 11.3.3
allocating buffers. *See* buffers
allocating USB bandwidth
  transfer management, 5.11.1 to 5.11.1.5
  USB System role, 10.3.2
alternate settings for interfaces
  configuration requirements, 10.3.1
  GetInterface() request, 9.4.4
  in interface descriptors, 9.6.5
  SetInterface() request, 9.4.10
  USBDI mechanisms, 10.5.2.10
  USB support for, 9.2.3
American National Standard/Electronic Industries Association, 6.7.1
American Standard Test Materials, 6.7.1
ANSI/EIA-364-C (12/94), 6.7.1
applications
  in source-to-sink connectivity, 5.12.4.4
  USB suitability for, 3.3
architectural overview of USB
  architectural extensions, 4.10
  bus protocol, 4.4
  bus topology, 4.1.1
  data flow types, 4.7 to 4.7.5
  hub architecture, 4.8.2.1, 11.1.1, 11.12.2
  mechanical and electrical specifications, 4.2 to 4.2.2, 6.1
  physical interface, 4.2 to 4.2.2
  power, 4.3 to 4.3.2
  robustness and error handling, 4.5 to 4.5.2
  system configuration, 4.6 to 4.6.3
  USB devices, 4.1.1.2, 4.8 to 4.8.2.2
  USB host, 4.1.1.1, 4.9
architectural overview of USB *(continued)*
  USB system description, 4.1 to 4.1.1.2
assigning addresses. *See* addresses; bus enumeration
ASTM-D-4565, 6.6.3, 6.7.1
ASTM-D-4566, 6.6.3, 6.7.1
asynchronous data transfers, 2.0 *glossary,* 4.9
asynchronous endpoints
  connection requirements, 5.12.4.4
  feedback for isochronous transfers, 5.12.4.2
  overview, 5.12.4.1.1
asynchronous RA, 2.0 *glossary,* 5.12.4.4. *See also* RA (rate adaptation)
asynchronous SRC, 2.0 *glossary*. *See also* SRC
Attached device state
  in bus enumeration process, 9.1.2
  overview, 9.1.1.1
  visible device state table, 9.1.1 *Table 9-1*
attaching devices. *See* dynamic insertion and removal
attenuation, 7.1.17
attributes of devices in configuration descriptors, 9.6.3
attributes of endpoints in endpoint descriptors, 9.6.6
audio connectivity, 5.12.4.4.1
Audio Device Class Specification Revision 1.0, 9.6
audio devices, defined, 2.0 *glossary*
automatic port color indicators, 11.5.3
available time in frames and microframes
  bulk transfers and, 5.8.4
  bus bandwidth reclamation, 5.11.5
  control transfers and, 5.5.4
  interrupt transfer bus access constraints, 5.7.4
  isochronous transfers and, 5.6, 5.6.4
AWG, 2.0 *glossary,* 6.6.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

B

babble
Collision conditions and detection, 11.8.3
defined, 2.0 *glossary*
EOF2 timing points and, 11.2.5
EOF and babble detection, 11.2.5.1
error detection and recovery, 8.7.4
transaction tracking and, 11.18.7
background of USB development, 3.1 to 3.3
backwards compatibility of USB 2.0, 3.1
bAlternateSetting field (interface descriptors), 9.6.5, 11.23.1
bandwidth
allocating for pipes, 4.4, 4.7.5
bandwidth reclamation, 5.11.5
defined, 2.0 *glossary*
transfer management, 4.7.5, 5.11.1 to 5.11.1.5, 10.3.2
USB system role in, 10.3.2
battery-powered hubs, 7.2.1
bcdDevice field (device descriptors), 9.6.1
bcdUSB field (device descriptors), 9.2.6.6, 9.6.1, 11.23.1
bcdUSB field (device qualifier descriptors), 9.6.2, 11.23.1
bConfigurationValue field
configuration descriptors, 9.6.3, 11.23.1
other speed configuration descriptors, 9.6.4, 11.23.1
B connectors. *See* Series "A" and "B" connectors
bDescLength field (hub descriptors), 11.23.2.1
bDescriptorType field
configuration descriptors, 9.6.3, 11.23.1
device descriptors, 9.6.1, 11.23.1
device qualifier descriptors, 9.6.2, 11.23.1
endpoint descriptors, 9.6.6, 11.23.1
bDescriptorType field *(continued)*
hub descriptors, 11.23.2.1, 11.24.2.5, 11.24.2.10
interface descriptors, 9.6.5, 11.23.1
other speed configuration descriptors, 9.6.4, 11.23.1
string descriptors, 9.6.7
bDeviceClass field
device descriptors, 9.6.1, 11.23.1
device qualifier descriptors, 9.6.2, 11.23.1
bDeviceProtocol field
device descriptors, 9.6.1, 11.23.1
device qualifier descriptors, 9.6.2, 11.23.1
bDeviceSubClass field
device descriptors, 9.6.1, 11.23.1
device qualifier descriptors, 9.6.2, 11.23.1
"beginning" encoding, 11.18.4
bEndpointAddress field (endpoint descriptors), 9.6.6, 11.23.1
best case full-speed budgets, 11.18.1, 11.18.4
bHubContrCurrent field (hub descriptors), 11.23.2.1
bi-directional communication flow, 5.6.2, 5.8.2
big endian, defined, 2.0 *glossary*
bInterfaceClass field (interface descriptors), 9.6.5, 11.23.1
bInterfaceNumber field (interface descriptors), 9.6.5, 11.23.1
bInterfaceProtocol field (interface descriptors), 9.6.5, 11.23.1
bInterfaceSubClass field (interface descriptors), 9.6.5, 11.23.1
bInterval field (endpoint descriptors), 9.6.6, 11.23.1
bit cells, decoding, 7.1.15.1
bitmaps of hub and port status changes, 11.12.4
bit ordering, 8.1
bits, defined, 2.0 *glossary*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

bit stuffing
bit stuffing errors, 11.3.3, 11.15, 11.22
bit stuff violations, 8.7.1
calculating transaction times, 5.11.3
defined, 2.0 *glossary*
high-speed signaling and, 7.1
microframe pipeline and, 11.18.2
overview, 7.1.9
bit times
bit time designations, 11.3
bit time zero, 11.3
before EOF, 11.2.5
in transaction completion prediction, 11.3.3
bLength field
configuration descriptors, 9.6.3, 11.23.1
device descriptors, 9.6.1, 11.23.1
device qualifier descriptors, 9.6.2, 11.23.1
endpoint descriptors, 9.6.6, 11.23.1
interface descriptors, 9.6.5, 11.23.1
other speed configuration descriptors, 9.6.4, 11.23.1
string descriptors, 9.6.7
blinking indicators. *See* indicators
blocking packets in Collision conditions, 11.8.3
blunt cut termination, 6.4.2, 6.4.3
bmAttributes field
configuration descriptors, 9.6.3, 11.23.1
endpoint descriptors, 9.6.6, 11.23.1
hub descriptors, 11.13
other speed configuration descriptors, 9.6.4, 11.23.1
bMaxPacketSize0 field
device descriptors, 9.6.1, 11.23.1
device qualifier descriptors, 9.6.2, 11.23.1
bMaxPower field, 9.6.3
configuration descriptors, 11.23.1
bMaxPower field *(continued)*
other speed configuration descriptors, 9.6.4, 11.23.1
bmRequestType field
hub class requests, 11.24.2
overview, 9.3.1
Setup data format, 9.3
standard device requests, 9.4
bNbrPorts field (hub descriptors), 11.23.2.1
bNumConfigurations field
device descriptors, 9.6.1, 11.23.1
device qualifier descriptors, 9.6.2, 11.23.1
bNumEndpoints field (interface descriptors), 9.6.5, 11.23.1
bNumInterfaces field
configuration descriptors, 9.6.3, 11.23.1
other speed configuration descriptors, 9.6.4, 11.23.1
bPwrOn2PwrGood field, 11.11, 11.23.2.1
bRequest field
hub class requests, 11.24.2
overview, 9.3.2
Setup data format, 9.3
standard device requests, 9.4
standard hub requests, 11.24.1
bReserved field (device qualifier descriptor), 9.6.2
broadcast mode of hub operation, 11.1.2.1
B/S or b/S, defined, 2.0 *glossary*
bString field (string descriptors), 9.6.7
budgets, best case full-speed budget, 11.18.1, 11.18.4
buffers
buffer impedance, 7.1.1.1
buffer match tests, 11.17.1
bulk/control transfer buffering requirements, 11.17.4
calculating sizes in functions and software, 5.11.4
clearing, 11.17.5, 11.24.2.3
client pipes and, 10.5.1.2.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

buffers *(continued)*
client role in, 10.3.3, 10.5.3
defined, 2.0 *glossary*
elasticity buffer, 11.7.1.3
endpoint buffer size, 4.4
identifying location and length, 10.3.4
interrupt transfers and, 5.7.3
isochronous transfers and, 5.12.4.2
non-periodic transaction buffers, 11.14.1, 11.14.2.2, 11.17, 11.17.4
non-USB isochronous application, 5.12.1
packet buffers, 2.0 *glossary*
periodic transaction buffers, 11.14.2.1
prebuffering data, 5.12.5
rate matching and, 5.12.8
rise and fall times for full-speed buffers, 7.1.2.1
Transaction Translator buffers
overview, 11.14.1
resetting, 11.24.2.9
space required, 11.19
underrun or overrun states and error counts, 10.2.6
USBD role in allocating, 10.5.1.2.1
bulk transfers. *See also* non-periodic transactions
buffering requirements, 11.14.2.2, 11.17.4
bus access constraints, 5.8.4
data format, 5.8.1
data sequences, 5.8.5
defined, 2.0 *glossary,* 5.4
direction, 5.8.2
failures, 11.17.5
NAK rates for endpoints, 9.6.6
non-periodic transactions, 11.17 to 11.17.5
overview, 4.7.2, 5.8
packet size, 5.8.3, 9.6.6
scheduling, 11.14.2.2
split transaction examples, A.1, A.2
split transaction notation for, 11.15
bulk transfers. *(continued)*
state machines, 8.5.1, 8.5.1.1, 8.5.2, 11.17.2
transaction format, 8.5.2
transaction organization within IRPs, 5.11.2
USBD pipe mechanism responsibilities, 10.5.3.1.3
bus access for transfers
bulk transfer constraints, 5.8.4
bus access periods, 5.12.8
bus bandwidth reclamation, 5.11.5
calculating buffer sizes, 5.11.4
calculating bus transaction times, 5.11.3
client software role in, 5.11.1.1
control transfer constraints, 5.5.4
HCD role in, 5.11.1.3
Host Controller role in, 5.11.1.5
interrupt transfer constraints, 5.7.4
isochronous transfer constraints, 5.6.4
transaction list, 5.11.1.4
transaction tracking, 5.11.2
transfer management, 5.1.1 to 5.11.1.5
transfer type overview, 5.4
USBD role in, 5.11.1.2
bus clock, 5.12.2, 5.12.3, 5.12.8
bus enumeration
defined, 2.0 *glossary*
device initialization, 10.5.1.1
enumeration handling, 11.12.6
overview, 4.6.3, 9.1.2
re-enumerating sub-trees, 10.5.4.5
staged power switching in functions, 7.2.1.4
USB System Software role, 4.9
bus-powered devices and functions
configuration descriptors, 9.6.3
defined, 4.3.1
device states, 9.1.1.2
high-power bus-powered functions, 7.2.1.4

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

bus-powered devices and functions *(continued)*
low-power bus-powered functions, 7.2.1.3
power budgeting, 9.2.5.1
bus-powered hubs
configuration, 11.13
defined, 4.3.1, 7.2.1
device states, 9.1.1.2
overview, 7.2.1.1
power switching, 11.11
voltage drop budget, 7.2.2
bus protocol overview, 4.4
Bus_Reset receiver state, 11.6.3, 11.6.3.9
bus states
evaluating after reset, 7.1.7.3
global suspend, 7.1.7.6.1
Host Controller role in state handling, 10.2.1
signaling levels and, 7.1.7.1, 7.1.7.2
Transaction Translator tracking, 11.14.1
bus timing/electrical characteristics, 7.3.2
bus topology, 5.2 to 5.2.5
client-software-to-function relationship, 5.2.5
defined, 4.1
devices, 5.2.2
hosts, 5.2.1
illustrated, 4.1.1
logical bus topology, 5.2.4
physical bus topology, 5.2.3
bus transaction timeout in isochronous transfers, 5.12.7
bus turn-around time, 2.0 glossary, 7.1.18 to 7.1.18.2, 8.7.2, 11.18.2
busy (ready/x) state, 11.17.5
bypass capacitors, 7.2.4.1, 7.2.4.2
bytes, defined, 2.0 *glossary*

C

cable assemblies, 6.4 to 6.4.4
cable attenuation, 7.1.17
cable delay
electrical characteristics, 7.3.2 *Table 7-12*
high-/full-speed cables, 6.4.2
hub differential delay, differential jitter, and SOP distortion, 7.3.3 *Figure 7-52*
hub EOP delay and EOP skew, 7.3.3 *Figure 7-53*
hub signaling timings, 7.1.14.1
inter-packet delay and, 7.1.18.1
low-speed cables, 6.4.3, 7.1.1.2
overview, 7.1.16
propagation delay, 6.4.1, 6.7 *Table 6-7,* 7.1.1.2
skew delay, 6.7 *Table 6-7,* 7.1.3, 7.3.3 *Figure 7-53*
cables
attenuation, 7.1.17
cable assemblies, 6.4 to 6.4.4
cable delay (*See* cable delay)
captive cables
high-/full-speed captive cable assemblies, 6.4.2
inter-packet delay and, 7.1.18.1
low-speed captive cable assemblies, 6.4.3
maximum capacitance, 7.1.6.1
termination, 7.1.5.1
color choices, 6.4
construction, 6.6.2
description, 6.6.1
detachable cables
cable delay, 7.1.16
connectors and, 6.2
detachable cable assemblies, 6.4.1
inter-packet delay and, 7.1.18.1
low-speed detachable cables, 6.4.4
maximum capacitance, 7.1.6.1
termination, 7.1.5.1
voltage drop budget, 7.2.2
electrical characteristics and standards, 4.2.1, 6.6.3, 6.7, 7.3.2 *Table 7-12*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

cables *(continued)*
end-to-end signal delay, 7.1.19.1
environmental characteristics, 6.6.4, 6.7
flyback voltage, 7.2.4.2
high-/full-speed cables, 6.4.2
impedance, 6.4.1, 6.4.2, 6.7 *Table 6-7*
input capacitance, 7.1.6.1
length, 6.4.1, 6.4.2, 6.4.3
listing, 6.6.5
low-speed cables, 6.4.3, 6.4.4, 7.1.1.2
mechanical configuration and material requirements, 6.6 to 6.6.5, 6.7
overview, 6.3
prohibited cable assemblies, 6.4.4
pull-out standards, 6.7 *Table* 6-7
shielding, 6.6, 6.6.1
termination, 7.1.5.1
voltage drop budget, 7.2.2
calculations
buffering for rate matching, 5.12.8
buffer sizes in functions and software, 5.11.4
bus transaction times, 5.11.3
capabilities, defined, 2.0 *glossary*
capacitance
after dynamic attach, 7.2.4.1
decoupling capacitance, 7.3.2 *Table* 7-7
input capacitance, 7.1.6.1, 7.3.2 *Table* 7-7
low-speed buffers, 7.1.1.2, 7.1.2.1
low-speed cable capacitive loads, 6.4.3
lumped capacitance guidelines for transceivers, 7.1.6.2
optional edge rate control capacitors, 7.1.6.1
pull-up resistors and, 7.1.5.1
single-ended capacitance, 7.1.1.2
small capacitors, 7.1.6.1
target maximum droop and, 7.2.4.1
unmated contact capacitance, 7.3.2 *Table 7-12*
capacitive load, 6.7 *Table 6-7*
captive cables
high-/full-speed captive cable assemblies, 6.4.2
inter-packet delay and, 7.1.18.1
low-speed captive cable assemblies, 6.4.3
maximum capacitance, 7.1.6.1
rise and fall times, 7.1.2.1, 7.1.2.2
TDR measurements and, 7.1.6.2
termination, 7.1.5.1
change bits
device states, 11.12.2
hub and port status change bitmap, 11.12.4
hub status, 11.24.2.6
over-current status change bits, 11.12.5
port status change bits, 11.24.2.7.2 to 11.24.2.7.2.5
Status Change endpoint defined, 11.12.1
change propagation, host state handling of, 10.2.1
characteristics of devices, 2.0 *glossary,* 9.6.3, 9.6.4
Chirp J and K bus states, 7.1.4.2, 7.1.7.2, 7.1.7.5, C.1, C.2.4
C_HUB_LOCAL_POWER, 11.11, 11.24.2, 11.24.2.1, 11.24.2.6, 11.24.2.7.1.6
C_HUB_OVER_CURRENT, 11.24.2, 11.24.2.1
C_HUB_OVER_POWER, 11.24.2.6
classes of devices. *See* device classes
Class field, 9.2.3, 9.6.5
class-specific descriptors, 9.5, 11.23.2.1
class-specific requests
hub class-specific requests, 11.24.2 to 11.24.2.13
time limits for completing, 9.2.6.5
USBDI mechanisms, 10.5.2.8
Cleared timer status, C.0
ClearFeature() request, CLEAR_FEATURE
ClearHubFeature() request, 11.24.2.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

ClearFeature() request, CLEAR_FEATURE *(continued)*
ClearPortFeature() request, 11.24.2.2
endpoint status and, 9.4.5
hub class requests, 11.24.2
hub requests, 11.24.1
overview, 9.4.1
standard device request codes, 9.4
ClearHubFeature() request
clearing hub features, 11.24.2.6
hub class requests, 11.24.2
hub class-specific requests, 11.24.2.1
clearing pipes, 10.5.2.2
ClearPortFeature() request
clearing status change bits, 11.12.2, 11.24.2.7.2
C_PORT_CONNECTION, 11.24.2.7.2.1
C_PORT_ENABLE, 11.24.2.7.2.2
C_PORT_OVER-CURRENT, 11.24.2.7.2.4
C_PORT_RESET, 11.24.2.7.2.5
C_PORT_SUSPEND, 11.24.2.7.2.3
hub class requests, 11.24.2, 11.24.2.2
PORT_CONNECTION, 11.24.2.7.1.1
PORT_ENABLE, 11.5.1.4, 11.24.2.7.1.2
PORT_HIGH_SPEED, 11.24.2.7.1.8
PORT_INDICATOR, 11.24.2.2, 11.24.2.7.1.10
PORT_LOW_SPEED, 11.24.2.7.1.7
PORT_OVER_CURRENT, 11.24.2.7.1.4
PORT_POWER, 11.24.2.13
PORT_POWER, 11.5.1.2, 11.24.2.7.1.6
PORT_RESET, 11.24.2.7.1.5
PORT_SUSPEND, 11.5.1.10
ClearTTBuffer() request, CLEAR_TT_BUFFER
checking for busy state, 11.17.5
hub class-specific requests, 11.24.2, 11.24.2.3
client pipes, 10.5.1.2.2
client software
in bus topology, 5.2, 5.2.1, 5.2.5
client software-to-function relationships, 5.2, 5.2.5
in communication flow, 5.3
control transfers and, 5.5
defined, 2.0 *glossary*
as implementation focus area, 5.1
notification identification, 10.3.4
role in configuration, 10.3.1
role in data transfers, 10.3.3
service clock and, 5.12.2
in source-to-sink connectivity, 5.12.4.4
in transfer management, 5.11.1, 5.11.1.1
clock model
buffering for rate matching, 5.12.8
bus clock, 5.12.2
clock encoding scheme in electrical specifications overview, 4.2.1
clock synchronization, 5.12.3
clock-to-clock phase differences, 5.12.3
clock tolerance, 11.7.1.3
defined, 5.12
frame clocks, 11.18.3
hub clock source, 11.2.3
in non-USB isochronous application, 5.12.1
overview, 5.12.2
receive clock, 11.7.1.2, 11.7.1.3
sample clock, 5.12.2
service clock, 5.12.2
transmit clock, 11.7.1.3
using SOF tokens as clocks, 5.12.5
clock timings, 7.3.2 *Table 7-8,* 7.3.2 *Table 7-9,* 7.3.2 *Table 7-10*
CMOS driver circuit, 7.1.1.1
CMOS implementations, 7.1.1.3
codes. *See specific types of codes*
Collision conditions, 11.8.3

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

color choices
cables, 6.4
indicator lights on devices, 11.5.3 to 11.5.3.1
plugs, 6.5.4.1
receptacles, 6.5.3.1
commanded stalls, 8.4.5
commands. *See* requests
common mode range for differential input sensitivity, 7.1.4.1
Communication Cables (UL Subject-444), 6.6.5, 6.7.1
communication flow, 5.3 to 5.3.3
Compare_BC_buff algorithm, 11.17.1
completed operations, 9.2.6
completed transactions, 11.3.3
complete-split transactions
buffering, 11.14.2.1, 11.17
bulk/control transactions, 11.17, 11.17.1
CSPLIT transaction tokens, 8.4.2.3
defined, 11.14.1.2
isochronous transactions, 11.21
notation for, 11.15
overview, 11.14.1
scheduling, 11.14.2.1, 11.18.4
space for, 11.18.6.3
split transaction overview, 8.4.2, 8.4.2.1
TT state searching, 11.18.8
completion times for hub requests, 11.24.1
composite devices, 5.2.3
compound devices
bus-powered hubs, 7.2.1.1
in bus topology, 5.2.3
defined, 4.8.2.2
hub descriptors for, 11.23.2.1
power configuration, 11.13
self-powered hubs, 7.2.1.2
conditions in state machine transitions, 8.5, 11.15
conductor resistance unbalance, 6.6.3
conductors
mechanical specifications, 4.2.2
power and signal conductors in cables, 6.3, 6.6.2
resistance, 6.6.3
configuration
bus enumeration, 4.6.3, 9.1.2
configuration management, 10.5.4.1.1
Configured device state, 9.1.1.5
control transfers and, 5.5.4
descriptors, 5.3.1.1, 9.4.3, 9.5, 9.6.1 to 9.6.4, 11.23.1 (*See also* descriptors)
device attachment, 4.6.1
device configuration, 10.3.1
device removal, 4.6.2, 10.5.4.1.4
function configuration, 10.3.1
hubs, 11.13
information in device characteristics, 4.8.1
initial device configuration, 10.5.4.1.2
interrupt transfers and, 5.7.4
modifying device configuration, 10.5.4.1.3
multiple configurations, 9.6.1
multiple interfaces, 9.2.3
operations overview, 9.2.3
other-speed configurations, 9.6.2
power distribution and, 7.2.1
remote wakeup capabilities, 9.2.5.2
requests
configuration requests, 5.11.1.2
GetConfiguration() request, 9.4.2
SetConfiguration() request, 9.4.7
required configurations before usage, 10.3.1
USB configuration, 10.3.1
USBDI mechanisms for getting current settings, 10.5.2.4
USBD role in, 5.11.1.2, 10.5.4.1 to 10.5.4.1.4
Configuration = 0 signal/event, 11.5 *Table 11-5*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

CONFIGURATION descriptor, 9.4 *Table 9-5*
configuration descriptors, 9.4.3, 9.6.4, 11.23.1
Configured device state
in bus enumeration process, 9.1.2
overview, 9.1.1.5
standard device requests and, 9.4.1 to 9.4.11
visible device state table, 9.1.1 *Table 9-1*
configuring software, defined, 2.0 *glossary*
Connect bus state, 7.1.7.1, 7.1.7.3
connecting devices. *See* dynamic insertion and removal
connection status, 11.24.2.7.2, 11.24.2.7.2.1
connectivity
audio connectivity, 5.12.4.4.1
hub fault recovery mechanisms, 11.1.2.3
Hub Repeater responsibilities, 11.1
hubs, 11.1, 11.1.2 to 11.1.2.3
packet signaling connectivity, 11.1.2.1
resume connectivity, 11.1.2.2
source/sink connectivity, 5.12.4.4
synchronous data connectivity, 5.12.4.4.2
tearing down, 11.2.5
connectors
input capacitance, 7.1.6.1
inrush current and, 7.2.4.1
interface and mating drawings, 6.5.3, 6.5.4
keyed connector protocol, 6.2
mechanical configuration and material requirements, 4.2.2, 6.5 to 6.5.4.3
orientation, 6.5.1
reference times, 7.1.6.2
Series "A" and Series "B" plugs, 6.5.4
Series "A" and Series "B" receptacles, 6.5.3
standards for, 6.7
termination data, 6.5.2
connectors *(continued)*
USB Icon, 6.5
construction, cable, 6.6.2
contact arcing, minimizing, 7.2.4.1
contact capacitance standards, 6.7 *Table 6-7*
contact current rating standards, 6.7 *Table 6-7*
contact materials, 6.5.3.3, 6.5.4.3
control endpoints, 2.0 *glossary*. *See also* control transfers
controlling hubs, defined, 7.1.7.7
control mechanisms
device states and control information, 11.12.2
Host Controller control flow management, 4.9
of USB host, 10.1.2
control pipes, 2.0 *glossary*. *See also* control transfers; message pipes; pipes
control transfers. *See also* non-periodic transactions
buffering, 11.14.2.2, 11.17.4
bus access constraints, 5.5.4
control pipes in device characteristics, 4.8.1
data format, 5.5.1
data sequences, 5.5.5
defined, 2.0 *glossary,* 5.4
device requests, 9.3
direction, 5.5.2
error handling on last data transaction, 8.5.3.3
failures, 11.17.5
full-speed limits, 5.5.4 *Table 5-2*
high-speed limits, 5.5.4 *Table 5-3*
low-speed limits, 5.5.4 *Table 5-1*
NAK rates for endpoints, 9.6.6
non-periodic transactions, 11.17 to 11.17.5
overview, 4.7.1, 5.5
packet size, 5.5.3, 9.6.6
protocol stalls, 8.4.5
reporting status results, 8.5.3.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

control transfers *(continued)*
scheduling, 11.14.2.2
simultaneous transfers, 5.5.4
split transaction examples, A.1, A.2
split transaction notation for, 11.15
stages, 2.0 *glossary,* 5.5
STALL handshakes returned by control pipes, 8.5.3.4
state machines, 8.5.1, 8.5.1.1, 8.5.2, 11.17.2
transaction format, 8.5.3
transaction organization within IRPs, 5.11.2
USBD pipe mechanism responsibilities, 10.5.3.1.4
variable-length data stage, 8.5.3.2
converting split transactions, 11.14.1
corrupted transfers and requests
in control transfers, 8.5.3
corrupted ACK handshake, 8.5.3.3, 8.6.4
corrupted CRCs, 10.2.6
corrupted IN tokens, 8.4.6.1
corrupted PIDs, 8.3.1
corrupted SOF packets in isochronous transfers, 5.12.6
in data toggle, 8.6.3
error detection and recovery, 8.7 to 8.7.4
function response to OUT transactions, 8.4.6.3
host response to IN transactions, 8.4.6.2
NAK or STALL handshake, 8.6.3
costs of implementation, 3.3
C_PORT_CONNECTION
clearing, 11.24.2.2
defined, 11.24.2.7.2.1
hub class feature selectors, 11.24.2
Port Change field, 11.24.2.7.2
port status changes, 11.24.2.7.1.10
SetPortFeature() request, 11.24.2.13
C_PORT_ENABLE
ClearPortFeature() request, 11.24.2.2
defined, 11.24.2.7.2.2
hub class feature selectors, 11.24.2
Port Change field, 11.24.2.7.2
SetPortFeature() request, 11.24.2.13
C_PORT_OVER_CURRENT
clearing, 11.24.2.2
defined, 11.24.2.7.2.4
hub class feature selectors, 11.24.2
over-current conditions, 11.11.1, 11.12.5
Port Change field, 11.24.2.7.2
SetPortFeature() request, 11.24.2.13
C_PORT_RESET
clearing, 11.24.2.2
defined, 11.24.2.7.2.5
hub class feature selectors, 11.24.2
Port Change field, 11.24.2.7.2
SetPortFeature() request, 11.24.2.13
C_PORT_SUSPEND
clearing, 11.24.2.2
defined, 11.24.2.7.2.3
hub class feature selectors, 11.24.2
Port Change field, 11.24.2.7.2
resume conditions and, 11.4.4
SetPortFeature() request, 11.24.2.13
CRCs
in bulk transfers, 8.5.2
corrupted CRCs, 10.2.6
CRC16 handling, 11.15, 11.18.5, 11.20.3, 11.20.4, 11.21.3, 11.21.4
CRC check failures, 11.15, 11.20.3, 11.20.4, 11.21.3, 11.21.4
in data packets, 8.3.5.2, 8.4.4
defined, 2.0 *glossary*
in error detection, 8.7.1
overview, 8.3.5
protection in isochronous transfers, 5.12.7
resending, 8.6.4
in token packets, 8.3.5.1, 8.4.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

CRCs *(continued)*
USB robustness and, 4.5, 4.5.1
cross-over points of data lines, 7.1.13.2.1
cross-over voltage in signaling, 7.1.2.1
crystal capacitive loading, 7.1.11
CSPLIT (complete-split transactions). *See* complete-split transactions
CTI, 2.0 *glossary,* 3.1
current
current averaging profile, 7.2.3
current spikes during suspend/resume, 7.2.3
high-speed current driver, 7.1 *Table 7-1*
high-speed signaling and, 7.1.1.3
supply current, 7.3.2 *Table 7-7*
current frame in hub timing, 11.2.3.1
current limiting
bus-powered hubs, 7.2.1.1
dynamic attach and detach, 7.2.4.1
in over-current conditions, 11.12.5
power control during suspend/resume, 7.2.3
remote wakeup and, 7.2.3
self-powered functions, 7.2.1.5
cyclic redundancy check. *See* CRCs

D

D+ or D- lines
average voltage, 7.1.2.1
high-speed signaling and, 7.1, 7.1.1.3
impedance, 7.1.6.1
pull-up resistors and, 7.1
signaling levels and, 7.1.7.1
signal termination, 7.1.5.1
during signal transitions, 7.1.4.1
single-ended capacitance, 7.1.1.2
standardized contact terminating assignments, 6.5.2
test mode, 7.1.20
data
data defined, 5.12.4
data encoding/decoding, 7.1.8
data prebuffering, 5.12.5
data *(continued)*
data processing role of Host Controller, 10.2.4
DATA0/DATA1/DATA2 PIDs
in bulk transfers, 5.8.5, 8.5.2
comparing sequence bits, 8.6.2
in control transfers, 8.5.3
in data packets, 8.4.4
high-bandwidth transactions and, 5.9.1, 5.9.2
high-speed DATA2 PIDs, 8.3.1 *Table 8-1*
in interrupt transactions, 5.7.5, 8.5.4, 11.20.4
synchronization and, 8.6
Transaction Translator response generation, 11.18.5
data field in packets, 8.3.4, 8.4.4
data flow model. *See* transfers
data flow types. *See* transfer types
data formats. *See also specific types of transfers*
bulk transfers, 5.8.1
control transfers, 5.5.1
interrupt transfers, 5.7.1
isochronous transfers, 5.6.1, 5.12.4
overview, 5.4
Data J state. *See* J bus state
Data K bus state. *See* K bus state
data packets
bus protocol overview, 4.4
data CRCs, 8.3.5.2
in isochronous transfers, 8.5.5
packet field formats, 8.3 to 8.3.5.2
packet overview, 8.4.4
spreading over several frames, 5.5.4
data payload
bulk transfers, 5.8.3
calculating transaction times, 5.11.3
defined, 5.3.2
interrupt transfers, 5.7.3
isochronous transfers, 5.6.3

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

data payload *(continued)*
maximum sizes, 8.4.4
non-zero data payload, 5.6.3
packet size constraints, 5.5.3, 5.6.3
data phases
aborting, 11.18.6.1
transaction notation for, 11.15
data PIDs. *See* DATA0/DATA1/DATA2 PIDs; DATA0/DATA1 PIDs; MDATA PIDs
data rates
adaptive endpoints, 5.12.4.1.3
asynchronous endpoints, 5.12.4.1.1
in buffering calculations, 5.12.8
data-rate tolerance, 7.1.11
defined, 5.12.4
in electrical specifications overview, 4.2.1
feedback for isochronous transfers, 5.12.4.2
full-speed source electrical characteristics, 7.3.2 *Table 7-9*
high-speed source electrical characteristics, 7.3.2 *Table 7-8*
low-speed source electrical characteristics, 7.3.2 *Table 7-10*
overview, 7.1.11
sample clock and, 5.12.2
synchronous endpoints, 5.12.4.1.2
data recovery unit, 11.7.1.2
data retry indicators in control transfers, 5.5.5
data sequences
bulk transfers, 5.8.5
control transfers, 5.5.5
interrupt transfers, 5.7.5
isochronous transfers, 5.6.5
data signaling, 7.1.7.4 to 7.1.7.4.2
data signal rise and fall time. *See* rise and fall times
data source jitter, 7.1.13.1 to 7.1.13.1.2, 7.1.14.2, 7.1.15.1
data source signaling, 7.1.13 to 7.1.13.2.2
Data stage
in control transfers, 5.5, 5.5.5, 8.5.3
error handling on last data transaction, 8.5.3.3
length of data, 9.3.5
packet size constraints, 5.5.3
variable-length data stages, 8.5.3.2
data toggle
bulk transfers, 5.8.5
in bulk transfers, 8.5.2
corrupted ACK handshake, 8.6.4
data corrupted or not accepted, 8.6.3
in data packets, 8.4.4
data toggle sequencing, 8.5.5
high bandwidth transactions and, 5.9.1
initialization via SETUP token, 8.6.1
in interrupt transactions, 8.5.4
interrupt transfers and, 5.7.5
low-speed transactions, 8.6.5
overview, 8.6
successful data transactions, 8.6.2
data transfers. *See* data packets; Data stage; transfers
DC electrical characteristics, 7.3.2 *Table 7-7*
DC output voltage specifications, 7.1.6.2
DC resistance of plugs, 6.6.3
debounce intervals in connection events, 7.1.7.3
debouncing connections, 11.8.2
declarations in state machines
global declarations, B.1
Host Controller declarations, B.2
Transaction Translator declarations, B.3
decoupling capacitance, 7.3.2 *Table 7-7*
default addresses of devices, 2.0 *glossary,* 9.1.1.4, 10.5.1.1
Default bus state, 7.1.7.5
Default Control Pipe
in bus enumeration process, 9.1.2
in communication flow, 5.3

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Default Control Pipe *(continued)*
control transfer packet size constraints, 5.5.3
defined, 4.4, 5.3.2
endpoint zero requirements, 5.3.1.1
as message pipe, 5.3.2.2
size description in descriptors, 9.6.1
Default device state
overview, 9.1.1.3
standard device requests and, 9.4.1 to 9.4.11
visible device state table, 9.1.1 *Table 9-1*
default pipes, 2.0 *glossary,* 10.5.1.2.1
delays. *See* cable delay; differential delay; propagation delay
delivery rates in isochronous transfers, 4.7.4
DEOP signal/event, 11.7.2.3 *Table 11-11*
descriptor index, 9.4.3, 9.4.8
descriptors
accessing, 11.23.1
in bus enumeration process, 9.1.2
class-specific descriptors, 9.5, 11.23.2.1
configuration descriptors, 9.6.3, 9.6.4, 10.3.1, 10.5.2.4
control transfers and, 5.5, 5.5.3
defined, 9.5
descriptor index, 9.4.3, 9.4.8
device class definitions, 9.7, 9.7.1
device descriptors, 9.4 *Table 9-5,* 9.6.1 to 9.6.5
endpoint descriptors, 9.6.6
getting descriptors, 9.4.3, 10.5.2.3
hub descriptors, 11.23 to 11.23.2.1, 11.24.2.5, 11.24.2.10
interface descriptors, 9.2.3, 9.6.5
isochronous transfer capabilities, 5.12
listing remote wakeup capabilities, 9.2.5.2
other speed configuration descriptor, 9.6.4
overview, 9.5 to 9.7.3
descriptors *(continued)*
setting descriptors, 5.3.1.1, 9.4.8, 10.5.2.12
speed dependent descriptors, 9.2.6.6, 9.6.4
string descriptors, 9.6.7
USBDI mechanisms for getting descriptors, 10.5.2.3
vendor-specific descriptors, 9.5
deserialization of transmissions, 10.2.2
detachable cables
cable delay, 7.1.16
connectors and, 6.2
detachable cable assemblies, 6.4.1
inter-packet delay and, 7.1.18.1
low-speed detachable cables, 6.4.4
maximum capacitance, 7.1.6.1
termination, 7.1.5.1
voltage drop budget, 7.2.2
detached devices, 9.1.1.1, 9.1.2
detaching devices. *See* dynamic insertion and removal
detecting connect and disconnect conditions, 7.1.7.3, 7.1.20
detecting errors. *See* error detection and handling
detecting hub and port status changes, 7.1.7.5, 11.12.2, 11.12.3, 11.12.4
detecting over-current conditions, 7.2.1.2.1
detecting speed of devices. *See* speed detection
Detection mechanism, 7.1.5.2
Dev_Do_BCINTI state machine, 8.5.2 *Figure 8-34*
Dev_Do_BCINTO state machine, 8.5.2 *Figure 8-32*
Dev_Do_IN state machine, 8.5 *Figure 8-25*
Dev_Do_IsochI state machine, 8.5.5 *Figure 8-43*
Dev_Do_IsochO state machine, 8.5.5 *Figure 8-41*
Dev_Do_OUT state machine, 8.5 *Figure 8-24*
Dev_HS_BCO state machine, 8.5.1.1 *Figure 8-29*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Dev_HS_ping state machine, 8.5.1.1 *Figure 8-28*

device addresses, 2.0 *glossary*. *See also* addresses; devices

device classes. *See also* USB device framework

- class codes, 9.2.3
- defined, 4.8
- descriptors, 9.2.3, 9.6.1, 9.7
- device characteristics, 4.8.1
- device class definitions, 9.7
- device qualifier descriptors, 9.6.2
- getting class-specific descriptors, 9.5
- hub class-specific requests, 11.24.2 to 11.24.2.13
- interfaces and endpoint usage, 9.7.2
- requests, 9.7.3
- standard, class, and vendor information, 4.8.1

Device Class Specification for Audio Devices Revision 1.0, 9.6

DEVICE descriptor, 9.4 *Table 9-5*

device descriptors

- descriptor types, 9.4 *Table 9-5*
- device class descriptors, 9.2.3, 9.7
- device qualifier descriptors, 9.6.2
- GetDescriptor() request, 9.4.3
- getting class-specific descriptors, 9.5
- hubs, 11.23.1
- overview, 9.6.1
- speed dependent descriptors, 9.2.6.6
- standard definitions, 9.6.1 to 9.6.5

device drivers, 5.12.4.4, 10.3.1

device endpoints, 2.0 *glossary,* 5.3.1.1. *See also* endpoints

device-initiated resume. *See* remote wakeup

Device layer

- descriptors, 9.5 to 9.7.3
- device states, 9.1 to 9.1.2
- generic USB device operations, 9.2 to 9.2.7
- standard device requests, 9.4 to 9.4.11

Device layer *(continued)*

- in USB device framework, 9
- USB device requests, 9.3 to 9.3.5

Device_Process_trans state machine, 8.5 *Figure 8-23*

device qualifier descriptors, 9.2.6.6, 9.4.3, 9.4 *Table 9-5,* 9.6.1, 9.6.2

Device release numbers, 9.6.1

DEVICE_REMOTE_WAKEUP, 9.4 *Table 9-6*

DeviceRemovable field (hub descriptors), 11.23.2.1

device resources, 2.0 *glossary*. *See also* buffers; endpoints

devices. *See also* USB device framework

- address assignment, 9.1.2, 9.2.2
- characteristics and configuration (*See also* device descriptors)
- configuration, 4.8.2.2, 9.2.3
- data-rate tolerance, 7.1.11
- descriptors, 9.5 to 9.7.3, 9.6.1
- device characteristics, 4.8.1
- device classes, 4.8, 9.7
- device descriptions, 4.8.2 to 4.8.2.1
- device speed, 7.1.5 to 7.1.5.2, 7.1.7.3, 11.8.2
- host role in configuration, 10.3.1
- optional endpoints, 5.3.1.2
- USBD role in configuration, 10.5.4.1 to 10.5.4.1.4

data transfer, 9.2.4

- communication flow requirements, 5.3
- control transfers and, 5.5
- detailed communication flow illustrated, 5.3
- differing bus access for transfers, 5.11
- jitter budget table, 7.1.15.1
- PING flow control, 8.5.1, 8.5.1.1
- response to IN transactions, 8.4.6.1
- response to OUT transactions, 8.4.6.3

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

devices *(continued)*
data transfer *(continued)*
response to SETUP transactions, 8.4.6.4
role in bulk transfers, 8.5.2
device event timings, 7.3.2 *Table 7-14*
devices defined, 2.0 *glossary*
device state machines, 8.5
dynamic attach and detach, 9.2.1
power distribution, 7.2.4 to 7.2.4.2
removing, 10.5.2.6, 10.5.4.1.4
USBDI mechanisms, 10.5.2.5, 10.5.2.6
generic USB device operations, 9.2 to 9.2.7
port indicators, 11.5.3 to 11.5.3.1
power distribution, 7.2.1, 9.2.5
bus-powered devices, 4.3.1, 7.2.1.1
dynamic attach and detach, 7.2.4 to 7.2.4.2
high-power bus-powered functions, 7.2.1.4
low-power bus-powered functions, 7.2.1.3
power supply and, 4.3.1
self-powered devices, 4.3.1, 7.2.1.2, 7.2.1.5
suspend/resume conditions, 7.2.3
voltage drop budget, 7.2.2
requests
host communication, 10.1.1
request errors, 9.2.7
request processing, 9.2.6 to 9.2.6.6
standard device requests, 9.4 to 9.4.11
USB device requests, 9.3 to 9.3.5
state machines, 8.5, 8.5.2, 8.5.5
status
device states, 9.1 to 9.1.2, 11.12.2
getting device status, 9.4.5
getting port status, 11.24.2.7.1.1
subtree devices after wakeup, 10.5.4.5
turn-around timers, 8.7.2
devices *(continued)*
types of devices
composite devices, 5.2.3
compound devices, 4.8.2.2, 5.2.3
functions, 4.8.2.2
hubs, 4.8.2.1
mapping physical and virtual devices, 5.12.4.4
virtual devices, 2.0 *glossary*
in USB topology, 4.1.1.2, 5.2, 5.2.2, 9.0
device software, defined, 2.0 *glossary*
device state machines, 8.5. *See also specific state machines under* Dev_
diameter of cables, 6.6.2
diamond symbols in state machines, 8.5, 11.15
dielectric withstanding voltage standards, 6.7 *Table 6-7*
Differential 0 bus state, 7.1.7.2
Differential 1 bus state, 7.1.7.1, 7.1.7.2
Differential 2 bus state, 7.1.7.1
differential data jitter, 7.3.3 *Figure 7-49,* 7.3.3 *Figure 7-52*
differential delay, 7.3.2 *Table 7-11,* 7.3.3 *Figure 7-52*
differential-ended components in upstream ports, 11.6.1, 11.6.2
differential envelope detectors, 7.1
differential input receivers, 1, 7.1, 7.1.4.1, 7.1.6, 7.1 *Table 7-1*
differential output drivers, USBD as, 7.1.1
differential signaling, 7.1.7.1, 7.1.7.2, 7.1.7.4.1
differential termination impedance, 7.1.6.2
differential-to-EOP transition skew, 7.3.3 *Figure 7-50*
dimensional inspection standards, 6.7 *Table 6-7*
Direction bit, 9.3.1, 9.3.4
direction of communication flow, 5.4
bmRequestType field, 9.3.1
bulk transfers, 5.8.2
bus protocol overview, 4.4
control transfers, 5.5.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

direction of communication flow *(continued)*
interrupt transfers, 5.7.2
isochronous transfers, 5.6.2
disabled ports, 11.5, 11.5.1.4, 11.24.2.7.1, 11.24.2.7.2
Disabled state, 11.5, 11.5.1.4
disabling features, 9.4.1
discarding packets, 11.3.2
Disconnect_Detect signal/event, 11.5.2, 11.5 *Table 11-5*
Disconnected state
connect and disconnect signaling, 7.1.7.3
detecting, 7.1, 7.1.4.2, 7.1.20
downstream ports, 11.5, 11.5.1.3
signaling levels and, 7.1.7.1, 7.1.7.2
disconnecting devices. *See* dynamic insertion and removal
disconnection envelope detectors, 7.1.7.3, 7.1 *Table 7-1*
disconnect timer, 11.5.2
distortion, minimizing in SOP, 7.1.7.4.1
DLL lock, 7.1
documents, applicable standards, 6.7.1
down counters in hub timing, 11.2.3.1
downstream facing ports and hubs
Disconnect state detection, 7.1
downstream connectivity defined, 11.1.2.1
downstream defined, 2.0 *glossary*
downstream facing port state machine, 11.5
downstream plugs, 6.2
downstream ports defined, 4.8.2.1
driver speed and, 7.1.2.3
enumeration handling, 11.12.6
high-speed driver characteristics and, 7.1.1.3
high-speed signaling and, 7.1.7.6.1, 7.1.7.6.2, 11.1.1
in hub architecture, 11.1.1
hub delay, 7.3.3 *Figure 7-52*
hub descriptors, 11.23.2.1
downstream facing ports and hubs *(continued)*
hub EOP delay and EOP skew, 7.3.3 *Figure 7-53*
input capacitance, 7.1.6.1
jitter, 7.3.2 *Table 7-10*
multiple Transaction Translators, 11.14.1.3
port state descriptions, 11.5.1 to 11.5.1.14
reset state machines, C.1
signaling delays, 7.1.14.1
signaling speeds, 7.1
status changes, 11.12.6
test mode support, 7.1.20
transceivers, 7.1, 7.1.7.1, 7.1.7.2
downstream facing transceivers, high-speed signaling and, 7, 7.1
downstream packets (HSD1), 8.5, 11.15
drain wires, 6.5.2, 6.6.1, 6.6.2
dribble, defined, 7.1.9.1
drift, 5.12.1, 5.12.3
driver characteristics
full-speed driver characteristics, 7.1.1.1
full-speed source electrical characteristics, 7.3.2 *Table 7-9*
high-speed driver characteristics, 7.1.1.3
high-speed source electrical characteristics, 7.3.2 *Table 7-8*
low-speed driver characteristics, 7.1.1.2, 7.1 *Table 7-1*
low-speed source electrical characteristics, 7.3.2 *Table 7-10*
overview, 7.1.1
drivers
defined, 2.0 *glossary*
role in configuration, 10.3.1
in source-to-sink connectivity, 5.12.4.4
droop, 7.2.3, 7.2.4.1
dual pin-type receptacles, 6.9
durability standards, 6.7 *Table 6-7*
DWORD, defined, 2.0 *glossary*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

dynamic insertion and removal, 9.2.1
attaching devices, 4.6.1
defined, 2.0 *glossary*
detecting insertion and removal, 4.9, 9.2.1
Hub Repeater responsibilities, 11.1
hub support for, 11.1
power control, 7.2.3, 7.2.4 to 7.2.4.2
power-on and connection events timing, 7.1.7.3
removing devices, 4.6.2
USB robustness and, 4.5

E

E field (End), 8.4.2.2
E2PROM defined, 2.0 *glossary*
ease-of-use considerations, 1.1
EBEmptied signal/event, 11.7.1.4 *Table 11-10*
edges of signals
cable delay, 7.1.16
data source jitter, 7.1.13.1.1
edge transition density, 8.2
optional edge rate control capacitors, 7.1.6.1
EEPROM, defined, 2.0 *glossary*
elasticity buffer, 11.7.1.3
Electrical Connector/Socket Test Procedures, 6.7.1
Electrically Erasable Programmable Read Only Memory (EEPROM), 2.0 *glossary*
Electrical Performance Properties of Insulation and Jacket for Telecommunication Wire and Cable, 6.7.1
electrical specifications, 6.1, 7
applicable documents, 6.7.1
bus timing/electrical characteristics, 7.3.2
cables, 6.3, 6.4 to 6.4.4, 6.6 to 6.6.5
connectors, 6.2, 6.5 to 6.5.4.3
overview, 4.2.1, 6
PCB reference drawings, 6.9
physical layer specifications, 7.3 to 7.3.3
electrical specifications *(continued)*
power distribution, 7.2 to 7.2.1.5, 7.2.3, 7.2.4 to 7.2.4.2
signaling, 7.1 to 7.1.20
standards for, 6.7, 7.3.1
timing waveforms, 7.3.3
USB grounding, 6.8
embedded hubs, 4.8.2.2, 5.2.3
EMI, USB grounding and, 6.8
enabled ports
connectivity and, 11.1.2.1
downstream ports, 11.5, 11.5.1.6
getting port status, 11.24.2.7.1
PORT_ENABLE bit, 11.24.2.7.1.2
port status change bits, 11.24.2.7.2
Enabled state, 11.5, 11.5.1.6
Enable Transmit state, 11.7.1.4.3
encoding data, 7.1.8, 11.18.4
"end" encoding, 11.18.4
End field (E), 8.4.2.2
End-of-Frame (EOF). *See* EOFs
End of High-speed Packet (HSEOP), 7.1.7.2, 7.1.7.4.2
End-of-Packet (EOP). *See* EOPs
End-of-Packet bus state, 7.1.7.1, 7.1.7.2, 7.1.7.4.1, 7.1.7.4.2
end-of-packet delimiter. *See* EOPs
ENDP field, 8.3.2.2, 8.3.5.1, 8.4.1
endpoint addresses, 2.0 *glossary,* 5.3.1, 9.6.6
ENDPOINT descriptor, 9.4 *Table 9-5*
endpoint descriptors, 9.4.3, 9.6.1, 9.6.5, 9.6.6
endpoint direction, defined, 2.0 *glossary*
endpoint field (ENDP), 8.3.2.2, 8.3.5.1, 8.4.1
ENDPOINT_HALT, 9.4 *Table 9-6*
endpoint numbers, 2.0 *glossary,* 5.3.1
endpoints
addresses, 9.6.6
characteristics, 5.3.1
description in descriptors, 9.4.3, 9.6.1, 9.6.5, 9.6.6

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

endpoints *(continued)*
in device class definitions, 9.7.2
direction of flow, 5.3.1
endpoint address field, 8.3.2.2
endpoint aliasing, 8.3.2
endpoint zero requirements, 4.8.1, 5.3.1.1, 5.3.1.2, 5.3.2
explicit feedback endpoints, 9.6.5, 9.6.6
getting endpoint status, 9.4.5
high-bandwidth endpoints, 2.0 *glossary,* 5.7.4
high-speed signaling attributes, 9.6.6
Hub Controller endpoint organization, 11.12.1
in interfaces, 9.2.3, 9.6.3, 9.6.5
logical devices as collections of endpoints, 5.3
message pipes and, 5.3.2.2
non-endpoint zero requirements, 5.3.1.2
number matching, 9.6.6
overview, 5.3.1
pipes and, 4.4, 5.3.2
programmable data rates, 2.0 *glossary*
reflected endpoint status, 10.5.2.2
role in data transfers, 4.7
samples, 2.0 *glossary*
specifying in *wIndex* field, 9.3.4
state machines, 8.5
stream pipes and, 5.3.2.1
synchronization frame, 9.4.11
Transfer Types, Synchronization Types, and Usage Types, 9.6.6
endpoint synchronization type, 5.12.4, 5.12.4.1
Endpoint Type field (ET), 8.4.2.2
endpoint type field (ET), 8.4.2.2
endpoint zero
Default Control Pipe and, 5.3.2
in device characteristics, 4.8.1
non-endpoint zero requirements, 5.3.1.2
endpoint zero *(continued)*
requirements, 5.3.1.1
end-to-end signal delay, 7.1.19 to 7.1.19.2
end users, 2.0 *glossary,* 3.3
entering test mode, 7.1.20
entry points into state machines, 8.5
enumeration. *See* bus enumeration
envelope detectors, 2.0 *glossary,* 7.1, 7.1.4.2, 7.1.7.3, 7.1 *Table 7-1*
environmental characteristics for cables, 6.6.4
environmental compliance standards, 6.7
EOF1 or EOF2 signal/event
frame and microframe timers, 11.2.3.2, 11.2.5 to 11.2.5.2
host behavior at end-of-frame, 11.3
in Hub Repeater state machine, 11.7.2.3 *Table 11-11*
in transmitter state machine, 11.6.4 *Table 11-9*
EOFs
advancing, 11.2.3.2
defined, 2.0 *glossary*
in frame and microframe timer synchronization, 11.2.3.2
host behavior at end-of-frame, 11.3 to 11.3.3
Host Controller frame and microframe generation, 10.2.3
in transaction completion prediction, 11.3.3
EOI signal/event
defined, 11.7.1.4 *Table 11-10*
in downstream port state machine, 11.5 *Table 11-5*
in internal port state machine, 11.4
in receiver state machine, 11.6.3 *Table 11-8*
in transmitter state machine, 11.6.4
EOP bus state, 7.1.7.1, 7.1.7.2, 7.1.7.4.1, 7.1.7.4.2
EOPs
defined, 2.0 *glossary*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

EOPs *(continued)*
differential-to-EOP transition skew and EOP width, 7.3.3 *Figure 7-50*
EOP delimiter, 8.3
EOP dribble defined, 11.7.1.1
EOP width, 7.1.13.2 to 7.1.13.2.2, 7.3.3 *Figure 7-50*
error detection through bus turn-around timing, 8.7.2
extra bits and, 7.1.9, 7.1.9.1
false EOPs, 2.0 *glossary,* 8.7.3, 11.15
handshake packets and, 8.4.5
high-speed signaling and, 7.1
hub EOP delay and EOP skew, 7.3.3 *Figure 7-53*
hub/repeater electrical characteristics, 7.3.2 *Table 7-11*
hub signaling at EOF1, 11.3.1
intervals between IN token and EOP, 11.3.3
propagation delays, 7.1.14.1
EOR signal/event, 11.6.3 *Table 11-8*
equations
buffering for rate matching, 5.12.8
buffer sizes in functions and software, 5.11.4
bus transaction times, 5.11.3
ERR handshake
interrupt transactions, 11.20.4
isochronous transactions, 11.21.1, 11.21.4
Transaction Translator response generation, 11.18.5
error detection and handling. *See also* corrupted transfers and requests
"3 strikes and you're out" mechanism, 11.17.1
babble and loss of activity recovery, 8.7.4
bit stuff violations, 8.7.1
bulk transfers and, 5.8.5, 8.5.2
bus turn-around timing, 8.7.2
busy (ready/x) state, 11.17.5
control transfers and, 5.5.5, 8.5.3.1
error detection and handling. *(continued)*
corrupted ACK handshake, 8.5.3.3, 8.6.4
corrupted SOF packets in isochronous transfers, 5.12.6
CRCs, 8.3.5, 8.7.1, 11.15, 11.20.3, 11.20.4, 11.21.3, 11.21.4
data corrupted or not accepted, 8.6.3
error count tally, 10.2.6, 11.17.1
error handling for transfers, 5.4
error handling on last data transaction, 8.5.3.3
false EOPs, 2.0 *glossary,* 8.7.3
HC_Data_or_error state machine, 11.20.2
high bandwidth transactions, 5.9.2
Host Controller role in, 10.2.6
Hub Repeater responsibilities, 11.1
hub role in, 11.1.2.3
interrupt transfers and, 5.7.5
isochronous transfers and, 5.6.4, 5.6.5, 5.12.7
notation for error cases, 11.15
overview, 8.7
packet error categories, 8.7.1
periodic transactions, 11.18.4
PID check bits, 8.7.1
Port Error conditions, 11.8.1
port indicators, 11.5.3 to 11.5.3.1
Request Errors, 9.2.7
sample size and, 5.12.8
short packets and error conditions, 5.3.2
split transaction sequencing, 11.21.3
status values for, 11.15
synchronous data connectivity, 5.12.4.4.2
timeouts, 8.7.2, 11.17.1
Transaction Translator error handling, 11.22
USBD role in, 10.5.4.4
USB robustness and, 4.5.1, 4.5.2
ERR PID, 8.3.1 *Table 8-1,* 8.4.5

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

ESD, USB grounding and, 6.8
ET field (Endpoint Type), 8.4.2.2
event notifications, USBD and, 10.5.4.3
example declarations in state machines, B.1, B.2, B.3
exception handling. *See* error detection and handling
Exception Window, 7.1.6.2
exiting test mode, 7.1.20
exit points from state machines, 8.5
explicit feedback endpoints, 9.6.5, 9.6.6
extended descriptor definitions, 9.7.1
extensibility of USB architecture, 4.10
extension cable assemblies, 6.4.4
externally-powered hubs, 7.2.1. *See also* self-powered hubs
extraction force standards, 6.7 *Table 6-7*
eye pattern templates
  defined, 2.0 *glossary*
  error rates and jitter tolerance, 7.1.14.2, 7.1.15.2
  high-speed receiver characteristics and, 7.1.4.2
  overview, 7.1.2.2
  transmit eye patterns, 7.1, 7.1.2

F

failed data transactions, 8.6.3
false EOPs, 2.0 *glossary,* 8.7.3, 11.15
fault detection. *See* error detection and handling
features
  hub class feature selectors, 11.24.2
  SetFeature() request, 9.4.9
  setting hub features, 11.24.2.12
  standard feature selectors, 9.4 *Table 9-6*
feedback endpoints, 9.6.6
feedback for isochronous transfers, 5.12.4.2, 5.12.4.3, 9.6.5
ferrite beads, 7.1.6.2
fields. *See names of specific fields*
flammability
  cables, 6.6.4
flammability *(continued)*
  Series "A" and Series "B" plugs, 6.5.4.1
  Series "A" and Series "B" receptacles, 6.5.3.1
  standards, 6.7 *Table 6-7*
flexibility of USB devices, 3.3
flow control mechanisms
  in bus protocol overview, 4.4
  handshake packets and, 8.4.5
  non-periodic transactions, 11.14.2.2
  USB robustness and, 4.5
flow sequences
  non-periodic transactions, 11.17 to 11.17.5
  periodic transactions, 11.18 to 11.18.8, 11.20 to 11.20.4, 11.21.1
  split transaction notation for, 11.15
flyback voltage, 7.2.4.2
format of USB device requests, 9.3
formulas
  buffering for rate matching, 5.12.8
  buffer sizes in functions and software, 5.11.4
  bus transaction times, 5.11.3
frame and microframe intervals
  full-speed source electrical characteristics, 7.3.2 *Table 7-9*
  high-speed source electrical characteristics, 7.3.2 *Table 7-8*
  low-speed source electrical characteristics, 7.3.2 *Table 7-10*
  repeatability, 7.1.12
frame and microframe numbers
  buffering for rate matching, 5.12.8
  frame and microframe number field, 8.4.3
  frame number field, 8.3.3
  frame numbers, 8.3.3
  generating frames and microframes, 10.2.3
  illustrated, 8.4.3.1
  SOF tracking, 5.12.6

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

frame and microframe timers
frame wander, 11.2.5.2
hub frame timer, 11.2 to 11.2.5.2
timing skew, 11.2.5.1 to 11.2.5.2
TT loss of synchronization, 11.22.1
frame clocks, 5.12.3, 5.12.4.1.2, 11.18.3
frame pattern, defined, 2.0 *glossary*
frames and microframes. *See also* frame and microframe timers
allocating bandwidth, 4.7.5, 5.11.1 to 5.11.1.5, 10.3.2
available time in frames and microframes, 5.5.4, 5.6, 5.6.4, 5.7.4, 5.8.4, 5.11.5
babble and loss of activity recovery, 8.7.4
bandwidth reclamation, 5.11.5
best case full-speed budgets, 11.18.1, 11.18.4
bit time zero, 11.3
clock tracking and microframe SOFs, 5.12.4.1.2
control transfer reserved portions, 5.5.4
data prebuffering and, 5.12.5
defined, 2.0 *glossary*, 5.3.3
error handling in transfers, 5.12.7
frame and microframe intervals, 7.1.12, 7.3.2 *Table 7-8,* 7.3.2 *Table 7-9,* 7.3.2 *Table 7-10*
frame and microframe numbers (*See* frame and microframe numbers)
frame and microframe timer ranges, 11.2.1 to 11.2.2
frame wander, defined, 11.2.5.2
generation role of Host Controller, 10.2.3
generation role of Transaction Translator, 11.18.3
host behavior at end-of-frame, 11.3
interrupt transfer limitations, 5.7.4
isochronous transactions, 5.6.3, 5.6.4, 5.12.4.2, 8.5.5
jitter, 11.2.4
maximum allowable transactions, 5.4.1, 11.18.6.3
frames and microframes. *(continued)*
microframe numbers, 8.4.3.1
microframe pipelines
buffer space, 11.19
clearing and aborting transactions, 11.18.6
defined, 11.18.2
periodic split transactions, 11.14.2.1, 11.18
resetting, 11.24.2.9
transaction tracking, 11.18.7
multiple transactions, 5.6.4, 5.7.4, 5.9, 5.9.2, 9.6.6
organization of transactions within, 5.11.2
overview, 8.4.3.1
samples per frame in isochronous transfers, 5.12.4.2
SOF packets, 8.4.3
SOF tracking, 5.12.6
split transactions and, 5.10
synch frame requests, 9.4.11
timers, 11.2 to 11.2.5.2
timer synchronization, 11.2.3 to 11.2.3.3, 11.22.1
toggle sequencing, 8.5.5
zeroth microframe, 9.4.11, 11.14.2.3
freeing pending start-splits, 11.18.6.2
frequency-locked clocks, 5.12.3
Fs. *See* SRC
Fsus state, 11.4, 11.4.3
full-duplex, defined, 2.0 *glossary*
full-speed buffers, 7.1.2.1
full-speed cables. *See* high-/full-speed cables
full-speed driver characteristics, 7.1.1.1, 7.1 *Table 7-1*
full-speed functions and hubs
bulk transfers and, 5.8.4
cable and resistor connections, 7.1.5.1
connect detection, 7.1.7.3
control transfers and, 5.5.3, 5.5.4, 5.5.4 *Table 5-2*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

full-speed functions and hubs *(continued)*
data-rate tolerance, 7.1.11
defined, 2.0 *glossary*
detachable cables and, 6.4.1
full-speed port transceiver, 7.1.7.1
full-speed source electrical characteristics, 7.3.2 *Table 7-9*
full- *vs.* low-speed port behavior, 11.8.4
getting port status, 11.24.2.7.1
hub/repeater electrical characteristics, 7.3.2 *Table 7-11*
hub support for, 11.1
input capacitance, 7.1.6.1
interrupt transfers and, 5.7.3, 5.7.4 *Table 5-7*
isochronous transfers and, 5.6.4
maximum data payload, 8.4.4
optional endpoints, 5.3.1.2
in physical bus topology, 5.2.3
reset states and, C.2.2
sampling rates, 5.12.4.2
signal termination, 7.1.5.1
SOF PID and, 8.4.3
speed detection and, 11.8.2
Transmit state and, 11.5.1.7
full-speed signaling
babble and loss of activity recovery, 8.7.4
best case full-speed budgets, 11.18.1, 11.18.4
bus transactions and, 4.4
calculating transaction times, 5.11.3
data rates, 4.2.1
data signaling overview, 7.1.7.4.1
data source jitter, 7.1.13.1.1
defined, 2.0 *glossary*
differential receivers, 7.1 *Table 7-1*
downstream and upstream facing ports, 7.1
driver characteristics, 7.1.1
driver requirements, 7.1.2.3
endpoint zero requirements, 5.3.1.1
full-speed signalling *(continued)*
EOF timing points, 11.2.5.2
EOP width, 7.1.13.2.1
errors, 8.6.4
frame timer ranges, 11.2.2
full-speed loads, 7.1.2.1
high-speed devices operating at full-speed, 5.3.1.1
host behavior at end-of-frame, 11.3 to 11.3.3
hub class descriptors and, 11.23.1
intervals between IN token and EOP, 11.3.3
isochronous transaction limits, 5.6.3
J and K states, 7.1.7.1
jitter budget table, 7.1.15.1
propagation delays, 7.1.14.1
receiver characteristics, 7.1.4.1
reset signaling, 7.1.7.5
sampling rates, 5.12.4.2
scheduling, 11.14.2.3
speed detection, 9.1.1.3
Transaction Translator and, 4.8.2.1, 11.18.3, 11.18.5
Full Suspend (Fsus) state, 11.4, 11.4.3
function address field (ADDR), 8.3.2.1, 8.4.2.2
functional stall, 8.4.5, 8.5.3.4
Function layer
detailed communication flow, 5.3
illustrated, 5.1
interlayer communications model, 10.1.1
functions. *See also* devices; full-speed functions and hubs; high-speed functions and hubs; low-speed functions and hubs
address assignment, 9.1.2, 9.2.2
characteristics and configuration (*See also* device descriptors)
configuration, 4.8.2.2, 9.2.3
data-rate tolerance, 7.1.11
descriptors, 9.5 to 9.7.3, 9.6.1
device characteristics, 4.8.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

functions. *(continued)*
characteristics and configuration *(continued)*
device classes, 4.8, 9.7
device speed, 7.1.7.3, 11.8.2
host role in configuration, 10.3.1
optional endpoints, 5.3.1.2
data transfer
communication flow requirements, 5.3
control transfers and, 5.5
detailed communication flow illustrated, 5.3
differing bus access for transfers, 5.11
jitter budget table, 7.1.15.1
overview, 9.2.4
PING flow control and OUT transactions, 8.5.1, 8.5.1.1
response to IN transactions, 8.4.6.1
response to OUT transactions, 8.4.6.3
response to SETUP transactions, 8.4.6.4
role in bulk transfers, 8.5.2
device event timings, 7.3.2 *Table 7-14*
devices defined, 2.0 *glossary*
dynamic attach and detach, 9.2.1
power distribution, 7.2.4 to 7.2.4.2
removing, 10.5.2.6, 10.5.4.1.4
USB mechanisms, 10.5.2.5, 10.5.2.6
generic USB device operations, 9.2 to 9.2.7
overview, 4.8.2.2
power distribution, 7.2.1, 9.2.5
bus-powered devices, 4.3.1, 7.2.1.1
dynamic attach and detach, 7.2.4 to 7.2.4.2
high-power bus-powered functions, 7.2.1.4
low-power bus-powered functions, 7.2.1.3
power supply and, 4.3.1
functions. *(continued)*
power distribution *(continued)*
self-powered functions, 7.2.1.2, 7.2.1.5
suspend/resume conditions, 7.2.3
voltage drop budget, 7.2.2
requests
host communication with, 10.1.1
request errors, 9.2.7
request processing, 9.2.6 to 9.2.6.6
standard device requests, 9.4 to 9.4.11
USB device requests, 9.3 to 9.3.5
status, 9.1 to 9.1.2, 9.4.5
types of devices
compound devices, 4.8.2.2
functions, 4.8.2.2
mapping physical and virtual devices, 5.12.4.4
virtual devices, 2.0 *glossary*
in USB topology, 4.1.1.2, 5.2.2, 5.2.3, 9.0
function-to-host transfers. *See* IN PID

G

gang-mode power control, 11.23.2.1
garbling messages in Collision conditions, 11.8.3
Generate End of Packet Towards Upstream Port state (GEOPTU), 11.6.4, 11.6.4.5
Generate Resume state, 11.4, 11.4.4
generic USB device operations, 9.2 to 9.2.7
GEOPTU state, 11.6.4, 11.6.4.5
GetConfiguration() request, GET_CONFIGURATION
hub requests, 11.24.1
overview, 9.4.2
returning interface descriptors, 9.6.5
standard device request codes, 9.4
GetDescriptor() request, GET_DESCRIPTOR, 11.23.1
device_qualifer descriptors, 9.6.2
endpoint descriptors, 9.6.6

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

GetDescriptor() request *(continued)*
GetDescriptor(CONFIGURATION) request, 9.5, 9.6.6
GetHubDescriptor() request, 11.24.2.5
hub class requests, 11.24.2
hub descriptors, 11.24.2.5
hub requests, 11.24.1
interface descriptors, 9.6.5
other_speed_configuration descriptors, 9.6.4
overview, 9.4.3
standard device request codes, 9.4
GetHubDescriptor() request, 11.24.2, 11.24.2.5
GetHubStatus() request, 11.24.2, 11.24.2.6
GetInterface() request, GET_INTERFACE
alternate settings for interfaces, 9.2.3
hub requests, 11.24.1
interface descriptors, 9.6.5
overview, 9.4.4
standard device request codes, 9.4
GetPortStatus() request
class-specific requests, 11.24.2
overview, 11.24.2.7 to 11.24.2.7.2.5
PORT_INDICATOR, 11.24.2.7.1.10
during test mode, 11.24.2.13
GET_STATE, 11.24.2
GetStatus() request, GET_STATUS
GetHubStatus() request, 11.24.2.6
GetPortStatus() request, 11.24.2.7
hub class requests, 11.24.2
overview, 9.4.5
PORT, 11.12.6
standard device request codes, 9.4
GetTTState() request, GET_TT_STATE
hub class requests, 11.24.2
overview, 11.24.2.8
STOP_TT, 11.24.2.11
global declarations in state machines, B.1
global resumes
frame and microframe timer synchronization, 11.2.3.3
global resumes *(continued)*
hub support, 11.9
signaling, 7.1.7.7
global suspend, 7.1.7.6.1, 11.9
glossary, 2.0
GND leads
cable electrical characteristics, 7.3.2 *Table 7-12*
captive cable assemblies, 6.4.2, 6.4.3
detachable cables, 6.4.1
electrical specifications overview, 4.2.1
standardized contact terminating assignments, 6.5.2
GResume state, 11.4, 11.4.4
grounding, 6.8

H

halted pipes, 10.5.2.2
Halt feature
bulk transfers, 5.8.5
control transfers, 5.5.5, 8.5.3.4
functional stalls, 8.4.5
GetStatus() request, 9.4.5
interrupt transfers, 5.7.5, 8.5.4
isochronous transfers, 5.6.5
responses to standard device requests, 9.4
handshakes. *See also* ACKs; NAKs; STALLs
ACK PID, 8.3.1 *Table 8-1*
bulk transfers, 8.5.2
bus protocol overview, 4.4
defined, 2.0 *glossary*
detection handshakes, 7.1.7.5, 7.1.7.6
function response to IN transactions, 8.4.6.1
function response to OUT transactions, 8.4.6.3
function response to SETUP transactions, 8.4.6.4
handshake responses, 8.4.6 to 8.4.6.4
host response to IN transactions, 8.4.6.2
isochronous transfers, 5.6.5, 5.12.7

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

handshakes *(continued)*
NAK PID, 5.9.1, 8.3.1 *Table 8-1,* 8.5.1, 8.5.1.1
NYET PID, 8.3.1 *Table 8-1,* 8.4.5, 8.5.1, 8.5.2
overview, 8.3.1 *Table 8-1,* 8.4.5
packet field formats, 8.3 to 8.3.5.2
PING flow control and OUT transactions, 8.5.1, 8.5.1.1
STALL PID, 8.3.1 *Table 8-1*
total allocation of bit times, 11.3.3
transaction notation for, 11.15
hardwired cable assemblies, 6.4.2
HCD (Host Controller Driver)
defined, 2.0 *glossary,* 5.3
HCDI (Host Controller Driver Interface), 10.1.1, 10.4
overview, 10.4
software interface overview, 10.3
in transfer management, 5.11.1, 5.11.1.3
in USB topology, 5.2.1, 10.1.1
HC_Data_or_error state machine, 11.20.2
HCDI (Host Controller Driver Interface), 10.1.1, 10.4
HC_Do_BCINTI state machine, 8.5.2 *Figure 8-33*
HC_Do_BCINTO state machine, 8.5.2 *Figure 8-31*
HC_Do_BICS state machine, 11.17.2
HC_Do_BISS state machine, 11.17.2
HC_Do_BOCS state machine, 11.17.2
HC_Do_BOSS state machine, 11.17.2
HC_Do_complete state machine, 11.16.1.1.2
HC_Do_IntICS state machine, 11.20.2
HC_Do_IntISS state machine, 11.20.2
HC_Do_intOCS state machine, 11.20.2
HC_Do_IntOSS state machine, 11.20.2
HC_Do_IsochICS state machine, 11.21.2
HC_Do_IsochISS state machine, 11.21.2
HC_Do_IsochI state machine, 8.5.5 *Figure 8-42*
HC_Do_IsochOSS state machine, 11.21.2
HC_Do_IsochO state machine, 8.5.5 *Figure 8-40*
HC_Do_nonsplit state machine, 8.5 *Figure 8-26*
HC_Do_Start state machine, 11.16.1.1.1
HC_HS_BCO state machine, 8.5.1 *Figure 8-27*
HC_Process_command state machine, 11.16.1.1
HEOP signal/event, 11.6.4 *Table 11-9,* 11.7.2.1, 11.7.2.3 *Table 11-11*
hierarchical state machines, 8.5, 11.15
high-bandwidth endpoints, 2.0 *glossary,* 5.7.4, 5.9 to 5.9.2, 5.12.3
high-bandwidth transactions, 5.12.7, 8.5.5
high-/full-speed cables
cable delay, 7.1.16
cable impedance tests, 6.7 *Table 6-7*
captive cable assemblies, 6.4.2
construction, 6.6, 6.6.2
description, 6.6.1
listing, 6.6.5
signal pair attenuation, 6.7 *Table 6-7*
specifications, 6.3
standards for, 6.6.3, 6.6.4, 6.7
high-powered devices
bus-powered functions, 7.2.1, 7.2.1.4
high-power ports, 7.2.1
voltage drop budget, 7.2.2
High-speed Detection Handshake, 7.1.7.5, 7.1.7.6
high-speed driver characteristics, 7.1 *Table 7-1*
high-speed functions and hubs
in application space taxonomy, 3.2
bulk transfers, 5.8.4
connect detection, 7.1.7.3
control transfers, 5.5.3, 5.5.4, 5.5.4 *Table 5-3*
data signaling rates, 7.1.11
defined, 2.0 *glossary*
detachable cables, 6.4.1
device_qualifier descriptors, 9.6.1, 9.6.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

high-speed functions and hubs *(continued)*
frame and microframe numbers, 8.4.3.1
full-speed operation, 5.3.1.1
getting port status, 11.24.2.7.1
high-speed repeaters, 11.2.5, 11.7.2.1
high-speed source electrical characteristics, 7.3.2 *Table 7-8*
hub/repeater electrical characteristics, 7.3.2 *Table 7-11*
input capacitance, 7.1.6.2
interrupt transfers, 5.7.3, 5.7.4 *Table 5-8*
isochronous transfers, 5.6.4
maximum data payload, 8.4.4
NAK mechanisms, 8.5.1, 8.5.1.1
other_speed_configuration descriptors, 9.6.4
performance, 1.1
in physical bus topology, 5.2.3
port selector state machine, 11.7.1.4 to 11.7.1.4.4
reset signaling, 7.1.7.5
reset state machines, C.2.3
sampling rates, 5.12.4.2
signal termination, 7.1
SOF PID and, 8.4.3, 8.4.3.1
speed detection and, 7.1.5.2, 11.8.2
test mode support, 7.1.20
high-speed port selector state machine, 11.7.1.4 to 11.7.1.4.4
high-speed signaling
bit stuffing, 7.1.9.2
bus transactions and, 4.4
calculating transaction times, 5.11.3
Chirp J and K states, 7.1.4.2, 7.1.7.2
control transfers, 5.5.4
current drivers, 7.1 *Table 7-1*
data handling, 11.14.1.1
data rates, 4.2.1
data signaling overview, 7.1.7.4.2
defined, 2.0 *glossary*
Differential 0/Differential 1 bus states, 7.1.7.2
high-speed signalling *(continued)*
differential data receivers, 7.1 *Table 7-1*
"disallowed" situations, 7.1.2.3
Disconnect state, 7.1.7.2
downstream and upstream facing ports, 7.1
driver characteristics, 7.1.1, 7.1.1.1, 7.1.1.3, 7.1.2.3
End of High-speed Packet (HSEOP), 7.1.7.2, 7.1.7.4.2
endpoints
attributes, 9.6.6
endpoint zero requirements, 5.3.1.1
high-bandwidth endpoints, 2.0 *glossary*, 5.9 to 5.9.2
EOF timing points, 11.2.5.1
EOP width, 7.1.13.2.2
error detection, 8.7.3, 8.7.4
error handling, 5.12.7, 8.6.4
eye patterns, 7.1.2.2
frame and microframe generation, 10.2.3
full-speed signaling and, 5.3.1.1, 7.1.1.1
high-speed devices operating at full-speed, 5.3.1.1
hub architecture and, 11.1.1
hub class descriptors and, 11.23.1
Idle state, 7.1.7.2
isochronous transaction limits, 5.6.3
J and K states, 7.1.7.2, 7.1.7.4.2
jitter, 7.1.13.1.2, 7.1.15.2
microframes and, 5.3.3, 11.2.1
overview, 7.1
packet repeaters, 11.7.1 to 11.7.1.4.4
PING flow control protocol and, 5.5.4, 5.8.4
propagation delays, 7.1.14.2
receiver characteristics, 7.1.4.2
reset signaling, 7.1.7.5, 7.1.7.6
resume signaling, 7.1.7.7, 11.9
sampling rates, 5.12.4.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

high-speed signaling *(continued)*
signaling levels, 7.1.7.2
split transactions, 5.10, 8.4.2 to 8.4.2.3
Squelch state, 7.1.7.2
Start of High-speed Packet (HSSOP), 7.1.7.4.2
Start of high-speed Packet (HSSOP), 7.1.7.2
Suspended state and, 7.1.7.6, 11.9
synch frame requests, 9.4.11
timer range, 11.2.1
toggle sequencing, 8.5.5
Transaction Translator, 4.8.2.1, 11.1, 11.14 to 11.14.2.3
types of transactions, 11.17
HJ signal/event, 11.6.3 *Table 11-8*
HK signal/event, 11.6.3 *Table 11-8*
host, 10
in bus topology, 4.1.1.1, 5.2, 5.2.1
collecting status and activity statistics, 10.1.4
components, 10.1.1
control mechanisms, 10.1.2
EOF1 and EOF2 timing points, 11.2.5 to 11.2.5.2
host behavior at end-of-frame, 11.3 to 11.3.3
host-to-hub communications, 11.1
resource management, 10.3.2
responsibilities and capabilities, 10.1.1
role in assigning addresses, 9.2.2
role in configuration, 9.2.3, 10.3.1
synchronizing hub (micro)frame timer to host (micro)frame period, 11.2
data flow, 10.1.3
common data definitions, 10.3.4
data-rate tolerance, 7.1.11
data transfer mechanisms, 10.1.3, 10.3.3
detailed communication flow illustrated, 5.3
host *(continued)*
data flow *(continued)*
host response to IN transactions, 8.4.6.2
interlayer communications model, 10.1.1
role in bulk transfers, 8.5.2
defined, 2.0 *glossary,* 4.9
electrical considerations, 10.1.5
jitter budget table, 7.1.15.1
over-current protection, 7.2.1.2.1
over-current recovery, 11.12.5
hardware and software, 10.0
host accuracy variations, 11.2.1 to 11.2.2
Host Controller Driver (HCD), 10.4 (*See also* HCD)
Host Controller responsibilities, 4.9, 10.2 (*See also* Host Controller)
host jitter, 11.2.1
host tolerance, hub (micro)frame timer and, 11.2
operating system environment guides, 10.6
overview of USB Host, 10.1 to 10.1.5
power management overview, 4.3.2
software mechanisms, 10.3 to 10.3.4
split transaction scheduling, 11.18.4
state machines, 8.5.1, 8.5.2, 8.5.5
status in USBD pipe state, 10.5.2.2
turn-around timers, 8.7.2
Universal Serial Bus Driver (USBD), 10.5 to 10.5.5 (*See also* USBD (USB Driver))
USB System Software responsibilities, 4.9 (*See also* USB System Software)
Host Controller, 4.9
best case full-speed budgets, 11.18.1, 11.18.4
in bus topology, 5.2.1
calculating buffer sizes in functions and software, 5.11.4

Host Controller *(continued)*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

data transfer mechanisms, 10.1.3
bulk transfers, 5.8.3, 11.17 to 11.17.5
control transfers, 5.5.3, 5.5.4, 11.17 to 11.17.5
data processing, 10.2.4
data-rate tolerance, 7.1.11
high bandwidth transfers, 5.9.1, 5.9.2, 11.20.2
interrupt transfers, 5.7.3, 5.9.1
isochronous transfers, 5.6.3, 5.9.2, 11.21.2
non-periodic transactions, 11.17 to 11.17.5
periodic transactions, 11.18 to 11.18.8
role in transfer management, 5.11.1, 5.11.1.5
split transactions, 11.16.1 to 11.16.1.1.2
tracking transactions, 5.11.2
transaction list, 5.11.1.4
transmission error handling, 10.2.6
declarations in state machines, B.2
defined, 2.0 *glossary,* 4.9
frame and microframe generation, 10.2.3
HCD and HCDI overview, 10.4 (*See also* HCD; HCDI)
host behavior at end-of-frame, 11.3
host-system interface, 10.2.9
as implementation focus area, 5.1
implemented in USB Bus interface, 10.1.1
multiple Host Controllers, 4.10
passing preboot control to operating system, 10.5.5
port resets, 10.2.8.1
protocol engine, 10.2.5
remote wakeup and, 10.2.7
requirements, 10.2
root hub and, 10.2.8
serializer/deserializer, 10.2.2
Host Controller *(continued)*
state handling, 10.2.1
state machines, 8.5, 11.16 to 11.16.1.1.2, 11.17.2, 11.20.2, 11.21.2, B.2
status and activity monitoring, 10.1.4
test mode support, 7.1.20
Transaction Translator and, 11.14.1, 11.14.1.2
USB System interaction, 10.1.1
Host Controller Driver. *See* HCD (Host Controller Driver)
Host Controller Driver Interface (HCDI), 10.1.1, 10.4
host resources, 2.0 *glossary*
host side bus interface. *See* Host Controller
host software
in bus topology, 5.2.1
as component of USB System, 10.1.1
pipes and, 10.5.1.2
status and activity monitoring, 10.1.4
host-to-function transfers. *See* OUT PID
hot plugging. *See* dynamic insertion and removal
HSD1 packets (downstream packets), 8.5, 11.15
HS_Drive_Enable, 7.1.1.3
HSEOP (End of High-speed Packet), 7.1.7.2, 7.1.7.4.2
HS_Idle signal/event, 11.6.2, 11.6.3 *Table 11-8*
HS signal/event, 11.6.3 *Table 11-8,* 11.6.4 *Table 11-9*
HSSOP (Start of High-speed Packet), 7.1.7.2, 7.1.7.4.2
HSU2 packets (upstream packets), 8.5, 11.15
Hub address field, 8.4.2.2
Hub Change field, 11.4.4, 11.24.2.6
hub class definitions
additional endpoints, 11.12.1
feature selectors, 11.24.2
request codes, 11.24.2
hub class definitions *(continued)*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

root hub and, 10.4
Hub Controller, 11.12 to 11.12.6
control commands, 11.1
defined, 4.8.2.1
endpoint organization, 11.12.1
hub and port change information processing, 11.12.3, 11.12.4
in hub architecture, 11.1.1, 11.12.2
internal port connection, 11.4
over-current reporting and recovery, 11.12.5
power distribution and, 7.2.1.1
role in host-to-hub communications, 11.1
status commands, 11.1
hub descriptors, 11.12.2
Hub Repeater
Collision conditions, 11.8.3
connectivity setup and tear-down, 11.1
data recovery unit, 11.7.1.2
defined, 4.8.2.1
dynamic insertion and removal, 11.1
elasticity buffer, 11.7.1.3
electrical characteristics, 7.3.2 *Table 7-11*
fault detection and recovery, 11.1
high-speed packet repeaters, 11.7.1 to 11.7.1.4.4
in hub architecture, 11.1.1
hub signaling timings, 7.1.14.1
internal port connection, 11.4
packet signaling connectivity, 11.1.2.1
repeater state descriptions, 11.2.3.3, 11.7 to 11.7.6
squelch circuit, 11.7.1.1
Wait for End of Packet (WFEOP), 11.7.6
Wait for End of Packet from Upstream Port state (WFEOPFU), 11.7.4
Wait for Start of Packet (WFSOP), 11.7.5
Wait for Start of Packet from Upstream Port state (WFSOPFU), 11.7.3
Hub Repeater state machine, 11.2.3.3, 11.7.2
hubs, 11. *See also* Hub Controller; Hub Repeater; ports
accuracy variations, 11.2.1 to 11.2.2
architecture, 4.1.1.2, 11.1
bus states
bus state evaluation, 11.8 to 11.8.4.1
collision, 11.8.3
connect/disconnect detection, 11.1
full- *vs.* low-speed behavior, 11.8.4
low-speed keep-alive, 7.1.7.1, 11.8.4.1
port error, 11.8.1
reset behavior, 11.10
speed detection, 11.8.2
status change detection, 7.1.7.5, 11.12.2, 11.12.4
in bus topology, 5.2.3, 5.2.4
characteristics and configuration, 11.13
clearing features, 11.24.2.6
data-rate tolerance, 7.1.11
descriptors, 11.12.2, 11.23 to 11.23.2.1, 11.24.2.10
full- *vs.* low-speed behavior, 11.8.4
input capacitance, 7.1.6.1
speed detection of devices, 11.8.2
typical configuration illustrated, 4.8.2.1
connectivity behavior, 11.1, 11.1.2 to 11.1.2.3
controlling hubs, 7.1.7.7
defined, 2.0 *glossary*
downstream ports, 11.5 to 11.5.1.15
dynamic insertion and removal role, 4.6.1, 4.6.2
embedded hubs, 4.8.2.2
enumeration handling, 11.12.6
fault detection and recovery, 11.1
Hub Controller, 4.8.2.1, 11.1, 11.12 to 11.12.6 (*See also* Hub Controller)
hub drivers, 10.3.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

hubs *(continued)*
Hub Repeater, 4.8.2.1, 7.3.2 *Table 7-11,* 11.1, 11.7 to 11.7.6 (*See also* Hub Repeater)
hub tier, 2.0 *glossary*
intermediate hubs, 7.1.7.7
internal ports, 11.4 to 11.4.4
labeling on port connectors, 11.5.3.1
overview, 4.8.2.1, 11.1 to 11.1.2.3
port indicators, 11.5.3 to 11.5.3.1
power management, 11.1
bus-powered hubs, 4.3.1, 7.2.1.1
hub port power control, 11.11
multiple gangs, 11.11.1
over-current reporting and recovery, 11.12.5
power source and sink requirements, 7.2.1
self-powered hubs, 7.2.1.2
surge limiting, 7.2.4.1
requests, 11.24 to 11.24.2.13
root hubs, 2.0 *glossary*
signaling and timing
high-speed rise and fall times, 7.1.2.2
high-speed signal isolation, 8.6.5
high-speed support, 7
host behavior at end-of-frame, 11.3 to 11.3.3
hub chain jitter, 11.2.1
hub differential delay, 7.3.3 *Figure 7-52*
hub EOP delay and EOP skew, 7.3.3 *Figure 7-53*
hub event timings, 7.3.2 *Table 7-13*
hub frame timer, 11.2 to 11.2.5.2
hub (micro)frame timer, 11.2
hub signaling timings, 7.1.14 to 7.1.14.2
hub switching skews, 7.1.9.1
jitter budget table, 7.1.15.1
low-speed keep-alive strobe, 7.1.7.1, 11.8.4.1
hubs *(continued)*
signaling and timing *(continued)*
power-on and connection events timing, 7.1.7.3
reset signaling, 7.1.7.5
resume signaling, 7.1.7.7
signaling delays, 7.1.14.1
suspend and resume signaling, 11.9
tracking frame and microframe intervals, 7.1.12
sync pattern, 7.1.10
test mode support, 7.1.20
Transaction Translator, 4.8.2.1, 11.1, 11.14 to 11.14.2.3
upstream ports, 11.6 to 11.6.4.6
Hub State Machine, 11.1.1
Hub Status field, 11.24.2.6
humidity life standards, 6.7 *Table 6-7*
hybrid powered hubs, 7.2.1.2
hysteresis in single-ended receivers, 7.1.4.1

I

iConfiguration field
configuration descriptors, 9.6.3, 11.23.1
other speed configuration descriptors, 9.6.4, 11.23.1
Icon for USB plugs and receptacles, 6.5, 6.5.1
Idle bus state
data signaling overview, 7.1.7.4.1, 7.1.7.4.2
downstream facing ports in high-speed, 7.1.7.6.1, 7.1.7.6.2
high-speed driver characteristics and, 7.1.1.3
high-speed signaling, 7.1
high-speed TDR measurements, 7.1.6.2
hub connectivity and, 11.1.2.1
Idle-to-K state transition, 7.1.14.1
NRZI data encoding, 7.1.8
signaling levels and, 7.1.7.1, 7.1.7.2
idle pipes, 5.3.2, 10.5.2.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

idProduct field (device descriptors), 9.6.1
idVendor field (device descriptors), 9.6.1
iInterface field (interface descriptors), 9.6.5, 11.23.1
iManufacturer field (device descriptors), 9.6.1
impedance
cable impedance tests, 6.7 *Table 6-7*
detachable cable assemblies, 6.4.1
differential cable impedance, 7.1, 7.3.2 *Table 7-12*
Exception Window, 7.1.6.2
full-speed connections, 7.1.1.1
high-/full-speed captive cable assemblies, 6.4.2
high-speed driver characteristics, 7.1.1.3
input impedance, 7.3.2 *Table 7-7*
input impedance of ports, 7.1.6.1
Termination Impedance, 7.1.6.2
test mode, 7.1.20
Through Impedance, 7.1.6.2
zero impedance voltage sources, 7.1.1
Implementers Forum, 1.4, 2.0 *glossary*
implementer viewpoints of data flow models, 5.1
implicit feedback for transfers, 5.12.4.3, 9.6.5
Inactive state, 11.4, 11.4.1, 11.6.4, 11.6.4.1, 11.7.1.4.1
in-band signaling, 10.1.2
incident rise times, 7.1.6.2
indicators
descriptors, 11.23.2.1
lights on devices, 11.5.3 to 11.5.3.1
port status indicators, 11.24.2.7.1, 11.24.2.7.1.10
selectors, 11.24.2.12
initial frequency inaccuracies, 7.1.11
initialization of USBD, 10.5.1.1
initial states, 8.5, 11.15
injection molded thermoplastic insulator material, 6.5.3.1, 6.5.4.1
inner shielding in cables, 6.6.2
IN PID, 8.3.1 *Table 8-1*
aborting IN transactions, 11.18.6.1
ACK handshake and, 8.4.5
ADDR field, 8.3.2.1
bit times and, 11.3.3
bulk transfers, 8.5.2, 8.5.2 *Figure 8-33,* 8.5.2 *Figure 8-34,* 11.17.1, 11.17.2
complete-split flow sequence, 11.20.1
control transfers, 8.5.2 *Figure 8-33,* 8.5.2 *Figure 8-34,* 8.5.3, 8.5.3.1, 11.17.1, 11.17.2
ENDP field, 8.3.2.2
error handling on last data transaction, 8.5.3.3
function response to, 8.4.6.1
high bandwidth transactions and, 5.9.2
host response to, 8.4.6.2
interrupt transfers, 8.5.2 *Figure 8-33,* 8.5.2 *Figure 8-34,* 8.5.4, 11.20.4
intervals between IN token and EOP, 11.3.3
isochronous transfers, 8.5.5, 8.5.5 *Figure 8-42,* 8.5.5 *Figure 8-43,* 11.21 to 11.21.4
low-speed transactions, 8.6.5
NAK handshake and, 8.4.5
prebuffering data, 5.12.5
scheduling IN transactions, 11.18.4
split transaction conversion, 8.4.2.1
split transaction examples, A.2, A.4, A.6
STALL handshake and, 8.4.5
start-split flow sequence, 11.20.1
state machines
bulk/control state machines, 8.5.2 *Figure 8-33,* 8.5.2 *Figure 8-34,* 11.17.2
interrupt state machines, 8.5.2 *Figure 8-33,* 8.5.2 *Figure 8-34,* 11.20.2
isochronous state machines, 8.5.5 *Figure 8-42,* 8.5.5 *Figure 8-43,* 11.21.2
token CRCs, 8.3.5.1
token packets, 8.4.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

IN PID *(continued)*
Transaction Translator response generation, 11.18.5
input capacitance, 7.1.2.1, 7.3.2 *Table 7-7*
input characteristics (signaling), 7.1.6.1
input impedance, 7.3.2 *Table 7-7*
input impedance of ports, 7.1.6.1
input levels, 7.3.2 *Table 7-7*
inputs (Series "B" receptacles), 6.2
input sensitivity of differential input receivers, 7.1.4.1
input transitions in state machines, 8.5
inrush current limiting
bus-powered hubs, 7.2.1.1
dynamic attach and detach, 7.2.4.1
remote wakeup and, 7.2.3
self-powered functions, 7.2.1.5
suspend/resume and, 7.2.3
inserting devices. *See* dynamic insertion and removal
insertion force standards, 6.7 *Table 6-7*
instancing of USBD, 10.5
insulation
cables, 6.6.2
insulator materials, 6.5.3.1, 6.5.4.1
resistance standards, 6.7 *Table 6-7*
interconnect model, 4.1, 5.12.4.4
interface class codes, 9.6.5
INTERFACE descriptor, 9.4 *Table 9-5*
interface descriptors
GetDescriptor() request, 9.4.3
hubs, 11.23.1
INTERFACE_POWER descriptor, 9.4 *Table 9-5*
overview, 9.6.5
interface numbers, 9.2.3, 9.6.5
INTERFACE_POWER descriptor, 9.4 *Table 9-5*
interfaces
alternate interfaces, 9.2.3, 10.5.2.10
alternate settings, 9.6.5
interfaces *(continued)*
in configuration descriptors, 9.6.3, 9.6.4
defined, 9.2.3
in device class definitions, 9.7.2
as endpoint sets, 5.3
getting interface status, 9.4.4, 9.4.5
host-system interface, 10.2.9
interface class codes, 9.6.5
interface descriptors, 9.4.3, 9.6.5, 11.23.1
interface numbers, 9.2.3, 9.6.5
interface subclass codes, 9.6.5
plug interface and mating drawings, 6.5.4
setting interface state, 9.4.10, 10.5.2.1
specifying in *wIndex* field, 9.3.4
interfaces of plugs, 6.5.3
interface state control, 10.5.2.1
interface subclass codes, 9.6.5
interlayer communications model, 4.1, 10.1.1
intermediate hubs, 7.1.7.7
internal clock source jitter, 7.1.13.1.1
internal ports
Full Suspend (Fsus) state, 11.4.3
Generate Resume (GResume) state, 11.4.4
Inactive state, 11.4.1
Suspend Delay state, 11.4.2
internal port state machine, 11.4
interpacket delay, 7.1.18 to 7.1.18.2
interpacket gaps, 11.18.2
interrupt endpoints, 11.12.1
interrupt requests, defined, 2.0 *glossary*
interrupt transfers. *See also* periodic transactions
aborting, 11.18.6.1
bus access constraints, 5.7.4
data format, 5.7.1
data sequences, 5.7.5
defined, 2.0 *glossary,* 5.4

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

interrupt transfers. *(continued)*
direction, 5.7.2
flow sequences, 11.18.8, 11.20 to 11.20.4
full-speed transfer limits, 5.7.4
high- bandwidth endpoints, 5.9.1
high-speed transfer limits, 5.7.4 *Table 5-8*
low-speed transfer limits, 5.7.4
multiple transactions and, 9.6.6
overview, 4.7, 4.7.3, 5.7
packet size, 5.7.3, 9.6.6
periodic transactions, 11.18 to 11.18.8
required bus access period, 5.7.4
scheduling and buffering, 11.14.2.1
sequencing, 11.20.3, 11.20.4
split transactions
examples, A.3, A.4
host handling of, 5.10
notation for, 11.15
state machines, 8.5.2, 11.20 to 11.20.4
transaction format, 8.5.4
transaction organization within IRPs, 5.11.2
Transaction Translator response generation, 11.18.5
USBD pipe mechanism responsibilities, 10.5.3.1.2
intervals
debounce intervals in connection events, 7.1.7.3
frame and microframe intervals, 7.1.12, 7.3.2 *Table 7-8,* 7.3.2 *Table 7-9,* 7.3.2 *Table 7-10*
Resetting state and Resuming state intervals, 11.5.1.10
resume and recovery intervals for devices, 9.2.6.2
service and polling intervals, 2.0 *glossary,* 9.6.6, 10.3.3
timeout intervals, 8.7.3
between IN token and EOP, 11.3.3
interwoven tinned copper wire, 6.6.2
IN transactions. *See* IN PID
I/O buffers. *See* buffers
I/O Request Packets. *See* IRPs
iProduct field (device descriptors), 9.6.1
IRPs. *See also* requests; transfers
aborting/retiring, 5.3.2, 10.5.3.2.1
class-specific requests, 9.2.6.5
client software role in, 5.11.1.1
defined, 2.0 *glossary,* 5.3.2
HCD tracking of, 5.11.1.3
multiple data payloads in, 5.3.2
pipes and, 5.3.2
queuing IRPs, 10.5.3.2.3
request processing overview, 9.2.6
reset/resume recovery time, 9.2.6.2
set address processing, 9.2.6.3
STALLS and, 5.3.2
standard device requests, 9.2.6.4, 9.4 to 9.4.11
timing, 9.2.6.1, 9.2.6.3, 9.2.6.4
transaction organization within IRPs, 5.11.2
USB device requests, 9.3 to 9.3.5
USBD role in, 10.1.1
IRQs, defined, 2.0 *glossary*
iSerialNumber field (device descriptors), 9.6.1
isochronous data, defined, 2.0 *glossary*
isochronous devices, defined, 2.0 *glossary*
isochronous sink endpoints, defined, 2.0 *glossary*
isochronous source endpoints, defined, 2.0 *glossary*
isochronous transactions. *See also* periodic transactions
isochronous transfers
buffering, 5.12.8, 11.14.2.1
bus access constraints, 5.6.4
clock model, 5.12.2
clock synchronization, 5.12.3
connectivity, 5.12.4.4
data format, 5.6.1
data prebuffering, 5.12.5

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

isochronous transfers *(continued)*
data sequences, 5.6.5
defined, 2.0 *glossary,* 5.4
direction, 5.6.2
endpoint attributes, 9.6.6
endpoint synchronization frame, 9.4.11
error handling, 5.12.7
feedback, 5.12.4.2
high-bandwidth endpoints, 2.0 *glossary,* 5.9.2
illustrated, 5.12.2
implicit feedback, 5.12.4.3, 9.6.5
isochronous device framework, 5.12.4
multiple transactions and, 9.6.6
non-USB example isochronous application, 5.12.1
non-zero data payload, 5.6.3
overview, 4.7.4, 5.6, 11.21 to 11.21.4
packet size, 5.6.3, 9.6.6
periodic transactions, 11.18 to 11.18.8
rate matching, 5.12.8
scheduling, 11.14.2.1
sequencing, 11.21.3, 11.21.4
SOF tracking, 5.12.6
special considerations, 5.12 to 5.12.8
specifying required bus access period, 5.6.4
split transactions
examples, A.5, A.6
flow sequences, 11.18.8
host controller and, 5.10
notation for, 11.15
state machines, 8.5.5, 11.21 to 11.21.4
synchronization types, 5.12.4.1.1, 5.12.4.1.2, 5.12.4.1.3
transaction format, 8.5.5
transaction organization within IRPs, 5.11.2
Transaction Translator response generation, 11.18.5
USBD pipe mechanism responsibilities, 10.5.3.1.1
USB features and, 3.3
isochronous transfers *(continued)*
USB System Software role, 4.9
ISO transfers. *See* isochronous transfers
ITCW (interwoven tinned copper wire), 6.6.2

J

J bus state
data signaling overview, 7.1.7.4.1
high-speed signaling and, 7.1, 7.1.1.3
J-to-K state transition, 7.1.14.1
NRZI data encoding, 7.1.8
signaling levels and, 7.1.7.1, 7.1.7.2
test mode, 7.1.20
J signal/event
differential transmissions, 11.6.1
receiver state machine, 11.6.3 *Table 11-8*
transmission sequence, 7
transmitter state machine, 11.6.4 *Table 11-9*
jacketing in cables, 6.6.2
jitter
clock jitter, 5.12.3
cumulative jitter in high-speed signaling, 7.1.14.2
data source jitter, 7.1.13.1 to 7.1.13.1.2
defined, 2.0 *glossary*
differential data jitter, 7.3.3 *Figure 7-49*
differential jitter, 7.3.3 *Figure 7-52*
frame and microframe jitter, 11.2.4
full-speed source electrical characteristics, 7.3.2 *Table 7-9*
high-speed source electrical characteristics, 7.3.2 *Table 7-8*
host jitter, 11.2.1
hub chain jitter, 11.2.1
hub/repeater electrical characteristics, 7.3.2 *Table 7-11*
internal clock source jitter, 7.1.13.1.1
low-speed source electrical characteristics, 7.3.2 *Table 7-10*
in non-USB isochronous application, 5.12.1
output driver jitter, 7.1.15.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

jitter *(continued)*
receiver data jitter, 7.1.15 to 7.1.15.2, 7.3.3 *Figure 7-51*
service jitter, 2.0 *glossary*
test mode, 7.1.20
Transaction Translator frame clock and, 11.18.3
joint symbols in state machine transitions, 8.5, 11.15

K

K bus state
data signaling overview, 7.1.7.4.1
high speed signaling and, 7.1.1.3
high-speed signaling and, 7.1
Idle-to-K state transition, 7.1.14.1
K-to-J state transition, 7.1.14.1
NRZI data encoding, 7.1.8
signaling levels and, 7.1.7.1, 7.1.7.2, 7.1.7.4.2
test mode, 7.1.20
K signal/event
differential transmissions, 11.6.1
downstream port state machine, 11.5 *Table 11-5*
receiver state machine, 11.6.3 *Table 11-8*
transmission sequence, 7
transmitter state machine, 11.6.4 *Table 11-9*
kB/S and kb/S, defined, 2.0 *glossary*
keep-alive strobe, 7.1.7.1, 7.1.7.6, 11.8.4.1
keyed connector protocol, 6.2

L

labeling on port connectors, 11.5.3.1
LANGID code array, 9.6.7
language IDs in string descriptors, 9.6.7
latency
constraints for transfers, 5.4
packets, 11.7.1.3
latest host packet, 11.3.1
least-significant bit (LSb), 2.0 *glossary,* 8.1
least-significant byte (LSB), 2.0 *glossary,* 8.1
length of cables, 6.4.1, 6.4.2, 6.4.3
listing, UL listing for cables, 6.6.5
little endian order, 2.0 *glossary,* 8.1
LOA, 2.0 *glossary,* 8.7.4
load capacitance, 6.4.3, 6.7 *Table 6-7*
Local Power Source field, 11.24.2.6
Local Power Status Change field, 11.24.2.6
Local Power Status field, 11.24.2.7.1.6
local power supplies, 7.2.1.2, 7.2.1.5
locking hub frame timer, 11.2.3
Lock signal/event, 11.7.2.3 *Table 11-11*
logical bus topology, 5.2, 5.2.4
logical devices
in bus topology, 5.2.2
as collections of endpoints, 5.3
detailed communication flow illustrated, 5.3
unique addresses and endpoints, 5.3.1
Logical Power Switching Mode field, 11.11, 11.11.1, 11.23.2.1
logo location on connectors, 6.5.1
LOI, 6.5.3.1, 6.5.4.1
loss, cable, 7.1.17
loss of bus activity. *See* LOA
loss of synchronization, 11.22.1
low level contact resistance standards, 6.7 *Table 6-7*
low-power bus-powered functions, 7.2.1, 7.2.1.3
low-power hubs, 7.2.2
low-power ports, 7.2.1
low-speed buffers, 7.1.2.1
low-speed cables
cable environmental characteristics, 6.6.4
captive cable assemblies, 6.4.3
configuration overview, 6.6
construction, 6.6.2
description, 6.6.1
detachable cables, 6.4.4
listing, 6.6.5
specifications, 6.3

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

low-speed cables *(continued)*
standards for, 6.6.3, 6.7
low-speed driver characteristics, 7.1.1.2, 7.1 *Table 7-1*
low-speed functions and hubs
cable and resistor connections, 7.1.5.1
connect detection, 7.1.7.3
control transfers and, 5.5.3, 5.5.4, 5.5.4 *Table 5-1*
data-rate tolerance, 7.1.11
defined, 2.0 *glossary*
detachable cables and, 6.4.1
detecting, 11.24.2.7.1.7
full- *vs.* low-speed port behavior, 11.8.4
getting port status, 11.24.2.7.1
high-/full-speed captive cable assemblies, 6.4.2
hub/repeater electrical characteristics, 7.3.2 *Table 7-11*
hub support for, 11.1
input capacitance, 7.1.6.1
interrupt transfers and, 5.7.3, 5.7.4 *Table 5-6*
low-speed captive cable assemblies, 6.4.3
maximum data payload, 8.4.4
optional endpoints, 5.3.1.2
in physical bus topology, 5.2.3
signal termination, 7.1.5.1
speed detection and, 11.8.2
Transmit state and, 11.5.1.7
low-speed keep-alive strobe, 7.1.7.1, 7.1.7.6, 11.8.4.1
low-speed signaling
babble and loss of activity recovery, 8.7.4
bus transactions and, 4.4
calculating transaction times, 5.11.3
data rates, 4.2.1
data signaling overview, 7.1.7.4.1
data source jitter, 7.1.13.1.1
data toggle synchronization and retry, 8.6.5
low-speed signalling *(continued)*
defined, 2.0 *glossary*
differential receivers, 7.1 *Table 7-1*
downstream and upstream facing ports, 7.1
driver characteristics, 7.1.1, 7.1.2.3, 7.1 *Table 7-1*
EOP width, 7.1.13.2.1
errors, 8.6.4
host behavior at end-of-frame, 11.3 to 11.3.3
hub class descriptors and, 11.23.1
intervals between IN token and EOP, 11.3.3
J and K states, 7.1.7.1
jitter budget table, 7.1.15.1
low-speed loads, 7.1.2.1
propagation delays, 7.1.14.1
receiver characteristics, 7.1.4.1
reset signaling, 7.1.7.5
scheduling, 11.14.2.3
speed detection, 9.1.1.3
transactions illustrated, 8.6.5
Transaction Translator, 4.8.2.1, 11.18.5
low-speed source electrical characteristics, 7.3.2 *Table 7-10*
LSb and LSB
in bit ordering, 8.1
defined, 2.0 *glossary*
LS signal/event, 11.5
lumped capacitance guidelines for transceivers, 7.1.6.2

M

male plug contact materials, 6.5.4.3
manual port color indicators, 11.5.3
Manufacturer's logo location, 6.5.1
Manufacturer's names in device descriptors, 9.6.1
mapping physical and virtual devices, 5.12.4.4
marking on cables, 6.6.2
master clock, 5.12.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

material requirements
cables, 6.6 to 6.6.5
connectors, 6.5 to 6.5.4.3
mating area materials, 6.5.3.3, 6.5.4.3
MaxPower field, 9.6.4, 11.13
MB/S, defined, 2.0 *glossary*
Mb/S, defined, 2.0 *glossary*
MDATA PID
in data packets, 8.4.4
defined, 8.3.1 *Table 8-1*
high-bandwidth endpoints and, 2.0 *glossary,* 5.9.2
interrupt IN sequencing, 11.20.4
Transaction Translator response generation, 11.18.5
measurement planes in speed signaling eye patterns, 7.1.2.2
mechanical specifications, 6, 6.1
applicable documents, 6.7.1
architectural overview, 6.1
cable assembly, 6.4 to 6.4.4
cables, 6.3, 6.6 to 6.6.5
connectors, 6.2, 6.5 to 6.5.4.3
overview, 4.2.2, 6
PCB reference drawings, 6.9
standards for, 6.7, 6.7.1
USB grounding, 6.8
message pipes. *See also* control pipes; pipes
in bus protocol overview, 4.4
defined, 2.0 *glossary*, 5.3.2
overview, 5.3.2.2
microframe pipelines. *See under* frames and microframes
microframes. *See* frames and microframes
microframe timers. *See* frame and microframe timers
microphone non-USB isochronous application, 5.12.1
microphone USB isochronous application, 5.12.2
"middle" encoding, 11.18.4
modifying device configuration, 10.5.4.1.3
monotonic transitions, 7.1.2.1, 7.1.2.2
most-significant bit (MSb), 2.0 *glossary,* 8.1
most-significant byte (MSB), 2.0 *glossary,* 8.1, 11.24.2.13
MSb and MSB, 2.0 *glossary,* 8.1
multiple gangs, hubs and, 11.11.1
multiple Transaction Translators, 11.14.1.3, 11.23.1, 11.24.2.8

N

NAKs
in bulk transfers, 8.5.2, 11.17.1
busy endpoints, 5.3.2
in control transfers, 8.5.3.1, 11.17.1
data corrupted or not accepted, 8.6.3
defined, 2.0 *glossary*
function response to IN transactions, 8.4.6.1
function response to OUT transactions, 8.4.6.3
high bandwidth transactions and, 5.9.1
in interrupt transfers, 5.7.4, 8.5.4, 11.20.4
NAK limiting via Ping flow control, 8.5.1, 8.5.1.1
non-periodic transactions, 11.14.2.2, 11.17.1
overview, 8.3.1 *Table 8-1,* 8.4.5
Ready/NAK status, 11.15
test mode, 7.1.20
NEC Article 800 for communications cables, 6.6.4
next frame in hub timing, 11.2.3.1
No Acknowledge packet. *See* NAKs
nominal cable diameter, 6.6.2
nominal cable temperatures, 6.6.4
nominal twist ratio in signal pair, 6.6.2
non-acceptable cables, 6.4.4
non-periodic transactions
buffers, 11.14.1, 11.14.2.2, 11.19
failures, 11.17.5
overview, 11.17 to 11.17.5
scheduling, 11.14.2.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Non Return to Zero Invert.
See NRZI encoding

non-twisted power pair in cables, 6.6.1, 6.6.2

non-USB isochronous application, 5.12.1

non-zero data payloads, 5.6.3

Not Configured state, 11.5, 11.5.1.1

Not Packet state, 11.7.1.4.4

NRZI encoding, 7.1.8

- bit stuffing, 7.1.9
- defined, 2.0 *glossary*
- in electrical specifications overview, 4.2.1
- high-speed signaling and, 7.1
- sync pattern, 7.1.7.4.2, 7.1.10

NYET handshake

- aborting, 11.18.6.1
- bulk/control transactions, 8.5.2, 11.17.1
- defined, 8.3.1 *Table 8-1,* 8.4.5
- isochronous transactions, 11.21.1
- Ping flow control and, 8.5.1
- Transaction Translator response generation, 11.18.5

O

objects, defined, 2.0 *glossary*

offsets between host and hub, 11.2

Old status, 11.15

one-way propagation delay, 7.1.16, 7.3.2 *Table 7-9,* 7.3.2 *Table 7-10*

open architecture, USB development and, 1.2

open-circuit voltage, 7.1.1

operating systems

- companion specifications, 10.6
- device configuration, 10.3.1
- interaction with USBD, 10.5
- passing preboot control to, 10.5.5

operating temperatures for cables, 6.6.4

operations, generic USB device operations, 9.2 to 9.2.7

optional edge rate control capacitors, 7.1.6.1

OTHER_SPEED_CONFIGURATION

- descriptor, 9.2.6.6, 9.4 *Table 9-5*, 9.4.3, 9.6.2, 9.6.4

(other speed) device_qualifier descriptor, 9.2.6.6

other_speed requests, 9.2.6.6

out-of-band signaling, 10.1.2

OUT PID, 8.3.1 *Table 8-1*

- aborting OUT transactions, 11.18.6.1
- ACK handshake and, 8.4.5
- ADDR field, 8.3.2.1
- in bulk transfers, 8.5.2, 8.5.2 *Figure 8-31,* 8.5.2 *Figure 8-32,* 11.17.1, 11.17.2
- complete-split flow sequence, 11.20.1
- in control transfers, 8.5.2 *Figure 8-31,* 8.5.2 *Figure 8-32,* 8.5.3, 8.5.3.1, 11.17.1, 11.17.2
- in data toggle, 8.6.1
- ENDP field, 8.3.2.2
- function response to OUT transactions, 8.4.6.3
- high bandwidth transactions and, 5.9.2
- high-speed PING flow control protocol, 5.8.4
- in interrupt transfers, 8.5.2 *Figure 8-31,* 8.5.2 *Figure 8-32,* 8.5.4, 11.20.3
- in isochronous transfers, 8.5.5, 8.5.5 *Figure 8-40,* 8.5.5 *Figure 8-41,* 11.21 to 11.21.4
- NAK handshake and, 8.4.5
- OUT/DATA transactions, 8.5.1, 8.5.1.1
- PING flow control and OUT transactions, 8.5.1, 8.5.1.1
- prebuffering data, 5.12.5
- scheduling OUT transactions, 11.18.4
- split transaction conversion, 8.4.2.1, 8.4.2.2
- split transaction examples, A.1, A.3, A.5
- STALL handshake and, 8.4.5
- start-split flow sequence, 11.20.1
- state machines
  - bulk/control state machines, 8.5.2 *Figure 8-31,* 8.5.2 *Figure 8-32,* 11.17.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

OUT PID *(continued)*
state machines *(continued)*
interrupt state machines, 8.5.2 *Figure 8-31,* 8.5.2 *Figure 8-32,* 11.20.2
isochronous state machines, 8.5.5 *Figure 8-40,* 8.5.5 *Figure 8-41,* 11.21.2
token CRCs, 8.3.5.1
in token packets, 8.4.1
Transaction Translator response generation, 11.18.5
output driver jitter, 7.1.15.1
output levels, 7.3.2 *Table 7-7*
output receptacles, 6.2
output rise and fall times, 7.1.2.1
output transitions in state machines, 8.5
outside plating, 6.5.3.2, 6.5.4.2
over-current conditions
C_PORT_OVER-CURRENT bit, 11.24.2.7.2.4
getting port status, 11.24.2.7.1
over-current gangs, 11.11.1
over-current protection in self-powered hubs, 7.2.1.2.1
port indicators, 11.5.3
PORT_OVER-CURRENT bit, 11.24.2.7.1.4
port status change bits, 11.24.2.7.2
protection mode descriptors, 11.23.2.1
reporting and recovery, 11.12.5
signaling, 11.11.1
Over-current Indicator Change field, 11.24.2.6
Over-current Indicator field, 11.24.2.6
Over-current Reporting Mode field, 11.11.1
Over-current signal/event, 11.5 *Table 11-5*
over-sampling state machine DPLLs, 7.1.15.1
oxygen index, 6.5.3.1, 6.5.4.1

P

packet buffers, defined, 2.0 *glossary*
packet field formats, 8.3
address fields, 8.3.2 to 8.3.2.2
packet field formats *(continued)*
cyclic redundancy checks (CRC), 8.3.5 to 8.3.5.2
data field, 8.3.4, 8.4.4
frame number field, 8.3.3, 8.4.3
packet identifier field, 8.3.1 (*See also* PIDs)
packet formats
data packets, 4.4, 8.3 to 8.3.5.2, 8.4.4, 8.5.2, 8.5.5
handshake packets, 8.4.5 (*See also* handshakes)
handshake responses, 8.4.6 to 8.4.6.4
overview, 8.4
packet field formats, 8.3 to 8.3.5.2
SOF packets, 5.12.2, 5.12.4.1.1, 5.12.4.1.2, 8.4.3
token packets, 4.4, 8.3.5.1, 8.4.1, 8.5.2, 8.5.5
packet identifier field (PID). *See* PIDs
packet IDs. *See* PIDs
packet nullification, 11.3.2
packets. *See also* packet field formats; packet formats; packet size
bit stuffing, 7.1.9 to 7.1.9.2
blocking in Collision conditions, 11.8.3
bus protocol overview, 4.4
data packets defined, 4.4
data signaling overview, 7.1.7.4 to 7.1.7.4.2
defined, 2.0 *glossary*
error detection and recovery, 8.7 to 8.7.4
handshake packets defined, 4.4
high-speed packet repeaters, 11.7.1 to 11.7.1.4.4
hub connectivity and, 11.1.2.1
inter-packet delay, 7.1.18 to 7.1.18.2
one transaction per frame in isochronous transfers, 5.12.7
packet field formats, 8.3 to 8.3.5.2
packet formats, 8.4 to 8.4.6.4
packet nullification, 11.3.2
packet size (*See* packet size)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

packets. *(continued)*
packet voltage levels, 7.1.7.4.1
short packets, 5.3.2
splitting samples across packets, 5.12.8
SYNC field, 8.2
test mode packets, 7.1.20
token packets defined, 4.4
total latency, 11.7.1.3
transaction formats, 8.5 to 8.5.5
packet size
in buffering calculations, 5.12.8
bulk transfer constraints, 5.8.3
characteristics for transfers, 5.4
control transfer constraints, 5.5.3
determining missing packet size, 5.12.7
in device descriptors, 9.6.1
in device_qualifer descriptor, 9.6.2
in endpoint descriptors, 9.6.6
interrupt transfer constraints, 5.7.3
isochronous transfer constraints, 5.6.3
packet voltage levels, 7.1.7.4.1
parasitic loading, 7.1, 7.1.1.3, 7.1.6.2
partitioning of power, 7.2.1.1
paths, notations for, 11.15
pattern synchronization, 9.4.11
PBT (polybutylene terephthalate), 6.5.3.1, 6.5.4.1
PCB reference drawings, 6.9
PC industry, USB and, 3.3
PC-to-telephone interconnects, 1.1
pending start-splits, 11.18.6.2
Pending status, 11.15, 11.17.1
pending transactions, 11.18.6
performance criteria for electrical, mechanical and environmental compliance, 6.7
periodic buffer sections, 11.14.1
periodic transactions. *See also* interrupt transfers; isochronous transfers
after loss of synchronication, 11.22.2
buffer space required, 11.14.1, 11.19
periodic transactions. *(continued)*
handling requirements, 11.18.6 to 11.18.6.3
interrupt transaction sequencing, 11.20.3, 11.20.4
IN transaction sequencing, 11.20.4, 11.21.4
isochronous split transactions, 11.21 to 11.21.4
OUT transaction sequencing, 11.20.3, 11.21.3
overview, 11.18 to 11.18.8
periodic transaction pipelines, 11.14.2.1
scheduling and buffering, 11.14.2.1
split transaction flow sequences, 11.18.8
peripheral devices, 4.8.2.2. *See also* devices; functions
per-port current limiting, 11.12.5
per-port power switching, 11.11, 11.12.5
PET (polyethylene terephthalate), 6.5.3.1, 6.5.4.1
phase delay for SOF packets, 11.3.1
phase differences, clock synchronization and, 5.12.3
phase-locked clocks, 5.12.3
Phase Locked Loop, defined, 2.0 *glossary*
phases, defined, 2.0 *glossary*
Physical and Environmental Performance Properties of Insulation and Jacket for Telecommunication Wire and Cable, 6.7.1
physical bus topology, 5.2, 5.2.3, 6.1
physical devices
connectivity illustrated, 5.12.4.4
defined, 2.0 *glossary*
as implementation focus area, 5.1
logical components in bus topology, 5.2.2
physical interface, 4.2 to 4.2.2
physical shock standards, 6.7 *Table 6-7*
PID errors, defined, 8.3.1
PIDs
corrupted PIDs, 8.3.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

PIDs *(continued)*
in data packets, 8.4.4
data PIDs
DATA0/DATA1/DATA2 PIDs
in bulk transfers, 5.8.5, 8.5.2
comparing sequence bits, 8.6.2
in control transfers, 8.5.3
in data packets, 8.4.4
high-bandwidth transactions and, 5.9.1, 5.9.2
high-speed data PIDs, 8.3.1 *Table 8-1*
in interrupt transactions, 5.7.5, 8.5.4, 11.20.4
synchronization and, 8.6
Transaction Translator response generation, 11.18.5
error handling in high-speed transactions, 5.12.7
high bandwidth transactions and, 5.9.2
MDATA PIDs
in data packets, 8.4.4
defined, 8.3.1 *Table 8-1*
high-bandwidth endpoints and, 5.9.2
interrupt IN sequencing, 11.20.4
Transaction Translator response generation, 11.18.5
defined, 2.0 *glossary*
full- *vs.* low-speed port behavior, 11.8.4
in handshake packets, 8.4.5
handshake PIDs, 8.3.1 *Table 8-1*
ACK PIDs
in bulk transfers, 8.5.2, 11.17.1
in control transfers, 8.5.3, 8.5.3.1, 11.17.1
corrupted ACK handshake, 8.5.3.3, 8.6.4
in data toggle, 8.6, 8.6.1, 8.6.2
defined, 2.0 *glossary*
function response to OUT transactions, 8.4.6.3
host response to IN transactions, 8.4.6.2
overview, 8.3.1 *Table 8-1,* 8.4.5
PIDs *(continued)*
PING flow control and OUT transactions, 8.5.1, 8.5.1.1
Ready/ACK status, 11.15
in request processing, 9.2.6
NAK PIDs
in bulk transfers, 8.5.2, 11.17.1
busy endpoints, 5.3.2
in control transfers, 8.5.3.1, 11.17.1
data corrupted or not accepted, 8.6.3
defined, 2.0 *glossary*
function response to IN transactions, 8.4.6.1
function response to OUT transactions, 8.4.6.3
handshake PIDs
NAK PIDs
high bandwidth transactions and, 5.9.1
in interrupt transfers, 5.7.4, 8.5.4, 11.20.4
NAK limiting via Ping flow control, 8.5.1, 8.5.1.1
non-periodic transactions, 11.14.2.2, 11.17.1
overview, 8.3.1 *Table 8-1,* 8.4.5
Ready/NAK status, 11.15
test mode, 7.1.20
NYET PIDs
aborting, 11.18.6.1
bulk/control transactions, 8.5.2, 11.17.1
defined, 8.3.1 *Table 8-1,* 8.4.5
isochronous transactions, 11.21.1
Ping flow control and, 8.5.1
Transaction Translator response generation, 11.18.5
STALL PIDs
in bulk transfers, 5.8.5, 8.5.2, 11.17.1
in control transfers, 8.5.3.1, 11.17.1
data corrupted or not accepted, 8.6.3
functional and commanded stalls, 8.4.5
function response to IN transactions, 8.4.6.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

PIDs *(continued)*
function response to OUT transactions, 8.4.6.3
in interrupt transfers, 5.7.5, 8.5.4, 11.20.4
overview, 8.3.1 *Table 8-1,* 8.4.5
protocol stalls, 8.4.5
Ready/Stall status, 11.15
Request Error responses, 9.2.7
responses to standard device requests, 9.4
returned by control pipes, 8.5.3.4
incorrect PID errors, 11.15
overview, 8.3.1
PID check bits, 8.7.1
PID errors, 8.3.1
special PIDs
ERR PIDs, 8.3.1 *Table 8-1,* 8.4.5
PING PIDs, 8.3.1 *Table 8-1,* 8.3.2.2, 8.3.5.1, 8.4.1, 8.4.5, 8.5.2, 8.5.3.1
PRE PIDs
inter-packet delays and, 7.1.18.1
low-speed port behavior and, 11.8.4
low-speed transactions, 8.6.5
overview, 8.3.1 *Table 8-1*
Transmit state and, 11.5.1.7
Reserved PIDs, 8.3.1 *Table 8-1*
SPLIT PIDs, 8.3.1 *Table 8-1,* 8.3.2.1, 8.3.5.1, 8.4.2 to 8.4.2.3
in start-of-frame packets, 8.4.3
in token packets, 8.4.1
token PIDs, 8.3.1 *Table 8-1*
OUT PIDs
aborting OUT transactions, 11.18.6.1
ACK handshake and, 8.4.5
ADDR field, 8.3.2.1
in bulk transfers, 8.5.2, 8.5.2 *Figure 8-31,* 8.5.2 *Figure 8-32,* 11.17.1, 11.17.2
complete-split flow sequence, 11.20.1
in control transfers, 8.5.2 *Figure 8-31,* 8.5.2 *Figure 8-32,* 8.5.3, 8.5.3.1, 11.17.1, 11.17.2

PIDs *(continued)*
in data toggle, 8.6.1
ENDP field, 8.3.2.2
function response to OUT transactions, 8.4.6.3
high bandwidth transactions and, 5.9.2
high-speed PING flow control protocol, 5.8.4
in interrupt transfers, 8.5.2 *Figure 8-31,* 8.5.2 *Figure 8-32,* 8.5.4, 11.20.3
in isochronous transfers, 8.5.5, 8.5.5 *Figure 8-40,* 8.5.5 *Figure 8-41,* 11.21 to 11.21.4
NAK handshake and, 8.4.5
OUT/DATA transactions, 8.5.1, 8.5.1.1
overview, 8.3.1 *Table 8-1*
PING flow control and OUT transactions, 8.5.1, 8.5.1.1
prebuffering data, 5.12.5
scheduling OUT transactions, 11.18.4
split transaction conversion, 8.4.2.1, 8.4.2.2
split transaction examples, A.1, A.3, A.5
STALL handshake and, 8.4.5
start-split flow sequence, 11.20.1
state machines, 8.5.2, 8.5.5, 11.17.2, 11.20.2, 11.21.2
token CRCs, 8.3.5.1
in token packets, 8.4.1
Transaction Translator response generation, 11.18.5
IN PIDs
aborting IN transactions, 11.18.6.1
ACK handshake and, 8.4.5
ADDR field, 8.3.2.1
bit times and, 11.3.3
bulk transfers, 8.5.2, 8.5.2 *Figure 8-33,* 8.5.2 *Figure 8-34,* 11.17.1, 11.17.2
complete-split flow sequence, 11.20.1
control transfers, 8.5.2 *Figure 8-33,* 8.5.2 *Figure 8-34,* 8.5.3, 8.5.3.1, 11.17.1, 11.17.2
ENDP field, 8.3.2.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

PIDs *(continued)*
error handling on last data transaction, 8.5.3.3
token PIDs
IN PIDs
function response to, 8.4.6.1
high bandwidth transactions and, 5.9.2
host response to, 8.4.6.2
interrupt transactions, 8.5.4
interrupt transfers, 8.5.2 *Figure 8-33,* 8.5.2 *Figure 8-34,* 11.20.4
intervals between IN token and EOP, 11.3.3
isochronous transfers, 8.5.5, 8.5.5 *Figure 8-42,* 8.5.5 *Figure 8-43,* 11.21 to 11.21.4
low-speed transactions, 8.6.5
NAK handshake and, 8.4.5
overview, 8.3.1 *Table 8-1*
prebuffering data, 5.12.5
scheduling IN transactions, 11.18.4
split transaction conversion, 8.4.2.1
split transaction examples, A.2, A.4, A.6
STALL handshake and, 8.4.5
start-split flow sequence, 11.20.1
state machines, 8.5.2, 8.5.5, 11.17.2, 11.20.2, 11.21.2
token CRCs, 8.3.5.1
token packets, 8.4.1
Transaction Translator response generation, 11.18.5
SETUP PIDs
ACK handshake and, 8.4.5
ADDR field, 8.3.2.1
in control transfers, 8.5.3
in data toggle, 8.6.1
ENDP field, 8.3.2.2
function response to SETUP transactions, 8.4.6.4
overview, 8.3.1 *Table 8-1*
token CRCs, 8.3.5.1
PIDs *(continued)*
in token packets, 8.4.1
SOF PIDs (*See also* SOFs)
frame number field, 8.3.3
frames and microframes, 8.4.3.1
overview, 8.3.1 *Table 8-1*
start-of-frame packets, 8.4.3
PID to PID_invert bits check failure, 11.15
PING flow control protocol
high-speed signaling, 5.8.4
high-speed signaling and, 5.5.4
as limited to high-speed transactions, 8.5.1
NAK limiting and, 8.5.1, 8.5.1.1
NYET handshake and, 8.4.5
PING PIDs, 8.3.1 *Table 8-1,* 8.3.2.2, 8.3.5.1, 8.4.1, 8.4.5, 8.5.2, 8.5.3.1
pins
dual pin-type receptacles, 6.9
inrush current and, 7.2.4.1
single pin-type receptacles, 6.9
pipes
aborting or resetting, 10.5.2.2, 10.5.3.2.1
active, stalled, or idle status, 10.5.2.2
allocating bandwidth for, 4.7.5
characteristics and transfer types, 5.4
client pipes, 10.5.1.2.2
data transfer mechanisms, 10.1.3
Default Control Pipe, 4.4 (*See also* Default Control Pipe)
default pipes, 10.5.1.2.1
defined, 2.0 *glossary,* 4.4
in device characteristics, 4.8.1
identification, 10.3.4
overview, 5.3.2, 10.5.1, 10.5.3
pipe state control, 10.5.2.2
pipe usage, 10.5.1.2
Policies, 10.3.1, 10.3.3, 10.5.3.2.2
queuing IRPs, 10.5.3.2.3
role in data transfers, 4.7
service and polling intervals, 10.3.3

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

pipes *(continued)*
stream and message pipes, 4.4, 5.3.2, 5.3.2.1, 5.3.2.2
supported pipe types, 10.5.3.1 to 10.5.3.1.4
USBD pipe mechanism responsibilities, 10.5.1 to 10.5.3.2.4
USB robustness and, 4.5
pipe state control, 10.5.2.2
pipe status, 10.5.2.2
PK signal/event, 11.5 *Table 11-5*
plating
plug contact materials, 6.5.4.3
plug shell materials, 6.5.4.2
receptacle contact materials, 6.5.3.3
receptacle shell materials, 6.5.3.2
PLL, defined, 2.0 *glossary*
plugs
DC resistance, 6.6.3
interface and mating drawings, 6.5.3
keyed connector protocol, 6.2
materials, 6.5.4.1, 6.5.4.2, 6.5.4.3
orientation, 6.5.1
Series "A" and Series "B" plugs, 6.5.4
standards for, 6.7
termination data, 6.5.2
USB Icon, 6.5, 6.5.1
Policies
defined, 10.3.1
setting, 10.3.3
USBDI mechanisms, 10.5.3.2.2
polling
defined, 2.0 *glossary*
endpoints, 9.6.6
setting intervals for pipes, 10.3.3
polybutylene terephthalate (PBT), 6.5.3.1, 6.5.4.1
polyethylene terephthalate (PET), 6.5.3.1, 6.5.4.1
POR signal/event
Bus_Reset state and, 11.6.3.9
defined, 2.0 *glossary*
POR signal/event *(continued)*
in receiver state machine, 11.6.3 *Table 11-8*
Port Change field, 11.4.4, 11.24.2.7.2
PORT_CONNECTION
defined, 11.24.2.7.1.1
hub class feature selectors, 11.24.2
Port Status field, 11.24.2.7.1
PORT_ENABLE
clearing, 11.24.2.2
defined, 11.24.2.7.1.2
hub class feature selectors, 11.24.2
Port Error condition, 11.24.2.7.2.2
PORT_HIGH_SPEED and, 11.24.2.7.1.8
Port Status field, 11.24.2.7.1
Port Error condition, 11.8.1, 11.24.2.7.2.2
Port_Error signal/event, 11.5 *Table 11-5*
Port field, 8.4.2.2
PORT_HIGH_SPEED
Port Status field, 11.24.2.7.1
speed detection and, 11.8.2, 11.24.2.7.1.8
PORT_INDICATOR
clearing port features, 11.24.2.2
getting indicator status, 11.24.2.7.1.10, 11.24.2.13
hub class feature selectors, 11.24.2
indicator selectors, 11.24.2.12
Port Status field, 11.24.2.7.1
port indicators
descriptors, 11.23.2.1
lights on devices, 11.5.3 to 11.5.3.1
port status indicators, 11.24.2.7.1, 11.24.2.7.1.10
selectors, 11.24.2.12
PORT_LOW_SPEED
defined, 11.24.2.7.1.7
hub class feature selectors, 11.24.2
Port Status field, 11.24.2.7.1
speed detection and, 11.8.2, 11.24.2.7.1.8

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

PORT_OVER_CURRENT
defined, 11.24.2.7.1.4
hub class feature selectors, 11.24.2
over-current conditions and, 11.11.1, 11.12.5
port indicators, 11.24.2.7.1.10
Port Status field, 11.24.2.7.1
PORT_POWER
clearing, 11.24.2.2
defined, 11.24.2.7.1.6
hub class feature selectors, 11.24.2
Port Status field, 11.24.2.7.1
SetPortFeature() request, 11.24.2.13
shared power switching and, 11.11.1
PortPwrCtrlMask field
hub descriptors for, 11.23.2.1
multiple gangs and, 11.11.1
power switching settings, 11.11
PORT_RESET
defined, 11.24.2.7.1.5
hub class feature selectors, 11.24.2
Port Status field, 11.24.2.7.1
SetPortFeature() request, 11.24.2.13
ports. *See also* hubs
in bus topology, 5.2.3
clearing features, 11.24.2.2
data source signaling, 7.1.13 to 7.1.13.2.2
defined, 4.8.2.1
disconnect timer, 11.5.2
downstream facing ports, 4.8.2.1, 11.5 to 11.5.1.15
full- *vs.* low-speed port behavior, 11.8.4
in hub architecture, 11.1.1
hub configuration and, 11.13
hub descriptors, 11.23 to 11.23.2.1
hub internal ports, 11.4 to 11.4.4
indicators, 11.5.3 to 11.5.3.1, 11.24.2.7.1, 11.24.2.7.1.10
indicator selectors, 11.24.2.12
input capacitance, 7.1.6.1
ports. *(continued)*
over-current reporting and recovery, 11.12.5
port expansion considerations in USB development, 1.1
port selector state machine, 11.7.1.4 to 11.7.1.4.4
power control, 11.11
resetting, 10.2.8.1
root ports, 2.0 *glossary*
setting port features, 11.24.2.13
signal speed support, 7
status
bus state evaluation, 11.8 to 11.8.4.1
detecting status changes, 7.1.7.5, 11.12.2, 11.12.3
hub and port status change bitmap, 11.12.4
port status bits, 11.24.2.7.1 to 11.24.2.7.2.5
testing mode, 11.24.2.7.1.9, 11.24.2.13
upstream facing ports, 4.8.2.1, 11.6 to 11.6.4.6
port selector state machine, 11.7.1.4 to 11.7.1.4.4
port status change bits, 11.24.2.7.1 to 11.24.2.7.2.5
Port Status field, 11.24.2.7.1
PORT_SUSPEND
clearing, 11.24.2.2
defined, 11.24.2.7.1.3
hub class feature selectors, 11.24.2
Port Status field, 11.24.2.7.1
SetPortFeature() request, 11.24.2.13
PORT_TEST
hub class feature selectors, 11.24.2
overview, 11.24.2.7.1.9
Port Status field, 11.24.2.7.1
specific test modes, 11.24.2.13
power budgeting, 7.2.1.4, 9.2.5.1
power conductors in cables, 6.3
power control circuits in bus-powered hubs, 7.2.1.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

power distribution and management. *See also* over-current conditions; power switching
classes of devices, 7.2.1 to 7.2.1.5
bus-powered devices or hubs, 4.3.1, 7.2.1.1
high-power bus-powered functions, 7.2.1.4
low-power bus-powered functions, 7.2.1.3
self-powered devices or hubs, 4.3.1, 7.2.1.2, 7.2.1.5
configuration characteristics
hub descriptors for power-on sequence, 11.23.2.1
information in device characteristics, 4.8.1
power consumption in configuration descriptors, 9.6.3
power source capability in configuration, 9.1.1.2
dynamic attach and detach, 7.2.3, 7.2.4 to 7.2.4.2
Host Controller role in, 4.9
host role in, 10.1.5
hub support for, 11.1
loss of power, 7.2.1.2
over-current conditions, 7.2.1.2.1, 11.12.5 (*See also* over-current conditions)
overview, 4.3.1, 4.3.2, 9.2.5
power budgeting, 7.2.1.4, 9.2.5.1
power status
control during suspend/resume, 7.2.3
device states, 9.1.1.2
port power states, 11.24.2.7.1.6, 11.24.2.13
power switching, 11.11, 11.11.1
remote wakeup, 7.2.3, 9.2.5.2
USB System role, 10.5.4.2
USB System Software role, 4.9
voltage drop budget, 7.2.2
Powered device state, 9.1.1.2, 9.1.1 *Table 9-1*
Powered Off state, 11.5, 11.5.1.2
powered-on ports, 11.24.2.13
Power On Reset. *See* POR signal/event
power-on sequence, 11.23.2.1
power pair construction, 6.6.2
power pins, 7.2.4.1
Power_source_off signal/event, 11.5 *Table 11-5*
power switching
bus-powered hubs, 7.2.1.1
getting port status, 11.24.2.7.1
hub descriptors for, 11.23.2.1
hub port power control, 11.11
power-on and connection events timing, 7.1.7.3
power switching gangs, 11.11.1
staged power switching, 7.2.1.4
preamble packet. *See* PRE PID
preambles, 8.6.5
preboot control, passing to operating systems, 10.5.5
prebuffering data, 5.12.5
prepared termination, 6.4.2, 6.4.3
PRE PID, 8.3.1 *Table 8-1*
inter-packet delays and, 7.1.18.1
low-speed port behavior and, 11.8.4
low-speed transactions, 8.6.5
overview, 8.3.1 *Table 8-1*
Transmit state and, 11.5.1.7
priming elasticity buffer, 11.7.1.3
Priming state, 11.7.1.4.2
product descriptions in device descriptors, 9.6.1
Product IDs in device descriptors, 9.6.1
programmable data rate, defined, 2.0 *glossary*
prohibited cable assemblies, 6.4.4
Promoters, USB Implementers Forum, 1.4, 2.0 *glossary*
propagation delay
cable delay, 7.1.16
detachable cables, 6.4.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

propagation delay *(continued)*
end-to-end signal delay, 7.1.19 to 7.1.19.2
EOF point advancement and, 11.2.3.2
full- and low-speed signals, 7.1.14.1
full-speed source electrical characteristics, 7.3.2 *Table 7-9*
high-/full-speed cables, 6.4.2
high-speed signaling, 7.1.1.3, 7.1.14.2
low-speed cables, 6.4.3, 7.1.1.2
low-speed source electrical characteristics, 7.3.2 *Table 7-10*
tests, 6.7 *Table 6-7*
propagation delay skew, 6.7 *Table 6-7*
protected fields in packets, 8.3.5
protocol codes
defined, 9.2.3
in device descriptors, 9.6.1
in device qualifer descriptors, 9.6.2
in interface descriptors, 9.6.5
protocol engine requirements of Host Controller, 10.2.5
protocol errors, detecting, 10.2.6
Protocol field, 9.2.3
Protocol layer, 8
bit ordering, 8.1
bus protocol, 4.4
data toggle synchronization and retry, 8.6 to 8.6.5
error detection and recovery, 8.7 to 8.7.4
packet field formats, 8.3 to 8.3.5.2
packet formats, 8.4 to 8.4.6.4
SYNC field, 8.2
transaction formats, 8.5 to 8.5.5
protocols
defined, 2.0 *glossary*
protocol codes, 9.2.3, 9.6.1, 9.6.2, 9.6.5
protocol stall, 8.4.5, 8.5.3.4
PS signal/event, 11.5 *Table 11-5*
pull-up and pull-down resistors
buffer impedance measurement, 7.1.1.1
high-speed signaling and, 7.1
power control during suspend/resume, 7.1.7.6, 7.2.3
signaling speeds and, 7.1 *Table 7-1*
signal termination, 7.1.5.1

Q

quantization in high-speed microframe timer range, 11.2.1
queuing IRPs, 10.5.3.2.3

R

RA (rate adaptation)
asynchronous RA, 2.0 *glossary*
audio connectivity and, 5.12.4.4.1
defined, 2.0 *glossary*
in source-to-sink connectivity, 5.12.4.4
synchronous data connectivity, 5.12.4.4.2
synchronous RA, 2.0 *glossary*
random vibration standards, 6.7 *Table 6-7*
rate adaptation. *See* RA (rate adaptation)
rate matchers
asynchronous endpoints, 5.12.4.1.1
buffering for rate matching, 5.12.8
client software role, 5.12.4.4
in non-USB isochronous application, 5.12.1
synchronous endpoints, 5.12.4.1.2
ratings, full-speed, 6.4.1, 6.4.2
read/write sequences
in bulk transfers, 8.5.2
in control transfers, 8.5.3, 8.5.3.1
Ready/ACK status, 11.15
Ready/Data status, 11.15
Ready/lastdata status, 11.15
Ready/moredata status, 11.15
Ready/NAK status, 11.15
Ready/Stall status, 11.15
Ready status, 11.15
Ready/trans-err status, 11.15
real-time clock, 5.12.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

real-time data transfers. *See* isochronous transfers

receive clock, 11.7.1.2, 11.7.1.3

receiver eye pattern templates. *See* eye pattern templates

receivers

- differential data receivers, 7.1 *Table 7-1*
- receive phase of signaling, 7.1.1
- receiver characteristics, 7.1.4 to 7.1.4.2
- receiver jitter, 7.1.15 to 7.1.15.2, 7.3.2 *Table 7-9,* 7.3.2 *Table 7-10,* 7.3.3 *Figure 7-51*
- receiver sequence bits, 8.6, 8.6.2
- receiver state machine, 11.6, 11.6.3
- sensitivity requirements, 7.1.2.2 *Figure 7-15,* 7.1.2.2 *Figure 7-16,* 7.1.2.2 *Figure 7-18*
- single- ended receivers, 7.1 *Table 7-1*
- squelch detection, 7.1, 7.1.4.2

ReceivingHJ state, 11.6.3.2

ReceivingHK state, 11.6.3.5

ReceivingIS state, 11.6.3.1

ReceivingJ state, 11.6.3, 11.6.3.3

ReceivingK state, 11.6.3, 11.6.3.6

ReceivingSE0 state, 11.6.3, 11.6.3.8

receptacles

- interface and mating drawings, 6.5.3
- keyed connector protocol, 6.2
- materials, 6.5.3.1, 6.5.3.2, 6.5.3.3
- PCB reference drawings, 6.9
- Series "A" and Series "B" receptacles, 6.5.3
- standards for, 6.7
- termination data, 6.5.2
- USB Icon, 6.5, 6.5.1

Recipient bits, 9.4.5

reclocking, defined, 11.7.1

recovering from errors. *See* error detection and handling

recovery intervals for devices, 9.2.6.2

re-enumerating sub-trees, 10.5.4.5

reflected endpoint status, 10.5.2.2

registers in hub timing, 11.2.3.1

regulators in bus-powered hubs, 7.2.1.1

regulatory compliance, 7.0

regulatory requirements for USB devices, 7.3.1

reliable delivery in isochronous transfers, 5.12

remote wakeup

- in configuration descriptors, 9.6.3
- Host Controller role, 10.2.7
- inrush current and, 7.2.3
- overview, 9.2.5.2
- resume signaling, 7.1.7.7, 9.1.1.6
- timing relationships, 11.9
- USB System role in, 10.5.4.5

Remote Wakeup field, 9.4.5

removable devices, 11.23.2.1

removing devices. *See* dynamic insertion and removal

RepeatingSE0 state, 11.6.4, 11.6.4.3

replacing configuration information, 10.5.4.1.3

reporting rates for feedback, 5.12.4.2

request codes, 9.4 *Table 9-4,* 11.24.2

Request Errors, 9.2.7

requests. *See also* PIDs; *names of specific requests*

- bRequest field, 9.3.2
- class-specific requests, 9.2.6.5, 10.5.2.8, 11.24 to 11.24.2.13
- completion times for hub requests, 11.24.1
- control transfers and, 5.5
- defined, 2.0 *glossary*
- in device class definitions, 9.7.3
- hub standard and class-specific requests, 11.24 to 11.24.2.13
- information requirements for, 10.3.4
- overview, 9.2.6
- port status reporting, 11.12.3
- request processing timing, 9.2.6.1
- reset/resume recovery time, 9.2.6.2
- set address processing, 9.2.6.3

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

requests. *(continued)*
standard device requests, 9.2.6.4, 9.4 to 9.4.11
standard feature selectors, 9.4 *Table 9-6*
standard hub requests, 11.24 to 11.24.2.13
standard request codes, 9.4 *Table 9-4*
USBD command mechanisms, 10.5.2 to 10.5.2.12
USB device requests, 9.3 to 9.3.5
vendor-specific requests, 10.5.2.9
required data sequences for transfers, 5.4
Reserved PID, 8.3.1 *Table 8-1*
reserved portions of frames, 5.5.4
Reserved test mode, 9.4.9
Reset bus state
downstream ports, 11.5, 11.5.1.5
high- and full-speed operations, 5.3.1.1
high-speed detection and, 7.1.5.2
high-speed signaling and, 7.1.7.6
in power-on and connection events, 7.1.7.3
reset signaling, 7.1.7.5
signaling levels and, 7.1.7.1
reset condition
in bus enumeration process, 9.1.2
C_PORT_RESET bit, 11.24.2.7.2.5
Default device state and, 9.1.1.3
device characteristics, 9.2.1
getting port status, 11.24.2.7.1
hub reset behavior, 11.10
PORT_RESET bit, 11.24.2.7.1.5
port status change bits, 11.24.2.7.2
remote wakeup and, 10.5.4.5
reset handshake, C.2.4
reset recovery time, 7.1.7.5, 9.2.6.2
reset signaling, 7.1.7.5
resetting pipes, 10.5.2.2
SetPortFeature(PORT_RESET) request, 11.24.2.13
state machine diagams, C.0
reset condition *(continued)*
USB System and, 10.2.8.1
reset devices, communicating with, 10.5.1.1
reset handshake, C.2.4
Resetting state, 11.5.1.5
ResetTT() request, RESET_TT, 11.24.2, 11.24.2.9
resistance ratings, 6.6.3
resistors
high-speed signaling and, 7.1
pull-up and pull-down resistors, 7.1.1.1, 7.1.5.1, 7.1.7.6, 7.2.3
series damping resistors, 7.1.1.1
speed detection and, 9.1.1.3
resonators, data-rate tolerance and, 7.1.11
resource management, USB System role in, 10.3.2
Restart_E state, 11.5, 11.5.1.13
Restart_S state, 11.5, 11.5.1.12
Resume bus state
downstream ports, 11.5, 11.5.1.10
overview, 7.1.7.7
receivers, 11.6.3, 11.6.3.7
reset signaling and, 7.1.7.5
signaling levels and, 7.1.7.1
Resume_Event signal/event, 11.4
resume intervals for devices, 9.2.6.2
resume signaling
after loss of synchronication, 11.22.2
hub support, 11.1.2.2, 11.9
power control during suspend/resume, 7.2.3
remote wakeup and, 10.5.4.5
resume conditions in Hub Controller, 11.4.4
single-ended transmissions, 11.6.1
retire, defined, 2.0 *glossary*
retiring IRPs. *See* aborting/retiring transfers
RFI, USB grounding and, 6.8
rise and fall times
data source jitter, 7.1.13.1.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

rise and fall times *(continued)*
full-speed connections, 7.1.1.1
full-speed source electrical characteristics, 7.3.2 *Table 7-9*
high-speed signaling, 7.1.2.2
high-speed source electrical characteristics, 7.3.2 *Table 7-8*
low-speed source electrical characteristics, 7.3.2 *Table 7-10*
overview, 7.1.2.1 to 7.1.2.2
SE0 from low-speed devices, 7.1.14.1
testing, 7.1.20
robustness of USB, 3.3, 4.5 to 4.5.2
root hub
in bus topology, 5.2.3
defined, 2.0 *glossary*
HCDI presentation of, 10.4
Host Controller and, 10.2.8, 10.2.8.1
state handling, 10.2.1
root port hub, defined, 7.2.1
root ports, 2.0 *glossary,* 11.9
round trip times, 7.1.6.2
Rptr_Enter_WFEOPFU signal/event, 11.5 *Table 11-5*
Rptr_Exit_WFEOPFU signal/event, 11.5 *Table 11-5*
Rptr_WFEOP signal/event, 11.6.4 *Table 11-9*
Run timer status, C.0
Rx_Bus_Reset signal/event, 11.6.4 *Table 11-9,* 11.7.1.4 *Table 11-10,* 11.7.2.3 *Table 11-11*
Rx_Resume signal/event, 11.5 *Table 11-5,* 11.7.2.3 *Table 11-11*
Rx_Suspend signal/event, 11.4, 11.5 *Table 11-5,* 11.6.4 *Table 11-9,* 11.7.2.3 *Table 11-11*

S

S field (Start), 8.4.2.2
sample clock
buffering for rate matching, 5.12.8
defined, 5.12.2
synchronous endpoints, 5.12.4.1.2
sampled analog devices, 5.12.4
sample declarations in state machines, B.1, B.2, B.3
Sample Rate Conversion. *See* SRC
samples
defined, 2.0 *glossary*
sample size in buffering calculations, 5.12.8
samples per (micro)frame in isochronous transfers, 5.12.4.2
SC field (Start/Complete field), 8.4.2.2, 8.4.2.3
scheduling
access to USB interconnect, 4.1
host split transaction scheduling , 11.18.4
microframe pipeline scheduling, 11.18.2
periodic split transaction scheduling, 11.18
transaction schedule in bus protocol overview, 4.4
Transaction Translator transaction scheduling, 11.14.2 to 11.14.2.3
SE0sent signal/event, 11.6.4 *Table 11-9*
SE0 signal/event
in data signaling, 7.1.7.4.1
downstream port state machine, 11.5 *Table 11-5*
Not Configured state, 11.5.1.1
propagation delays, 7.1.14.1
pull-down resistors and, 7.1.7.3
receiver state machine, 11.6.3 *Table 11-8*
reset signaling, 7.1.7.5
SE0 interval of EOP, 7.3.2 *Table 7-9,* 7.3.2 *Table 7-10*
signaling levels and, 7.1.7.1
single-ended transmissions, 11.6.1
test mode, 7.1.20
SE0 width, 7.1.13.2.1, 7.1.14.1, 7.3.2 *Table 7-9,* 7.3.2 *Table 7-10*
SE1 signal/event, 7.1.1, 11.6.1
selective resume signaling, 11.9

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

selective suspend signaling
defined, 9.1.1.6
hub support, 11.9
overview, 7.1.7.6.2
self-powered devices and functions
configuration descriptors, 9.6.3
defined, 4.3.1, 7.2.1
device states, 9.1.1.2
overview, 7.2.1.5
Self Powered field, 9.4.5
self-powered hubs
configuration, 11.13
defined, 7.2.1
device states, 9.1.1.2
over-current protection, 7.2.1.2.1
overview, 7.2.1.2
power switching, 11.11
self-recovery, USB robustness and, 4.5
SendEOR state, 11.5, 11.5.1.11
SendJ state, 11.6.4, 11.6.4.4
Send Resume state (Sresume), 11.6.4, 11.6.4.6
sequence of transactions in frames, 5.11.2
Serial Interface Engine (SIE), 10.1.1, 10.2.2
serializer/deserializer, 10.2.2
serial numbers in device descriptors, 9.6.1
Series "A" and "B" connectors
detachable cables and, 6.4.1
keyed connector protocol, 6.2
plugs
injection molded thermoplastic insulator material, 6.5.4.1
interface drawings, 6.5.4
plug (male) contact materials, 6.5.4.3
plug shell materials, 6.5.4.2
receptacles
injection molded thermoplastic insulator material, 6.5.3.1
interface and mating drawings, 6.5.3
Series "A" and "B" connectors *(continued)*
receptacles *(continued)*
PCB reference drawings, 6.9
receptacle contact materials, 6.5.3.3
receptacle shell materials, 6.5.3.2
standards for, 6.7
USB Icon, 6.5, 6.5.1
series damping resistors, 7.1.1.1
service, defined, 2.0 *glossary*
service clock, 5.12.2, 5.12.8
service intervals, 2.0 *glossary*, 10.3.3
service jitter, defined, 2.0 *glossary*
service periods of data, 5.12.1
service rates, defined, 2.0 *glossary*
SetAddress() request, SET_ADDRESS
hub requests, 11.24.1
overview, 9.4.6
reset recovery time and, 7.1.7.5
standard device request codes, 9.4
time limits for completing processing, 9.2.6.3
SetConfiguration() request, SET_CONFIGURATION
hub requests, 11.24.1
overview, 9.4.7
Powered-off state and, 11.5.1.2
setting configuration in descriptors, 9.1.1.5, 9.6.3
standard device request codes, 9.4
SetDescriptor() request, SET_DESCRIPTOR
getting endpoint descriptors, 9.6.6
hub class requests, 11.24.2
hub requests, 11.24.1
interface descriptors and, 9.6.5
overview, 9.4.8
SetHubDescriptor() request, 11.24.2.10
standard device request codes, 9.4
SetDeviceFeature(DEVICE_REMOTE_WAKEUP) request, 10.5.4.5

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

SetFeature() request, SET_FEATURE, 9.4.5, 9.4.9
hub class requests, 11.24.2
hub requests, 11.24.1
overview, 9.4.9
SetHubFeature() request, 11.24.2.12
SetPortFeature() request, 11.24.2.13
standard device request codes, 9.4
TEST_MODE, 7.1.20, 9.4.9
TEST_SELECTOR, 9.4.9
SetHubDescriptor() request, 11.24.2, 11.24.2.10
SetHubFeature() request, 11.24.2, 11.24.2.6, 11.24.2.12
SetInterface() request, SET_INTERFACE, 9.2.3, 9.4, 9.4.10, 9.6.5, 11.24.1
SetPortFeature() request
hub class requests, 11.24.2, 11.24.2.13
PORT_CONNECTION, 11.24.2.7.1.1
PORT_ENABLE, 11.24.2.7.1.2
PORT_HIGH_SPEED, 11.24.2.7.1.8
PORT_INDICATOR, 11.24.2.7.1.10, 11.24.2.12
PORT_LOW_SPEED, 11.24.2.7.1.7
PORT_OVER_CURRENT, 11.24.2.7.1.4
PORT_POWER
Disconnected state and, 11.5.1.3
port power settings, 11.11
port power states, 11.24.2.7.1.6, 11.24.2.13
requirements, 11.24.2.13
PORT_RESET
completion, 9.2.6
C_PORT_ENABLE bit, 11.24.2.7.2.2
evaluating device speed during, 11.8.2
initiating port reset, 11.24.2.7.1.5, 11.24.2.13
in port enabling, 11.24.2.7.1.2
requirements, 11.24.2.13
Resetting state and, 11.5.1.5
SetPortFeature() request *(continued)*
PORT_SUSPEND, 10.5.4.5
selective suspend, 7.1.7.6.2
suspending ports, 11.5.1.9, 11.24.2.7.1.3
PORT_TEST, 11.24.2.7.1.9, 11.24.2.13
power-off conditions and, 11.13
TEST_MODE, 7.1.20
SetTest signal/event, 11.5 *Table 11-5*
SETUP PID, 8.3.1 *Table 8-1*
ACK handshake and, 8.4.5
ADDR field, 8.3.2.1
in control transfers, 8.5.3
in data toggle, 8.6.1
ENDP field, 8.3.2.2
function response to, 8.4.6.4
overview, 8.3.1 *Table 8-1*
split transaction examples, A.1
token CRCs, 8.3.5.1
in token packets, 8.4.1
Setup stage
in control transfer data sequences, 5.5.5
in control transfers, 5.5, 8.5.3
data format for USB device requests, 9.3
SETUP transactions. *See* SETUP PID
shell
conductors, 6.5.2
plug shell materials, 6.5.4.2
receptacle shell materials, 6.5.3.2
shielding
grounding, 6.8
low-speed and high-/full-speed cables, 6.6
outer and inner cable shielding, 6.6.1
shielded cables illustrated, 6.4.1
standardized contact terminating assignments, 6.5.2
short circuits, USB withstanding capabilities, 7.1.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

short packets
defined, 9.4.3
detecting, 10.2.6
multiple data payloads and, 5.3.2
SIE (Serial Interface Engine), 10.1.1, 10.2.2
signal conductors in cables, 6.3
signal edges. *See* edges of signals
signaling
bit stuffing, 7.1.9 to 7.1.9.2
cable attenuation, 7.1.17
connect and disconnect signaling, 7.1.7.3
data encoding/decoding, 7.1.8
data rate, 7.1.11
data signaling, 7.1.7.4 to 7.1.7.4.2
delay
bus turn-around time and inter-packet delay, 7.1.18 to 7.1.18.2
cable delay, 7.1.16
cable skew delay, 7.1.3
maximum end-to-end signal delay, 7.1.19 to 7.1.19.2
high-speed driver characteristics, 7.1.1.3
hub signaling timings, 7.1.14 to 7.1.14.2
in-band and out-of-band, 10.1.2
input characteristics, 7.1.6.1
jitter, 7.1.13.1 to 7.1.13.2, 7.1.15 to 7.1.15.2 (*See also* jitter)
low-speed (1.5Mb/S) driver characteristics, 7.1.1.2
(micro)frame intervals and repeatability, 7.1.12
overview, 7.1
receiver characteristics, 7.1.4 to 7.1.4.2, 7.1.15.1
reset signaling, 7.1.7.5
resume signaling, 7.1.7.7
rise and fall time, 7.1.2.1 to 7.1.2.2
signal attenuation, 7.1.17
signal edges (*See* edges of signals)
signaling levels, 7.1.7 to 7.1.7.5
signalling *(continued)*
signal integrity, 4.5
signal termination, 7.1.5.1
source signaling, 7.1.13 to 7.1.13.2.2
suspend signaling, 7.1.7.6
sync pattern, 7.1.10
USB driver characteristics, 7.1.1
signal matching, 7.1.2.1
signal pair attenuation, 6.4.1, 6.7 *Table 6-7*
signal pair construction, 6.6.2
signal pins, 7.2.4.1
signal swing, 7.1.2.1
signal termination, 7.1, 7.1.5.1
simple states, notation for, 11.15
Single-ended 0 bus state (SEO)
in data signaling, 7.1.7.4.1
pull-down resistors and, 7.1.7.3
reset signaling, 7.1.7.5
signaling levels and, 7.1.7.1
test mode, 7.1.20
single-ended capacitance, 7.1.1.2
single-ended components in upstream ports, 11.6.1
single-ended receivers, 7.1.4.1, 7.1.6.1, 7.1 *Table 7-1*
"single packets" and split transactions, 5.12.3
single pin-type receptacles, 6.9
sink endpoints
adaptive sink endpoints, 5.12.4.1.3
audio connectivity, 5.12.4.4.1
connectivity overview, 5.12.4.4
feedback for isochronous transfers, 5.12.4.2
synchronization types, 5.12.4.1
synchronous data connectivity, 5.12.4.4.2
skew
cable skew delay, 6.7 *Table 6-7,* 7.1.3, 7.3.2 *Table 7-12*
differential-to-EOP transition skew, 7.3.3 *Figure 7-50*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

skew *(continued)*
hub EOP delay and EOP skew, 7.3.3 *Figure 7-53*
hub/repeater electrical characteristics, 7.3.2 *Table 7-11*
hub switching skew, 7.1.9.1
Idle-to-K state transition, 7.1.14.1
minimizing signal skew, 7.1.1
timing skew accumulation, 11.2.5.1 to 11.2.5.2
slips in synchronous data, 5.12.4.4.2
small capacitors, 7.1.6.1
SOF PID, 8.3.1 *Table 8-1*
frame number field, 8.3.3
frames and microframes, 8.4.3.1
start-of-frame packets, 8.4.3
SOFs
after loss of synchronication, 11.22.2
bus clock and, 5.12.2
defined, 2.0 *glossary*
in downstream port state machine, 11.5
error recovery in isochronous transfers, 5.12.7
frame and microframe intervals, 7.1.12
frame and microframe timer synchronization, 11.2, 11.2.3 to 11.2.3.3
frame clock tracking and microframe SOFs, 5.12.4.1.2
high-speed SOF in connect detection, 7.1.7.3
Host Controller frame and microframe generation, 10.2.3
loss of consecutive SOFs, 11.2.5
loss of TT synchronization, 11.22.1
microframe synchronization and, 11.2.4
overview, 8.4.3
tracking, 5.12.6, 5.12.7
using as clocks, 5.12.5
soft-start circuits, 7.2.4.1
software interfaces. *See* client software; HCDI; host; USBDI; USB System software
SOHP, 11.7.2.1
solderability standards, 6.7 *Table 6-7*
solder tails, 6.5.3.3, 6.5.4.3
SOP bus state, 7.1.7.1, 7.1.7.2, 7.1.7.4.1, 7.1.7.4.2
SOP_FD signal/event
generating, 11.4.4
in Hub Repeater state machine, 11.7.2.3 *Table 11-11*
SOP_FU signal/event, 11.7.2.3 *Table 11-11*
SOPs, 8.3
error detection through bus turn-around timing, 8.7.2
frame and microframe timer synchronization, 11.2.3 to 11.2.3.3
Idle-to-K state transition, 7.1.14.1
SOP distortion, 7.3.3 *Figure 7-52*
timeout periods and, 7.1.19.1
SORP signal/event, 11.7.1.4 *Table 11-10*
source endpoints
adaptive source endpoints, 5.12.4.1.3
audio connectivity, 5.12.4.4.1
connectivity overview, 5.12.4.4
feedback for isochronous transfers, 5.12.4.2
synchronization types, 5.12.4.1
synchronous data connectivity, 5.12.4.4.2
source jitter, 7.3.2 *Table 7-8,* 7.3.2 *Table 7-9,* 7.3.2 *Table 7-10*
source/sink connectivity, 5.12.4.4
special PIDs
ERR PID, 8.3.1 *Table 8-1,* 8.4.5
PING PID, 8.3.1 *Table 8-1,* 8.3.2.2, 8.3.5.1, 8.4.1, 8.4.5, 8.5.2, 8.5.3.1
PRE PID
defined, 8.3.1 *Table 8-1*
inter-packet delays and, 7.1.18.1
low-speed port behavior and, 11.8.4
low-speed transactions, 8.6.5
Transmit state and, 11.5.1.7
Reserved PID, 8.3.1 *Table 8-1*
SPLIT PID, 8.3.1 *Table 8-1,* 8.3.2.1, 8.3.5.1, 8.4.2 to 8.4.2.3

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

specific-sized data payloads, 5.3.2
speed
downstream facing ports and hubs, 7.1
high-speed devices operating at full-speed, 5.3.1.1
hubs and signaling speeds, 11.1.1
measurement planes in speed signaling eye patterns, 7.1.2.2
pull-up and pull-down resistors and, 7.1 *Table 7-1*
speed dependent descriptors, 9.2.6.6
upstream facing ports and hubs, 7.1
speed detection
attached devices, 11.8.2
detecting low-speed functions and hubs, 11.24.2.7.1.7
detecting speed of devices, 7.1.7.3
other_speed_configuration descriptor, 9.6.4
PORT_HIGH_SPEED, 11.24.2.7.1.8
reset condition and Default device state, 9.1.1.3
speed indication bits, 7.1.5.2
termination and, 7.1.5.1
SPI, defined, 2.0 *glossary*
SPLIT PID, 8.3.1 *Table 8-1*, 8.3.2.1, 8.3.5.1, 8.4.2 to 8.4.2.3
splitting sample across packets, 5.12.8
split transactions. *See also* complete-split transactions; start-split transactions
best case full-speed budgets, 11.18.1, 11.18.4
bulk/control transactions
control transfers, 5.5.4
IN examples, A.2
OUT and SETUP examples, A.1
sequencing, 11.17.3
state machines, 11.17.2
data handling, 11.14.1.1
data packet types, 8.4.4
defined, 2.0 *glossary,* 5.10
failures, 11.17.5
host controller and, 11.14.1.2
split transactions *(continued)*
host scheduling, 11.18.4
IN transactions
bulk/control examples, A.2
interrupt examples, A.4
interrupt transaction sequencing, 11.20.4
isochronous examples, A.6
isochronous transaction sequencing, 11.21.4
interrupt transactions
IN examples, A.4
flow sequences and state machines, 11.20 to 11.20.4
OUT examples, A.3
isochronous transactions
IN examples, A.6
OUT examples, A.5
overview, 11.21 to 11.21.4
microframe pipeline, 11.18.2
non-periodic transactions
IN examples, A.2
OUT and SETUP examples, A.1
overview, 11.17 to 11.17.5
sequencing, 11.17.3
state machines, 11.17.2
notation for, 11.15
OUT transactions
bulk/control examples, A.1
interrupt examples, A.3
interrupt sequencing, 11.20.3
isochronous examples, A.5
isochronous sequencing, 11.21.3
periodic transactions
interrupt IN examples, A.4
interrupt OUT examples, A.3
interrupt transaction state machines, 11.20 to 11.20.4
isochronous IN examples, A.6
isochronous OUT examples, A.5
isochronous transaction state machines, 11.21 to 11.21.4

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

split transactions *(continued)*
periodic transactions *(continued)*
overview, 11.18 to 11.18.8
SETUP transactions, A.1
"single packets" and, 5.12.3
state machines
bulk/control state machines, 11.17.2
interrupt state machines, 11.20 to 11.20.4
isochronous transaction state machines, 11.21 to 11.21.4
overview, 11.16
split transaction state machines, 8.5
token packets, 8.4.2 to 8.4.2.3
Transaction Translator, 11.1.1, 11.14.1
squelch circuit, 7.1.20, 11.7.1.1
squelch detection
error detection and, 8.7.3, 8.7.4
turn-around timing and, 8.7.2
Squelch signal/event, 11.7.1.4 *Table 11-10*
Squelch state, 7.1, 7.1.4.2, 7.1.7.2
SRC
asynchronous SRC, 2.0 *glossary*
audio connectivity and, 5.12.4.4.1
defined, 2.0 *glossary*
synchronous SRC, 2.0 *glossary*
Sresume state, 11.6.4, 11.6.4.6
SSPLIT. *See* start-split transactions (SSPLIT)
staged power switching, 7.2.1.4
stages in control transfers, defined, 2.0 *glossary*, 5.5. *See also* Data stage; Setup stage; Status stage
STALLs, 8.3.1 *Table 8-1*
in bulk transfers, 5.8.5, 8.5.2, 11.17.1
in control transfers, 8.5.3.1, 11.17.1
data corrupted or not accepted, 8.6.3
functional and commanded stalls, 8.4.5
function response to IN transactions, 8.4.6.1
function response to OUT transactions, 8.4.6.3
STALLs *(continued)*
in interrupt transfers, 5.7.5, 8.5.4, 11.20.4
overview, 8.4.5
protocol stalls, 8.4.5
Ready/Stall status, 11.15
Request Error responses, 9.2.7
responses to standard device requests, 9.4
returned by control pipes, 8.5.3.4
standard device information, 4.8.1
standard device requests, 9.2.6.4, 9.4 to 9.4.11, 11.24.1
standards (applicable documents), 6.7.1
standard USB descriptor definitions, 9.6.1 to 9.6.5
Start/Complete field (SC), 8.4.2.2
Started timer status, C.0
Start field (S), 8.4.2.2
Start-of-Frame. *See* SOFs
Start of High-speed Packet (HSSOP), 7.1.7.2, 7.1.7.4.2
Start-of-Packet. *See* SOPs
Start-of-Packet bus state (SOP), 7.1.7.1, 7.1.7.2, 7.1.7.4.1, 7.1.7.4.2
start-of-packet delimiter. *See* SOPs
star topology, 5.2.3
start-split transactions (SSPLIT)
after loss of synchronication, 11.22.2
buffering, 11.14.2.1, 11.14.2.2, 11.14.2.3, 11.17
bulk/control split transactions, 11.17, 11.17.1
defined, 8.4.2, 11.14.1.2
freeing pending transactions, 11.18.6.2
isochronous transactions, 11.21
notation for, 11.15
overview, 11.14.1
scheduling, 11.14.2.1, 11.14.2.2, 11.14.2.3, 11.18.4
split transaction overview, 8.4.2.1
SSPLIT token, 8.4.2.2
tracking, 11.18.7

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

state handling. *See* bus states; status

state machines. *See also names of specific state machines under Dev_, HC_, and TT_*

actions in, 8.5, 11.15

bulk/control transaction state machines, 8.5.2, 11.17.2

conditions in, 8.5

device state machines, 8.5, 8.5.2, 8.5.5

diamond symbols in, 8.5, 11.15

downstream facing port state machines, 11.5

endpoint state machines, 8.5

example declarations, B.1, B.2, B.3

Host Controller state machines, 8.5, 11.16 to 11.16.1.1.2, 11.17.2, 11.20.2, 11.21.2

host state machines, 8.5.1, 8.5.2, 8.5.5

Hub Repeater state machine, 11.2.3.3, 11.7.2, 11.7.2.3 *Table 11-11*

Hub state machine, 11.1.1

initial states in, 8.5

input transitions in, 8.5

internal port state machine, 11.4

interrupt transaction state machines, 8.5.2, 8.5.2 *Figure 8-33,* 8.5.2 *Figure 8-34,* 11.20 to 11.20.4

isochronous transaction state machines, 8.5.5 *Figure 8-42,* 8.5.5 *Figure 8-43,* 11.21.2

notation in, 8.5, 11.15

output transitions in, 8.5

over-sampling state machine DPLLs, 7.1.15.1

overview, 8.5

port indicator colors, 11.5.3

port selector state machine, 11.7.1.4 to 11.7.1.4.4

receiver state machine, 11.6, 11.6.3 *Table 11-8*

reset protocol diagrams, C.0

resetting TT state machines, 11.24.2.9

split transaction state machine overview, 11.16

state hierarchy, 8.5

state machines. *(continued)*

states defined, 8.5

Transaction Translator state machines, 11.16, 11.16.2 to 11.16.2.1.7, 11.24.2.9

transitions in, 8.5

transmitter state machine, 11.6, 11.6.4, 11.6.4.2, 11.6.4 *Table 11-9*

static output swing of USBD, 7.1.1

status. *See also* status change bits

device states, 10.5.2.7, 11.12.2

Host Controller role in, 4.9

host's role in monitoring status and activity, 10.1.4

hub and port status change bitmap, 11.12.4

hub and port status changes, 7.1.7.5, 11.12.6

hub status, 11.24.2.6

notification of completion status, 10.3.4

port change information processing, 11.12.3

port indicators, 11.5.3 to 11.5.3.1

port status change bits, 11.24.2.7.2 to 11.24.2.7.2.5

USBD event notifications, 10.5.4.3

USBD status reporting and error recovery, 10.5.4.4

status change bits. *See also* Status Change endpoint

detecting changes, 11.12.2

device states, 11.12.2

hub and port status change bitmap, 11.12.4

hub status, 11.24.2.6

over-current status change bits, 11.12.5

port status change bits, 11.24.2.7.2 to 11.24.2.7.2.5

Status Change endpoint

defined, 11.12.1

device states and, 11.12.2

hub and port status change bitmap, 11.12.4

hub configuration and, 11.13

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Status Change endpoint *(continued)*
hub descriptors, 11.23.1
Status stage
in control transfers, 5.5, 5.5.5, 8.5.3
reporting status results, 8.5.3.1
StopTT() request, STOP_TT
hub class requests, 11.24.2
overview, 11.24.2.11
storage temperatures for cables, 6.6.4
stranded tinned conductors, 6.6.2
streaming real time transfers. *See* isochronous transfers
stream pipes
bulk transfers and, 5.8.2
in bus protocol overview, 4.4
defined, 2.0 *glossary,* 5.3.2
interrupt transfers and, 5.7.2
isochronous transfers and, 5.6.2
overview, 5.3.2.1
STRING descriptor, 9.4 *Table 9-5*
string descriptors
GetDescriptor() request, 9.4.3
as optional, 9.5
overview, 9.6.7
stuffed bits. *See* bit stuffing
subclasses
device_qualifer descriptor codes, 9.6.2
device subclass codes, 9.2.3, 9.6.1
interface subclass codes, 9.2.3, 9.6.5
SubClass field, 9.2.3
substrate materials
plug contact materials, 6.5.4.3
plug shell materials, 6.5.4.2
receptacle contact materials, 6.5.3.3
receptacle shell materials, 6.5.3.2
subtree devices after wakeup, 10.5.4.5
successful transfers, 8.6.2, 10.3.4
supply current, 7.3.2 *Table 7-7*
supply voltage
DC electrical characteristics, 7.3.2 *Table 7-7*
supply voltage *(continued)*
oscillators, 7.1.11
surge limiting, 7.2.4.1
Suspend bus state
global suspend, 7.1.7.6.1
overview, 7.1.7.6
power control during suspend/resume, 7.2.3
reset signaling, 7.1.7.5
resume signaling, 7.1.7.7
selective suspend, 7.1.7.6.2
Suspend Delay state, 11.4, 11.4.2
suspended devices
global suspend, 7.1.7.6.1
hub support for suspend signaling, 11.9
power control during suspend/resume, 7.2.3
power-on and connection events, 7.1.7.3
remote wakeup, 9.2.5.2, 10.2.7, 10.5.4.5
reset state machines, C.2.1
resume signaling, 7.1.7.7
selective suspend, 7.1.7.6.2
single-ended transmissions, 11.6.1
Suspend bus state, 7.1.7.6
Suspended device state, 9.1.1.6
suspended hubs
hub reset behavior, 11.10
resume signaling and, 11.1.2.2
suspended ports
C_PORT_SUSPEND, 11.24.2.7.2.3
getting port status, 11.24.2.7.1
port status change bits, 11.24.2.7.2
PORT_SUSPEND, 11.24.2.7.1.3, 11.24.2.13
Suspended state, 9.1.1.6, 9.1.1 *Table 9-1,* 11.5, 11.5.1.9. *See also* Suspend bus state; Suspend state
suspend sequencing, 11.22.2
Suspend state, 11.6.3, 11.6.3.4
switching thresholds for single-ended receivers, 7.1.4.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

SYNC field
in data signaling, 7.1.7.4.1
in electrical specifications overview, 4.2.1
high-speed signaling and, 7.1
overview, 8.2
squelch detection and, 11.7.1.1
SynchFrame() request, SYNCH_FRAME, 9.4, 9.4.11, 11.24.1
synchronization. *See also* synchronization types
clock synchronization, 5.12.3
data-per-time synchronization, 5.12.7
data toggle synchronization, 8.4.4, 8.6 to 8.6.5
endpoint synchronization frame, 9.4.11
frame and microframe timer synchronization, 11.2, 11.2.3 to 11.2.3.3
jitter, 2.0 *glossary* (*See also* jitter)
physical and virtual devices, 5.12.4.4
SYNC field, 8.2
sync pattern, 7.1.10
Transaction Translator loss of synchronization, 11.18.6, 11.22.1
transmitter and receiver synchronization in isochronous transfers, 5.12
synchronization types
adaptive, 5.12.4.1.3
asynchronous, 5.12.4.1.1
defined, 2.0 *glossary,* 5.12.4
endpoints and, 9.6.6
overview, 5.12.4.1
synchronous, 5.12.4.1.2
synchronous data connectivity, 5.12.4.4.2
synchronous data devices, 5.12.4
synchronous endpoints, 5.12.4.1.2, 5.12.4.4
synchronous RA, 2.0 *glossary,* 5.12.4.4
synchronous SRC, 2.0 *glossary*
sync pattern, 7.1.7.4.2, 7.1.9, 7.1.10
system configuration. *See* configuration
System Programming Interface, defined, 2.0 *glossary*
system software. *See* USB System Software

T

TDM, defined, 2.0 *glossary*
TDR loading specification, 2.0 *glossary,* 7.1.6.2
telephone interconnects, 1.1
temperature
data-rate inaccuracies and, 7.1.11
ranges for cables, 6.6.4
templates. *See* receiver eye pattern templates
termination
blunt cut and prepared termination, 6.4.2, 6.4.3
DC electrical characteristics, 7.3.2 *Table 7-7*
defined, 2.0 *glossary*
detachable cable assemblies, 6.4.1
electrical specifications overview, 4.2.1
high-/full-speed captive cable assemblies, 6.4.2
low-speed captive cable assemblies, 6.4.3
signal termination, 7.1, 7.1.5.1
USB topology rules, 6.4.4
termination data, 6.5.2
Termination Impedance, 7.1.6.2
test criteria for electrical, mechanical and environmental compliance, 6.7
Test for Flammability of Plastic Materials for Parts in Devices and Appliances, 6.7.1
Testing state, 11.5.1.14
Test_J test mode, 7.1.20, 9.4.9, 11.24.2.13
Test_K test mode, 7.1.20, 9.4.9, 11.24.2.13
test mode, 7.1.20, 9.4.9, 11.24.2.7.1.9, 11.24.2.13
TEST_MODE, 7.1.20, 9.4.9, 9.4 *Table 9-6*
Test_Packet test mode, 7.1.20, 9.4.9, 11.24.2.13
test planes in high-speed signaling, 7.1.2.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Test_SE0_NAK test mode, 7.1.20, 9.4.9, 11.24.2.13

TEST_SELECTOR, 9.4.9

thermal shock standards, 6.7 *Table 6-7*

Thevenin resistance, 7.1.5.1

"think time," 11.18.2, 11.23.2.1

"three strikes and you're out" mechanism, 11.17.1

Through Impedance, 7.1.6.2

tiered topology

- EOF point advancement and, 11.2.3.2
- tiered star topology, 5.2.3
- tiers in bus typology, 4.1.1

Time Division Multiplexing (TDM), 2.0 *glossary*

Time Domain Reflectometer loading specification, 2.0 *glossary,* 7.1.6.2

timed states

- Disconnected state, 11.5.1.3
- Resuming state, 11.5.1.10

timeout

- bus transaction timeout, 5.12.7
- defined, 2.0 *glossary*
- detecting timeout conditions, 10.2.6
- high bandwidth transactions and, 5.9.1
- split transaction flow sequences, 11.18.8
- timeout intervals in error detection, 8.7.2, 8.7.3

timeouts, 11.15, 11.17.1

timing. *See also* cable delay; propagation delay; skew; synchronization; timing waveforms

- bus timing/electrical characteristics, 7.3.2
- bus transaction time calculations, 5.11.3
- bus turn-around timing, 8.7.2
- clock model, 5.12.2
- clock synchronization, 5.12.3
- completion times for hub requests, 11.24.1
- current frame timer, 11.2.3.1

timing *(continued)*

- data source signaling, 7.1.13 to 7.1.13.2.2
- device event timings, 7.3.2 *Table 7-14*
- frame and microframe intervals, 7.1.12
- frame and microframe timers, 11.2.3 to 11.2.3.3
- hub event timings, 7.3.2 *Table 7-13*
- hub frame timer, 11.2 to 11.2.5.2
- hub signaling timings, 7.1.14 to 7.1.14.2
- isochronous transfer feedback, 5.12.4.2
- isochronous transfer importance, 5.12
- low, full, and high-speed turn-around timing, 8.7.2
- next frame timer, 11.2.3.1
- in non-USB isochronous application, 5.12.1
- port disconnect timer, 11.5.2
- power-on and connection events timing, 7.1.7.3
- remote wakeup timing relationships, 11.9
- request processing timing, 9.2.6.1
- Resetting state and Resuming state intervals, 11.5.1.10
- Run, Clear, and Started timer status, C.0
- SE0 for EOP width timing, 7.1.13.2.1
- skew accumulating between host and hub, 11.2.5.1 to 11.2.5.2
- SOF PID timing information, 8.4.3
- SOF tokens as clocks, 5.12.5
- synchronization types, 5.12.4.1

timing waveforms, 7.3.3

- differential data jitter, 7.3.3 *Figure 7-49*
- differential-to-EOP transition skew and EOP width, 7.3.3 *Figure 7-50*
- hub differential delay, differential jitter, and SOP distortion, 7.3.3 *Figure 7-52*
- hub EOP delay and EOP skew, 7.3.3 *Figure 7-53*
- receiver jitter tolerance, 7.3.3 *Figure 7-51*

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

toggle mode. *See* data toggle
toggle sequencing, 8.5.5
token packets
in bulk transfers, 8.5.2
bus protocol overview, 4.4
CRCs, 8.3.5.1
defined, 2.0 *glossary*
in isochronous transfers, 8.5.5
overview, 8.4.1
packet field formats, 8.3 to 8.3.5.2
split transaction token packets, 8.4.2 to 8.4.2.3
token phases, notation for, 11.15
token PIDs, 8.3.1 *Table 8-1*. *See also* IN PID; OUT PID; SETUP PID; SOF PID
topology
bus topology, 4.1, 4.1.1, 5.2 to 5.2.5
EOF point advancement and, 11.2.3.2
hub tiers defined, 2.0 *glossary*
trace delays, 7.1.14.2
tracking transactions, 11.18.7
transaction completion prediction, 11.3.3
transaction list
defined, 5.11.1.4
HCD role in, 5.11.1.3
Host Controller and, 5.11.1.5
transactions. *See also specific types of transactions*
aborting, 11.18.6, 11.18.6.1
allocating bandwidth for, 5.11.1 to 5.11.1.5, 10.3.2
buffer size calculations, 5.11.4
bus protocol overview, 4.4
defined, 2.0 *glossary*
error detection and recovery, 8.7 to 8.7.4
maximum allowable transactions per microframe, 5.4.1, 11.18.6.3
multiple transactions in microframes, 5.9, 5.9.2
organization within IRPs, 5.11.2
packet sequences, 8.5
pending, 11.18.6
transactions. *(continued)*
PING flow control protocol and, 5.5.4
scheduling, 4.4, 11.14.2 to 11.14.2.3
split transactions, 5.5.4, 8.4.2 to 8.4.2.3, A.1, A.2, A.3, A.4
state machine overview, 8.5
timeout, 5.12.7
tracking transactions, 5.11.2
transaction completion prediction, 11.3.3
transaction formats
bulk transfers, 5.8.4, 8.5.2, A.1, A.2
control transfers, 5.5.4, 8.5.3 to 8.5.3.4, A.1, A.2
IN transactions, A.2, A.4, A.6
interrupt transfers, 8.5.4, A.3, A.4
isochronous transfers, 5.6.3, 5.12.6, 5.12.7, 8.5.5, A.5, A.6
non-periodic transactions, 11.17 to 11.17.5
OUT transactions, A.1, A.3, A.5
overview, 8.5
periodic and non-periodic transactions, 11.14.1, 11.18 to 11.18.8, 11.22.1
SETUP transactions, A.1
transaction list, 5.11.1.3, 5.11.1.4, 5.11.1.5
transaction time calculations, 5.11.3
Transaction Translator, 4.8.2.1, 11.18.7
Transaction Translator
aborting transactions, 11.18.6.1
buffers
buffer space required, 11.17.4, 11.19
clearing buffers, 11.17.5, 11.24.2.3
periodic and non-periodic buffer sections, 11.14.1
complete-split state searching, 11.18.8
data handling, 11.14.1.1
defined, 2.0 *glossary,* 4.8.2.1, 11.1
delay in bus times, 5.11.3
error handling, 11.22
frame and microframe jitter, 11.2.4

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Transaction Translator *(continued)*
freeing pending start-splits, 11.18.6.2
full-speed frame generation, 11.18.3
GET_TT_STATE, 11.24.2.8
host controller and, 11.14.1.2
hub architecture and, 11.1.1
hub class descriptors and, 11.23.1
loss of synchronization, 11.18.6, 11.22.1
low-speed signaling, 8.6.5
microframe pipelines and, 11.18.2
multiple TTs, 11.14.1.3, 11.23.1, 11.24.2.8
resetting, 11.24.2.9
response generation, 11.18.5
scheduling, 11.14.2 to 11.14.2.3
split transaction notation, 11.15
state machines
bulk/control state machines, 11.17.2
declarations, B.3
interrupt transaction state machines, 11.20.2
isochronous transaction state machines, 11.21.2
overview, 11.16, 11.16.2 to 11.16.2.1.7
stopping normal execution, 11.24.2.11
"think time," 11.18.2, 11.23.2.1
in transactions
bulk/control transactions, 11.17.2, 11.17.4
interrupt transactions, 11.20.2
isochronous transactions, 11.21.2, 11.21 to 11.21.4
non-periodic transactions, 11.17 to 11.17.5
periodic transactions, 11.18 to 11.18.8, 11.22.1
transaction tracking, 11.18.7
transceivers
downstream facing ports and hubs, 7.1.4.2, 7.1.7.1
full- and high-speed signaling, 7.1, 7.1.1.1
transceivers *(continued)*
lumped capacitance guidelines for transceivers, 7.1.6.2
transfer management, 5.11.1 to 5.11.1.5
allocating bandwidth, overview, 4.7.5, 5.11.1.1
client software, 5.11.1.1
HCD, 5.11.1.3
Host Controller, 5.11.1.5
illustrated, 5.11.1
transaction list, 5.11.1.4
USB driver, 5.11.1.2
USB System, 10.3.2
transfers, 5.0. *See also* transactions; *names of specific transfer types (i.e., bulk transfers)*
bulk transfers, 2.0 *glossary,* 4.7.2, 5.8 to 5.8.5, 8.5.2
bus access for transfers, 5.11 to 5.11.5
bus bandwidth reclamation, 5.11.5
calculating buffer sizes in functions and software, 5.11.4
calculating bus transaction times, 5.11.3
transaction tracking, 5.11.2
transfer management, 5.11.1 to 5.11.1.5
bus protocol overview, 4.4
bus topology, 5.2 to 5.2.5
communication flow, 4.1, 5.3 to 5.3.3
control transfers, 4.7.1, 5.5 to 5.5.5, 8.5.2, 8.5.3 to 8.5.3.4
data prebuffering, 5.12.5
data signaling overview, 7.1.7.4 to 7.1.7.4.2
defined, 2.0 *glossary*
error detection and recovery, 8.7 to 8.7.4
frames and microframes, 5.3.3
high-bandwidth transfers, 5.9.1, 5.9.2
high-speed transfer rates in 2.0, 1.1
Host Controller responsibilities, 4.9, 10.1.3
hub connectivity and, 11.1.2.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

transfers *(continued)*
implementer viewpoints, 5.1
interrupt transfers, 2.0 *glossary,* 4.7, 4.7.3, 5.7 to 5.7.5, 5.9.1, 8.5.2, 8.5.4
isochronous transfers, 2.0 *glossary,* 4.7.4, 5.6 to 5.6.5, 5.9.2, 5.12 to 5.12.8, 8.5.5
operations overview, 9.2.4
organization of transactions within frames, 5.11.2
overview, 5.0
periodic transfers, 5.6.4, 5.7.4
power management, 9.2.5
request processing, 9.2.6 to 9.2.6.6
standard device requests, 9.4 to 9.4.11
time limits for completing, 9.2.6.4, 9.2.6.5
transaction formats, 8.5 to 8.5.5
transfer types, 4.7 to 4.7.5, 5.4 to 5.8.5
USB device requests, 9.3 to 9.3.5
USBD role in, 10.1.1, 10.5.3 to 10.5.3.2.3
USB System role in, 10.3.3
transfer types. *See also* transactions; transfers; *names of specific transfer types (i.e., bulk transfers)*
allocating USB bandwidth, 4.7.5
bulk transfers, 2.0 *glossary*, 4.7.2, 5.8
in calculating transaction times, 5.11.3
control transfers, 4.7.1, 5.5
endpoint field indicators, 9.6.6
high-bandwidth transfers, 5.9.1, 5.9.2
interrupt transfers, 2.0 *glossary,* 4.7.3, 5.7
isochronous transfers, 2.0 *glossary,* 4.7.4, 5.6
for message pipes, 5.3.2.2
overview, 4.7 to 4.7.5, 5.4
pipes and, 4.4
split transactions and, 5.10
for stream pipes, 5.3.2.1
transfer types defined, 2.0 *glossary*
transitions in state machines, 8.5, 11.15
transmission envelope detectors, 7.1.4.2, 7.1 *Table 7-1*
transmit clock, 11.7.1.3
transmit eye patterns, 7.1, 7.1.2
transmit phase of signaling, 7.1.1
TransmitR state, 11.5.1.8
Transmit state, 11.5, 11.5.1.7
Transmitter/Receiver Test Fixture, 7.1.2.2 *Figure 7-12*
transmitters
Active state, 11.6.4.2
Generate End of Packet Towards Upstream Port state (GEOPTU), 11.6.4.5
Inactive state, 11.6.4.1
RepeatingSE0 state, 11.6.4.3
SendJ state, 11.6.4.4
Send Resume state (Sresume), 11.6.4.6
transmitter data jitter, 7.1.13.1.1
transmitter sequence bits, 8.6, 8.6.2
transmitter state descriptions, 11.6.4
transmitter state machine, 11.6, 11.6.4
transmitter state machine, 11.6, 11.6.4
transmit waveform requirements, 7.1.2.2 *Figure 7-13,* 7.1.2.2 *Figure 7-14,* 7.1.2.2 *Figure 7-17*
TrueRWU signal/event, 11.5 *Figure 11-10,* 11.5 *Table 11-5*
truncated packets, 11.3.2
TT. *See* Transaction Translator
TT_BulkCS state machine, 11.16.2.1.4
TT_BulkSS state machine, 11.16.2.1.3
TT_Do_BICS state machine, 11.17.2
TT_Do_BISS state machine, 11.17.2
TT_Do_BOCS state machine, 11.17.2
TT_Do_BOSS state machine, 11.17.2
TT_Do_complete state machine, 11.16.2.1.2
TT_Do_IntICS state machine, 11.20.2
TT_Do_IntISS state machine, 11.20.2
TT_Do_IntOCS state machine, 11.20.2
TT_Do_IntOSS state machine, 11.20.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

TT_Do_IsochISS state machine, 11.21.2
TT_Do_IsochOSS state machine, 11.21.2
TT_Do_Start state machine, 11.16.2.1.1
TT_Flags bits, 11.24.2.8
TT_IntCS state machine, 11.16.2.1.6
TT_IntSS state machine, 11.16.2.1.5
TT_IsochICS state machine, 11.21.2
TT_IsochSS state machine, 11.16.2.1.7
TT_Process_Packet state machine, 11.16.2.1
TT_Return_Flags field, 11.24.2.8
TT_specific_state field, 11.24.2.8
turn-around times
  defined, 2.0 *glossary*
  error detection, 8.7.2
  overview, 7.1.18 to 7.1.18.2
turning power on for ports, 11.11
twisted data pair in cables, 6.6.1
Tx_active signal/event, 11.6.3 *Table 11-8*
Tx_resume signal/event, 11.6.3 *Table 11-8*

U

UEOP signal/event, 11.7.2.3 *Table 11-11*
UL listing for cables, 6.6.5
UL STD-94, 6.7.1
UL Subject-444, 6.6.5, 6.7.1
unacceptable cables, 6.4.4
underplating
  plug contact materials, 6.5.4.3
  plug shell materials, 6.5.4.2
  receptacle contact materials, 6.5.3.3
  receptacle shell materials, 6.5.3.2
Underwriter's Laboratory, Inc., 6.6.5, 6.7.1
The Unicode Standard, Worldwide Character Encoding, 9.6.7
UNICODE string descriptors, 9.6.7
unique addresses
  assigning after dynamic insertion or removal, 4.6.3
  device initialization, 10.5.1.1
  operations overview, 9.2.2
  SetAddress() request, 9.4.6
unique addresses *(continued)*
  time limits for completing addressing, 9.2.6.3
Universal Serial Bus
  architectural extensions, 4.10
  backwards compatibility, 3.1
  bus protocol, 4.4
  clock model, 5.12, 5.12.2
  components, 5.1
  configuration, 4.6 to 4.6.3, 10.3.1
  data flow and transfers, 4.7 to 4.7.5, 5.1 to 5.10.8
  description, 4.1 to 4.1.1.2
  feature list, 3.3
  goals, 3.1
  high-speed applications, 3.2
  host hardware and software, 4.9, 10.2 to 10.6
  hubs, 11.1 to 11.16
  mechanical and electrical specifications, 6.1 to 6.9, 7.1 to 7.1.20, 7.3 to 7.3.3
  motivation for development, 1.1
  physical interface, 4.2 to 4.2.2
  power distribution, 4.3 to 4.3.2, 7.2 to 7.2.4.2
  protocol layer, 8.1 to 8.7
  range of USB data traffic workloads, 3.2
  robustness and error detection/recovery, 4.5 to 4.5.2
  USB device framework, 9.1 to 9.7.3
  USB devices, 4.8 to 4.8.2.2
  USB schedule, 4.1
Universal Serial Bus Driver. *See* USBD (USB Driver)
Universal Serial Bus Resources, 2.0 *glossary*
up counters in hub timing, 11.2.3.1
upgrade paths, 3.3
upstream facing ports and hubs
  defined, 4.8.2.1
  driver speed and, 7.1.2.3
  full-speed port transceiver, 7.1, 7.1.7.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

upstream facing ports and hubs *(continued)*
high-speed detection, 7.1.5.2
high-speed signaling and, 11.1.1
hub architecture, 11.1.1
hub EOP delay and EOP skew, 7.3.3 *Figure 7-53*
input capacitance, 7.1.6.1
jitter, 7.3.2 *Table 7-10*
low-speed source electrical characteristics, 7.3.2 *Table 7-10*
receivers, 11.6.3 to 11.6.3.9
reset on upstream port, 11.10
reset state machines, C.2
signaling delays, 7.1.14.1
signaling speeds and, 7.1
test mode support, 7.1.20
transmitters, 11.6 to 11.6.4.6
upstream connectivity defined, 11.1.2.1
upstream defined, 2.0 *glossary*
upstream hub delay, 7.3.3 *Figure 7-52*
upstream facing transceivers, signalling speeds and, 7, 7.1
upstream packets (HSU2), 8.5
upstream plugs, 6.2
Usage Types, 9.6.6
USB. *See* Universal Serial Bus
USB 2.0 Adopters Agreement, 1.4
USB Bus Interface layer
in bus topology, 5.2.2
detailed communication flow illustrated, 5.3
Host Controller implementation, 10.1.1
illustrated, 5.1
interlayer communications model, 10.1.1
USBD (USB Driver). *See also* USBDI (USB Driver Interface)
in bus topology, 5.2.1
command mechanisms, 10.5.1 to 10.5.2.12
as component of USB System, 10.1.1
configuration and, 10.3.1
USBD (USB Driver). *(continued)*
control mechanisms, 10.1.2
data transfer mechanisms, 10.1.3
defined, 2.0 *glossary,* 5.3, 10.5
driver characteristics, 7.1.1
driver speed and, 7.1.2.3
full-and low-speed drivers, 7.1.1.1, 7.1.1.2
HCD interaction with, 10.4
hub drivers, 10.3.1
initialization, 10.5.1.1
overview, 10.5.1
passing preboot control to operating system, 10.5.5
pipe mechanisms, 5.11.1.2, 10.5.1 to 10.5.3.2.4
request data format mechanisms, 10.3.4
service capabilities, 10.5.1.3
software interface overview, 10.3
in transfer management, 5.11.1, 5.11.1.2
USB System and, 10.5.4 to 10.5.4.5
USB device framework, 9, 9.7.2
descriptors, 9.5 to 9.7.3
device class definitions, 9.7 to 9.7.3
generic USB device operations, 9.2 to 9.2.7
address assignment, 9.2.2
configuration, 9.2.3
data transfer, 9.2.4
dynamic attachment and removal, 9.2.1
power management, 9.2.5
request error, 9.2.7
request processing, 9.2.6 to 9.2.6.6
USB device framework, 9, 9.7.2
standard descriptor definitions, 9.6 to 9.6.7
standard device requests, 9.4 to 9.4.11
USB device requests, 9.3 to 9.3.5
USB device states, 9.1 to 9.1.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

USB Device layer
detailed communication flow illustrated, 5.3
illustrated, 5.1
interlayer communications model, 10.1.1
USB devices. *See* devices
USBDI (USB Driver Interface)
adding devices, 10.5.2.5
alternate interface mechanisms, 10.5.2.10
getting descriptors, 10.5.2.3
removing devices, 10.5.2.6
role in request data format, 10.3.4
sending class commands, 10.5.2.8
sending vendor commands, 10.5.2.9
setting descriptors, 10.5.2.12
software interface overview, 10.3
USB host. *See* host
USB host controller. *See* Host Controller
USB Icon, 6.5, 6.5.1
USB-IF (USB Implementers Forum, Inc.), 1.4, 2.0 *glossary*
USB Implementers Forum, 1.4, 2.0 *glossary*
USB interconnect model, 4.1, 5.12.4.4
USB Logical Devices. *See* logical devices
USB Physical Devices. *See* physical devices
USB schedule, 4.1
USB Specification Release Number, 9.6.1
USB System. *See also* HCD; host software; USBD
allocating bandwidth, 10.3.2
buffers and, 10.2.9
data transfer role, 10.3.3
HCD component, 10.1.1
Host Controller interaction, 10.1.1
host software component, 10.1.1
power management, 10.5.4.2
remote wakeup, 10.2.7, 10.5.4.5
software interface overview, 10.3
state handling, 10.2.1
USB System. *(continued)*
status and activity monitoring, 10.1.4
USBD component, 10.1.1
USB System Software
asynchronous data transfers, 4.9
bus enumeration, 4.9
in bus topology, 5.2.1
in communication flow, 5.3
detecting hub and port status changes, 11.12.2
as implementation focus area, 5.1
interrupt transfer support, 5.7.3
isochronous transfer support, 4.9
power management, 4.9
role, 4.9

V

variable-length data stages, 8.5.3.2
variable-sized data payloads, 5.3.2
VBus leads
bypassing, 7.2.4.1
cable electrical characteristics, 7.3.2 *Table 7-12*
detachable cables, 6.4.1
in electrical specifications overview, 4.2.1
high-/full-speed captive cable assemblies, 6.4.2
low-speed captive cable assemblies, 6.4.3
standardized contact terminating assignments, 6.5.2
upstream port power supply and, 7.2.1
Vendor IDs in device descriptors, 9.6.1
vendor information in device characteristics, 4.8.1
vendor-specific descriptors, 9.5
vendor-specific requests, 10.5.2.9
version numbers in device descriptors, 9.6.1
version numbers in device_qualifer descriptor, 9.6.2
VHDL syntax, 11.15
V/I characteristics of full-speed connections, 7.1.1.1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

virtual devices, 2.0 *glossary,* 5.12.4.4
visible device states, 9.1.1
visual inspection standards, 6.7 *Table 6-7*
voltage
average voltage on D+/D- lines, 7.1.2.1
cross-over voltage in signaling, 7.1.2.1
DC output voltage specifications, 7.1.6.2
droop, 7.2.3, 7.2.4.1
flyback voltage, 7.2.4.2
full-speed connections, 7.1.1.1
high-speed signaling and, 7.1
open-circuit voltage, 7.1.1
ratings for cables, 6.6.3
reduction due to cable resistive effects, 7.2.3
reversing in high-speed signaling, 7.1.1.3
supply voltage, 7.3.2 *Table 7-7*
test mode, 7.1.20
voltage drop budget, 7.2.2
voltage drops, 7.2.1.1
voltage drop topology, 7.2.2
zero impedance voltage sources, 7.1.1

W

Wait for End of Packet from Upstream Port state (WFEOPFU), 11.7.2.3 *Figure 11-16,* 11.7.4
Wait for End of Packet (WFEOP) state, 11.7.2.3 *Figure 11-16,* 11.7.6
Wait for Start of Packet from Upstream Port state (WFSOPFU), 11.7.2.3 *Figure 11-16,* 11.7.3
Wait for Start of Packet (WFSOP) state, 11.7.2.3 *Figure 11-16,* 11.7.5
wander, defined, 11.2.5.2
waveforms
differential data jitter, 7.3.3 *Figure 7-49*
differential-to-EOP transition skew and EOP width, 7.3.3 *Figure 7-50*
full-speed driver signal waveforms, 7.1.1.1
hub differential delay, differential jitter, and SOP distortion, 7.3.3 *Figure 7-52*
waveforms *(continued)*
hub EOP delay and EOP skew, 7.3.3 *Figure 7-53*
maximum input waveforms for signaling, 7.1.1
receiver jitter tolerance, 7.3.3 *Figure 7-51*
testing, 7.1.20
WFEOPFU state, 11.5.1.6, 11.7.2.3 *Figure 11-16,* 11.7.4
WFEOP state, 11.7.2.3 *Figure 11-16,* 11.7.6
WFSOPFU state, 11.7.2.3 *Figure 11-16,* 11.7.3
WFSOP state, 11.7.2.3 *Figure 11-16,* 11.7.5
wHubChange field, 11.24.2.6
wHubCharacteristics field
hub descriptor, 11.23.2.1
multiple gangs and, 11.11.1
over-current reporting, 11.12.5
port indicator status, 11.5.3
power switching settings, 11.11
wHubStatus field, 11.24.2.6
wIndex field
hub class requests, 11.24.2.2
overview, 9.3.4
Setup data format, 9.3
standard device requests, 9.4
wire gauge in cables, 6.6.2
wire insulation in cables, 6.6.2
wiring assignments for conductors, 6.5.2
wLANGID[] field (string descriptors), 9.6.7
wLength field
hub class requests, 11.24.2
overview, 9.3.5
Setup data format, 9.3
standard device requests, 9.4
wMaxPacketSize field
bulk transfers and, 5.8.3
control transfer packet size, 5.5.3
endpoint descriptors, 9.6.6, 11.23.1
high bandwidth endpoints and, 5.9

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

wMaxPacketSize field *(continued)*

interrupt transfer packet size, 5.7.3

variable-length data stages, 8.5.3.2

words, defined, 2.0 *glossary*

working space, location and length of, 10.3.4

worst-case bit stuffing, 5.11.3

worst-case signal delay, 7.1.19.1, 7.1.19.2

wPortChange field, 11.24.2.7, 11.24.2.7.2

wPortStatus field, 11.24.2.7, 11.24.2.7.1

wTotalLength field

configuration descriptors, 9.6.3, 11.23.1

other speed configuration descriptors, 9.6.4, 11.23.1

wValue field

hub class requests, 11.24.2

overview, 9.3.3

Setup data format, 9.3

standard device requests, 9.4

Z

zero impedance voltage sources, 7.1.1

zeroth microframe, 9.4.11, 11.14.2.3, 11.18.3, 11.22.2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Note: All Engineering Change Notice's (ECN) and Errata documents as of September 01, 2012 that pertain to this core specification follow the last page of the specification starting on page 619.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Errata for “USB Revision 2.0 April 27, 2000”as of 12/7/2000

Check the http://www.usb.org website for the latest errata.

Chapter 5

Reversed Values in Transaction Overhead Tables:

Background: The tables in chapter 5 showing high-speed transaction overheads have typographical errors where the interpacket gap and minimum bus turnaround time values were reversed. No functional impact.

Change: p. 54, table 5-3 & p. 57, table 5-5 and p. 61, table 5-8: change “…8 bit interpacket gap, 88 bit min bus…” to “…88 bit interpacket gap, 8 bit min bus…”.

Table 5-3. High-speed Control Transfer Limits

	Protocol Overhead (173 bytes)	(Based on 480 Mb/s and 88 bit interpacket gap, 88 bit min bus turnaround, 32 bit sync, 8 bit EOP: (9x4 SYNC bytes, 9 PID bytes, 6 EP/ADDR+CRC,6 CRC16, 8 Setup data, 9x(1+11) byte interpacket delay (EOP, etc.))				
	Data Payload	**Max Bandwidth (bytes/second)**	**Microframe Bandwidth per Transfer**	**Max Transfers**	**Bytes Remaining**	**Bytes/ Microframe Useful Data**
	1	344000	2%	43	18	43
	2	672000	2%	42	150	84
	4	1344000	2%	42	66	168
	8	2624000	2%	41	79	328
	16	4992000	3%	39	129	624
	32	9216000	3%	36	120	1152
	64	15872000	3%	31	153	1984
Max		60000000				7500

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 5-8. High-speed Interrupt Transaction Limits

	Protocol Overhead		(Based on 480 Mb/s and 88 bit interpacket gap, 88 bit min bus turnaround, 32 bit sync, 8 bit EOP: (3x4 SYNC bytes, 3 PID bytes, 2 EP/ADDR+CRC bytes, 2 CRC16 and a 3x(1+11) byte interpacket delay(EOP, etc.))			
	Data Payload	Max Bandwidth (bytes/second)	Microframe Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/ Microframe Useful Data
	1	1064000	1%	133	52	133
	2	2096000	1%	131	33	262
	4	4064000	1%	127	7	508
	8	7616000	1%	119	3	952
	16	13440000	1%	105	45	1680
	32	22016000	1%	86	18	2752
	64	32256000	2%	63	3	4032
	128	40960000	2%	40	180	5120
	256	49152000	4%	24	36	6144
	512	53248000	8%	13	129	6656
	1024	49152000	14%	6	1026	6144
	2048	49152000	28%	3	1191	6144
	3072	49152000	42%	2	1246	6144
Max		60000000				7500

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 5-5. High-speed Isochronous Transaction Limits

	Protocol Overhead		(Based on 480 Mb/s and 88 bit interpacket gap, 88 bit min bus turnaround, 32 bit sync, 8 bit EOP: (2x4 SYNC bytes, 2 PID bytes, 2 EP/ADDR+addr+CRC5, 2 CRC16, and a 2x(1+11)) byte interpacket delay (EOP, etc.))			
	Data Payload	Max Bandwidth (bytes/second)	Microframe Bandwidth per Transfer	Max Transfers	Bytes Remaining	Bytes/ MicroFrame Useful Data
	1	1536000	1%	192	12	192
	2	2992000	1%	187	20	374
	4	5696000	1%	178	24	712
	8	10432000	1%	163	2	1304
	16	17664000	1%	138	48	2208
	32	27392000	1%	107	10	3424
	64	37376000	1%	73	54	4672
	128	46080000	2%	45	30	5760
	256	51200000	4%	25	150	6400
	512	53248000	7%	13	350	6656
	1024	57344000	14%	7	66	7168
	2048	49152000	28%	3	1242	6144
	3072	49152000	41%	2	1280	6144
Max		60000000				7500

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Chapter 7

Issue with Test_SE0_NAK Behavior:

Background: While in test mode Test_SE0_NAK, upstream facing ports "must respond to any IN token packet with a NAK handshake (only if the packet CRC is determined to be correct) within the normal device response time." The question has been raised as to the meaning of "any IN token" and whether a device must NAK even if the packet is not directed to the device or directed to any endpoint on that device. If it must, a device placed in this test mode will NAK even when an IN token is directed to another devices, such as a mouse or keyboard, resulting in possible problems for other devices connected to the bus during compliance testing. At this point, there are probably implementations that interpret the spec both ways, so this erratum expands the spec to allow either interpretation. This will have to be dealt with in compliance testing. There will be no impact on non-test-mode operation.

Change: p. 170, section 7.1.20, 1st bullet, 3rd sentence: Change complete sentence to: "In addition, while in this mode, upstream facing ports (and only upstream facing ports) must respond to any IN token packet directed to the device (with an endpoint number supported by the device), and optionally to any IN token packet which is not directed to the device or optionally with any endpoint number, with a NAK handshake (only if the packet CRC is determined to be correct) within the normal device response time. Note: This means that the host should send test packets to a device under test with the device's current address and endpoint number zero to ensure that the device will respond."

7.1.20 Test Mode Support

To facilitate compliance testing, host controllers, hubs, and high-speed capable functions must support the following test modes:

- Test mode Test_SE0_NAK: Upon command, a port's transceiver must enter the high-speed receive mode and remain in that mode until the exit action is taken. This enables the testing of output impedance, low level output voltage, and loading characteristics. ~~In addition, while in this mode, upstream facing ports (and only upstream facing ports) must respond to any IN token packet with a NAK handshake (only if the packet CRC is determined to be correct) within the normal allowed device response time.~~ In addition, while in this mode, upstream facing ports (and only upstream facing ports) must respond to any IN token packet directed to the device (with an endpoint number supported by the device), and optionally to any IN token packet which is not directed to the device or optionally with any endpoint number, with a NAK handshake (only if the packet CRC is determined to be correct) within the normal device response time. Note: This means that the host should send test packets to a device under test with the device's current address and endpoint number zero to ensure that the device will respond. This enables testing of the device squelch level circuitry and, additionally, provides a general purpose stimulus/response test for basic functional testing.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Issue with Test_Force_Enable Behavior:

Background: The question has been raised as to the meaning of the language that says "…the disconnect bit can be polled while varying the loading on the port, allowing the disconnect detection threshold voltage to be measured." Some implementations have allowed the port to exit the test mode and return to the disconnected port state when the disconnect threshold is exceeded. Others have let the port stay in the test mode and allowed the bit to be "live". This erratum allows either behavior. There will be no impact on non-test-mode operation.

Change: p. 171, section 7.1.20, bullet on Test_Force_Enable: Add sentence at end of paragraph, "Disconnect detection is optionally allowed to exit test_mode and return the port to the disconnected port state."

- Test mode Test_Force_Enable: Upon command, downstream facing hub ports (and only downstream facing hub ports) must be enabled in high-speed mode, even if there is no device attached. Packets arriving at the hub's upstream facing port must be repeated on the port which is in this test mode. This enables testing of the hub's disconnect detection; the disconnect detect bit can be polled while varying the loading on the port, allowing the disconnect detection threshold voltage to be measured. Disconnect detection is optionally allowed to exit test mode and return the port to the disconnected port state.

Issue with Test_Packet behavior:

Background: The question has been raised as to the minimum inter-packet gap allowed for an upstream facing port in test mode Test_Packet. The spec states that "The inter-packet timing must be no less than the minimum allowable inter-packet gap as defined in Section 7.1.18 and no greater than 125us." Unfortunately, an upstream facing port never sends back-to-back packets, so this parameter is actually not defined. Since an upstream facing port may send a packet in response to an incoming packet after a minimum of 8 bit times, some designers have interpreted this to mean the minimum inter-packet gap in the test mode should be 8 bit times. This erratum allows this interpretation. It will meet the needs of compliance testing, and there will be no impact on non-test-mode operation.

Change: p. 170, section 7.1.20, bullet on Test_Packet: Add sentence at end of paragraph, "For an upstream facing port, the inter-packet timing must be no less than 8 bit times."

- Test mode Test_Packet: Upon command, a port must repetitively transmit the following test packet until the exit action is taken. This enables the testing of rise and fall times, eye patterns, jitter, and any other dynamic waveform specifications.

  The test packet is made up by concatenating the following strings. (Note: For J/K NRZI data, and for NRZ data, the bit on the left is the first one transmitted. "S" indicates that a bit stuff occurs, which inserts an "extra" NRZI data bit. "* N" is used to indicate N occurrences of a string of bits or symbols.)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

•

NRZI Symbols (Fields)	NRZ Bit Strings	Number of NRZ Bits
{KJ * 15}, KK (SYNC)	{00000000 * 3}, 00000001	32
KKJKJKKK (DATA0 PID)	11000011	8
JKJKJKJK * 9	00000000 * 9	72
JJKKJJKK * 8	01010101 * 8	64
JJJJKKKK * 8	01110111 * 8	64
JJJJJJJKKKKKKK * 8	0, {111111S *15}, 111111	97
JJJJJJJK * 8	S, 111111S, {0111111S * 7}	55
{JKKKKKKK * 10}, JK	00111111, {S0111111 * 9}, S0	72
JJJKKKJJKKKKJKKK (CRC16)	0110110101110011	16
JJJJJJJJ (EOP)	01111111	8

A port in Test_Packet mode must send this packet repetitively. The inter-packet timing must be no less than the minimum allowable inter-packet gap as defined in Section 7.1.18 and no greater than 125 µs. For an upstream facing port, the inter-packet timing must be no less than 8 bit times.

Issue with TDR Loading Specification:

Background: On page 144, Section 7.1.6.2, several paragraphs from the end of the section, there is a sentence which reads “No single excursion, however, may exceed the Through limits for more than twice the TDR rise time (400 ps).” This is confusing, since the TDR risetime is required to be 400 ps, not 200 ps. This erratum clarifies the language. There is no negative impact caused by this clarification.

Change: Change the sentence to read “No single excursion, however, may exceed the Through limits for more than twice the TDR rise time (twice the 400 ps risetime is 800 ps).”

Through Impedance (Z_{HSTHRU}) is the impedance measured from 500 ps before the connector reference location until the time governed by the Termination impedance specification.

$$70\ \Omega \leq Z_{HSTHRU} \leq 110\ \Omega$$

In the Exception Window (a sliding 1.4 ns window inside the Through Impedance time window), the differential impedance may exceed the Through limits. No single excursion, however, may exceed the Through limits for more than twice the TDR rise time (twice the 400 ps risetime is 800 ps).~~No single excursion, however, may exceed the Through limits for more than twice the TDR rise time (400 ps).~~

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Chapter 8

Issue with Some PING STALL Response Omissions:

Background: The PING protocol state machines and “railroad diagrams” show (correctly) that STALL is an allowed response to PING. However, two sentences in the spec do not explicitly mention STALL. This change should cause no functional impact since the more explicit and detailed figures show the full set of allowed responses. Any potential impact is limited to bulk OUT and control endpoints in high-speed devices.

Change: p. 213, section 8.5.1, 4th paragraph, last sentence: change “… a NAK or an ACK handshake.” to “…. a NAK, STALL or an ACK handshake.”

The host controller queries the high-speed device endpoint with a PING special token. The PING special token packet is a normal token packet as shown in Figure 8-5. The endpoint either responds to the PING with a NAK, STALL, or ~~an~~ ACK handshake.

Change: p. 215, section 8.5.1.1, 4th paragraph, change “…. the allowed ACK, NAK or an NYET handshakes for the PING mechanism.” to “…. the allowed ACK, NAK, STALL or an NYET handshakes for the PING mechanism.”

Figure 8-27 shows the host controller state machine for the interactions and transitions between PING and OUT/DATA tokens and the allowed ACK, NAK, STALL, or ~~an~~ NYET handshakes for the PING mechanism.

Clarifying PING Related OUT NAK Responses:

Background: The PING protocol tries to limit NAK responses to OUT transactions. However, a description in the spec suggests that NAK responses to PING transactions are unusual, (which it not true). This change should cause no functional impact since the other descriptions of PING more clearly identify that PING can be NAK’d “forever”. Any potential impact is limited to bulk OUT and control endpoints in high-speed devices.

Change: p. 214, section 8.5.1.1, 2nd paragraph, 1st sentence: change “A NAK response is expected to be an unusual occurrence.” to “A NAK response to an OUT is expected to be an unusual occurrence.”
p. 214, section 8.5.1.1, 2nd paragraph, 3rd sentence: change “…allowed to NAK…” to “… allowed to NAK an OUT…”

A NAK response to an OUT is expected to be an unusual occurrence. A high-speed bulk/control endpoint must specify its maximum NAK rate in its endpoint descriptor. The endpoint is allowed to NAK an OUT at most one time each bInterval period. A NAK suggests that the endpoint responded to a previous OUT or PING with an inappropriate handshake, or that the endpoint transitioned into a state where it (temporarily) could not accept data. An endpoint can use a bInterval of zero to indicate that it never NAKs. An endpoint must always be able to accept a PING from the host, even if it never NAKs.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Incorrect Figure 8-27 (Host High-speed Bulk OUT/Control Ping State Machine):

Background: The Host High-speed Bulk OUT/Control Ping State Machine figure was incorrectly cut and pasted from the draft specification (where it was correct). The current (incorrect) figure is a total duplicate of figure 8-32. Any potential impact is limited to bulk OUT and control endpoints in high-speed devices.

Change: p. 214: Replace incorrect Figure 8-27 with below:

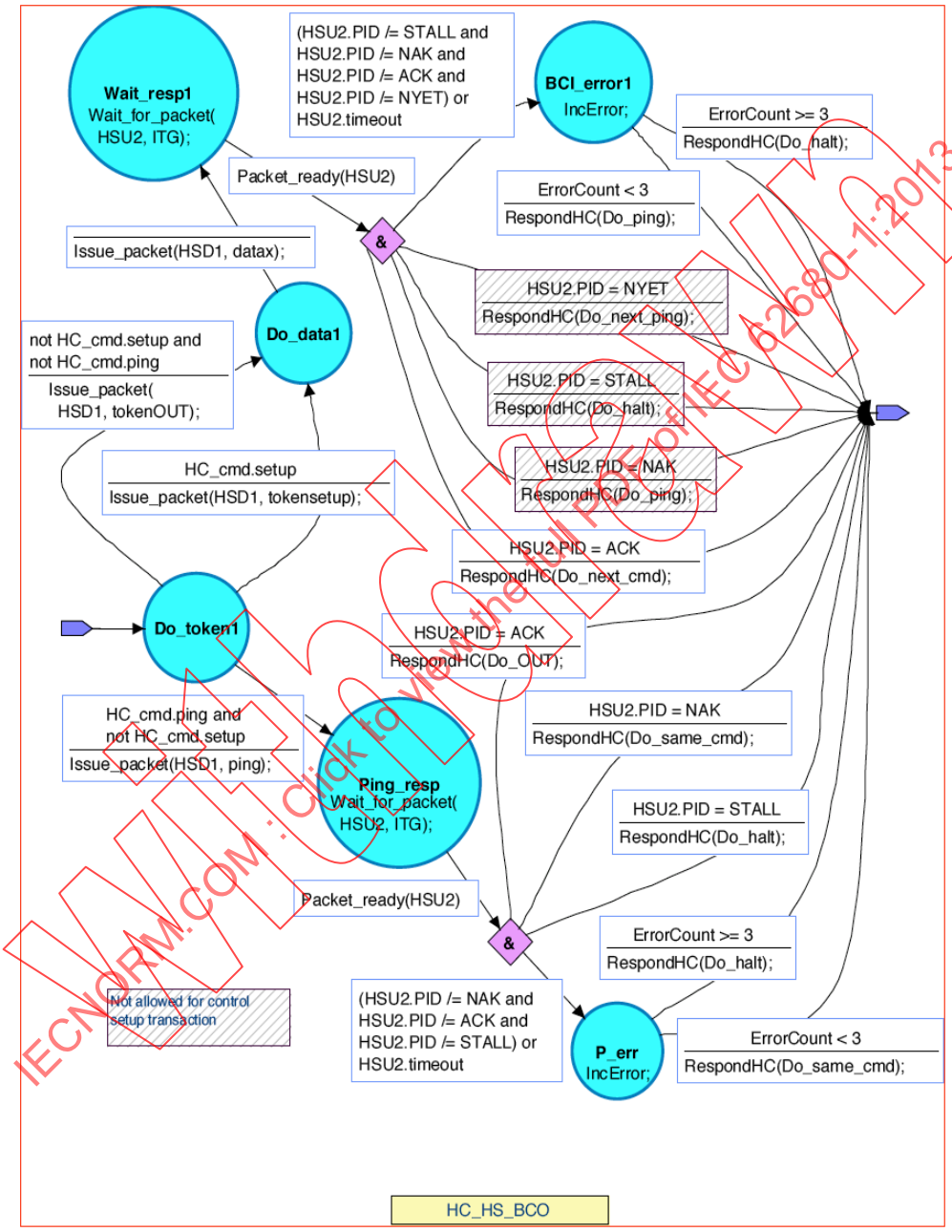


Figure 8-27. Host High-speed Bulk OUT/Control Ping State Machine

Clarifying Figure Title for Figure 8-31 & 8-32:

Background: The titles on figures 8-31 and 8-32 are misleading. The current titles suggest that the figure is uniformly applicable for all speeds, bulk/control/interrupt OUT transactions, (which is not true). No functional impact.

Change: p.218, figure 8-31: caption should be “FS Bulk, FS/LS Control, or HS/FS/LS Interrupt OUT Transaction Host State Machine”

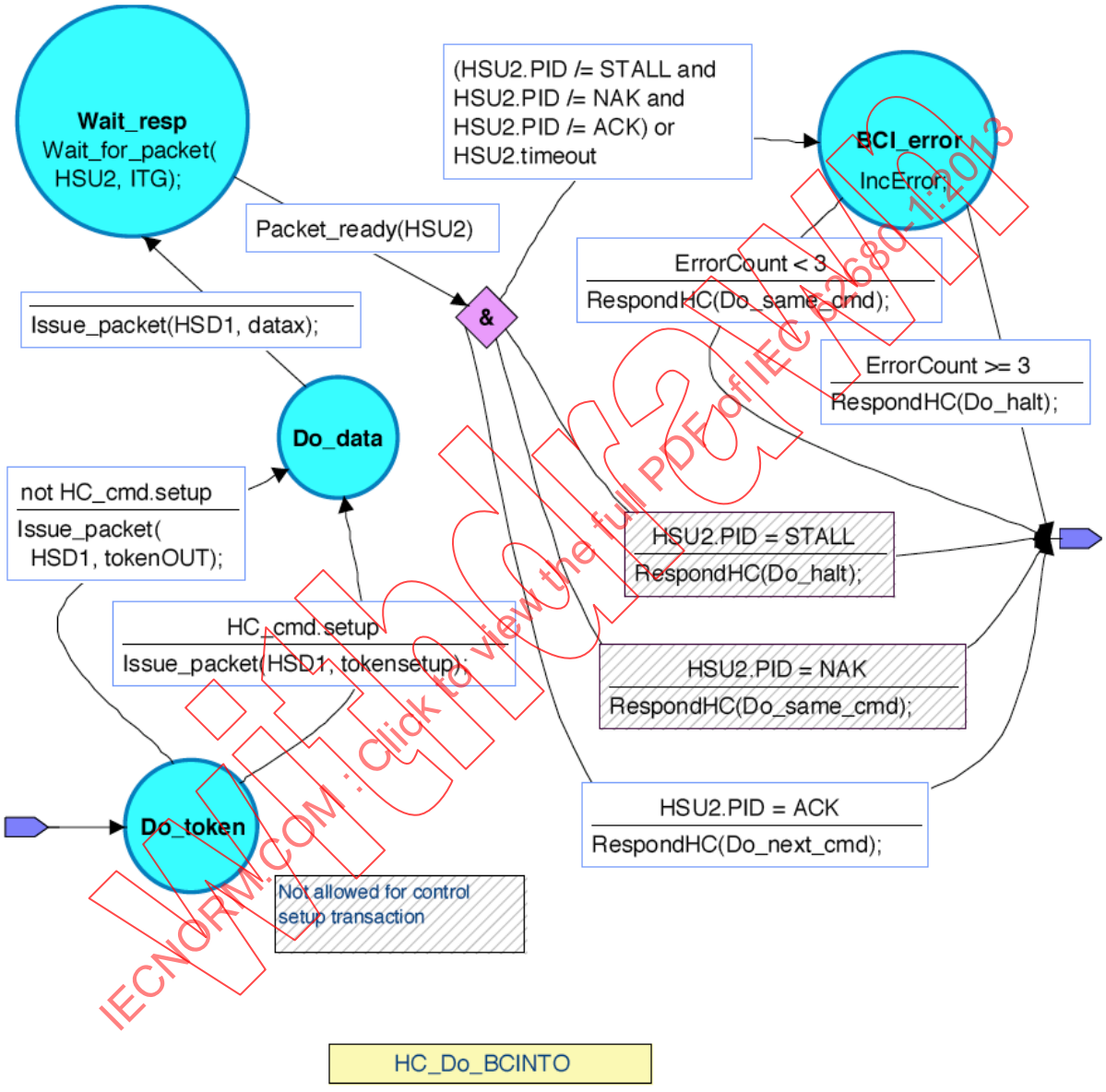


Figure 8-31. FS Bulk, FS/LS~~/~~ Control, or HS/FS/LS Interrupt~~/~~ OUT Transaction Host State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Change:p.219, figure 8-32: caption should be “FS Bulk, FS/LS Control, or HS/FS/LS Interrupt OUT Transaction Device State Machine”.

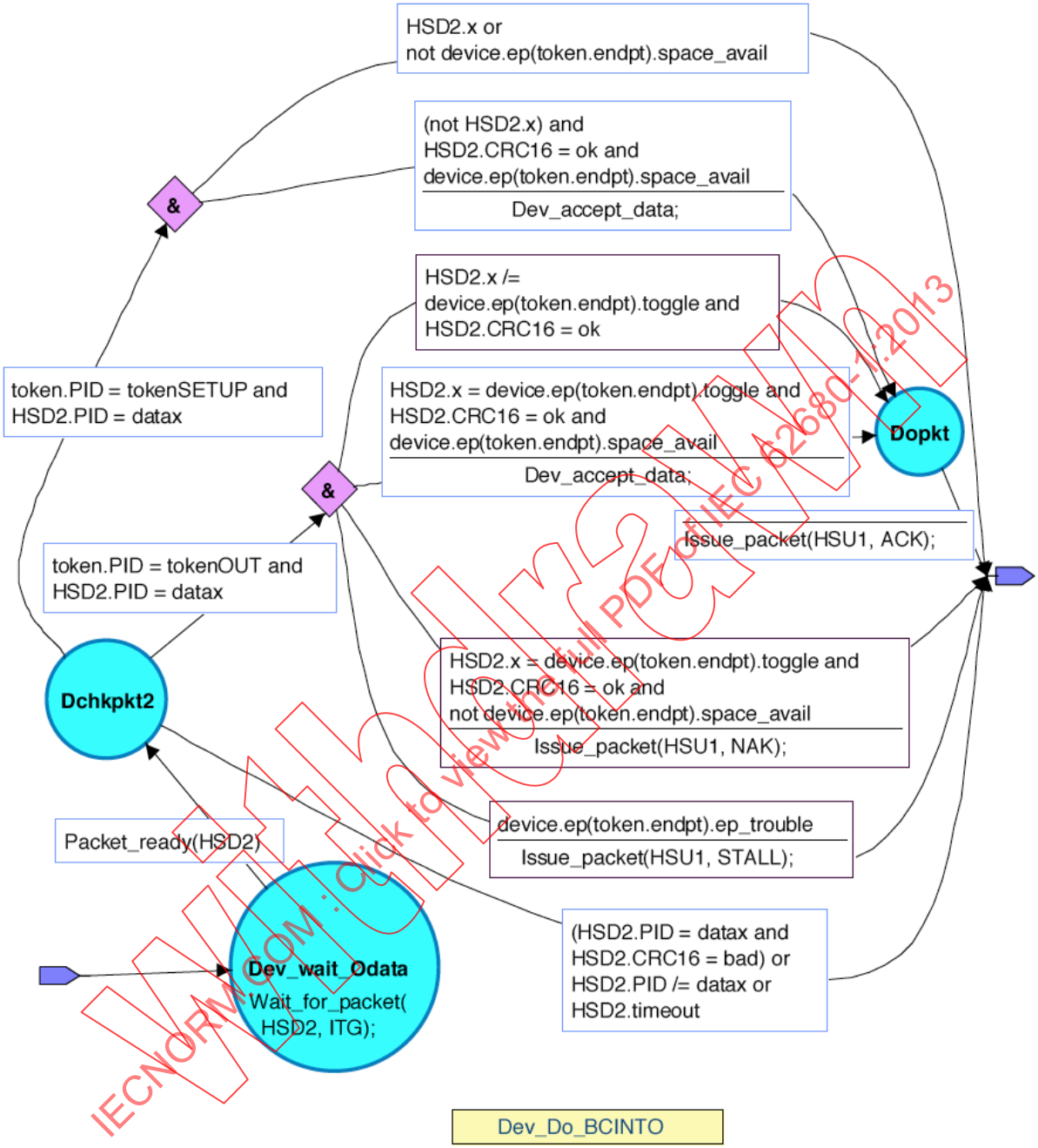


Figure 8-32. FS Bulk, ~~/~~FS/LS Control, ~~/~~or HS/FS/LS Interrupt OUT Transaction Host State Machine

Clarifying No Data PID Sequencing for Interrupt:

Background: Typographical error in the description of high-speed, high-bandwidth endpoints suggests that data PID sequencing is used for both isochronous and interrupt endpoints. Data PID sequencing is only applicable to isochronous endpoints. Also a reference to chapter 5 is incorrect. No functional impact.

Change: p. 230, last paragraph, first sentence: strike "… and interrupt…".
2nd sentence change reference from Section 5.9.1 to 5.9.2.

High-speed, high-bandwidth isochronous ~~and interrupt~~ endpoints support a similar but different data synchronization technique called data PID sequencing. That technique is used instead of data toggle synchronization. Section 5.9.~~1~~2 defines data PID sequencing.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Chapter 9

Background: The table summarizing the standard device requests doesn't reflect the test selector value for SET_FEATURE. This value is correctly described in other places in the spec. Also, the tabular representation of the test selector in the SET_FEATURE definition has the subcolumns reversed (incorrectly) while the text precisely (correctly) defines the organization. No functional impact.

Change: p. 248, table 9-3, row for "SET_FEATURE": change column for wIndex into two subcolumns; left sub-column has current text, new right sub-column has "Test Selector" to correspond with Test_mode support.

Table 9-3. Standard Device Requests

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B 00000001B 00000010B	CLEAR_FEATURE	Feature Selector	Zero Interface Endpoint	Zero	None
10000000B	GET_CONFIGURATION	Zero	Zero	One	Configuration Value
10000000B	GET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor
10000001B	GET_INTERFACE	Zero	Interface	One	Alternate Interface
10000000B 10000001B 10000010B	GET_STATUS	Zero	Zero Interface Endpoint	Two	Device, Interface, or Endpoint Status
00000000B	SET_ADDRESS	Device Address	Zero	Zero	None
00000000B	SET_CONFIGURATION	Configuration Value	Zero	Zero	None
00000000B	SET_DESCRIPTOR	Descriptor Type and Descriptor Index	Zero or Language ID	Descriptor Length	Descriptor
00000000B 00000001B 00000010B	SET_FEATURE	Feature Selector	Zero Interface Endpoint	Test Selector	None
00000001B	SET_INTERFACE	Alternate Setting	Interface	Zero	None
10000010B	SYNCH_FRAME	Zero	Endpoint	Two	Frame Number

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Change: p. 254, Section 9.4.9: the figure field wIndex reverses the “Test Selector” and “Zero Interface Endpoint” columns to more accurately reflect the order of bitfields described in the text.

9.4.9 Set Feature

This request is used to set or enable a specific feature.

bmRequestType	bRequest	wValue	wIndex		wLength	Data
00000000B 00000001B 00000010B	SET_FEATURE	Feature Selector	Zero Interface Endpoint ~~Test Selector~~	Test Selector~~Zero Interface Endpoint~~	Zero	None

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Chapter 11

Incorrect Cross Reference:

Background: Wrong cross reference was made for full speed microframe timer ranges. No functional impact.

Change: p. 293, section 11.2.1, 2nd paragraph, last sentence: Replace reference to "Table 11-2" with reference to "Table 11-1".

11.2.1 High-speed Microframe Timer Range

The range for a microframe timer must be from 59904 to 60096 high-speed bits. The nominal microframe interval is 60000 high-speed bit times. The hub microframe timer range specified above is 60000 +/- 96 high-speed bit times in order to accommodate host accuracy, hub accuracy, repeater jittter, and hub quantization. The +/-96 full-speed bit time variation is calculated in ~~Table 11-2~~Table 11-1.

Typographical Error in Full Speed EOF1/EOF Timing Reference:

Background: A reference to EOF1 was incorrectly used instead of EOF2 in describing the EOF points of full speed frames. Other figures clearly show the required relationships of EOF1 and EOF2. No functional impact.

Change: p. 298, section 11.2.5.2, first sentence: Replace text "..., the EOF1 point is 10 bit times before EOF and EOF1 is ..." with "... the EOF2 point is 10 bit times before EOF and EOF1 is ...".

11.2.5.2 Full-speed EOF1 and EOF2 Timing Points

When the hub operates as a full-/low-speed repeater, the EOF~~1~~2 point is 10 bit times before EOF and EOF1 is 32 bit times before EOF as shown in Figure 11-8.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Illegally Specified bInterval Values for Hub Class Descriptors:

Background: In section 11.23.1, the hub descriptors have several typographical errors. These errors were introduced when the descriptors were copied from the USB 1.1 specification. Most of the errors have an illegal USB2.0 value (FFH) for the bInterval of the hub polling period. This was the correct value for USB1.x hubs. One descriptor incorrectly repeats the same bAlternateSetting value for two different AlternateSettings. The changes allow both previous USB1.1 allowed values and compliant USB2.0 values to avoid silicon changes to hub designs in progress. The change only impacts USB2.0 hubs.

Change: p. 397, in the other_speed endpoint descriptor for the status change endpoint: the bInterval value should be 0CH (i.e. $2^{(12-1)}$ or 256ms). The current value of FFH in the specification is a holdover from the USB 1.1 for full-speed and is an illegal value for high-speed. Similarly, the endpoint descriptors for the status change endpoints on pages 398, 399, 401 and 402 should be changed to 0CH. The expectation is that compliance testing will accept values in the range of 0CH to 10H (12-16) or FFH for some time with an eventual convergence on the value of 0CH for long term compliance.
p. 398 in the hub descriptor other speed interface descriptor for multiple TT hub case (i.e. 2nd interface descriptor on page): the bAlternateSetting should be set to 1 (not 0).

11.23.1 Standard Descriptors for Hub Class

The hub class pre-defines certain fields in standard USB descriptors. Other fields are either implementation-dependent or not applicable to this class.

A hub returns different descriptors based on whether it is operating at high-speed or full-/low-speed. A hub can report three different sets of the descriptors: one descriptor set for full-/low-speed operation and two sets for high-speed operation.

A hub operating at full-/low-speed has a device descriptor with a bDeviceProtocol field set to zero(0) and an interface descriptor with a bInterfaceProtocol field set to zero(0). The rest of the descriptors are the same for all speeds.

A hub operating at high-speed can have one of two TT organizations: single TT or multiple TT. All hubs must support the single TT organization. A multiple TT hub has an additional interface descriptor (with a corresponding endpoint descriptor). The first set of descriptors shown below must be provided by all hubs. A hub that has a single TT must set the bDeviceProtocol field of the device descriptor to one(1) and the interface descriptor bInterfaceProtocol field set to 0.

A multiple TT hub must set the bDeviceProtocol field of the device descriptor to two (2). The first interface descriptor has the bInterfaceProtocol field set to one(1). Such a hub also has a second interface descriptor where the bInterfaceProtocol is set to two(2). When the hub is configured with an interface protocol of one(1), it will operate as a single TT organized hub. When the hub is configured with an interface protocol of two(2), it will operate as a multiple TT organized hub. The TT organization must not be changed while the hub has full-/low-speed transactions in progress.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Note: For the descriptors and fields shown below, the bits in a field are organized in a little-endian fashion; that is, bit location 0 is the least significant bit and bit location 7 is the most significant bit of a byte value.

Full-/Low-speed Operating Hub

Device Descriptor (full-speed information):

bLength	12H
bDescriptorType	1
bcdUSB	0200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
bNumConfigurations	1

Device_Qualifier Descriptor (high-speed information):

bLength	0AH
bDescriptorType	6
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	1 (for single TT) or 2 (for multiple TT)
bMaxPacketSize0	64
bNumConfigurations	1

Configuration Descriptor (full-speed information):

bLength	09H
bDescriptorType	2
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in full-/low-speed configuration

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

Other_Speed_Configuration Descriptor (High-speed information):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1 (for single TT) or 2 (for multiple TT)
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in high-speed configuration

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0 (for single TT) 1 (for multiple TT)
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	0CH~~FFH~~ (Maximum allowable interval)

Interface Descriptor (present if multiple TT hub):

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	~~0~~1
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	2
iInterface	i

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Endpoint Descriptor (present if multiple TT hub):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	0CH~~FFH~~ (Maximum allowable interval)

High-speed Operating Hub with Single TT

Device Descriptor (High-speed information):

bLength	12H
bDescriptorType	1
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	1
bMaxPacketSize0	64
bNumConfigurations	1

Device_Qualifier Descriptor (full-speed information):

bLength	0AH
bDescriptorType	6
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
bNumConfigurations	1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Configuration Descriptor (high-speed information):

bLength	09H
bDescriptorType	2
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in this configuration

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0 (single TT)
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	0CH~~FFH~~ (Maximum allowable interval)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Other_Speed_Configuration Descriptor (full-speed information):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in high-speed configuration

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

High-speed Operating Hub with Multiple TTs

Device Descriptor (High-speed information):

bLength	12H
bDescriptorType	1
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	2 (multiple TTs)
bMaxPacketSize0	64
bNumConfigurations	1

Device_Qualifier Descriptor (full-speed information):

bLength	0AH
bDescriptorType	6
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
bNumConfigurations	1

Configuration Descriptor (high-speed information):

bLength	09H
bDescriptorType	2
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in this configuration

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	1 (single TT)
iInterface	i

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	0CH~~FFH~~ (Maximum allowable interval)

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	1
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	2 (multiple TTs)
iInterface	i

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Endpoint Descriptor:

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	0CH~~FFH~~ (Maximum allowable interval)

Other_Speed_Configuration Descriptor (full-speed information):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in high-speed configuration

Interface Descriptor:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	i

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Endpoint Descriptor (for Status Change Endpoint):

bLength	07H
bDescriptorType	5
bEndpointAddress	Implementation-dependent; Bit 7: Direction = In(1)
bmAttributes	Transfer Type = Interrupt (00000011B)
wMaxPacketSize	Implementation-dependent
bInterval	FFH (Maximum allowable interval)

The hub class driver retrieves a device configuration from the USB System Software using the GetDescriptor() device request. The only endpoint descriptor that is returned by the GetDescriptor() request is the Status Change endpoint descriptor.

Inconsistency in ClearPortFeature Descriptions:

Background: The description of ClearPortFeature is inconsistent in some places in the spec. The description of the PORT_INDICATOR in section 11.24.2.7.1.10 is correct, but the state machine figure (11-11) omits a label for ClearPortFeature and the description of the ClearPortFeature request incorrectly specifies an explicit indicator value, where there is no value that would be meaningful. Also, the PORT_INDICATOR is not explicitly listed for SetPortFeature as a case for having the MSB of wIndex set to zero. This change only impacts hubs that provide port indicator control.

Change: p. 308, figure 11-11: Add "or ClearPortFeature(PORT_INDICATOR, indicator_selector = n/a)" to the label on the transition from Manual mode to Automatic mode.

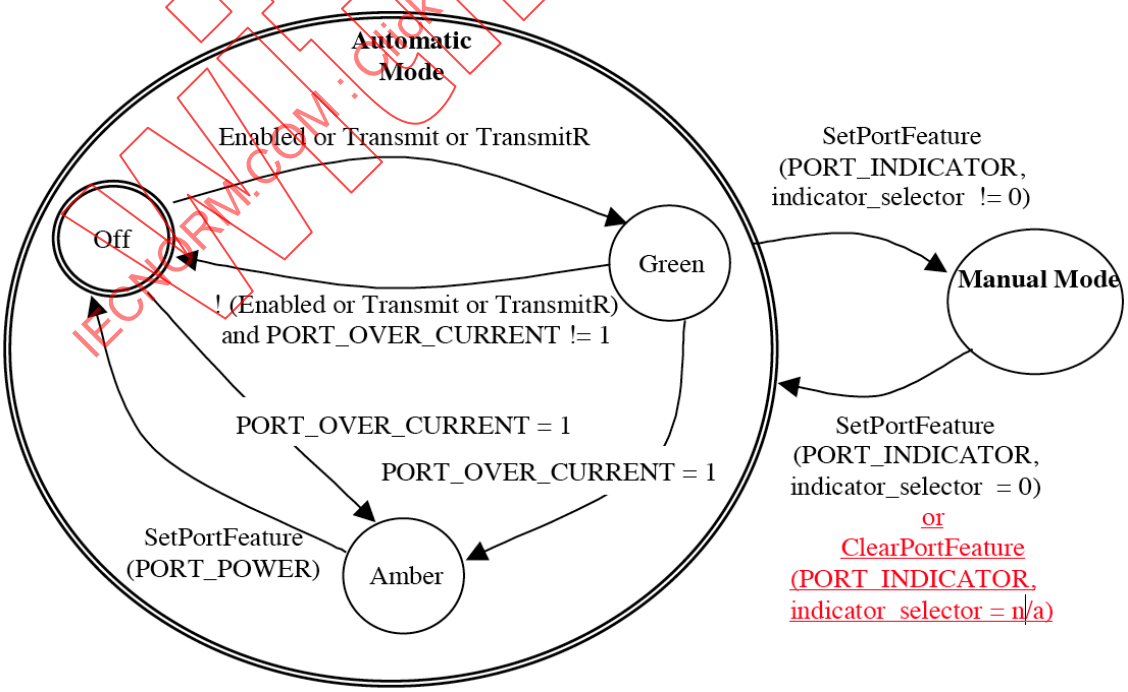


Figure11-11. Port Indicator State Diagram

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Change: p. 407, 3rd paragraph before section 11.24.2.3: replace the sentence beginning “The selector field…” with the sentences “Clearing the PORT_INDICATOR causes a transition of the port indicator state machine back to automatic mode (see Figure 11-11). The indicator_selector field is ignored for ClearPortFeature.”

Clearing the PORT_ENABLE feature causes the port to be placed in the Disabled state. If the port is in the Powered-off state, the hub should treat this request as a functional no-operation.

Clearing the PORT_POWER feature causes the port to be placed in the Powered-off state and may, subject to the constraints due to the hub’s method of power switching, result in power being removed from the port. Refer to Section 11.11 on rules for how this request is used with ports that are gang-powered.

~~The selector field identifies the port indicator selector when clearing a port indicator.~~ Clearing the PORT_INDICATOR causes a transition of the port indicator state machine back to automatic mode (see Figure 11-11). The indicator selector field is ignored for the ClearPortFeature. The selector field is in bits 15..8 of the *wIndex* field.

It is a Request Error if wValue is not a feature selector listed in Table 11-17, if wIndex specifies a port that does not exist, or if wLength is not as specified above. It is not an error for this request to try to clear a feature that is already cleared (hub should treat as a functional no-operation). If the hub is not configured, the hub's response to this request is undefined.

If the hub is not configured, the hub's response to this request is undefined.

Change: p. 418, 1st paragraph: add “… or PORT_INDICATOR.” to end of the last sentence.

11.24.2.13 Set Port Feature

This request sets a value reported in the port status.

bmRequestType	bRequest	wValue	wIndex		wLength	Data
00100011B	SET_FEATURE	Feature Selector	Selector	Port	Zero	None

The port number must be a valid port number for that hub, greater than zero. The port number is in the least significant byte (bits 7..0) of the *wIndex* field. The most significant byte of *wIndex* is zero, except when the feature selector is PORT_TEST or PORT_INDICATOR.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Clarifying Standard Hub Class Requests:

Background: GET_/SET_INTERFACE requests to the hub in the summary hub class request table do not currently indicate that they are required for multiple TT hub implementations. Since the hub descriptors report alternate interfaces and other descriptions indicate how these interfaces are used, these commands are clearly required.
This change only impacts hubs that support multiple TTs.

Change: p. 404, table 11-14: change Hub Response descriptions for GET_INTERFACE and SET_INTERFACE from “Undefined….” to “Standard for multiple TT hub operating at highspeed.”

Table 11-14. Hub Responses to Standard Device Requests

bRequest	Hub Response
CLEAR_FEATURE	Standard
GET_CONFIGURATION	Standard
GET_DESCRIPTOR	Standard
GET_INTERFACE	~~Undefined. Hubs are allowed to support only one interface.~~ Standard for multiple TT hub operating at highspeed.
GET_STATUS	Standard
SET_ADDRESS	Standard
SET_CONFIGURATION	Standard
SET_DESCRIPTOR	Optional
SET_FEATURE	Standard
SET_INTERFACE	~~Undefined. Hubs are allowed to support only one interface.~~ Standard for multiple TT hub operating at highspeed.
SYNCH_FRAME	Undefined. Hubs are not allowed to have isochronous endpoints.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Minor Typographical Errors:

Background: No functional changes.

Change: p. 410, section 11.24.2.7, 2nd paragraph after figure: change “refer to Table 11-20” to “refer to Table 11-22”

11.24.2.7 Get Port Status

This request returns the current port status and the current value of the port status change bits.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10100011B	GET_STATUS	Zero	Port	Four	Port Status and Change Status

The port number must be a valid port number for that hub, greater than zero.

The first word of data contains wPortStatus (refer to Table 11-21). The second word of data contains wPortChange (refer to ~~Table 11-20~~Table 11-22).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Change: p. 410 table 11-21, row for bit 10: change name from “High-speed Device Attached” to “Full- or High-speed Device Attached”.

Table 11-21. Port Status Field, wPortStatus

Bit	Description
0	**Current Connect Status:** (port_connection) This field reflects whether or not a device is currently connected to this port. 0 = No device is present. 1 = A device is present on this port.
1	**Port Enabled/Disabled:** (port_enable) Ports can be enabled by the USB System Software only. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by the USB System Software. 0 = Port is disabled. 1 = Port is enabled.
2	**Suspend:** (port_suspend) This field indicates whether or not the device on this port is suspended. Setting this field causes the device to suspend by not propagating bus traffic downstream. This field may be reset by a request or by resume signaling from the device attached to the port. 0 = Not suspended. 1 = Suspended or resuming.
3	**Over-current:** (port_over_current) If the hub reports over-current conditions on a per-port basis, this field will indicate that the current drain on the port exceeds the specified maximum. For more details, see Section 7.2.1.2.1. 0 = All no over-current condition exists on this port. 1 = An over-current condition exists on this port.
4	**Reset**: (port_reset) This field is set when the host wishes to reset the attached device. It remains set until the reset signaling is turned off by the hub. 0 = Reset signaling not asserted. 1 = Reset signaling asserted.
5-7	**Reserved** These bits return 0 when read.
8	**Port Power:** (port_power) This field reflects a port's logical, power control state. Because hubs can implement different methods of port power switching, this field may or may not represent whether power is applied to the port. The device descriptor reports the type of power switching implemented by the hub. 0 = This port is in the Powered-off state. 1 = This port is not in the Powered-off state.
9	**Low- Speed Device Attached:** (port_low_speed) This is relevant only if a device is attached. 0 = Full-speed or High-speed device attached to this port (determined by bit 10). 1 = Low-speed device attached to this port.
10	**Full- or High-speed Device Attached:** (PORT_HIGH_SPEED) This is relevant only if a device is attached. 0 = Full-speed device attached to this port. 1 = High-speed device attached to this port.
11	**Port Test Mode**: (PORT_TEST) This field reflects the status of the port's test mode. Software uses the SetPortFeature() and ClearPortFeature() requests to manipulate the port test mode. 0 = This port is not in the Port Test Mode. 1 = This port is in Port Test Mode.
12	**Port Indicator Control**: (PORT_INDICATOR) This field is set to reflect software control of the port indicator. For more details see Sections 11.5.3, 11.24.2.7.1.10, and 11.24.2.13. 0 = Port indicator displays default colors. 1 = Port indicator displays software controlled color.
13-15	**Reserved** These bits return 0 when read.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

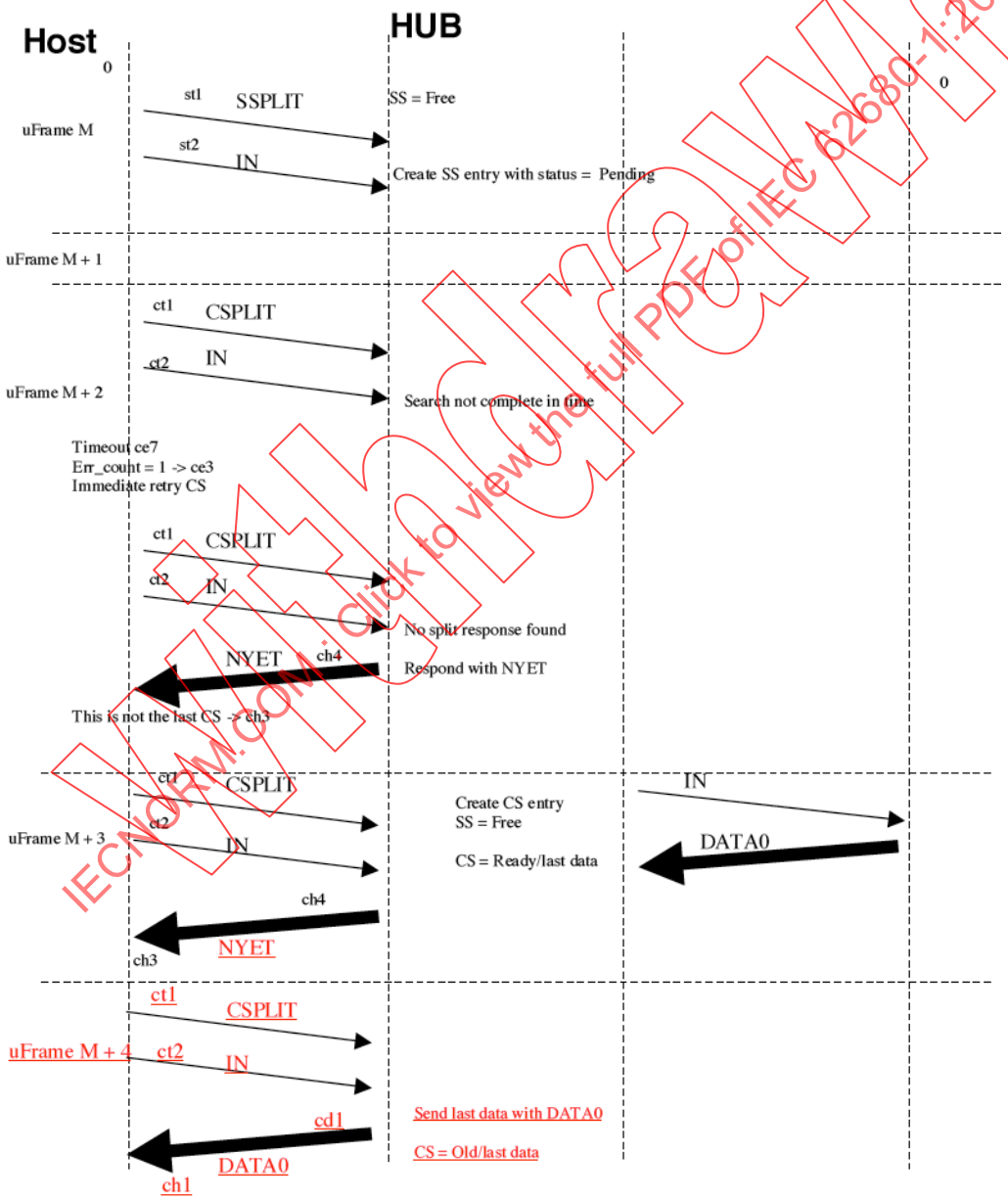
Chapter Appendix A

Missing Information in Example Split Transactions:

Background: One of the split transaction examples is missing some labels and a final complete split transaction. No functional impact.

Change: p. 531, section A.6, example 12 “HS CS too early…”: Add “NYET” label on final arrow from Hub to Host. Add new uframe M+4 with HS transaction arrows: 1) host to hub: “CSPLIT” and “IN” arrows, 2) hub to host “data” arrow.

12) HS CS too early (full-speed data not available yet)



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

USB 2.0 Specification Engineering Change Notice (ECN) #1: Mini-B connector

Date: 10/20/2000

Reason for ECN:
The USB 2.0 specified device-side connector – the B connector – is too large for use with a new generation of handheld and mobile devices, e.g., cell phones which would benefit from connectivity to the PC. This ECN incorporates a specification of a device-side mini connector (hereafter referred to as a mini-B connector). The new connector only applies to upstream facing ports, i.e., connectors on devices.

Summary of ECN:
The bulk of the ECN specifies the mechanical requirements for the mini-B plug, receptacle and cable assembly. It also identifies the usage scope of this connector. The last part of the ECN describes the minimum test criteria and performance requirements for the new connector.

Benefits of ECN:
The ECN enables standardization of miniature device-side USB connectors and consequent economies of scales and lower cost for a new and growing class of devices which will benefit from connectivity to the PC. The standardization also allows leveraging of the compliance test and certification model which is already in place for USB connectors.

Assessment of Impact on Current Specification and Current USB Products:
The connector specified in the ECN will not have any impact on hardware or software of existing USB products. The current USB spec already allows for vendor-specific device side connectors – such cable assemblies are called captive assemblies. All that the ECN does is to identify one such connector for use in devices which need the smaller size of connector. There is a potential for some end-user confusion because of two standard cable options; but this can be mitigated by appropriate end-user education.

Structure of ECN:
The ECN is in the form of a new Chapter 6 with the mini-B connector requirements inserted into the appropriate locations. This format enables specification of the new connector in context.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

6 Chapter 6 Mechanical

This chapter provides the mechanical and electrical specifications for the cables, connectors, and cable assemblies used to interconnect USB devices. The specification includes the dimensions, materials, electrical, and reliability requirements. This chapter documents minimum requirements for the external USB interconnect. Substitute material may be used as long as it meets these minimums.

6.1 Architectural Overview

The USB physical topology consists of connecting the downstream hub port to the upstream port of another hub or to a device. The USB can operate at three speeds. High-speed (480 Mb/s) and full-speed (12 Mb/s) require the use of a shielded cable with two power conductors and twisted pair signal conductors. Low-speed (1.5 Mb/s) recommends, but does not require the use of a cable with twisted pair signal conductors.

The connectors are designed to be hot plugged. The USB Icon on the plugs provides tactile feedback making it easy to obtain proper orientation.

6.2 Keyed Connector Protocol

To minimize end user termination problems, USB uses a “keyed connector” protocol. The physical difference in the Series “A” and “B” (or “mini-B”) connectors insures proper end user connectivity. The “A” connector is the principle means of connecting USB devices directly to a host or to the downstream port of a hub. All USB devices must have the standard Series “A” connector specified in this chapter. The “B” (or “mini-B”) connector allows device vendors to provide a standard detachable cable. This facilitates end user cable replacement. Figure 6-1 illustrates the keyed connector protocol.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

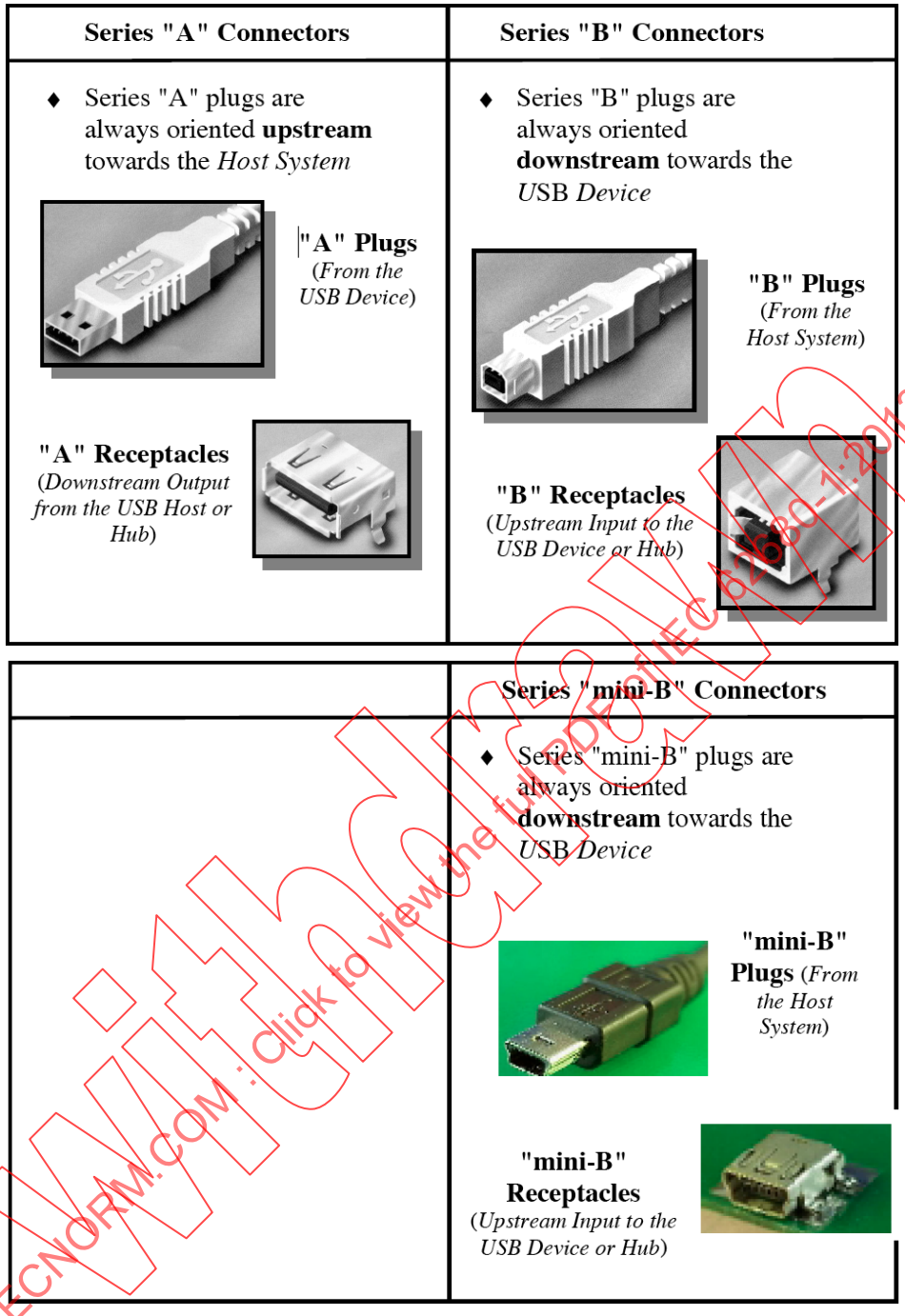


Figure 6-1. Keyed Connector Protocol

The following list explains how the plugs and receptacles can be mated:

- Series “A” receptacle mates with a Series “A” plug. Electrically, Series “A” receptacles function as outputs from host systems and/or hubs.
- Series “A” plug mates with a Series “A” receptacle. The Series “A” plug always is oriented towards the host system.
- Series “B” receptacle mates with a Series “B” plug (male). Electrically, Series “B” receptacles function as inputs to hubs or devices.
- Series “B” plug mates with a Series “B” receptacle. The Series “B” plug is always oriented towards the USB hub or device.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Series “mini-B” receptacle mates with a Series “mini-B” plug (male). Electrically, Series “mini-B” receptacles function as inputs to hubs or devices.
- Series “mini-B” plug mates with a Series “mini-B” receptacle. The Series “mini-B” plug is always oriented towards the USB hub or device.

6.3 Cable

USB cable consists of four conductors, two power conductors, and two signal conductors.

High-/full-speed cable consists of a signaling twisted pair, VBUS, GND, and an overall shield. High-/full-speed cable must be marked to indicate suitability for USB usage (see Section 6.6.2). High-/full-speed cable may be used with either low-speed, full-speed, or high-speed devices. When high-/full-speed cable is used with low-speed devices, the cable must meet all low-speed requirements.

Low-speed recommends, but does not require the use of a cable with twisted signaling conductors.

6.4 Cable Assembly

This specification describes three USB cable assemblies: standard detachable cable, high-/full-speed captive cable, and low-speed captive cable.

A standard detachable cable is a high-/full-speed cable that is terminated on one end with a Series “A” plug and terminated on the opposite end with a series “B” (or “mini-B”) plug. A high-/full-speed captive cable is terminated on one end with a Series “A” plug and has a vendor-specific connect means (hardwired or custom detachable) on the opposite end for the high-/full-speed peripheral. The low-speed captive cable is terminated on one end with a Series “A” plug and has a vendor-specific connect means (hardwired or custom detachable) on the opposite end for the low-speed peripheral. Any other cable assemblies are prohibited.

The color used for the cable assembly is vendor specific; recommended colors are white, grey, or black.

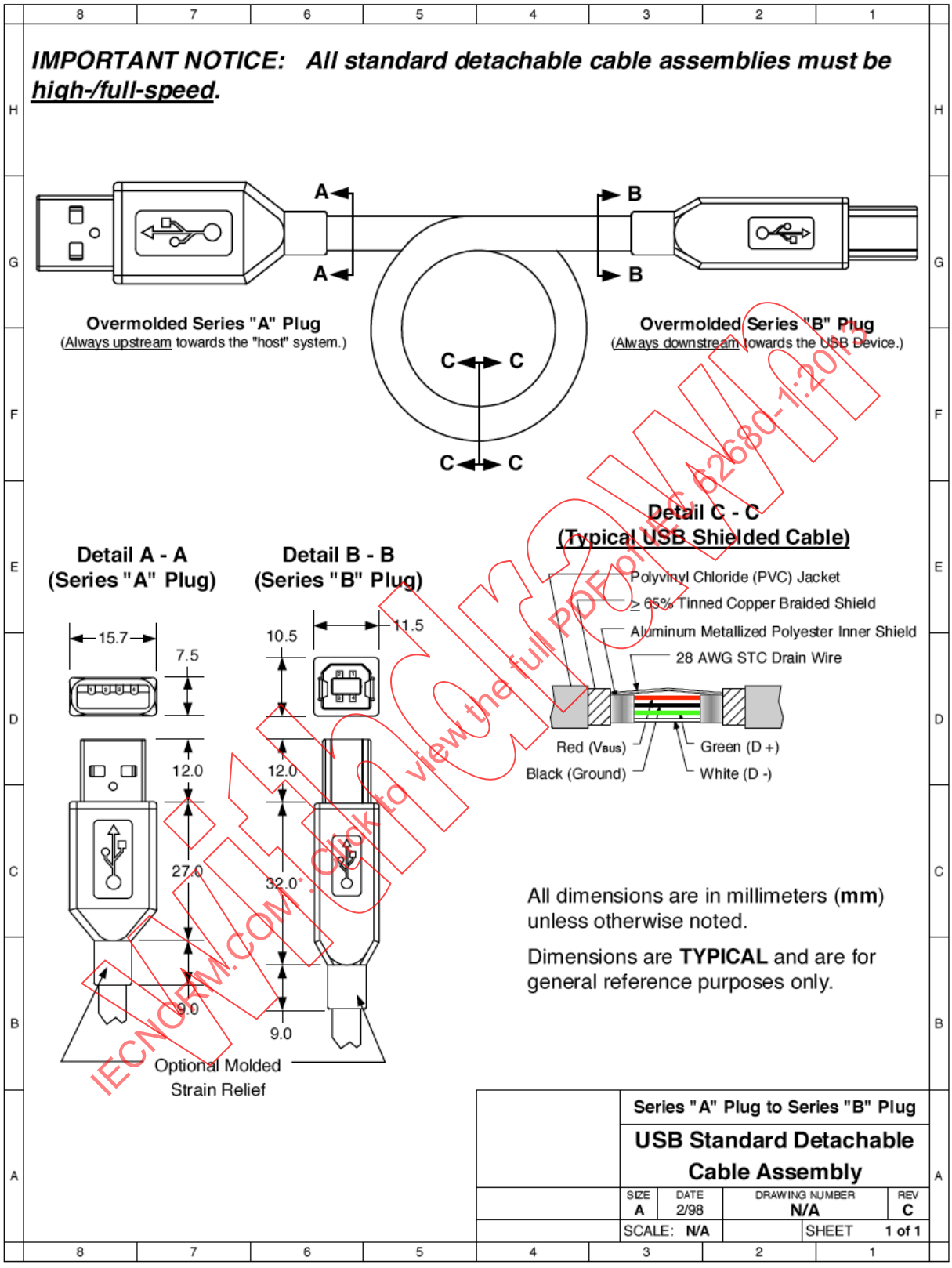
6.4.1 Standard Detachable Cable Assemblies

High-speed and full-speed devices can utilize the “B” (or “mini-B”) connector. This allows the device to have a standard detachable USB cable. This eliminates the need to build the device with a hardwired cable and minimizes end user problems if cable replacement is necessary.

Devices utilizing the “B” (or “mini-B”) connector must be designed to work with worst case maximum length detachable cable. Standard detachable cable assemblies may be used only on high-speed and full-speed devices. Using a high-/full-speed standard detachable cable on a low-speed device may exceed the maximum low-speed cable length.

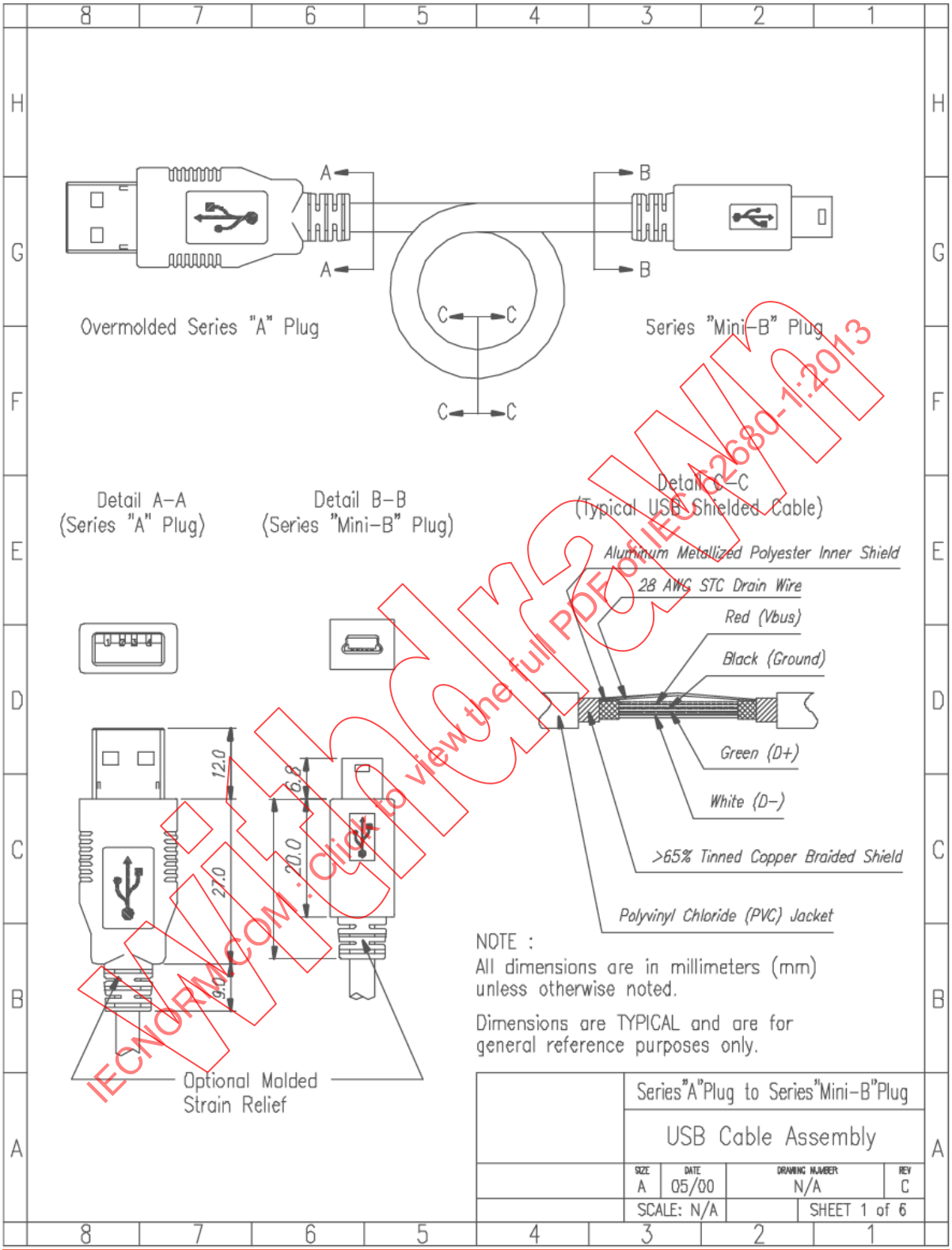
Figure 6-2 and Figure 6-3 illustrates ~~a~~ standard detachable cable assemblies~~y~~.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 6-2. USB Standard Detachable Cable Assembly



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-3. USB Standard Mini-connector Detachable Cable Assembly

Standard detachable cable assemblies must meet the following electrical requirements:

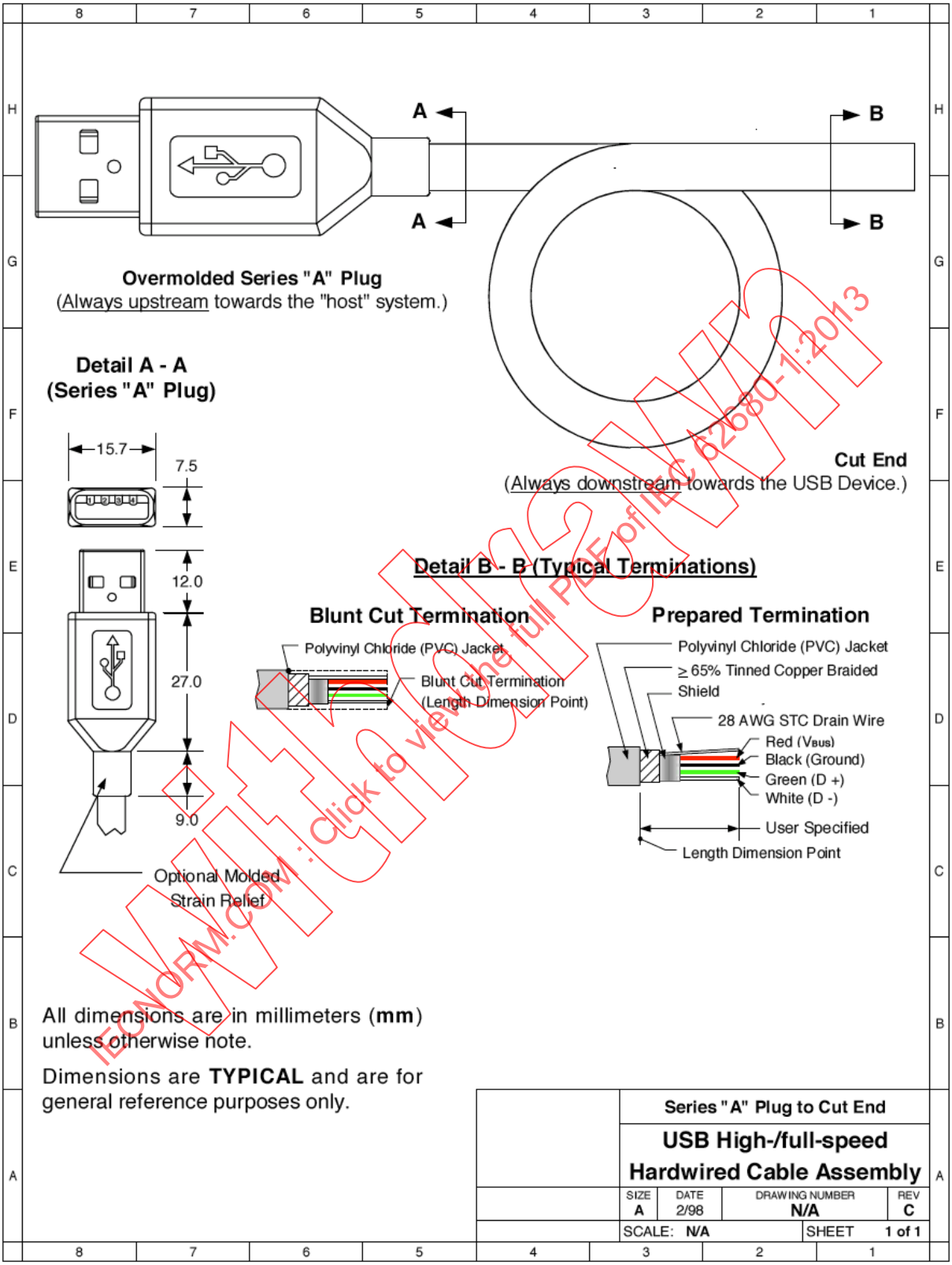
- The cable must be terminated on one end with an overmolded Series “A” plug and the opposite end is terminated with an overmolded Series “B” (or “mini-B”) plug.
- The cable must be rated for high-speed and full-speed.
- The cable impedance must match the impedance of the high-speed and full-speed drivers. The drivers are characterized to drive specific cable impedance. Refer to Section 7.1.1 for details.
- The maximum allowable cable length is determined by signal pair attenuation and propagation delay. Refer to Sections 7.1.14 and 7.1.17 for details.
- Differences in propagation delay between the two signal conductors must be minimized. Refer to Section 7.1.3 for details.
- The GND lead provides a common ground reference between the upstream and downstream ports. The maximum cable length is limited by the voltage drop across the GND lead. Refer to Section 7.2.2 for details. The minimum acceptable wire gauge is calculated assuming the attached device is high power.
- The VBUS lead provides power to the connected device. For standard detachable cables, the VBUS requirement is the same as the GND lead.

6.4.2 High-/full-speed Captive Cable Assemblies

Assemblies are considered captive if they are provided with a vendor-specific connect means (hardwired or custom detachable) to the peripheral. High-/full-speed hardwired cable assemblies may be used with either high-speed, full-speed, or low-speed devices. When using a high-/full-speed hardwired cable on a low-speed device, the cable must meet all low-speed requirements.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

~~Figure 6-3~~Figure 6-4 illustrates a high-/full-speed hardwired cable assembly.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-3. USB High-/full-speed Hardwired Cable Assembly

High-/full-speed captive cable assemblies must meet the following electrical requirements:

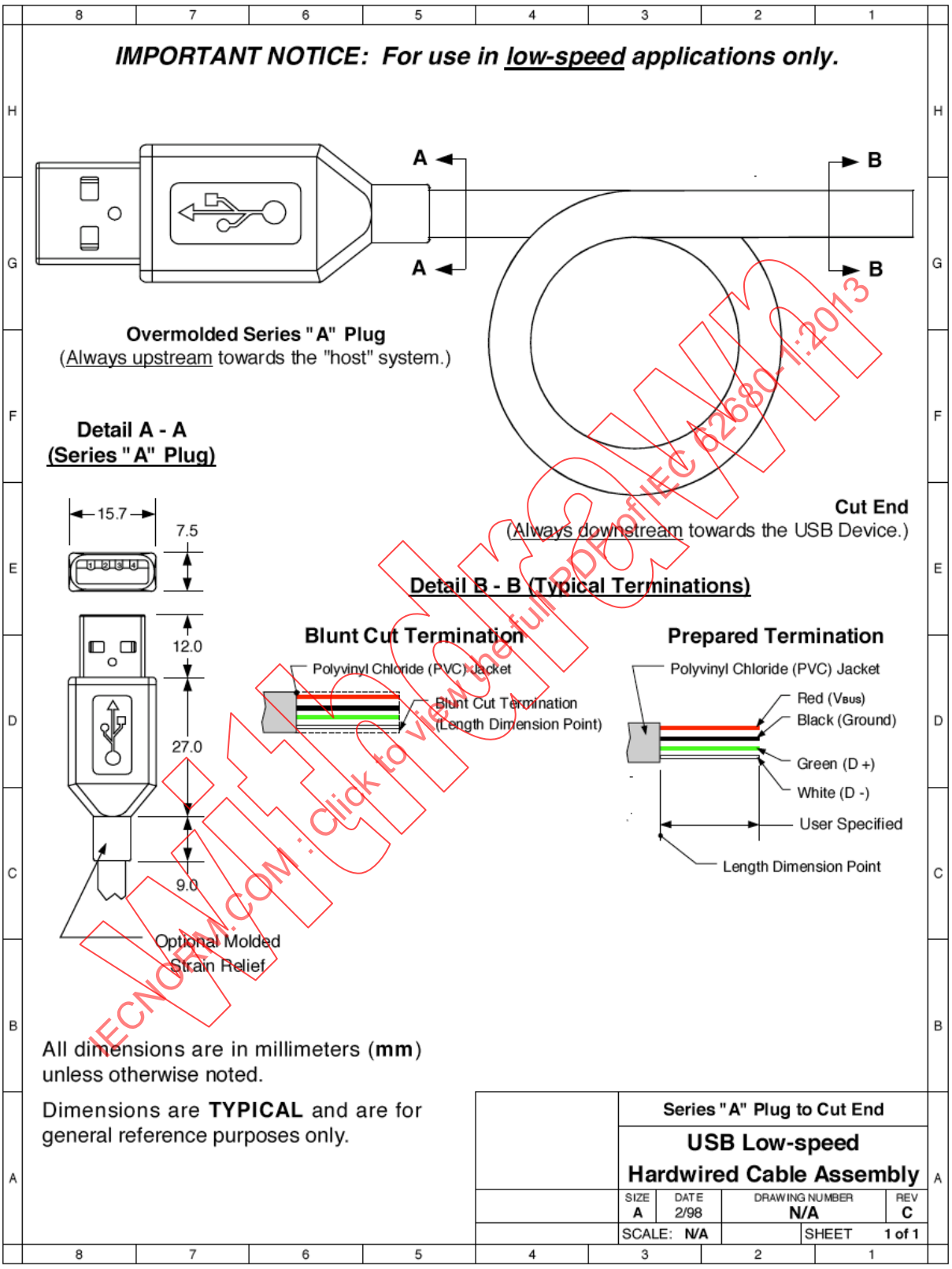
- The cable must be terminated on one end with an overmolded Series “A” plug and the opposite end is vendor specific. If the vendor specific interconnect is to be hot plugged, it must meet the same performance requirements as the USB “B” connector.
- The cable must be rated for high-speed and full-speed.
- The cable impedance must match the impedance of the high-speed and full-speed drivers. The drivers are characterized to drive specific cable impedance. Refer to Section 7.1.1 for details.
- The maximum allowable cable length is determined by signal pair attenuation and propagation delay. Refer to Sections 7.1.14 and 7.1.17 for details.
- Differences in propagation delay between the two signal conductors must be minimized. Refer to Section 7.1.3 for details.
- The GND lead provides a common reference between the upstream and downstream ports. The maximum cable length is determined by the voltage drop across the GND lead. Refer to Section 7.2.2 for details. The minimum wire gauge is calculated using the worst case current consumption.
- The VBUS lead provides power to the connected device. The minimum wire gauge is vendor specific.

6.4.3 Low-speed Captive Cable Assemblies

Assemblies are considered captive if they are provided with a vendor-specific connect means (hardwired or custom detachable) to the peripheral. Low-speed cables may only be used on low-speed devices.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

~~Figure 6-4~~Figure 6-5 illustrates a low-speed hardwired cable assembly.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure ~~6-4~~6-5. USB Low-speed Hardwired Cable Assembly

Low-speed captive cable assemblies must meet the following electrical requirements:

- The cable must be terminated on one end with an overmolded Series “A” plug and the opposite end is vendor specific. If the vendor specific interconnect is to be hot plugged, it must meet the same performance requirements as the USB “B” connector.
- Low-speed drivers are characterized for operation over a range of capacitive loads. This value includes all sources of capacitance on the D+ and D-lines, not just the cable. Cable selection must insure that total load capacitance falls between specified minimum and maximum values. If the desired implementation does not meet the minimum requirement, additional capacitance needs to be added to the device. Refer to Section 7.1.1.2 for details.
- The maximum low-speed cable length is determined by the rise and fall times of low-speed signaling. This forces low-speed cable to be significantly shorter than high-/full-speed. Refer to Section 7.1.1.2 for details.
- Differences in propagation delay between the two signal conductors must be minimized. Refer to Section 7.1.3 for details.
- The GND lead provides a common reference between the upstream and downstream ports. The maximum cable length is determined by the voltage drop across the GND lead. Refer to Section 7.2.2 for details. The minimum wire gauge is calculated using the worst case current consumption.
- The VBUS lead provides power to the connected device. The minimum wire gauge is vendor specific.

6.4.4 Prohibited Cable Assemblies

USB is optimized for ease of use. The expectation is that if the device can be plugged in, it will work. By specification, the only conditions that prevent a USB device from being successfully utilized are lack of power, lack of bandwidth, and excessive topology depth. These conditions are well understood by the system software.

Prohibited cable assemblies may work in some situations, but they cannot be guaranteed to work in all instances.

- **Extension cable assembly**
  A cable assembly that provides a Series “A” plug with a series “A” receptacle or a Series “B” plug with a Series “B” receptacle or a Series “mini-B” plug with a Series “mini-B” receptacle. This allows multiple cable segments to be connected together, possibly exceeding the maximum permissible cable length.
- **Cable assembly that violates USB topology rules**
  A cable assembly with both ends terminated in either Series “A” plugs or Series “B” (or “mini-B”) receptacles. This allows two downstream ports to be directly connected. Note: This prohibition does not prevent using a USB device to provide a bridge between two USB buses.
- **Standard detachable cables for low-speed devices**
  Low-speed devices are prohibited from using standard detachable cables. A standard detachable cable assembly must be high-/full-speed. Since a standard detachable cable assembly is high-/full-speed rated, using a long high-/full-speed cable exceeds the capacitive load of low-speed.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

6.5 Connector Mechanical Configuration and Material Requirements

The USB Icon is used to identify USB plugs and the receptacles. ~~Figure 6-5~~Figure 6-6 illustrates the USB Icon.

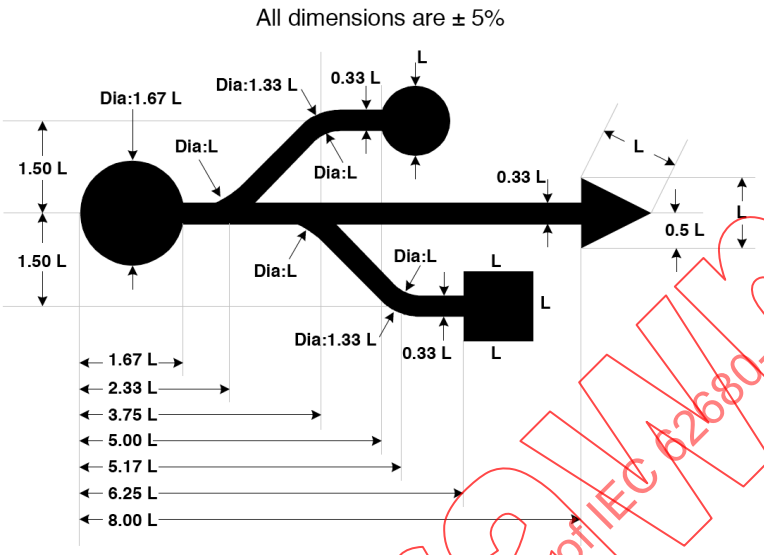


Figure ~~6-5~~6-6. USB Icon

11.24.3 USB Icon Location

The USB Icon is embossed, in a recessed area, on the topside of the USB plug. This provides easy user recognition and facilitates alignment during the mating process. The USB Icon and Manufacturer's logo should not project beyond the overmold surface. The USB Icon is required, while the Manufacturer's logo is recommended, for both Series "A" and "B" (or "mini-B") plug assemblies. The USB Icon is also located adjacent to each receptacle. Receptacles should be oriented to allow the Icon on the plug to be visible during the mating process. Figure ~~6-6~~6-7 and Figure 6-8 illustrate~~s~~ the typical plug orientations.

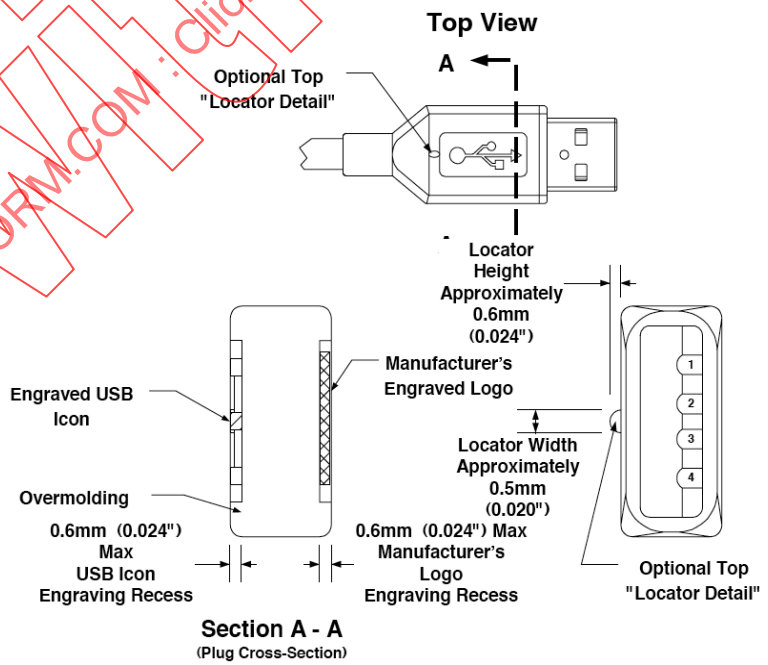
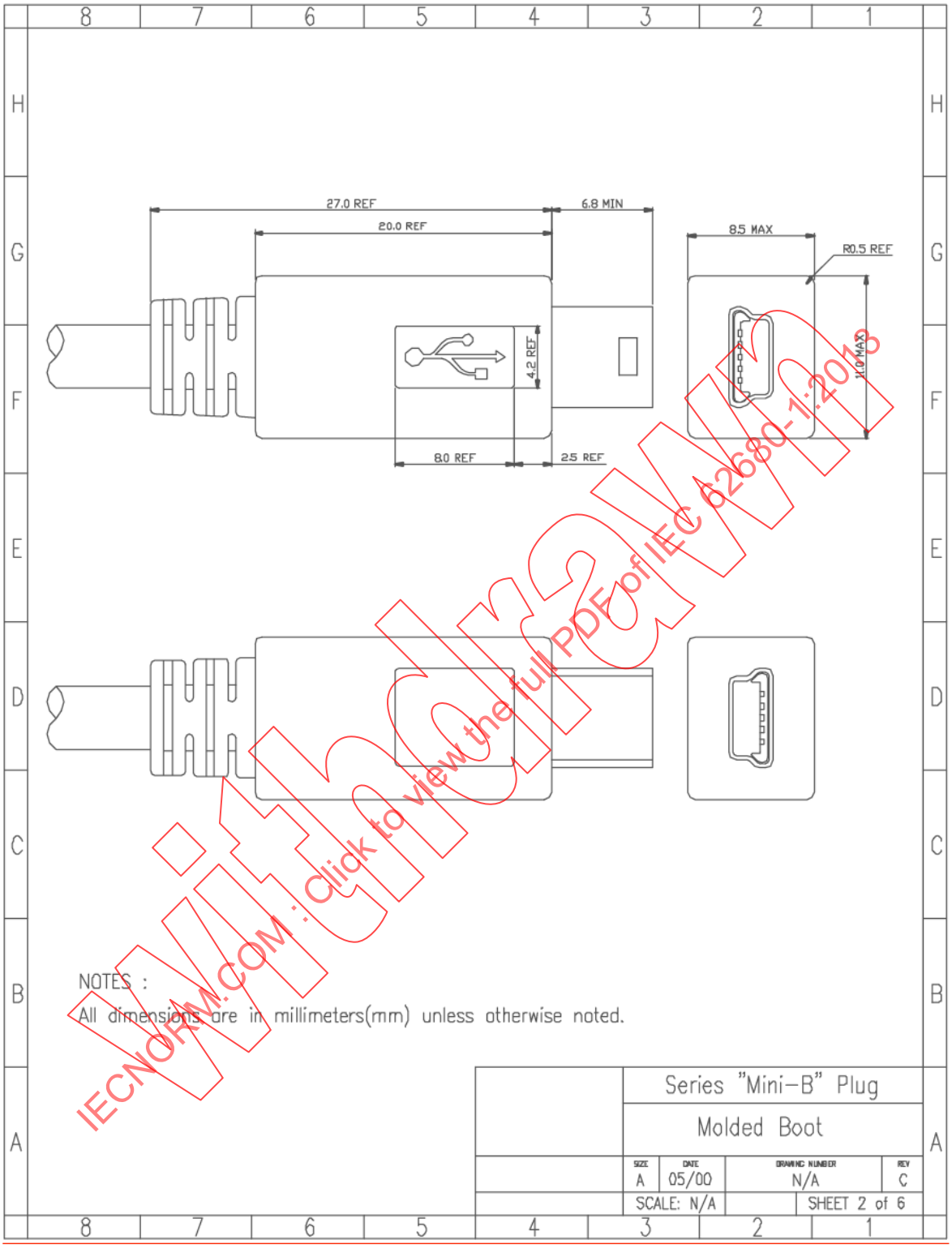


Figure ~~6-6~~6-7. Typical USB Plug Orientation

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 6-8. Typical USB "Mini-B" Connector Plug Orientation

6.5.2 USB Connector Termination Data

Table 6-1 provides the standardized contact terminating assignments by number and electrical value for Series "A" and Series "B" connectors.

Table 6-1. USB Series "A" and Series "B" Connector Termination Assignment

Contact Number	Signal Name	Typical Wiring Assignment
1	VBUS	Red
2	D-	White
3	D+	Green
4	GND	Black
Shell	Shield	Drain Wire

Table 6-2 provides the standardized contact terminating assignments by number and electrical value for Series "mini-B" connectors.

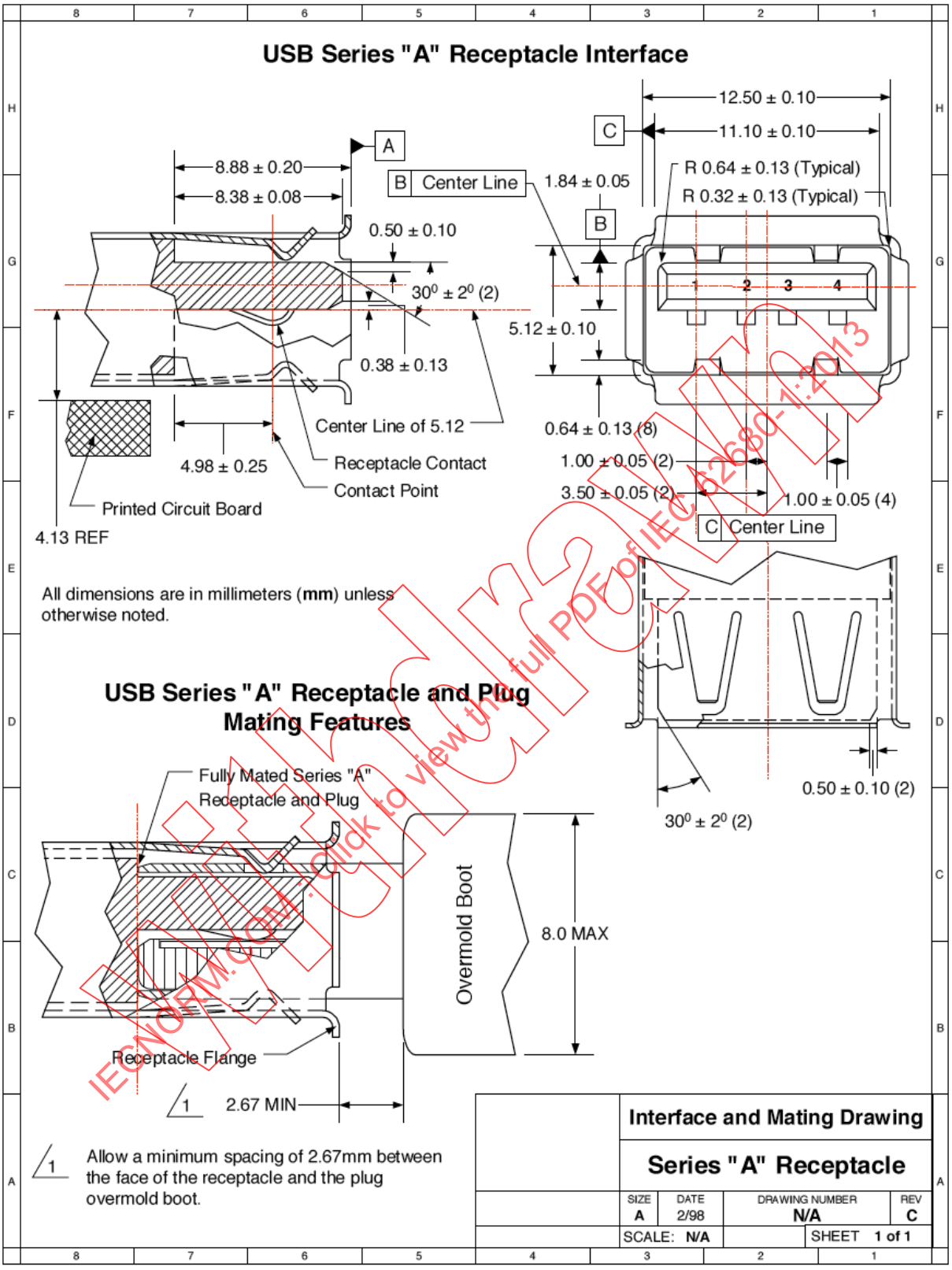
Table 6-2. USB Series "mini-B" Connector Termination Assignment

Contact Number	Signal Name	Typical Wiring Assignment
1	VBUS	Red
2	D-	White
3	D+	Green
4	ID	not connected
5	GND	Black
Shell	Shield	Drain Wire

6.5.3 Series "A" and Series "B" (or "Mini-B") Receptacles

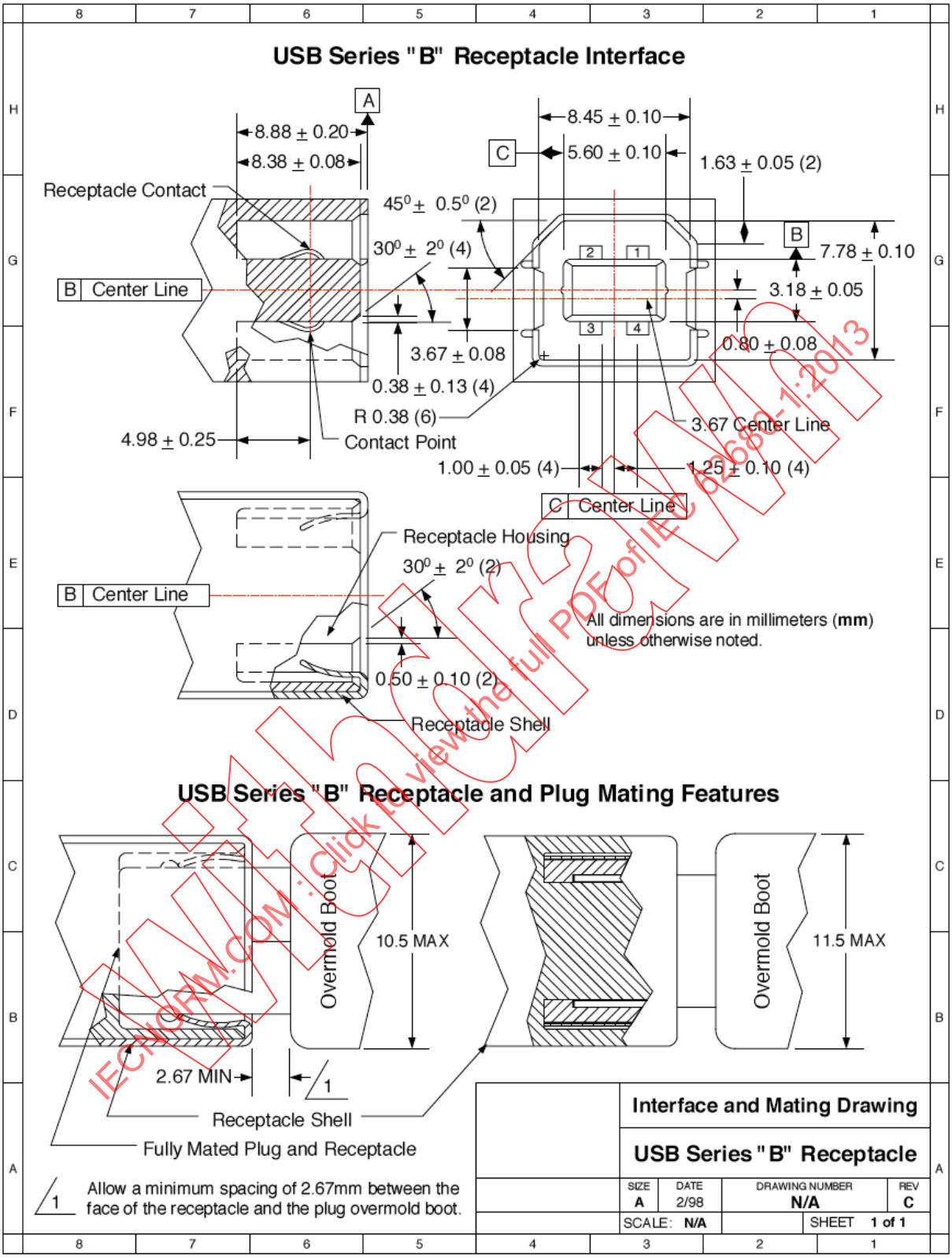
Electrical and mechanical interface configuration data for Series "A" and Series "B" receptacles are shown in Figure ~~6-7~~6-9 ~~and~~through Figure ~~6-8~~6-12. Also, refer to Figure ~~6-12~~6-17 through Figure 6-20~~; Figure 6-13, and Figure 6-14~~ at the end of this chapter for typical PCB receptacle layouts.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



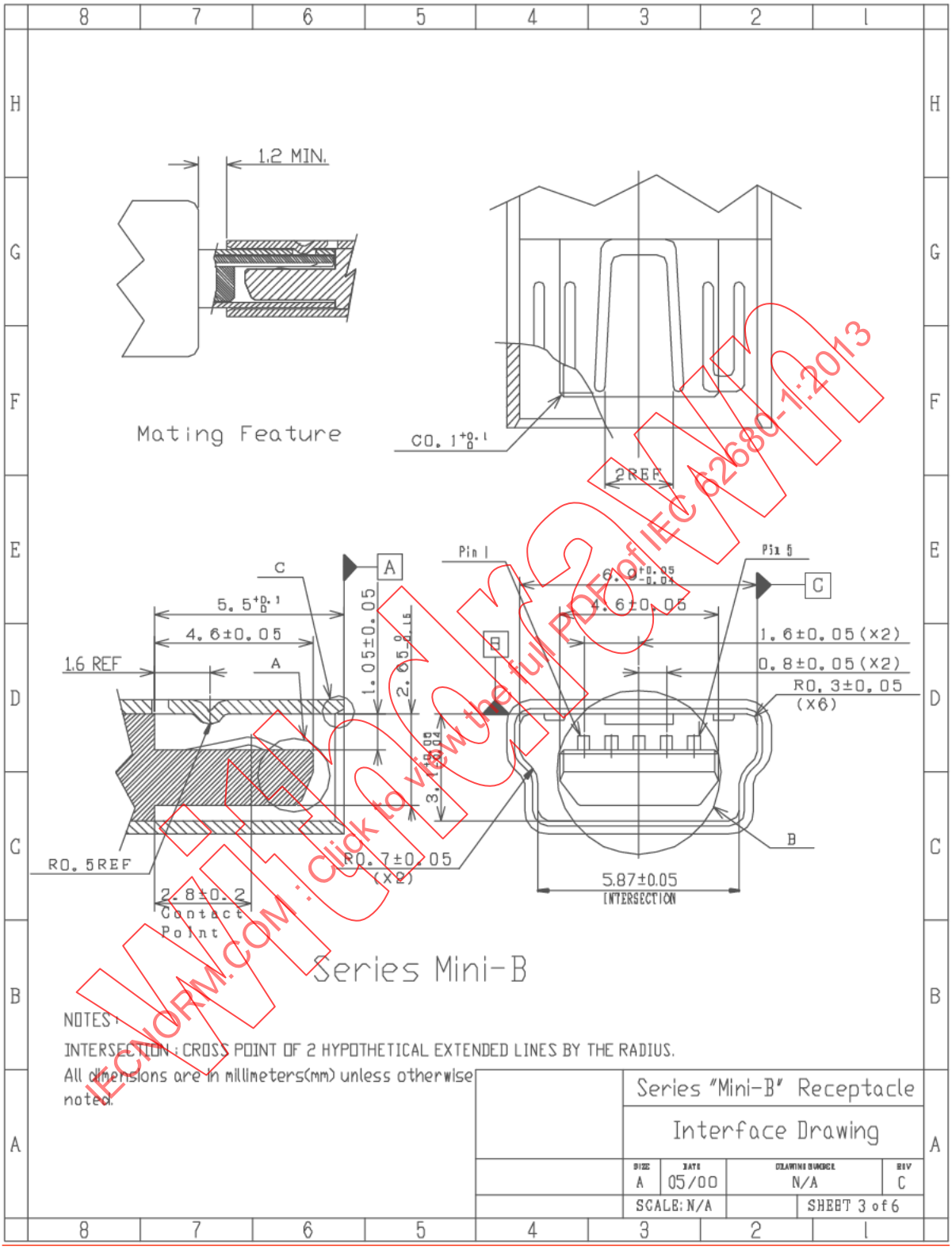
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure ~~6-7~~6-9. USB Series "A" Receptacle Interface and Mating Drawing



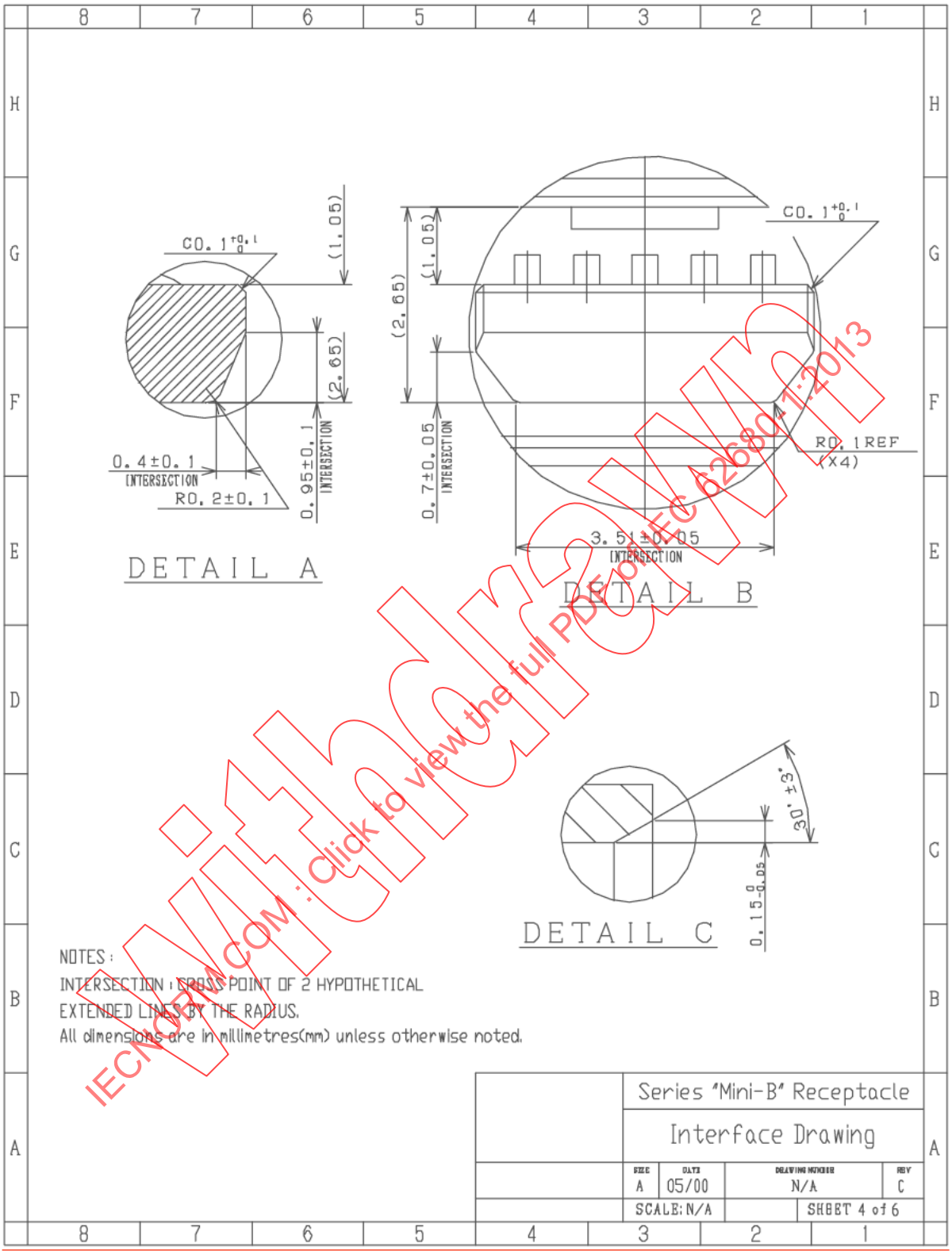
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure ~~6-86~~-10. USB Series "B" Receptacle Interface and Mating Drawing



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-11. USB Series "Mini-B" Receptacle Interface and Mating Drawing



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-12. USB Series "Mini-B" Receptacle Interface Drawing (Detail)

6.5.3.1 Receptacle Injection Molded Thermoplastic Insulator Material

Minimum UL 94-V0 rated, thirty percent (30%) glass-filled polybutylene terephthalate (PBT) or polyethylene terephthalate (PET) or better.

Typical Colors: Black, gray, and natural. The “mini-B” receptacle insulator must be black in color.

Flammability Characteristics: UL 94-V0 rated.

Flame Retardant Package must meet or exceed the requirements for UL, CSA, VDE, etc.

Oxygen Index (LOI): Greater than 21%. ASTM D 2863.

6.5.3.2 Receptacle Shell Materials

6.5.3.2.1 Series “A” and Series “B”

Substrate Material: 0.30 + 0.05 mm phosphor bronze, nickel silver, or other copper based high strength materials.

Plating:

1. Underplate: Optional. Minimum 1.00 micrometers (40 microinches) nickel. In addition, manufacturer may use a copper underplate beneath the nickel.
2. Outside: Minimum 2.5 micrometers (100 microinches) bright tin or bright tin-lead.

6.5.3.2.2 Series “mini-B”

Substrate Material: 0.3 mm minimum phosphor bronze, nickel silver, or other suitable material.

The plating information below describes an example of acceptable "mini-B" receptacle plating.

Plating:

1. Underplate: Optional. Minimum 1.00 micrometers (40 microinches) nickel. In addition, manufacturer may use a copper underplate beneath the nickel.
2. Outside: Minimum 2.5 micrometers (100 microinches) bright tin or bright tin-lead.

6.5.3.3 Receptacle Contact Materials

6.5.3.3.1 Series “A” and Series “B”

Substrate Material: 0.30 ± 0.05 mm minimum half-hard phosphor bronze or other high strength copper based material.

Plating: Contacts are to be selectively plated.

A. Option I
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.70 micrometers (28 microinches) palladium.
   3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

B. Option II
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.75 micrometers (30 microinches) palladium-nickel.
   3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

C. Option III

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
2. Mating Area: Minimum 0.75 micrometers (30 microinches) gold.
3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

6.5.3.3.2 Series “mini-B”

Substrate Material: 0.2 mm minimum half-hard phosphor bronze or other high strength suitable material.

Plating: Contacts are to be selectively plated.

The following underplate, mating area, and solder tails options below describe examples of acceptable "mini-B" receptacle contact plating.

D. Option I

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.70 micrometers (28 microinches) palladium.
3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

E. Option II

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.75 micrometers (30 microinches) palladium-nickel.
3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

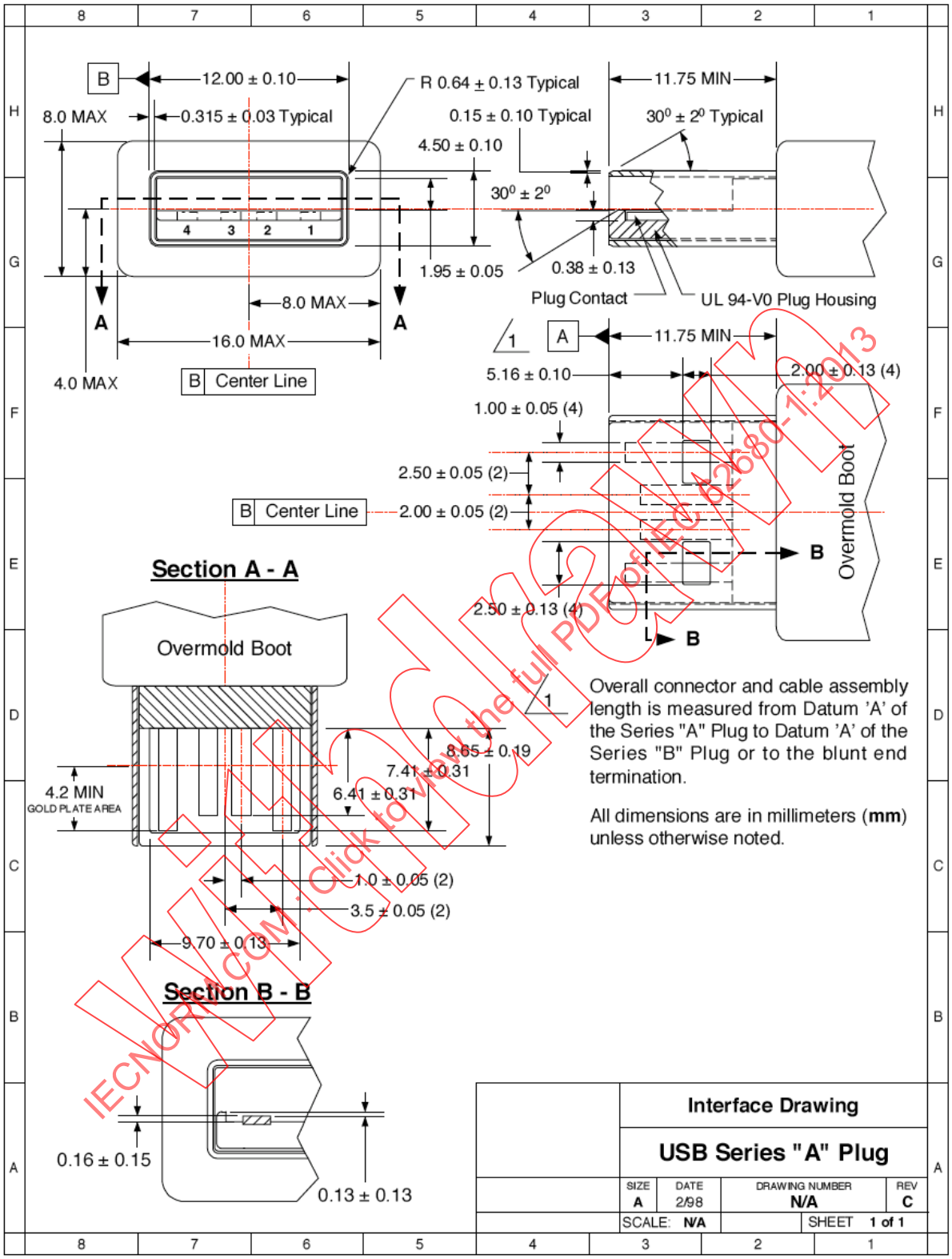
F. Option III

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
2. Mating Area: Minimum 0.75 micrometers (30 microinches) gold.
3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

6.5.4 Series “A” and Series “B” (or “Mini-B”) Plugs

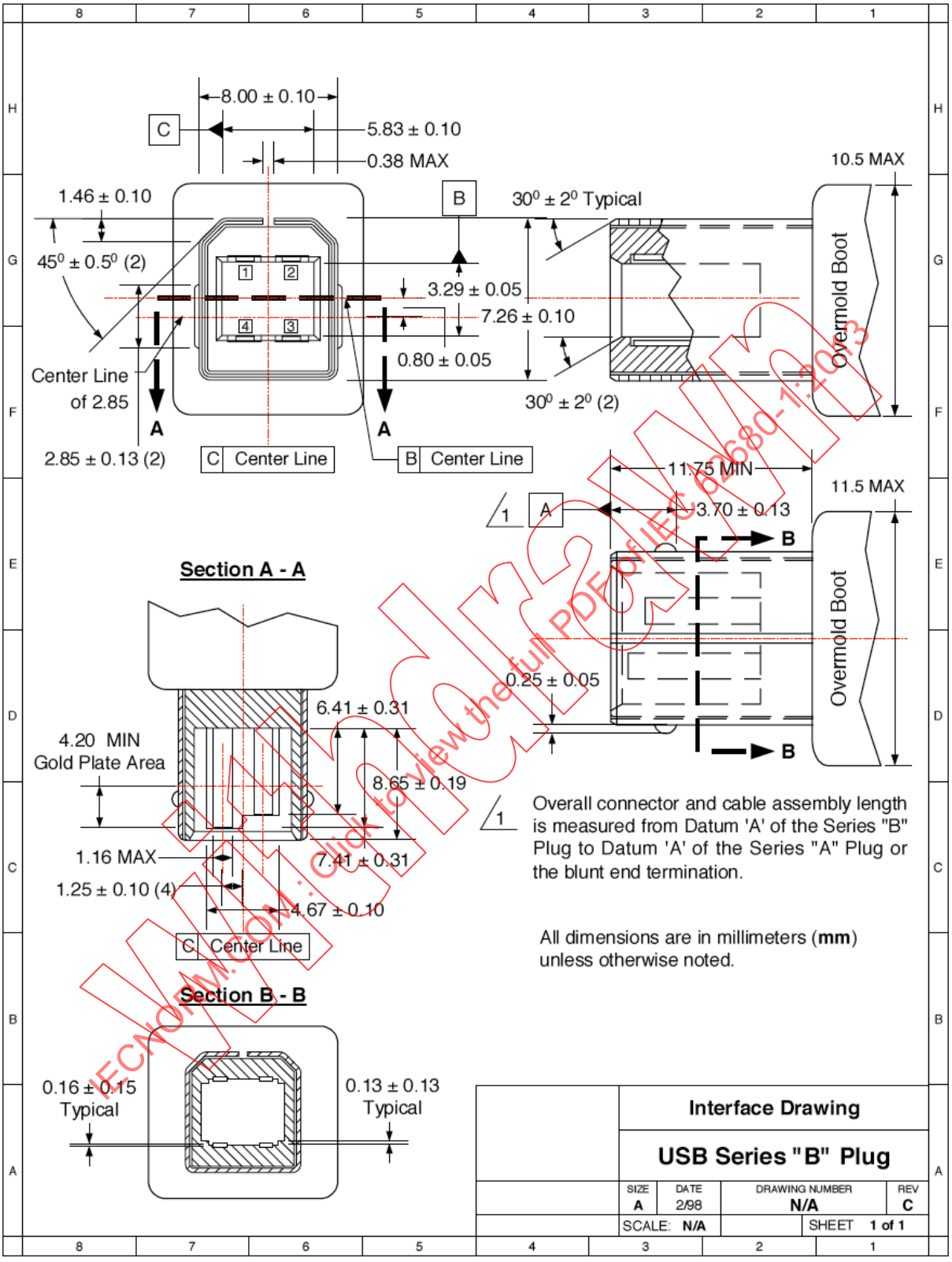
Electrical and mechanical interface configuration data for Series "A" and Series "B" (or “Mini-B”) plugs are shown in Figure ~~6-9~~6-13 through~~and~~ Figure ~~6-10~~6-15.

Withdrawn
FCNORM.COM : Click to view the full PDF of IEC 62680-1:2013



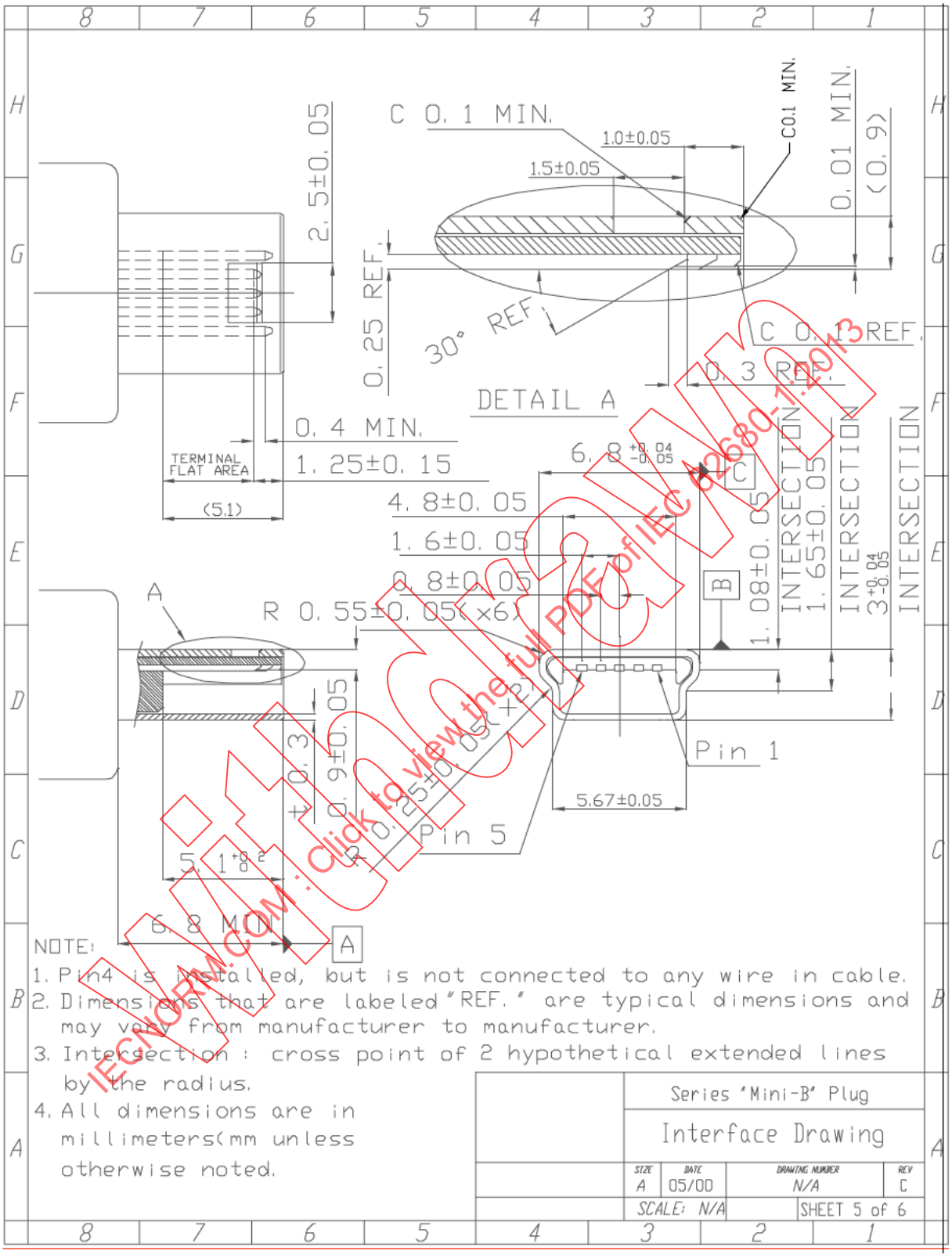
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure ~~6-96~~-13. USB Series "A" Plug Interface Drawing



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure ~~6-106~~6-14. USB Series "B" Plug Interface Drawing



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 6-15. USB Series "Mini-B" Plug Interface Drawing

6.5.4.1 Plug Injection Molded Thermoplastic Insulator Material

Minimum UL 94-V0 rated, thirty percent (30%) glass-filled polybutylene terephthalate (PBT) or polyethylene terephthalate (PET) or better.

Typical Colors: Black, gray, and natural. The “mini-B” plug insulator must be black in color.

Flammability Characteristics: UL 94-V0 rated.

Flame Retardant Package must meet or exceed the requirements for UL, CSA, and VDE. Oxygen Index (LOI): 21%. ASTM D 2863.

6.5.4.2 Plug Shell Materials

6.5.4.2.1 Series “A” and Series “B”

Substrate Material: 0.30 ± 0.05 mm phosphor bronze, nickel silver, or other suitable material.

Plating:

A. Underplate: Optional. Minimum 1.00 micrometers (40 microinches) nickel. In addition, manufacturer may use a copper underplate beneath the nickel.

B. Outside: Minimum 2.5 micrometers (100 microinches) bright tin or bright tin-lead.

6.5.4.2.2 Series “mini-B”

Substrate Material: 0.2 mm minimum phosphor bronze, nickel silver, or other suitable material.

The information below describes an example of acceptable "mini-B" plug plating.

Plating:

1. Underplate: Optional. Minimum 1.00 micrometers (40 microinches) nickel. In addition, manufacturer may use a copper underplate beneath the nickel.
2. Outside: Minimum 2.5 micrometers (100 microinches) bright tin or bright tin-lead.

6.5.4.3 Plug (Male) Contact Materials

6.5.4.3.1 Series “A” and Series “B”

Substrate Material: 0.30 ± 0.05 mm half-hard phosphor bronze.

Plating: Contacts are to be selectively plated.

A. Option I
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.70 micrometers (28 microinches) palladium.
   3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

B. Option II
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.75 micrometers (30 microinches) palladium-nickel.
   3. Wire Crimp/Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

C. Option III
   1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
   2. Mating Area: Minimum 0.75 micrometers (30 microinches) gold.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

6.5.4.3.2 Series “mini-B”

Substrate Material: 0.2 mm minimum half-hard phosphor bronze or other suitable material.

Plating: Contacts are to be selectively plated.

The underplate, mating area, and solder tails options below describe examples of acceptable "mini-B" plug contact plating.

D. Option I

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.70 micrometers (28 microinches) palladium.
3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

E. Option II

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.75 micrometers (30 microinches) palladium-nickel.
3. Wire Crimp/Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

F. Option III

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.
2. Mating Area: Minimum 0.75 micrometers (30 microinches) gold.
3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead over the underplate.

6.6 Cable Mechanical Configuration and Material Requirements

High-/full-speed and low-speed cables differ in data conductor arrangement and shielding. Low-speed recommends, but does not require, use of a cable with twisted data conductors. Low speed recommends, but does not require, use of a cable with a braided outer shield. Figure ~~6-116~~-16 shows the typical high-/full-speed cable construction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

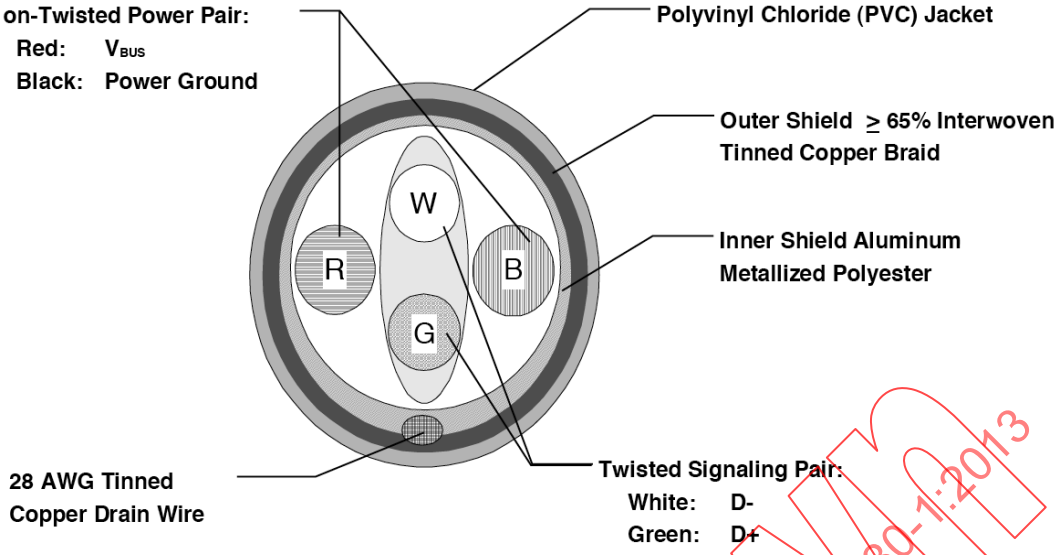


Figure ~~6-11~~6-16. Typical High-/full-speed Cable Construction

6.6.1 Description

High-/full-speed cable consists of one 28 to 20 AWG non-twisted power pair and one 28 AWG twisted data pair with an aluminum metallized polyester inner shield, 28 AWG stranded tinned copper drain wire, ≥ 65% tinned copper wire interwoven (braided) outer shield, and PVC outer jacket.

Low-speed cable consists of one 28 to 20 AWG non-twisted power pair and one 28 AWG data pair (a twist is recommended) with an aluminum metallized polyester inner shield, 28 AWG stranded tinned copper drain wire and PVC outer jacket. A ≥ 65% tinned copper wire interwoven (braided) outer shield is recommended.

6.6.2 Construction

Raw materials used in the fabrication of this cable must be of such quality that the fabricated cable is capable of meeting or exceeding the mechanical and electrical performance criteria of the most current USB Specification revision and all applicable domestic and international safety/testing agency requirements; e.g., UL, CSA, BSA, NEC, etc., for electronic signaling and power distribution cables in its category.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table ~~6-26-~~3. Power Pair

American Wire Gauge (AWG)	Nominal Conductor Outer Diameter	Stranded Tinned Conductors
28	0.381 mm (0.015”) 0.406 mm (0.016”)	7 x 36 19 x 40
26	0.483 mm (0.019”) 0.508 mm (0.020”)	7 x 34 19 x 38
24	0.610 mm (0.024”) 0.610 mm (0.024”)	7 x 32 19 x 36
22	0.762 mm (0.030”) 0.787 mm (0.031”)	7 x 30 19 x 34
20	0.890 mm (0.035”) 0.931 mm (0.037”)	7 x 28 19 x 32

Note: Minimum conductor construction must be stranded tinned copper.

Non-Twisted Power Pair:

A. Wire Gauge: Minimum 28 AWG or as specified by the user contingent upon the specified cable length. Refer to ~~Table 6-2~~Table 6-3.

B. Wire Insulation: Semirigid polyvinyl chloride (PVC).

1. Nominal Insulation Wall Thickness: 0.25 mm (0.010”)
2. Typical Power (V_{BUS}) Conductor: Red Insulation
3. Typical Ground Conductor: Black Insulation

Signal Pair:

A. Wire Gauge: 28 AWG minimum. Refer to ~~Table 6-3~~Table 6-4.

Table ~~6-36-~~4. Signal Pair

American Wire Gauge (AWG)	Nominal Conductor Outer Diameter	Stranded Tinned Conductors
28	0.381 mm (0.015”) 0.406 mm (0.016”)	7 x 36 19 x 40

Note: Minimum conductor construction must be stranded tinned copper.

B. Wire Insulation: High-density polyethylene (HDPE), alternately foamed polyethylene or foamed polypropylene

1. Nominal Insulation Wall Thickness: 0.31 mm (0.012”)
2. Typical Data Plus (+) Conductor: Green Insulation
3. Typical Data Minus (-) Conductor: White Insulation

C. Nominal Twist Ratio (not required for low-speed): One full twist every 60 mm (2.36”) to 80 mm (3.15”)

Aluminum Metallized Polyester Inner Shield (required for low-speed):

A. Substrate Material: Polyethylene terephthalate (PET) or equivalent material

B. Metallizing: Vacuum deposited aluminum

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

C. Assembly:

1. The aluminum metallized side of the inner shield must be positioned facing out to ensure direct contact with the drain wire.
2. The aluminum metallized inner shield must overlap by approximately one-quarter turn.

Drain Wire (required for low-speed):

A. Wire Gauge: Minimum 28 AWG stranded tinned copper (STC) non-insulated. Refer to ~~Table 6-4~~Table 6-5.

Table ~~6-4~~6-5. Drain Wire Signal Pair

American Wire Gauge (AWG)	Nominal Conductor Outer Diameter	Stranded Tinned Conductors
28	0.381 mm (0.015”) 0.406 mm (0.016”)	7 x 36 19 x 40

Interwoven (Braided) Tinned Copper Wire (ITCW) Outer Shield (recommended but not required for low-speed):

A. Coverage Area: Minimum 65%.

B. Assembly: The interwoven (braided) tinned copper wire outer shield must encase the aluminum metallized PET shielded power and signal pairs and must be in direct contact with the drain wire.

Outer Polyvinyl Chloride (PVC) Jacket:

A. Assembly: The outer PVC jacket must encase the fully shielded power and signal pairs and must be in direct contact with the tinned copper outer shield.

B. Nominal Wall Thickness: 0.64 mm (0.025”).

Marking: The cable must be legibly marked using contrasting color permanent ink.

A. Minimum marking information for high-/full-speed cable must include:
USB SHIELDED <Gauge/2C + Gauge/2C> UL CM 75 °C — UL Vendor ID.

B. Minimum marking information for low-speed cable shall include:
USB specific marking is not required for low-speed cable.

Nominal Fabricated Cable Outer Diameter:

This is a nominal value and may vary slightly from manufacturer to manufacturer as a function of the conductor insulating materials and conductor specified. Refer to ~~Table 6-5~~Table 6-6.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table ~~6-5~~6-6. Nominal Cable Diameter

Shielded USB Cable Configuration	Nominal Outer Cable Diameter
28/28	4.06 mm (0.160”)
28/26	4.32 mm (0.170”)
28/24	4.57 mm (0.180”)
28/22	4.83 mm (0.190”)
28/20	5.21 mm (0.205”)

6.6.3 Electrical Characteristics

All electrical characteristics must be measured at or referenced to +20 °C (68 °F).

Voltage Rating: 30 V rms maximum.

Conductor Resistance: Conductor resistance must be measured in accordance with ASTM-D-4566 Section 13. Refer to ~~Table 6-6~~Table 6-7.

Conductor Resistance Unbalance (Pairs): Conductor resistance unbalance between two (2) conductors of any pair must not exceed five percent (5%) when measured in accordance with ASTM-D-4566 Section 15.

The DC resistance from plug shell to plug shell (or end of integrated cable) must be less than 0.6 ohms.

Table ~~6-6~~6-7. Conductor Resistance

American Wire Gauge (AWG)	Ohms (Ω) / 100 Meters Maximum
28	23.20
26	14.60
24	9.09
22	5.74
20	3.58

6.6.4 Cable Environmental Characteristics

Temperature Range:

A. Operating Temperature Range: 0 °C to +50 °C
B. Storage Temperature Range: -20 °C to +60 °C
C. Nominal Temperature Rating: +20 °C

Flammability: All plastic materials used in the fabrication of this product shall meet or exceed the requirements of NEC Article 800 for communications cables Type CM (Commercial).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

6.6.5 Listing

The product shall be UL listed per UL Subject 444, Class 2, Type CM for Communications Cable Requirements.

6.7 Electrical, Mechanical, and Environmental Compliance Standards

~~Table 6-7~~Table 6-8 lists the minimum test criteria for all USB cable, cable assemblies, and connectors.

Table ~~6-7~~6-8. USB Electrical, Mechanical, and Environmental Compliance Standards

Test Description	Test Procedure	Performance Requirement
Visual and Dimensional Inspection	EIA 364-18 Visual, dimensional, and functional inspection in accordance with the USB quality inspection plans.	Must meet or exceed the requirements specified by the most current version of Chapter 6 of the USB Specification.
Insulation Resistance	EIA 364-21 The object of this test procedure is to detail a standard method to assess the insulation resistance of USB connectors. This test procedure is used to determine the resistance offered by the insulation materials and the various seals of a connector to a DC potential tending to produce a leakage of current through or on the surface of these members.	1,000 (100 for "mini-B" connector) MΩ minimum.
Dielectric Withstanding Voltage	EIA 364-20 The object of this test procedure is to detail a test method to prove that a USB connector can operate safely at its rated voltage and withstand momentary over-potentials due to switching, surges, and/or other similar phenomena.	The dielectric must withstand 500 (100 for "mini-B" connector) V AC for one minute at sea level.
Low Level Contact Resistance	EIA 364-23 The object of this test is to detail a standard method to measure the electrical resistance across a pair of mated contacts such that the insulating films, if present, will not be broken or asperity melting will not occur.	30(100 for "mini-B" connector) mΩ maximum when measured at 20 mV maximum open circuit at 100 mA. Mated test contacts must be in a connector housing.
Contact Current Rating	EIA 364-70 — Method B The object of this test procedure is to detail a standard method to assess the current carrying capacity of mated USB connector contacts.	1.5 (100 for "mini-B" connector) A at 250 V AC minimum when measured at an ambient temperature of 25 °C. With power applied to the contacts, the Δ T must not exceed +30 °C at any point in the USB connector under test.
Contact Capacitance	EIA 364-30 The object of this test is to detail a standard method to determine the capacitance between conductive elements of a USB connector.	2 pF maximum unmated per contact.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table ~~6-7~~6-8. USB Electrical, Mechanical, and Environmental Compliance Standards (Continued)

Test Description	Test Procedure	Performance Requirement
Insertion Force	EIA 364-13 The object of this test is to detail a standard method for determining the mechanical forces required for inserting a USB connector.	35 Newtons maximum at a maximum rate of 12.5 mm (0.492") per minute.
Extraction Force	EIA 364-13 The object of this test is to detail a standard method for determining the mechanical forces required for extracting a USB connector.	10 Newtons minimum at a maximum rate of 12.5 mm (0.492") per minute. For "mini-B" connector with Detent Latch at a maximum rate of 12.5 mm (0.492") per minute 7 N min. initial 3 N min. after 5000 cycles
Durability	EIA 364-09 The object of this test procedure is to detail a uniform test method for determining the effects caused by subjecting a USB connector to the conditioning action of insertion and extraction, simulating the expected life of the connectors. Durability cycling with a gauge is intended only to produce mechanical stress. Durability performed with mating components is intended to produce both mechanical and wear stress.	1,500 (5,000 for "mini-B" connector) insertion/extraction cycles at a maximum rate of 200 cycles per hour.
Cable Pull-Out	EIA 364-38 Test Condition A The object of this test procedure is to detail a standard method for determining the holding effect of a USB plug cable clamp without causing any detrimental effects upon the cable or connector components when the cable is subjected to inadvertent axial tensile loads.	After the application of a steady state axial load of 40 Newtons for one minute.
Physical Shock	EIA 364-27 Test Condition H The object of this test procedure is to detail a standard method to assess the ability of a USB connector to withstand specified severity of mechanical shock.	No discontinuities of 1 μs or longer duration when mated USB connectors are subjected to 11 ms duration 30 Gs half-sine shock pulses. Three shocks in each direction applied along three mutually perpendicular planes for a total of 18 shocks.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table ~~6-7~~6-8. USB Electrical, Mechanical, and Environmental Compliance Standards (Continued)

Test Description	Test Procedure	Performance Requirement
Random Vibration	EIA 364-28 Test Condition V Test Letter A This test procedure is applicable to USB connectors that may, in service, be subjected to conditions involving vibration. Whether a USB connector has to function during vibration or merely to survive conditions of vibration should be clearly stated by the detailed product specification. In either case, the relevant specification should always prescribe the acceptable performance tolerances.	No discontinuities of 1 μs or longer duration when mated USB connectors are subjected to 5.35 Gs RMS. 15 minutes in each of three mutually perpendicular planes.
Thermal Shock	EIA 364-32 Test Condition I The object of this test is to determine the resistance of a USB connector to exposure at extremes of high and low temperatures and to the shock of alternate exposures to these extremes, simulating the worst case conditions for storage, transportation, and application.	10 cycles –55 ˚C and +85 ˚C. The USB connectors under test must be mated.
Humidity Life	EIA 364-31 Test Condition A Method III The object of this test procedure is to detail a standard test method for the evaluation of the properties of materials used in USB connectors as they are influenced by the effects of high humidity and heat.	168 hours minimum (seven complete cycles). The USB connectors under test must be tested in accordance with EIA 364-31.
Solderability	EIA 364-52 The object of this test procedure is to detail a uniform test method for determining USB connector solderability. The test procedure contained herein utilizes the solder dip technique. It is not intended to test or evaluate solder cup, solder eyelet, other hand-soldered type, or SMT type terminations.	USB contact solder tails must pass 95% coverage after one hour steam aging as specified in Category 2.
Flammability	UL 94 V-0 This procedure is to ensure thermoplastic resin compliance to UL flammability standards.	The manufacturer will require its thermoplastic resin vendor to supply a detailed C of C with each resin shipment. The C of C shall clearly show the resin's UL listing number, lot number, date code, etc.
~~Flammability~~	~~UL 94 V-0~~ ~~This procedure is to ensure thermoplastic resin compliance to UL flammability standards.~~	~~The manufacturer will require its thermoplastic resin vendor to supply a detailed C of C with each resin shipment. The C of C shall clearly show the resin's UL listing number, lot number, date code, etc.~~

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table ~~6-7~~6-8. USB Electrical, Mechanical, and Environmental Compliance Standards (Continued)

Test Description	Test Procedure	Performance Requirement
Cable Impedance (Only required for high-/full-speed)	The object of this test is to insure the signal conductors have the proper impedance. Connect the Time Domain Reflectometer (TDR) outputs to the impedance/delay/skew test fixture (Note 1). Use separate 50 Ω cables for the plus (or true) and minus (or complement) outputs. Set the TDR head to differential TDR mode. Connect the Series "A" plug of the cable to be tested to the text fixture, leaving the other end open-circuited. Define a waveform composed of the difference between the true and complement waveforms, to allow measurement of differential impedance. Measure the minimum and maximum impedances found between the connector and the open circuited far end of the cable.	Impedance must be in the range specified in Table 7-9 (ZO).
Signal Pair Attenuation (Only required for high-/full-speed)	The object of this test is to insure that adequate signal strength is presented to the receiver to maintain a low error rate. 1. Connect the Network Analyzer output port (port 1) to the input connector on the attenuation test fixture (Note 2). 2. Connect the Series “A” plug of the cable to be tested to the test fixture, leaving the other end open-circuited. 3. Calibrate the network analyzer and fixture using the appropriate calibration standards over the desired frequency range. 4. Follow the method listed in Hewlett Packard Application Note 380-2 to measure the open-ended response of the cable. 5. Short circuit the Series “B” (or “mini-B”) end (or bare leads end, if a captive cable) and measure the short-circuit response. 6. Using the software in H-P App. Note 380-2 or equivalent, calculate the cable attenuation accounting for resonance effects in the cable as needed.	Refer to Section 7.1.17 for frequency range and allowable attenuation.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table ~~6-7~~6-8. USB Electrical, Mechanical, and Environmental Compliance Standards (Continued)

Test Description	Test Procedure	Performance Requirement
Propagation Delay	The purpose of the test is to verify the end to end propagation of the cable. 1. Connect one output of the TDR sampling head to the D+ and D- inputs of the impedance/delay/skew test fixture (Note 1). Use one 50 Ω cable for each signal and set the TDR head to differential TDR mode. 2. Connect the cable to be tested to the test fixture. If detachable, plug both connectors in to the matching fixture connectors. If captive, plug the series “A” plug into the matching fixture connector and solder the stripped leads on the other end to the test fixture. 3. Measure the propagation delay of the test fixture by connecting a short piece of wire across the fixture from input to output and recording the delay. 4. Remove the short piece of wire and remeasure the propagation delay. Subtract from it the delay of the test fixture measured in the previous step.	High-/full-speed. See Section 7.1.1.1, Section 7.1.4, Section 7.1.16, and Table 7-9 (TFSCBL). Low-speed. See Section 7.1.1.2, Section 7.1.16, and Table 7-9 (TLSCBL).
Propagation Delay Skew	This test insures that the signal on both the D+ and D- lines arrive at the receiver at the same time. 1. Connect the TDR to the fixture with test sample cable, as in the previous section. 2. Measure the difference in delay for the two conductors in the test cable. Use the TDR cursors to find the open-circuited end of each conductor (where the impedance goes infinite) and subtract the time difference between the two values.	Propagation skew must meet the requirements as listed in Section 7.1.3.
Capacitive Load Only required for low-speed	The purpose of this test is to insure the distributed inter-wire capacitance is less than the lumped capacitance specified by the low-speed transmit driver. 1. Connect the one lead of the Impedance Analyzer to the D+ pin on the impedance/delay/skew fixture (Note 1) and the other lead to the D- pin. 2. Connect the series "A" plug to the fixture, with the series “B” (or “mini-B”) end leads open-circuited. 3. Set the Impedance Analyzer to a frequency of 100 kHz, to measure the capacitance.	See Section 7.1.1.2 and Table 7-7 (CLINUA).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Note1: Impedance, propagation delay, and skew test fixture
This fixture will be used with the TDR for measuring the time domain performance of the cable under test. The fixture impedance should be matched to the equipment, typically 50 Ω. Coaxial connectors should be provided on the fixture for connection from the TDR.

Note 2: Attenuation text fixture
This fixture provides a means of connection from the network analyzer to the Series "A" plug. Since USB signals are differential in nature and operate over balanced cable, a transformer or balun (North Hills NH13734 or equivalent) is ideally used. The transformer converts the unbalanced (also known as single-ended) signal from the signal generator which is typically a 50 Ω output to the balanced (also known as differential) and likely different impedance loaded presented by the cable. A second transformer or balun should be used on the other end of the cable under test to convert the signal back to unbalanced form of the correct impedance to match the network analyzer.

6.7.1 Applicable Documents

American National Standard/Electronic Industries Association

ANSI/EIA-364-C (12/94)	Electrical Connector/Socket Test Procedures Including Environmental Classifications

American Standard Test Materials

ASTM-D-4565	Physical and Environmental Performance Properties of Insulation and Jacket for Telecommunication Wire and Cable, Test Standard Method
ASTM-D-4566	Electrical Performance Properties of Insulation and Jacket for Telecommunication Wire and Cable, Test Standard Method

Underwriters' Laboratory, Inc.

UL STD-94	Test for Flammability of Plastic materials for Parts in Devices and Appliances
UL Subject-444	Communication Cables

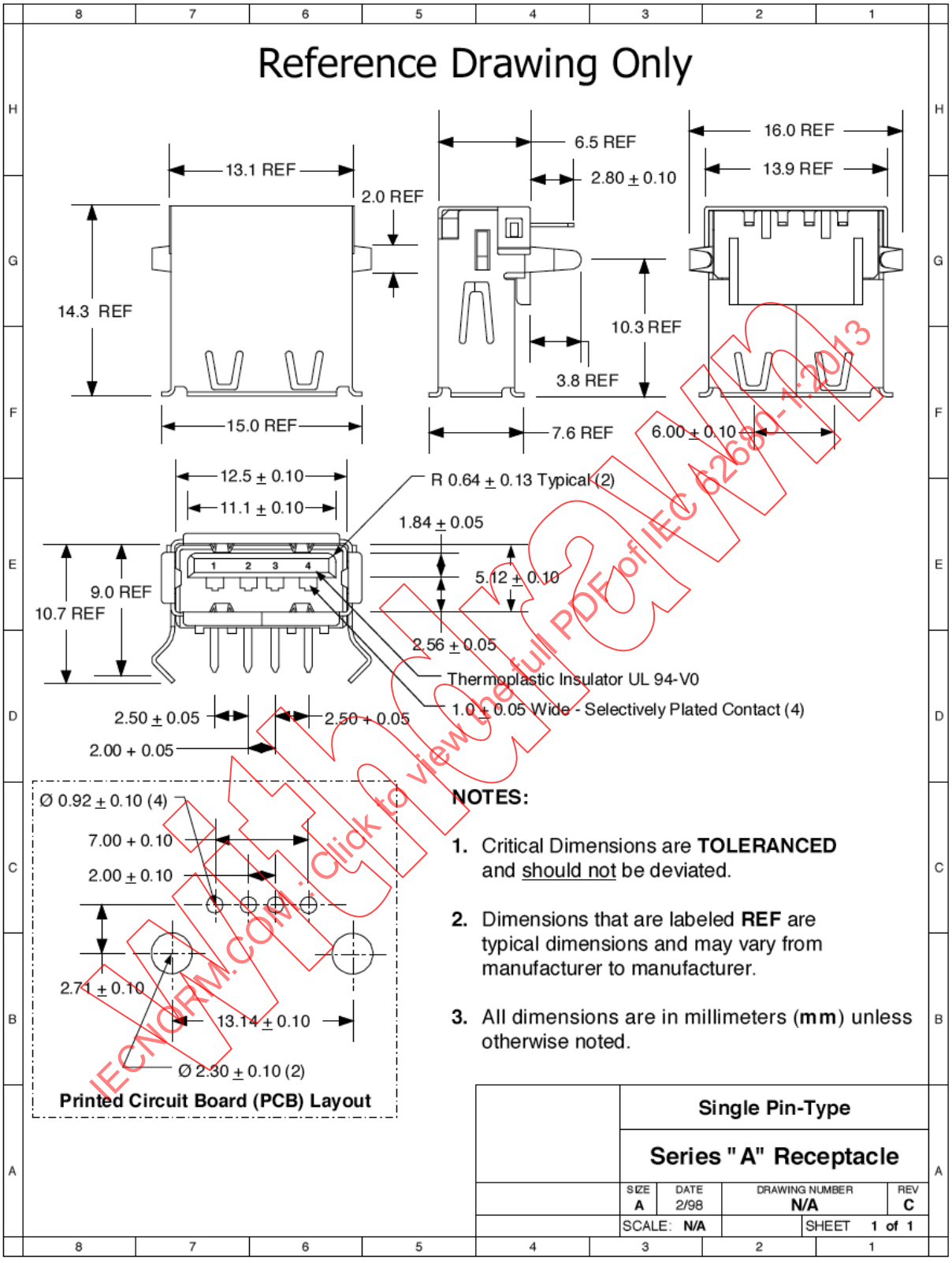
6.8 USB Grounding

The shield must be terminated to the connector plug for completed assemblies. The shield and chassis are bonded together. The user selected grounding scheme for USB devices, and cables must be consistent with accepted industry practices and regulatory agency standards for safety and EMI/ESD/RFI.

6.9 PCB Reference Drawings

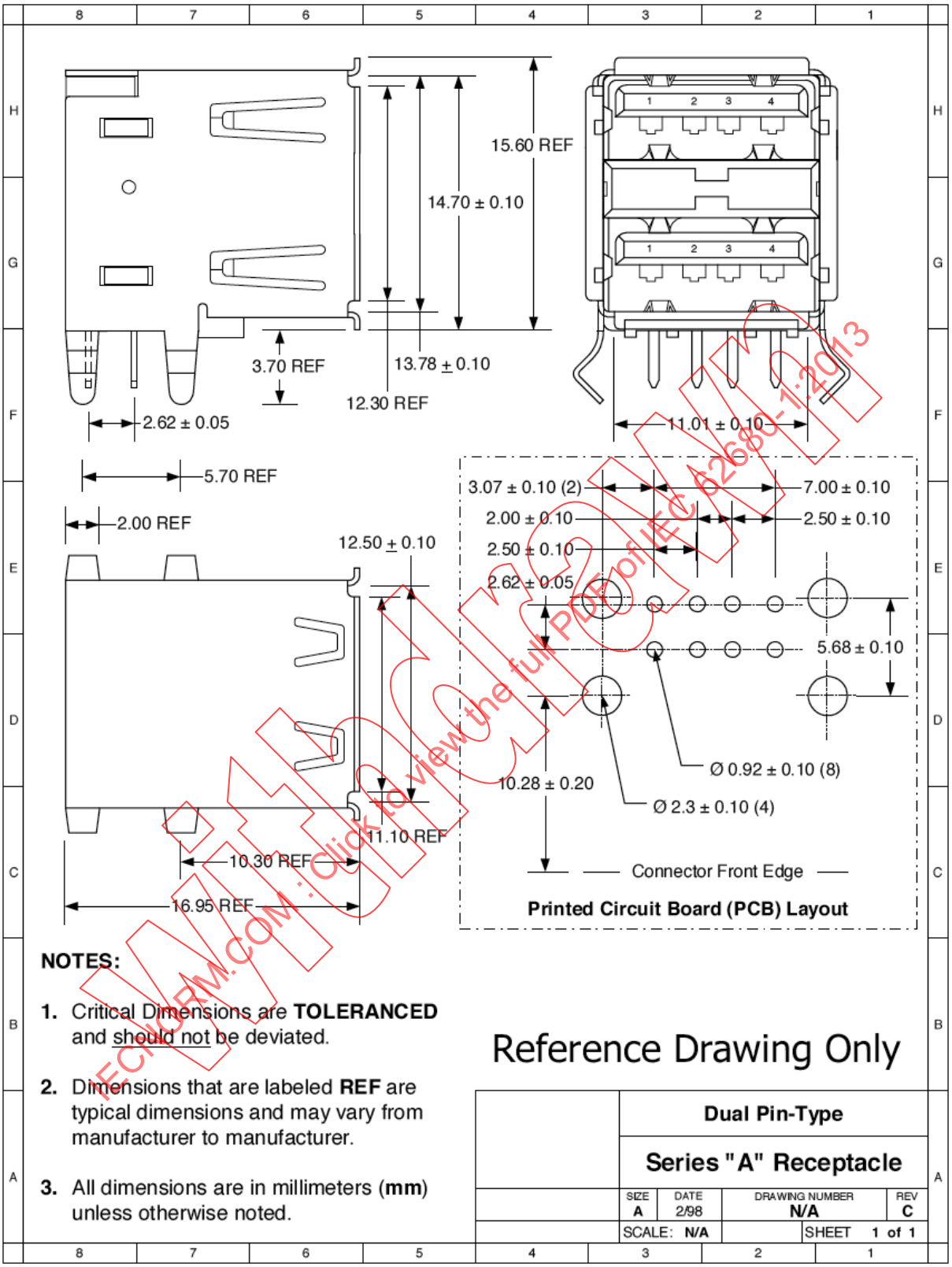
The drawings in ~~Figure 6-12, Figure 6-13, and~~ Figure 6-17 through Figure 6-20 describe typical receptacle PCB interfaces. These drawings are included for informational purposes only.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure ~~6-12~~6-17. Single Pin-type Series "A" Receptacle



NOTES:

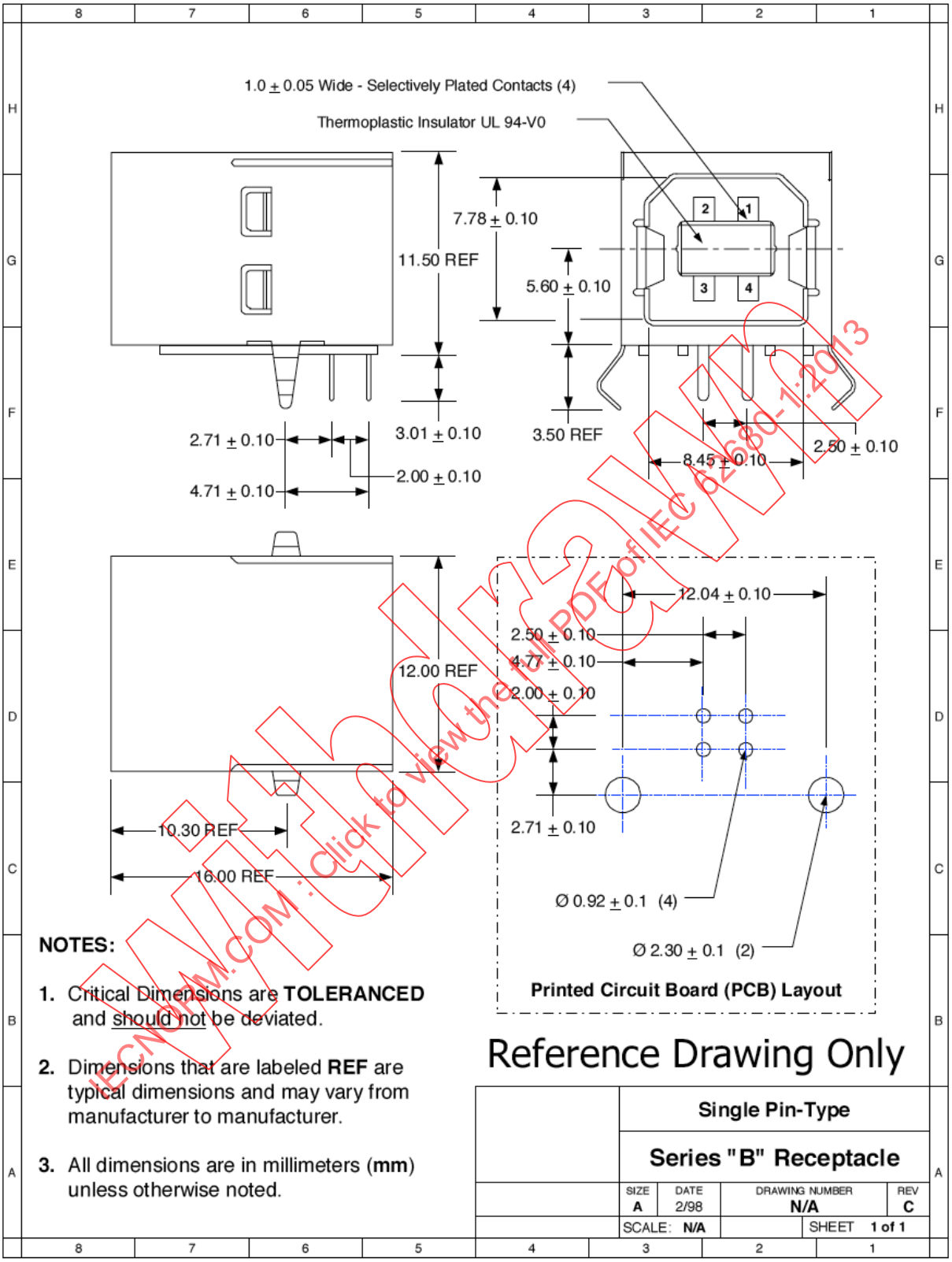
1. Critical Dimensions are **TOLERANCED** and should not be deviated.
2. Dimensions that are labeled **REF** are typical dimensions and may vary from manufacturer to manufacturer.
3. All dimensions are in millimeters (**mm**) unless otherwise noted.

Reference Drawing Only

	Dual Pin-Type			
	Series "A" Receptacle			
	SIZE A	DATE 2/98	DRAWING NUMBER N/A	REV C
	SCALE: N/A		SHEET 1 of 1	

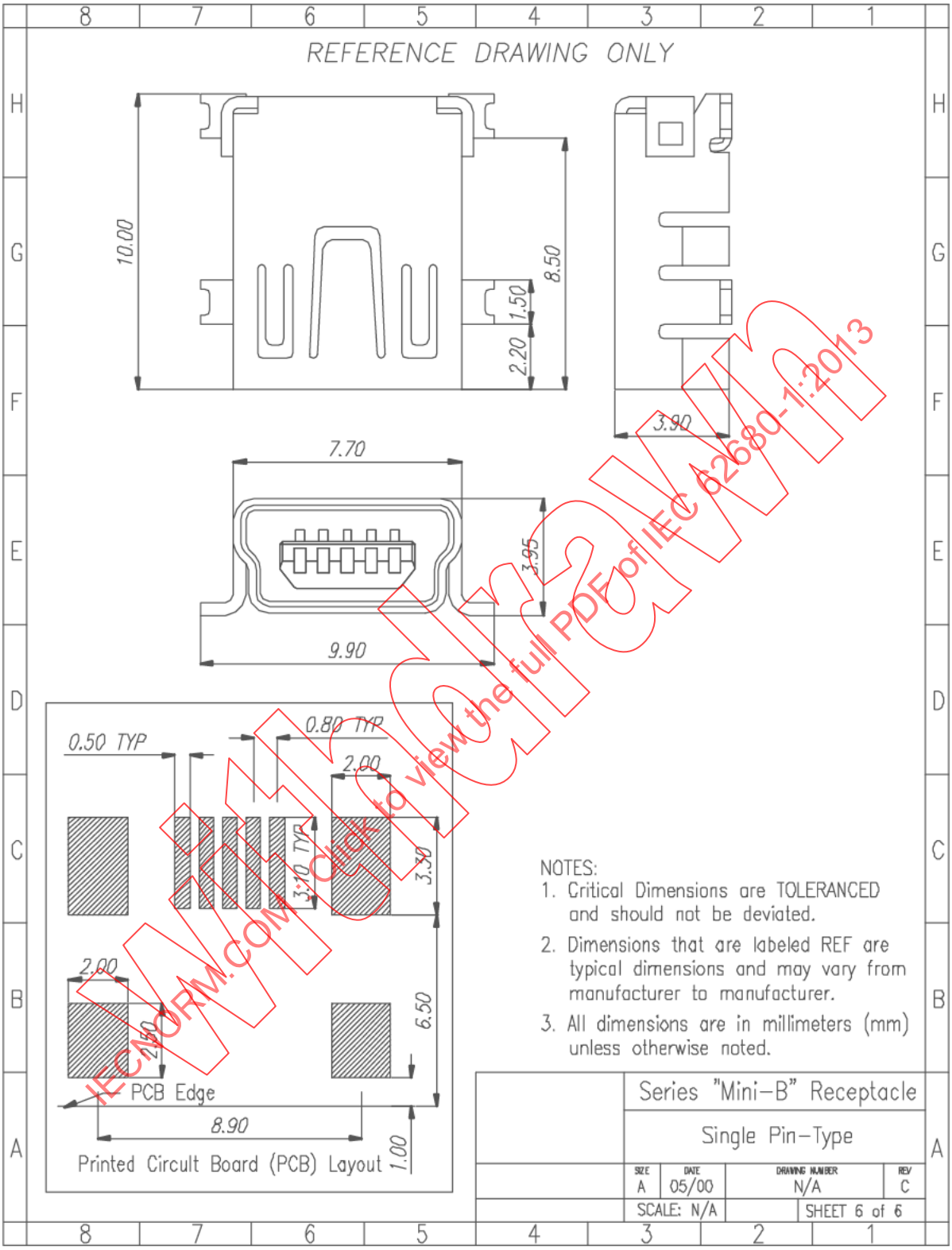
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure ~~6-136~~-18. Dual Pin-type Series "A" Receptacle



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure ~~6-14~~6-19. Single Pin-type Series "B" Receptacle



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-20. Single Pin-Type Series "Mini-B" Receptacle

USB ENGINEERING CHANGE NOTICE

Title: Pull-up/pull-down resistors

Applies to: Universal Serial Bus Specification Revision 2.0

Summary of ECN:
This ECN changes the range of the pull-up and pull-down resistors used to bias the bus. New values for the pull-down resistors are 14.25 kΩ to 24.80 kΩ. For the pull-up resistors, the values are 900 Ω to 1575 Ω when the bus is idle and 1425 Ω to 3090 Ω when the upstream device is transmitting.

Reasons for ECN:
This change allows the resistors to be moved onto the device or hub Silicon and not require laser trimming to achieve the required tolerances. This can result in a significant savings in cost, especially for a hub. It also has a significant impact on an OnTheGo, dual-role device that has to switch both the pullup and pull-down resistors.

Impact on existing peripherals and systems:
There is no impact to current devices that are compatible with the current versions of the USB specification. The change has no impact on interoperability.

Hardware implications:

1) Increasing the pull-down resistor increases the discharge time of the bus after a disconnect. Worstcase discharge time increases from 15.75 µs to 20 µs. This increase has no operational implications.
2) Decreasing the pull-up resistor value increases the amount of current used for bus biasing by a small percentage (from 218 µA max to 238 µA or 9%). On average, the amount of current used for the biasing is reduced as the pull-down resistor value has the largest impact on the total current. If the increase in bias current is a problem for a device, it is free to use the older tolerance (i.e. if a device would be affected then it does not have to implement the change and the wider tolerance on the connected device will not have a negative impact).
3) The original values for the pull-down resistors assumed that the output leakage current of an off output would be about 10 µA. With both the hub and device in the idle state, the pull-down on the low data line (D- for high and full-speed, D+ for low-speed) would be 20 µA * 15.75 kΩ = 315 mV. To keep the idle voltage at the same level, any hub/host that uses the higher value resistance has to reduce its leakage current proportionately. If, for example, a hub has a 24.80 kΩ pull-down, then to allow for the legacy device leakage of 10 µA, the leakage current on the hub has to be no more than 2.7 µA. This is well within the capabilities of contemporary Si processing and testing.
4) The pull-up resistors change from lower resistance to higher resistance when the bus goes from idle to active. This ensures that existing hub/host Si can drive the downstream device without increasing jitter or otherwise degrading signaling margin.
5) Devices that use this scheme for their pull-down resistors must be aware that the pull-up resistor on a LS device may be tied to Vbus. The increase in the pull-down resistance can cause the idle voltage on the bus to rise as high as 4.1 V. If the process used does not allow this sustained voltage on the bus, then the standard values of resistance should be used.

Software implications:
None.

Compliance testing implications:
None.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Specification Changes:

No changes outside of Chapter 7.

In Chapter 7, change all occurrences in text and in drawings of:

"15 kΩ ±5%"

to

"14.25 kΩ to 24.80 kΩ"

In Chapter 7, change all occurrences in text and in drawings of:

"1.5 kΩ ±5%"

to:

"1.5 kΩ (nominal)"

Note: No changes to test loads are proposed. All testing is to be done using the "nominal" values for pull-up (1.5K) and pull-down (15K) as currently specified.

Change sections 7.1.5.1, 7.1.5.2 as follows:

7.1.5.1 Low-/Full-speed Device Speed Identification

The USB is terminated at the hub and function ends as shown in Figure 7-20 and Figure 7-21. Full-speed and low-speed devices are differentiated by the position of the pull-up resistor on the downstream end of the cable:

Full-speed devices are terminated as shown in Figure 7-20 with the pull-up resistor on the D+ line.

Low-speed devices are terminated as shown in Figure 7-21 with the pull-up resistor on the D- line.

The pull-down terminators on downstream facing ports are resistors of 14.25 kΩ to 24.80 kΩ~~15 kΩ ±5%~~ connected to ground.

The design of the pull-up resistor must ensure that the signal levels satisfy the requirements specified in Table 7-2. In order to facilitate bus state evaluation that may be performed at the end of a reset, the design must be able to pull-up D+ or D- from 0 V to VIH (min) within the minimum reset relaxation time of 2.5 µs. A device ~~that has a detachable cable must~~may use a 1.5 kΩ ±5% resistor tied to a voltage source between 3.0 V and 3.6 V (V_{TERM}) to satisfy these requirements. Devices with captive cables may use alternative termination means; ~~h~~However, the Thevenin resistance of any termination must be no less than 900 Ω and the open-circuit voltage of the pull up must not exceed VBUS. These alternative means must produce an idle voltage (VIHZ) on D- or D+ of between 2.7 V and 3.6 V when terminated with a pull-down resistance of 15 kΩ±5% to ground.

Note: Thevenin resistance of termination does not include the 14.25 kΩ to 24.80 kΩ~~15 kΩ ±5%~~ resistor on the host/hub.

Alternative implementations for RPU are allowed on upstream facing ports on devices with detachable cables with the following restrictions:

When the bus is in the Idle state (including Suspend) or the upstream device is asserting SE0, the pull-up resistance must be between 900 Ω and 1575 Ω. The 1575 Ω maximum value ensures that the Idle voltage is at 2.7 V or higher even when a minimum value of pull down (14.25 kΩ) is present.

FCNVIEW.COM :Click to view the full PDF of IEC 62680-1:2013

- When the upstream device is transmitting data, the pull-up resistance shall switch to a value between 1425 Ω and 3090 Ω. The 1425 Ω minimum value ensures that the load presented by the switched resistance is no greater than the load presented by a 1.5 kΩ ±5%. The 3090 Ω maximum value is sufficient to ensure that the data line is held in a valid high state if the receiver has interpreted noise on the line as the start of a packet.
- When the upstream device is transmitting SE0, the pull-up resistance should be less than 3090 Ω but not less than 900 Ω. The resistance must be less than 3090 Ω to ensure that the data line will be pulled high within 2.5 µs if the upstream device stops driving the SE0 without driving the bus back to an Idle state.

If a variable RPU is implemented, then the resistance switches from its low-resistance value (900 Ω to 1575 Ω) to its high-resistance value (1425 Ω to 3090 Ω) when a J-K transition is detected on the bus. This change must occur within 0.5 bit times of the transition. The resistance is switched from its high-resistance value to its low-resistance value when SE0 is detected on the bus for more than 0.5 bit times, or when the bus has been in the J state for more than 7 bit times. An implementation may implement either or both of these schemes. Figure 7-22 illustrates an implementation of a variable RPU.

The nominal value for the pull-down resistors is 15 kΩ±5%. Values of up to 24.8 kΩ are allowed; however, when values exceeding 15.75 kΩ are used, allowance must be made for alternative pull-up means as described above. In particular, the pull-up resistor on the downstream device may be connected to VBUS instead of to a local 3.3 V supply with a resistor value as 7.2 kΩ. If a pull-down resistance of 24.8 kΩ is used, the idle voltage (VHIZ) on the D- or D+ lines can be as high as 4.1 V. Downstream ports using pull-down resistors greater than 15.75 kΩ and upstream ports of devices with captive cables using alternative termination means must be able to accommodate this idle voltage (VHIZ), which is greater than the value for (VHIZ) given in table 7.3.2.

The voltage source on the pull-up resistor must be derived from or controlled by the power supplied on the USB cable such that when VBUS is removed, the pull-up resistor does not supply current on the data line to which it is attached.

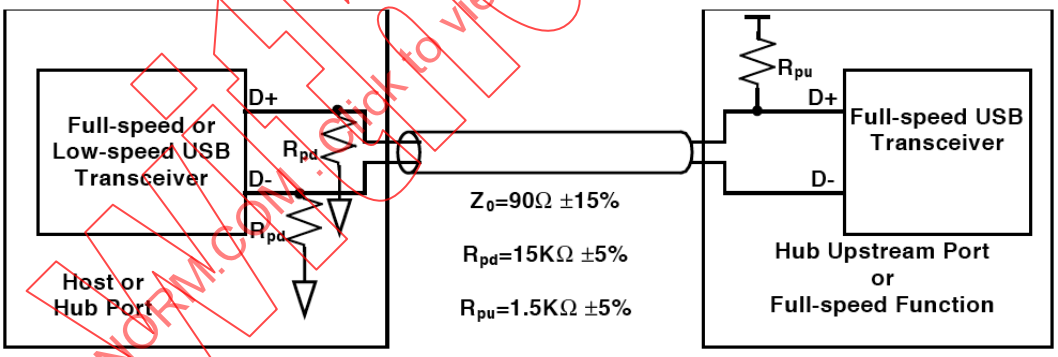


Figure 7-20. Full-speed Device Cable and Resistor Connections

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

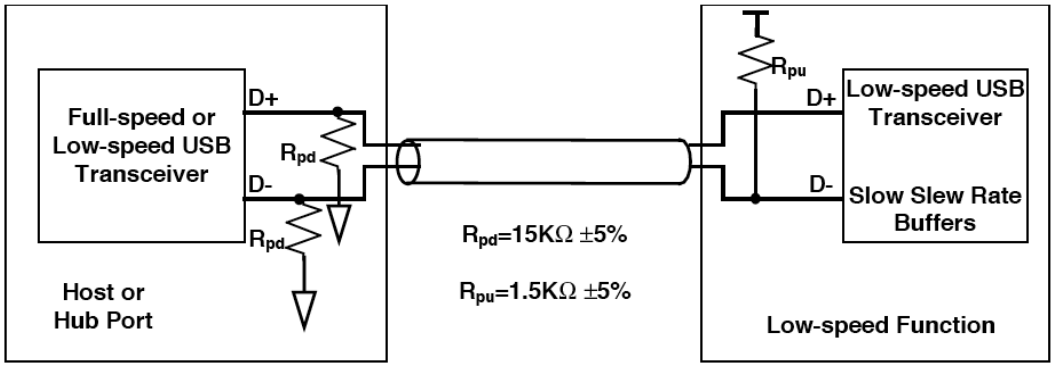
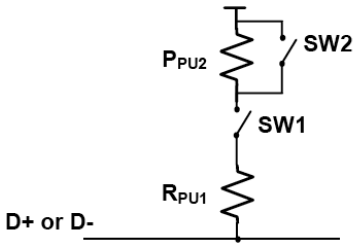


Figure 7-21. Low-speed Device Cable and Resistor Connections



Bus State	SW1	SW2
Idle	Closed	Closed
Receiving	Closed	Open
Transmitting	X	X
Vbus Off	Open	X

R_{PU1} = 900 Ω to 1.575 kΩ
R_{PU2} = 525 Ω to 1.515 kΩ

Figure 7-22. Variable Pull-up Example

7.1.5.2 High-speed Device Speed Identification

The high-speed ~~R~~reset and ~~D~~detection mechanisms follow the behavioral model for low~~-~~/full~~-~~speed. When reset is complete, the link must be operating in its appropriate signaling mode (low~~-~~speed, full~~-~~speed, or high~~-~~speed as governed by the preceding usage rules), and the speed indication bits in the port status register will correctly report this mode. Software need only initiate the assertion of reset and read the port status register upon notification of reset completion.

High-speed capable devices initially attach as full-speed devices. This means that for high-speed capable upstream facing ports, RPU (1.5 kΩ nominal~~±5%~~) must be connected from D+ to the 3.3 V supply (as ~~illustrated~~shown in Figure 7-1 or in 7-22) through a switch ~~which can be opened under SW control~~.

After the initial attachment, high-speed capable transceivers engage in a low level protocol during reset to establish a high-speed link and to indicate high-speed operation in the appropriate port status register. This protocol is described in Section 7.1.7.5.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

In Table 7-7 on page 178, change the following entry:

Table 7-7. DC Electrical Characteristics

Input Levels for Low-/full-speed:					
High (driven)	VIH	Note 4, Section 7.1.4	2.0		V
High (floating)	VIHZ	Note ~~4~~16, Section 7.1.4	2.7	3.6	V

Add note 16 to page 183 as follows:

Note 15: For high power devices (non-hubs) when enabled for remote wakeup.

Note 16: Measured at A connector with 1.5 kΩ±5% to 3.3 V ± 0.3 V connected to D+ or D- and at B connector with 15 kΩ±5% to ground connected to D+ and D-.

In Table 7-7 on page 179, change the following entries:

Table 7-7. DC Electrical Characteristics (Continued)

Terminations:					
Bus Pull-up Resistor on Upstream Facing Port	RPU	1.5 kΩ ±5% Section 7.1.5	1.425	1.575	kΩ
Bus Pull-down Resistor on Downstream Facing Port	RPD	1.5 kΩ ±5% Section 7.1.5	14.25	15.75	kΩ

to:

Terminations:					
Bus Pull-up Resistor on Upstream Facing Port (idle bus)	RPU	~~1.5 kΩ ±5%~~ Section 7.1.5	~~1.425~~0.900	1.575	kΩ
Bus Pull-up Resistor on Upstream Facing Port (upstream port receiving)	~~RPUA~~	Section 7.1.5	1.425	3.090	kΩ
Bus Pull-down Resistor on Downstream Facing Port	RPD	~~1.5 kΩ ±5%~~ Section 7.1.5	14.25	~~15.75~~24.80	kΩ

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Errata for "USB Revision 2.0 April 27, 2000" as of May 28, 2002:

Check the http://www.usb.org website for the latest errata.

Chapter 7

Clarify endpoint number used in Test_SE0_NAK:

Background: There is potential confusion over the endpoint number used in the IN packet sent to devices when in test mode Test_SE0_NAK. No known impact to host controllers or devices.

Change: p 170, section 7.1.20, 1st bullet (3rd sentence change):

- Test mode Test_SE0_NAK: Upon command, a port's transceiver must enter the high-speed receive mode and remain in that mode until the exit action is taken. This enables the testing of output impedance, low level output voltage, and loading characteristics. ~~In addition, while in this mode, upstream facing ports (and only upstream facing ports) must respond to any IN token packet with a NAK handshake (only if the packet CRC is determined to be correct) within the normal allowed device response time.~~In addition, while in this mode, upstream facing ports (and only upstream facing ports) must respond to any IN token packet directed to the device (with an endpoint number supported by the device), and optionally to any IN token packet at all, with a NAK handshake (only if the packet CRC is determined to be correct) within the normal device response time. This enables testing of the device squelch level circuitry and, additionally, provides a general purpose stimulus/response test for basic functional testing.

  Note: This means that the host should send test packets to a device under test with the device's current address and endpoint number zero to ensure that the device will respond.

Modify cable attenuation values to allow 5 meter cables:

Background: The USB 2.0 specification always intended to allow cables up to 5 meters in length. Analysis of the allowed cable attenuation values in the spec has shown that the value at 12.0 Mhz is transposed and should be .76 db not .67 db.

Change: p. 168, table 7-6: Change 12.0 MHz attenuation value from .67 to .76. A portion of the corrected table is shown below.

0.512	0.13
0.772	0.15
1.000	0.20
4.000	0.39
8.000	0.57
12.000	~~0.67~~0.76
24.000	0.95
48.000	1.35
96.000	1.9
200.00	3.2

Chapter 8

Clarify the intended behavior and specification of NAK'd OUTs:

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Background: There is a confusion over the correct usage and interpretation of NAK responses to Highspeed bulk/control OUT transactions. This change and a corresponding change in Chapter 9 clarify the required aspects of this USB feature. The intent was 1) to require a device to specify the maximum number of OUTs it could NAK in some period (this would be tested as part of compliance) and 2) to allow the host to limit the attempted OUT transactions to the number specified by the device. The current specification language is inconsistent and somewhat confusing in describing this. This change impacts high-speed devices with bulk OUT and control endpoints.

Change: p. 213, section 8.5.1, 5th paragraph, after 3rd sentence: Break for new paragraph. Change sentence starting “If a device responds with a NAK in a (micro)frame, the host . . .” to “If a device responds with a NAK to an OUT in a microframe, the host . . .”

Change:

A NAK handshake indicates that the endpoint does not have space for a *wMaxPacketSize* data payload. The host controller will retry the PING at some future time to query the endpoint again. A device can respond to a PING with a NAK for long periods of time. A NAK response is not a reason for the host controller to retire a transfer request. If a device responds with a NAK in a (micro)frame, the host controller may choose to issue the next transaction in the next *bInterval* specified for the endpoint. However, the device must be prepared to receive PINGs as sequential transactions, e.g., one immediately after the other.

To:

A NAK handshake indicates that the endpoint does not have space for a *wMaxPacketSize* data payload. The host controller will retry the PING at some future time to query the endpoint again. A device can respond to a PING with a NAK for long periods of time.

A NAK response is not a reason for the host controller to retire a transfer request. If a device responds with a NAK to an OUT in a (micro)frame, the host controller may choose to issue the next transaction in the next *bInterval* specified for the endpoint. However, the device must be prepared to receive PINGs as sequential transactions, e.g., one immediately after the other.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Chapter 9

Correct use of version 2.0 in the bcdUSB field of the device descriptor.

Background: There has been confusion over when 0200 should be used in the bcdUSB field in the device descriptor. Many people have assumed that 0200 is somehow associated with high speed peripherals and that full speed or low speed peripherals should not use it for their bcdUSB value. This is not correct. Any peripheral designed to the USB 2.0 specification should use 0200 as the bcdUSB value.

Change: Page 257, Section 9.6.1, 3rd paragraph.

Change paragraph:

The DEVICE descriptor of a high-speed capable device has a version number of 2.0 (0200H). If the device is full-speed only or low-speed only, this version number indicates that it will respond correctly to a request for the device_qualifier desciptor (i.e., it will respond with a request error).

To the following:

The DEVICE descriptor of a high-speed capable device has a version number of 2.0 (0200H). Full speed and low speed only devices designed to this specification should also use version number 2.0 (0200H). If the device is full-speed or low-speed only, this version number indicates that it will respond correctly to a request for the device_qualifier descriptor (i.e., it will respond with a request error).”

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Clarify the intended behavior and specification of NAK'd OUTs:

Background: There is a confusion over the correct usage and interpretation of NAK responses to Highspeed bulk/control OUT transactions. This change and a corresponding change in Chapter 9 clarify the required aspects of this USB feature. The intent was 1) to require a device to specify the maximum number of OUTs it could NAK in some period (this would be tested as part of compliance) and 2) to allow the host to limit the attempted OUT transactions to the number specified by the device. The current specification language is inconsistent and somewhat confusing in describing this. This change impacts high-speed devices with bulk OUT and control endpoints.

Change: p.265, table 9-13, description of bInterval: change bulk/control description from
"…at most 1 NAK each…"
to
"… after a NAK to an OUT that the next retry is after …".
At end of paragraph, add "See Section 8.5.1 for more PING detail

Table 9-13. Standard Endpoint Descriptor (Continued)

Offset	Field	Size	Value	Description
6	*bInterval*	1	Number	Interval for polling endpoint for data transfers. Expressed in frames or microframes depending on the device operating speed (i.e., either 1 millisecond or 125 µs units). For full-/high-speed isochronous endpoints, this value must be in the range from 1 to 16. The *bInterval* value is used as the exponent for a $2^{bInterval-1}$ value; e.g., a *bInterval* of 4 means a period of 8 (2^{4-1}). For full-/low-speed interrupt endpoints, the value of this field may be from 1 to 255. For high-speed interrupt endpoints, the *bInterval* value is used as the exponent for a $2^{bInterval-1}$ value; e.g., a *bInterval* of 4 means a period of 8 (2^{4-1}). This value must be from 1 to 16. For high-speed bulk/control OUT endpoints, the *bInterval* must specify the maximum NAK rate of the endpoint. A value of 0 indicates the endpoint never NAKs. Other values indicate ~~at most 1 NAK each~~ after a NAK to an OUT that the next retry is after *bInterval* number of microframes. This value must be in the range from 0 to 255. See Section 8.5.1 for more PING detail. See Chapter 5 description of periods for more detail.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Chapter 10

Add host software requirement to suspend downstream ports before suspending a hub:

Background: There is a very unlikely possibility of a race condition between a remote wakeup event and the process of suspending a hub that could create unintended signaling events. This issue can be avoided if system software suspends a downstream facing hub ports before suspending a hub.

Change: p. 276, section 10.2.7, Add new paragraph, “The USB system should suspend all downstream facing ports of a hub before suspending the hub.”

10.2.7 Remote Wakeup

The USB system should suspend all downstream facing ports of a hub before suspending the hub.

If USB System wishes to place the bus in the Suspended state, it commands the Host Controller to stop all bus traffic, including SOFs. This causes all USB devices to enter the Suspended state. In this state, the USB System may enable the Host Controller to respond to bus wakeup events. This allows the Host Controller to respond to bus wakeup signaling to restart the host system.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Chapter 11

Allow orange in addition to amber LEDs for downstream facing hub ports:

Background: Some LED vendors do not provide any Green/Amber (bi-color) LEDs. However, almost all LED vendors provide Green/Orange LEDs. Allow either Amber or Orange bi-color LEDs.

Change: Change references from “amber” to “amber (or orange)”

Different port indicator behavior for hubs with/without power switching.

Background: The specification currently indicates that port indicator states are not necessarily consistent in automatic mode for hubs with/without power switching. This spec allowance is being removed.

Change: p. 308, Table 11-6. Remove the column for power switching and the row for without power switching.

Table 11-6. Automatic Port State to Port Indicator Color Mapping

~~Power Switching~~	Downstream Facing Hub Port State			
	Powered-off	**Disconnected, Disabled, Not Configured, Resetting, Testing**	**Enabled, Transmit, or TransmitR**	**Suspended, Resuming, SendEOR, Restart_E, or Restart_S**
~~With~~	Off or amber if due to an over-current condition	~~Off~~	Green	Off
~~Without~~	~~Off~~	~~Off or amber if due to an over-current condition~~	~~Green~~	~~Off~~

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Port indicator behavior through a hub suspend/resume cycle.

Background: There has been some confusion over how the port indicators on a hub should behave if the hub is suspended and then resumed. The core specification states that port indicators should be off when a hub is suspended. This is true whether the indicator is in automatic or manual mode. Port indicator behavior should be consistent and this behavior may be necessary for bus-powered hubs. Furthermore, when the hub is resumed the port indicator should return to its previous state before the hub was suspended.

Change: p. 307, 2nd paragraph. Add the following to the end of the paragraph. When the hub is suspended the port indicator must be off regardless of its state before the hub was suspended. When the hub is resumed, the port indicator must return to its state before the hub was suspended, unless the port indicator is in automatic mode and a port status change requires the indicator to change.

11.5.2.2 Full-/low-speed Disconnect Detection

Each port is required to have a timer used for detecting disconnect when a full-/low-speed device is attached to the port. This timer is used to constantly monitor the port's single-ended receivers to detect a disconnect event. The reason for constant monitoring is that a noise event on the bus can cause the attached device to detect a reset condition on the bus after 2.5 μs of SE0 or SE1 on the bus. If the hub does not place the port in the disconnect state before the device resets, then the device can be at the Default Address state with the port enabled. This can cause systems errors that are very difficult to isolate and correct.

This timer must be reset whenever the D+ and D- lines on the port are not in the SE0 or SE1 state or when the port is not in the Enabled, Suspended, Disabled, Restart-E, or Restart_S states. This timer must be reset for 4ms upon entry to the Suspended and Disabled states. This timer times an interval TDDIS. The range of TDDIS is 2.0 μs to 2.5 as defined in Table 7-13. When this timer expires, it generates the Disconnect_Detect signal to the port state machine. When the hub is suspended the port indicator must be off regardless of its state before the hub was suspended. When the hub is resumed, the port indicator must return to its state before the hub was suspended, unless the port indicator is in automatic mode and a port status change requires the indicator to change.

Change: p. 308, Figure 11-11. Add the following note to the figure: When the hub is suspended the port indicator must be off regardless of its state before the hub was suspended. When the hub is resumed, the port indicator must return to its state before the hub was suspended, unless the port indicator is in automatic mode and a port status change requires the indicator to change.

Figure 11-11. Port Indicator State Diagram

Note: When the hub is suspended the port indicator must be off regardless of its state before the hub was suspended. When the hub is resumed, the port indicator must return to its state before the hub was suspended, unless the port indicator is in automatic mode and a port status change requires the indicator to change.

In **Manual Mode** the color of a port indicator (Amber, Green, or Off) is set by a system software USB Hub class request. In **Automatic Mode** the color of a port indicator is set by the port state information.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Clarify the treatment of remote wake up (RWU) in the hub repeater:

Background: There is some potential confusion in the understanding of remote wake up in the hub repeater. These changes make more explicit some of the handling behavior of the hub repeater. This change may impact USB 2.0 hubs.

Change:

p. 301, section 11.4.4: replace “and SOP_FD…Repeater” with “TrueRWU from a downstream port”.

11.4.4 Generate Resume (GResume)

This state is entered from the Fsus state when a resume condition exists in the Hub Controller. A resume condition exists if the C_PORT_SUSPEND bit is set in any port, or if the hub is enabled as a wakeup source and any bit is set in a Port Change field or the Hub Change field (as described in Figures 11-22 and 11-20, respectively).

In this state, the internal port generates signaling to emulate an ~~SOP_FD to the Hub Repeater~~TrueRWU from a downstream port.

p. 305, section 11.5.1.6: delete the third bullet “From the Suspended state…”.

11.5.1.6 Enabled

A port transitions to this state in any of the following situations:

- At the end of the Resetting state
- From the Transmit state or the TransmitR state when the Hub Repeater exits the WFEOPFU state
- ~~From the Suspended state if the upstream Receiver is in the Suspend state when a 'K' is detected on the port~~
- At the end of the SendEOR state
- From the Restart_E state when a persistent K or persistent SE0 has not been seen within 900 μs of entering that state

p. 307, sections 11.5.1.12 and .13 2nd paragraph, 2nd sentence: delete the phrase “…to the repeater…”.

11.5.1.12 Restart_S

A port enters the Restart_S state from the Suspended state when an SE0 or ‘K’ is seen at the port and the Receiver is in the Suspended state.

In this state, the port continuously monitors the bus state. If the bus is in the ‘K’ state for at least TDDIS, the port sets the C_PORT_SUSPEND bit, exits to the TransmitR, and generates a signal ~~to the repeater~~ called ‘TrueRWU’. If the bus is in the ‘SE0’ state for at least TDDIS, the port exits to the Disconnected state. Either of these transitions must happen within 900 μs after entering the Restart_S state; otherwise, the port must transition back to the Suspended state.

11.5.1.13 Restart_E

A port enters the Restart_E state from the Enabled state when an ‘SE0’ or ‘K’ is seen at the port and the Receiver is in the Suspended state.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

In this state, the port continuously monitors the bus state. If the bus is in the 'K' state for at least TDDIS, the port exits to the TransmitR state and generates a signal ~~to the repeater~~ called 'TrueRWU'. If the bus is in the 'SE0' state for at least TDDIS, the port exits to the Disconnected state. Either of these transitions must happen within 900 µs after entering the Restart_E state; otherwise the port must transition back to the Enabled state.

p.313, Figure 11-13: replace "Rptr_WFEOP" with "TrueRWUor".

p. 313, section 11.6.4, table 11-9: replace row "Rptr_WFEOP" with:

J	Internal	Transmitter transitions to sending a 'J' and transmits a 'J'
~~Rptr_WFEOP~~ TrueRWUor	~~Hub Repeater~~ Downstream port	Hub Repeater is in the WFOEP state Logic OR of TrueRWU from Internal port or from downstream ports.
K	Internal	Transmitter transmits a 'K'

p. 314, section 11.6.4.6: replace 1st sentence with. "The port enters this state from the Inactive state on TrueRWU from a downstream port."

11.6.4.6 Send Resume (Sresume)

The port enters this state from the Inactive state on TrueRWU from a downstream port~~if the Receiver is in the Suspend state and the Hub Repeater transitions to the WFEOP state~~. This indicates that a downstream device (or the port to the Hub Controller) has generated resume signaling causing upstream connectivity to be established.

p. 415, section 11.24.2.7.2.3: replace "Send_EOP" with "Send_EOR"

11.24.2.7.2.3 C_PORT_SUSPEND

This bit is set on the following transitions:

- On transition from the Resuming state to the ~~SendEOP~~Send_EOR state
- On transition from the Restart_S state to the Transmit state

This bit will be cleared by a ClearPortFeature(C_PORT_SUSPEND) request, or while the port is in the Powered-off state.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Clarify handling of complete-split isochronous IN DATAx PIDs:

Background: USB 2.0 specification is ambiguous about HC and TT requirements in expected DATA packet PIDs allowed during a high-speed complete-split isochronous IN DATA packet. Clarify that the DATAx packet PID returned by the TT for an isochronous complete split is the same DATAx packet PID returned by the TT for the corresponding full-speed transaction. An HC may therefore receive (and accept as valid) a DATA1 packet PID if the originating full-speed device sends a DATA1. No known impact on HC and Hub TT implementations in progress.
Note: It is also acceptable for the TT to always use DATA0 in this case – even if the device sends a DATA1.

Change: p. 386, Add an asterix in Figure 11-85 on the DATA0 (cd1 transition) “bubble”. Add text “*TT accepts a DATA0 or DATA1 data packet received from downstream full-speed bus and returns that packet in the corresponding complete-split on the high-speed bus. A TT may choose to always send a DATA0, even if it receives a DATA1.”

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Clarification of Allowed Behavior for Data Carve Up Around Microframe Boundaries:

Background: Chapter 11 currently implies that Transaction Translators use a byte data engine and should respond with all data except for the last two bytes received in microframe X in response to the complete split in microframe X+1 for an isochronous or interrupt IN transaction that spans X/X+1 microframe boundary on the full/low speed bus. Examples of this behavior are shown in figure 11-82 and 11-93 in Chapter 11. However, the common industry design for USB 2.0 hubs have used a DWORD based data engine. With these designs a transaction translator "effectively" is not aware of data until it has a full DWORD. This means that up to 5.99999 . . . bytes could be carried forward to the next microframe (the two reserved potential CRC bytes plus anything up to another full DWORD of data). As a result of this behavior a Transaction Translator could return as many as 192 data bytes in response to an Isochronous IN complete split. The question has arisen as to whether this should be legal behavior. Several hub implementations have already been designed with this behavior. Changing the size of the data path would add significant delays and cost to USB 2.0 hub implementations. The issue was also not clearly discussed across the EHCI and core specifications. Therefore, the core spec is being modified to allow the transaction translator to carry forward up to (but not including) a full DWORD of data in addition to the 2 potential CRC bytes when an Isochronous or Interrupt IN transaction spans a microframe boundary.

A change is also being added to the EHCI specification to clearly specify that host must be able to handle up to 192 data bytes in response to an Isochronous IN complete split.

Note: If the transaction does end in microframe X (no matter how close to the microframe X+1 boundary) the transaction translator must respond with all data to the complete split in microframe X+1.

Change: Ch. 11, p. 366, section 11.18.5, 1st paragraph. Add the following sentences to the end of the paragraph: "The TT must hold back at least the last two data bytes to prevent returning the CRC for an interrupt or isochronous IN transaction that spans the microframe boundary. The TT is allowed to hold back up to (but not including) an additional DWORD (4 bytes). The TT must provide all remaining data to the complete split in the microframe immediately following the microframe where the transaction ended (no matter how close to the microframe boundary the transaction ends).

11.18.5 TT Response Generation

The approach used for full-speed isochronous INs and interrupt INs/OUTs ensures that there is always an opportunity for the TT to return data/results whenever it has something to return from the full/low-speed transaction. Then whenever the full-/low speed handler starts the full-/low speed transaction, it simply accumulates the results in each microframe and then returns it in response to a complete-split from the host. The TT acts similar to an isochronous device in that it uses the microframe boundary to "carve up" the full-/low speed data to be returned to the host. The TT does not do any computation on how much data to return at what time. In response to the "next" high-speed complete-split, the TT simply returns the endpoint data it has received from the full-/low-speed bus in a microframe. The TT must hold back at least the last two data bytes to prevent returning the CRC for an interrupt or isochronous IN transaction that spans the microframe boundary. The TT is allowed to hold back up to (but not including) an additional DWORD (4 bytes). The TT must provide all remaining data to the complete split in the microframe immediately following the microframe where the transaction ended (no matter how close to the microframe boundary the transaction ends). (The boundary is defined as the beginning of the micro SOF based on the hubs internal timer.)

Change: Ch. 11, p. 383, section 11.20.4, last paragraph. Add the following sentences as a new paragraph below figure 11-82. "The transaction translator must hold back at least the last 2 bytes for an interrupt IN transaction spanning a microframe boundary. It may hold back up

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

to (but not including) the last 6 bytes (2 for the CRC and up to an additional DWORD to accommodate DWORD based data paths)."

Figure 11-82. Example of CRC16 Handling for Interrupt IN

The transaction translator must hold back at least the last 2 bytes for an interrupt IN transaction spanning a microframe boundary. It may hold back up to (but not including) the last 6 bytes (2 for the CRC and up to an additional DWORD to accommodate DWORD based data paths).

Change: Ch. 11, p. 392, section 11.21.4, last paragraph. Add the following sentences as a new paragraph below figure 11.93. "The transaction translator must hold back at least the last 2 bytes for an isochronous IN transaction spanning a microframe boundary. It may hold back up to (but not including) the last 6 bytes (2 for the CRC and up to an additional DWORD to accommodate DWORD based data paths). Thus a TT could respond with up to 192 bytes in response to an isochronous IN complete split."

Figure 11-93. Example of CRC16 Isochronous IN Data Packet Handling

The transaction translator must hold back at least the last 2 bytes for an isochronous IN transaction spanning a microframe boundary. It may hold back up to (but not including) the last 6 bytes (2 for the CRC and up to an additional DWORD to accommodate DWORD based data paths). Thus a TT could respond with up to 192 bytes in response to an isochronous IN complete split.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

How should a TT handle packet babble from a device on a control or bulk transfer?

Background: The core spec never specifically addresses the case where a device responds with more bytes than the maximum allowed size for a control or bulk transfer. In any such case the goal of the TT should be to ensure the behavior from the device's point of view is the same is it would be below a UHCI or OHCI host. Since an OHCI or UHCI host would not retry in these cases – neither must the TT. One option for the TT is simply to return the bytes as they were sent by the device. The host can detect and deal with the babble. However, if the TT uses a minimal buffering solution this may not be possible. The other option for the TT is to return a STALL result for the transfer WITHOUT retrying it locally.

Change: Ch. 11, p. 361, Section 11.17.6. Add the following section below section 11.17.5.

11.17.6 Bulk/Control packet babble handling.

If a TT receives a bulk or control packet from a device that exceeds the maximum byte size allowed by the spec for that transfer type it has two options:

1. Return the data to the host as received.
2. Return STALL WITHOUT retrying the transaction locally.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.18 Periodic Split Transaction Pipelining and Buffer Management

How should a host handle a NAK response from the device to a setup performed using the split transaction protocol?

Background: The transaction sequence diagrams in the core specification are ambiguous with respect to how a host controller should handle a NAK response to the complete split for a control setup transaction. Immediately retrying the start split without incrementing err_count will be inefficient in what is almost certainly a device protocol error case. In this case the host should increment err_count and only retry the start split if err_count is less than three.

Change: Ch. 11, p. 353, Figure 11-49. Add the following to the Host response to the NAK response (currently says only Retry start split). If the complete split is for a Setup increment err_count. If err_count >= 3 endpoint halt, if err_count < 3 retry start split.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Meaning of bHubContrCurrent field in Hub Descriptor:

Background: There has been some confusion over the meaning of the bHubContrCurrent field in the Hub Descriptor. Some vendors have assumed the field should be current usage of the hub controller electronics (whether from VBUS or from an external power supply). They felt it would be redundant with the bMaxPower field in the configuration descriptor to report consumption from VBUS again. Other vendors have assumed that field should reflect the consumption of the hub controller electronics from VBUS without the current draw of any embedded devices (bMaxPower for a hub can include the draw from embedded devices - as long as the embedded devices don't also report current draw in their own bMaxPower fields). The core spec is being clarified to indicate the second interpretation must be used. This allows software to intelligently perform power management in cases where embedded devices have their power requirements included with the hub's bMaxPower field.

Change: Ch. 11, p. 404, offset 6 in table Change the Description of the bHubContrCurrent from "Maximum current requirements of the Hub Controller electronics in mA" to "Maximum current requirements of the Hub Controller electronics from VBUS. This value must not include current draw for any embedded devices."

11.23.2.1 Hub Descriptor

5	bPwrOn2PwrGood	1	Time (in 2 ms intervals) from the time the power-on sequence begins on a port until power is good on that port. The USB System Software uses this value to determine how long to wait before accessing a powered-on port.
6	bHubContrCurrent	1	Maximum current requirements of the Hub Controller electronics ~~in mA.~~from VBUS. This value must not include current draw for any embedded devices.
7	DeviceRemovable	Variable, depend-ing on number of ports on hub	Indicates if a port has a removable device attached. This field is reported on byte-granularity. Within a byte, if no port exists for a given location, the field representing the port characteristics returns 0. Bit value definition: 0B - Device is removable. 1B - Device is non-removable This is a bitmap corresponding to the individual ports on the hub: Bit 0: Reserved for future use. Bit 1: Port 1 Bit 2: Port 2 .... Bit *n*: Port *n* (implementation-dependent, up to a maximum of 255 ports).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Downstream port behavior after Stop_TT request.

Background: There have been questions over whether a hub should continue to generate SOFs on an enabled downstream port controlled by a TT that has been stopped via a Stop_TT request. The hub should continue to generate SOFs in this situation.

Change: Ch. 11, p. 417, section 11.24.2.11. 1st paragraph. Add the following sentence to the end of the first paragraph. “Any downstream port that was generating SOFs controlled by a transaction translator that is being stopped should continue to do so.”

11.24.2.11 Stop_TT

This request stops the normal execution of the transaction translator so that the internal TT state can be retrieved via Get_TT_State. This request is provided for debugging purposes. Any downstream port that was generating SOFs controlled by a transaction translator that is being stopped should continue to do so.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00100011B	STOP_TT	Zero	TT_Port	Zero	None

The only standardized method to restart a TT after a Stop_TT request is via the Reset_TT request.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

USB 2.0 Spec Errata dated 12/07/2000

Standard Descriptors for Hub Class:

Background: The 12/07/2000 Errata for the USB 2.0 Specification contained several errata to the standard descriptors for the hub class. One additional change to these errata is still required. The Other_Speed_Configuration Descriptor for a full speed operating hub should always have bNumInterfaces set to 1 (not 1 for single TT and 2 for multiple TT). A multiple TT hub has two alternate settings not two interfaces.

Change: 12/07/2000 Errata, page 635. Change text in Other_Speed_Configuration Descriptor bNumInterfaces field to "1".

Other_Speed_Configuration Descriptor (High-speed information):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1 ~~(for single TT) or 2 (for multiple TT)~~
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	The maximum amount of bus power the hub will consume in high-speed configuration

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

USB ENGINEERING CHANGE NOTICE

Title: Interface Association Descriptors

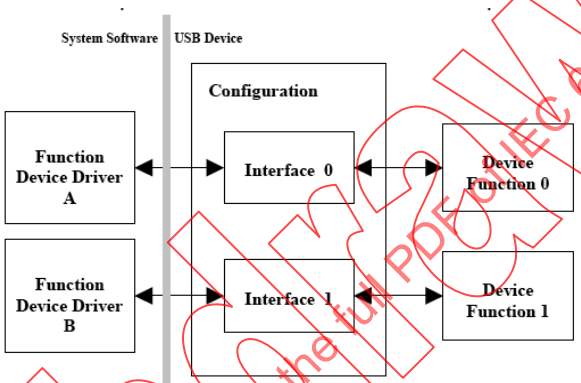
Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN

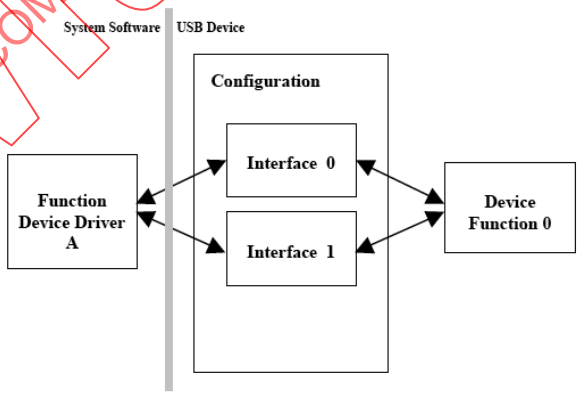
This ECN defines a new standard descriptor and interface numbering rules that allow a device to describe which interfaces are associated with the same device function. This allows the operating system to bind all of the appropriate interfaces to the same driver instance.

Reasons for ECN

The base configuration model assumed by the core USB framework was that there was always a 1:1 association between an interface and a function on a device. System software was designed to the intent of the core specification and assumes one driver per function (and one interface) (see figure below).



Several device class specifications have exceeded the core USB specification framework and defined device functions that utilize multiple interfaces (i.e. multiple interface descriptors). The model to support this still requires only one function driver per function, but also requires multiple interfaces getting bound to the same driver instance (see figure below). Unfortunately, there is no standard method to allow a device, via the device framework, to describe which interfaces in a configuration should be associated with the same function.



This change notice defines the necessary extensions to the device framework that allow the device to annotate which interfaces are associated with the same function. This device framework extension will be eventually required for all devices that utilize multiple interfaces per device function.

Impact on Existing Peripherals and Systems:

No Impact.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Hardware Implications:
None.

Software Implications:
No impact to existing operating system versions and existing device classes. The new descriptors are ignored by system software. The interfaces they describe don't change.

This feature must be supported by future implementations of devices that use multiple interfaces to manage a single device function.

There exists an impact to future/new device implementations for device classes that are not currently supported by the OS. Specifically, if a device implementation includes multiple functional units (with multiple interfaces per unit) then the device will only be correctly enumerated on an OS implementation that supports this new descriptor.

In order to more easily enhance existing OS implementations with the capability to handle devices that use this descriptor, a device class code will be allocated with the intent that all devices that use the interface association descriptor will use this class code in their device descriptor. This will allow easy installation of a new driver that knows how to parse and enumerate configurations that include the interface association descriptor. The class code for the IAD will be documented on the usb.org website.

It is under the responsibility of the existing Device Class working groups to determine whether their individual specifications need to be modified to work with or take advantage of this new framework extension.

Compliance Testing Implications:
The standard compliance toolset (USBCV) must be eventually updated to check the format (and use) of these new descriptors. In addition, some rules must be established for the compliance tools to determine which device should be using these descriptors and fail them for not using them.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Specification Changes

The following proposal is backward compatible with previous operating systems. It leaves current interface definitions alone and adds a new descriptor type. For older operating systems versions, the new descriptors will be ignored and the old mechanisms will prevail. For new operating system versions, the new descriptors will be in effect.

Add the following to table 9-5 (note, the core specification currently defines values 1-8. Since publication, values 9 & 10 have been allocated as noted below).

Table 9-5. Descriptor Types

Descriptor Types	Value
DEVICE	1
CONFIGURATION	2
STRING	3
INTERFACE	4
ENDPOINT	5
DEVICE_QUALIFIER	6
OTHER_SPEED_CONFIGURATION	7
INTERFACE_POWER	8
OTG	9
DEBUG	10
INTERFACE_ASSOCIATION	11

The following is the definition of an Interface Association Descriptor. At the next spec revision update, it should be included as a new section 9.6.6 (moving the current section 9.6.6 to 9.6.7, and so-on).

9.6.6 Interface Association

The Interface Association Descriptor is used to describe that two or more interfaces are associated to the same function. An 'association' includes two or more interfaces and all of their alternate setting interfaces. A device must use an Interface Association descriptor for each device function that requires more than one interface. An Interface Association descriptor is always returned as part of the configuration information returned by a GetDescriptor(Configuration) request. An interface association descriptor cannot be directly accessed with a GetDescriptor() or SetDescriptor() request. An interface association descriptor must be located before the set of interface descriptors (including all alternate settings) for the interfaces it associates. All of the interface numbers in the set of associated interfaces must be contiguous. Table 9-Z shows the standard interface association descriptor. The interface association descriptor includes function class, subclass and protocol fields. The values in these fields can be the same as the interface class, subclass and protocol values from any one of the associated interfaces. The preferred implementation, for existing device classes, is to use the interface class, subclass and protocol field values from the first interface in the list of associated interfaces.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 9-13. Standard Interface Association Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	INTERFACE ASSOCIATION Descriptor Type
2	*bFirstInterface*	1	Endpoint	Interface number of the first interface that is associated with this function.
3	*bInterfaceCount*	1	Number	Number of contiguous interfaces that are associated with this function.
4	*bFunctionClass*	1	Class	Class code (assigned by USB-IF). A value of zero is not allowed in this descriptor. If this field is FFH, the function class is vendor-specific. All other values are reserved for assignment by the USB-IF.
5	*bFunctionSubClass*	1	SubClass	Subclass code (assigned by USB-IF). If the *bFunctionClass* field is not set to FFH all values are reserved for assignment by the USB-IF.
6	*bFunctionProtocol*	1	Protocol	Protocol code (assigned by USB-IF). These codes are qualified by the values of the *bFunctionClass* and *bFunctionSubClass* fields.
7	*iFunction*	1	Index	Index of string descriptor describing this function.

Note: Since this particular feature was not included in earlier versions of the USB specification, there is issue with how existing USB OS implementations will support devices that use this descriptor. It is strongly recommended that device implementations utilizing the *interface association descriptor* use the Multi-interface Function Class codes in the device descriptor. This allows simple and easy identification of these devices and allows on some operating systems, installation of an upgrade driver that can parse and enumerate configurations that include the Interface Association Descriptor. The Multi-interface Function Class code is documented on the http://www.usb.org/developers/docs website.

9.6.~~6~~7 Endpoint

Each endpoint used for an interface has its own descriptor. This descriptor contains the information required by the host to determine the bandwidth requirements of each endpoint. An endpoint descriptor is always returned as part of the configuration information returned by a GetDescriptor(Configuration) request. An endpoint descriptor cannot be directly accessed with a GetDescriptor() or SetDescriptor() request. There is never an endpoint descriptor for endpoint zero. Table 9-13 shows the standard endpoint descriptor.

Table 9-~~13~~14. Standard Endpoint Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	ENDPOINT Descriptor Type

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

USB ENGINEERING CHANGE NOTICE

Title: Rounded Chamfer

Applies to: USB 2.0 Specification

Summary of ECN
This ECN defines a rounded chamfer design option for the mini-B plug, which is recommended for all new and existing designs. The previous flat chamfer design is allowed, but not recommended.

Reasons for ECN
Consumers benefit from this ECN because rounded chamfer mini-B plugs result in longer lasting mini-B receptacles. Manufacturers also benefit. Connectors with rounded chamfer plugs can achieve more mating cycles than connectors with flat chamfer plugs. This allows the rounded chamfer connectors to be used in applications that may require more durability than the flat chamfer plugs can achieve.

Assessment of Impact on Current Specification and Current USB Products
This ECN modifies the Mini-B Connector ECN to the USB 2.0 Specification by replacing Figure 6-15 with the two attached figures. Figure 1 below shows both the rounded chamfer and flat chamfer plug options. The rounded chamfer option includes a note recommending it for new and existing designs. The flat chamfer option includes a note indicating that it is allowed, but not recommended for new and existing designs. Users of current USB products are not affected by the change from flat chamfer to rounded chamfer mini-B plugs. Both types of plugs mate the same with mini-B receptacles. The only difference is the increased life of the receptacle when using the rounded chamfer plugs.

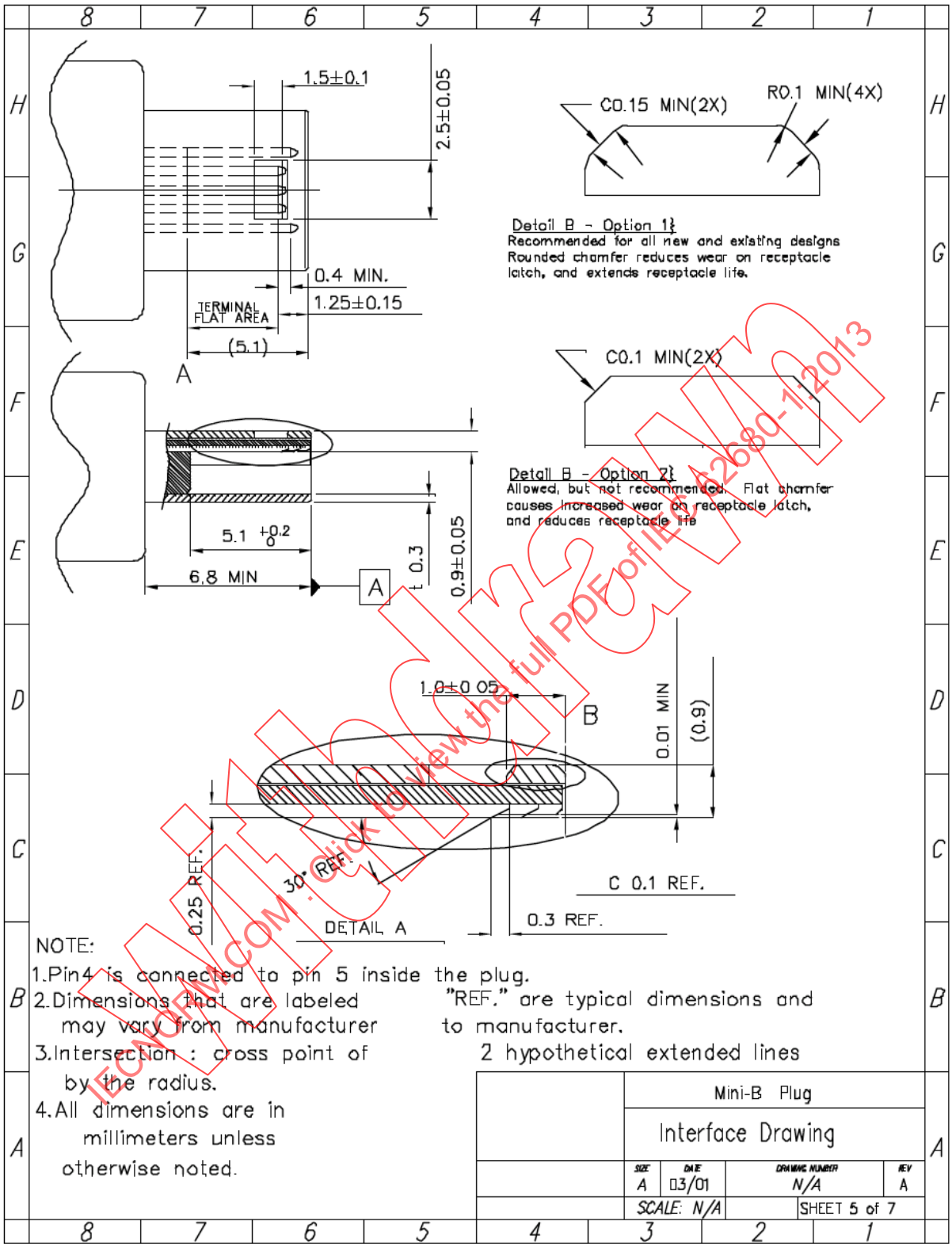
Hardware Implications
There are no issues. Receptacle life is increased with the use of rounded chamfer plugs.

Software Implications
None

Compliance Testing Implications
There are no compliance testing issues. Device mfrs are still allowed to use flat chamfer plugs if they so desire. The OTG checklist does include a question that asks device mfrs if the plug has a flat chamfer or a round chamfer.

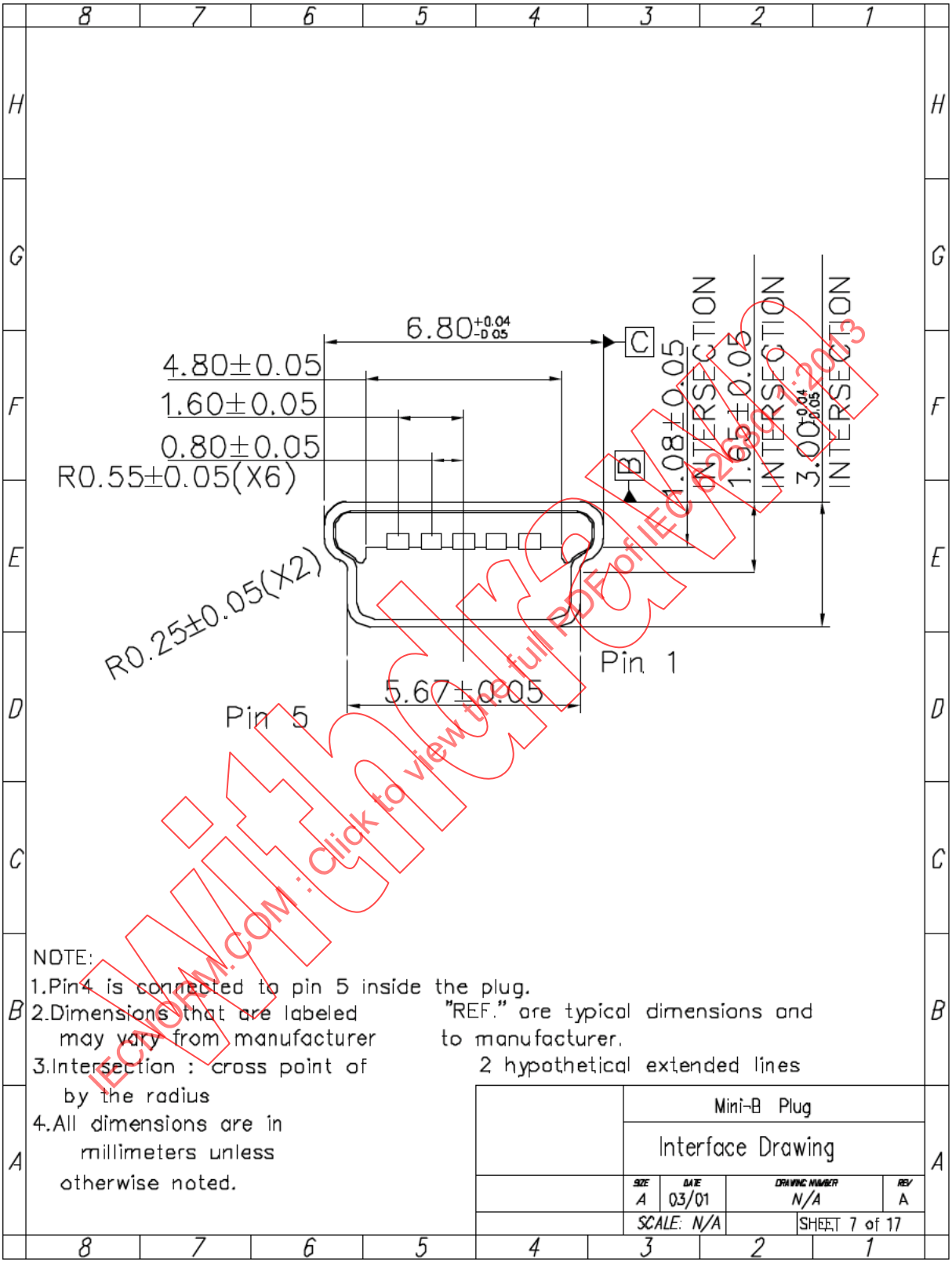
Specification Changes
This ECN modifies the Mini-B Connector ECN of the USB 2.0 Specification by replacing Figure 6-15 with the two attached figures.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 6-15. USB Series "Mini-B" Plug Interface Drawing (1 of 2)



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 6-15. USB Series "Mini-B" Plug Interface Drawing (2 of 2)

USB ENGINEERING CHANGE NOTICE

February 21, 2005

Title: UNICODE UTF-16LE for String Descriptors

Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN
This ECN more completely specifies the encoding scheme for string descriptors. String descriptors must use UNICODE UTF-16LE encodings.

Reasons for ECN
To ensure consistent coding and decoding of string descriptors, the spec has to be clear on the exact encoding to use. The original spec said to use UNICODE encodings, but there are multiple encoding schemes that make up UNICODE. UTF-16LE was the intended encoding scheme, and the one assumed by most developers.

Impact on Existing Peripherals and Systems
USB compliance testing checks for proper encoding of string descriptors on peripherals. There are no known cases of devices using an encoding other than UTF-16LE. There is at least one OS that has been using UTF-8 encoding and ignoring null bytes. Since most devices have been encoding ASCII characters resulting in the upper byte being NULL, this has not been a problem.

Hardware Implications
None expected.

Software Implications
Any software not using UTF-16LE will need to change.

Compliance Testing Implications
None. USB compliance testing already checks string descriptors for proper UTF-16LE encoding.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Specification Changes

Section 9.6.7 String, p. 267

Change This:

…

String descriptors use UNICODE encodings as defined by *The Unicode Standard, Worldwide Character Encoding, Version 3.0*, The Unicode Consortium, Addison-Wesley Publishing Company, Reading, Massachusetts (URL: http://www.unicode.com).

…

To this:

…

String descriptors use UNICODE UTF-16LE encodings as defined by *The Unicode Standard, Worldwide Character Encoding, Version 4.0*, The Unicode Consortium, Addison-Wesley Publishing Company, Reading, Massachusetts (URL: http://www.unicode.org).

…

9.6.7 String

String descriptors are optional. As noted previously, if a device does not support string descriptors, all references to string descriptors within device, configuration, and interface descriptors must be reset to zero.

String descriptors use UNICODE UTF-16LE encodings as defined by *The Unicode Standard, Worldwide Character Encoding, Version ~~3.0~~4.0*, The Unicode Consortium, Addison-Wesley Publishing Company, Reading, Massachusetts (URL: http://www.unicode.~~com~~org). The strings in a USB device may support multiple languages. When requesting a string descriptor, the requester specifies the desired language using a sixteen-bit language ID (LANGID) defined by the USB-IF. The list of currently defined USB LANGIDs can be found at http://www.usb.org/developers/docs.html. String index zero for all languages returns a string descriptor that contains an array of two-byte LANGID codes supported by the device. Table 9-15 shows the LANGID code array. A USB device may omit all string descriptors. USB devices that omit all string descriptors must not return an array of LANGID codes.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 9-16, p. 267

Change Description of bString from this:

UNICODE encoded string

To this:

UNICODE UTF-16LE encoded string

Table 9-16. UNICODE String Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	STRING Descriptor Type
2	*bString*	N	Number	UNICODE UTF-16LE encoded string

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

USB ENGINEERING CHANGE NOTICE

Title: USB 2.0 Link Power Management Addendum Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN

This ECN defines a power management feature for USB called Link Power Management (LPM) which is similar to the existing suspend/resume, but has transitional latencies of tens of microseconds between power states (instead of three to greater than 20 millisecond latencies of the USB 2.0 suspend/resume). It has several parts: (a) it introduces formal terminology for line (bus) states in terms of power states, (b) defines a new, fast mechanism for transitioning the bus on a root port from an enabled state (called L0, see below) to a new Sleep state (called L1, see below), (c) defines new, fast timing requirements for resuming a bus from L1 back to (L0), and (d) defines the method for extending the existing USB protocol to accommodate explicit L1 entry. The L0 (enabled) to L1 (link Sleep) transition is actively initiated by the host using a new USB-defined transaction. This allows a device to immediately detect the intent of the host and respond immediately, instead of the USB 2.0 defined method of 'discovery by loss of activity', which requires 3 orders of magnitude more elapsed time. The L1 to L0 transition is similar to the USB 2.0 resume, as it can be initiated by either the host port or the device (i.e. remote wake). However, the duration of resume signaling is redefined to be three orders of magnitude faster (i.e. tens of microseconds, not tens of milliseconds). The LPM feature may be utilized at any depth in a valid USB connection hierarchy. However, it is expected that the most value to be obtained will be when it is used with root-ports and possibly a single level of hub beyond the root-ports. This is motivated primarily by the fact that the goal is to minimize the transitional latencies. Supporting multiple layers in the hierarchy requires driving devices across the hierarchy of hubs in and out of an L1, in a consistent and coherent fashion takes additional time, which increases latencies to unacceptable durations. This is an addendum to the USB 2.0 specification. Note there are elements in this ECN that override parts of the USB 2.0 specification. These are called out explicitly where appropriate.

Reasons for ECN

USB 2.0 defined standard mechanisms for device power management, based on a simple protocol for detecting suspend and special signaling protocol for allowing either a host or device to initiate resume. USB is an increasingly popular standard bus to connect peripherals both outside and inside the box. The currently supported USB device power management mechanisms, although simple are large grain controls and not intended to be used for micro-managing or optimizing device power. For example, in order to 'suspend' a device, a host will suspend traffic on the port the device is connected. The device 'discovers' the suspend condition after 3 milliseconds of idle. The device is required to obey a minimum suspend residency of 2 milliseconds (i.e. can initiate a resume only after 2 additional milliseconds after detecting suspend). The resume timing is even more onerous in terms of latency required to bring the link back from suspend. The resume mechanism is engineered to accommodate driving resume events through 5 levels of hierarchy and backward compatibility with devices built to the resume mechanism defined in an older revision of the USB specification. Existing utilizations of USB suspend/resume are almost exclusively associated only with system state transitions (S3, S4, etc.), which don't, occur very often and don't require fast transition times; They attempt to be fast in human perception terms, but not in machine terms. The bus-imposed resume latencies are so long that the mechanism doesn't support response times that are useful in many applications, especially in hand-held platforms.

The existing suspend/resume mechanisms have been proven by platform architects and product groups to be inadequate for current and future generation mobile platforms. In order to keep USB a relevant technology for mobile platforms, it is necessary to update and add features to USB. Note that careful consideration has been taken in deriving the appropriate transitional latencies, where appropriate means reasonable complexity and cost for device implementations.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Impact on Existing Peripherals and Systems
The LPM enhancement requires changes to both host and device platforms to take advantage of the feature. As defined below, LPM simply adds a new feature and bus state that co-exists with the USB2.0 defined suspend/resume. As such, it is backward compatible in that a host will be able to determine whether a device supports LPM and a device that supports LPM will continue to work properly with older hosts. The only time LPM is used is when a device properly acknowledges a sleep transaction (described below). Similarly, the new resume timings are not used unless the link is transitioned to the L1 state via the sleep transaction.

Hardware Implications
There are no changes to the physical (analog) layer. The line state for L1 (Sleep) and USB 2.0 Suspend are identical (that is Full or Low-speed idle). There are no changes to resume signaling levels, end-of-resume signaling event (connect-speed dependent EOP), and subsequent line states. The device and host will require a new (link power state-specific) resume state machine (which may be able to use/share portions of the existing resume state machines) in order to accommodate the shorter event times. Depending on current implementations, a faster or different clock than is currently in use to detect the start of resume signaling may be needed; still this clock can be relatively slow as compared to the native clock used to drive signaling. A device must be able to decode a new extension transaction. The extension transaction uses an undefined PID which by the USB 2.0 definition must result in no response by addressed devices and ignored by other devices. The footprint of the new extension transaction borrows the method and form of split transactions. Since this can be applied to FS/LS devices, the expectation is that the new transaction format (see below) won't disturb existing device implementations. A host or hub must generate the new transaction when appropriate.

Software Implications
The mechanisms defined here are the 'bus-level' mechanisms required for the host port and device to properly operate. A host controller may allow host software to enable/disable this feature, or may provide other mechanisms to allow software to manage the feature. However, these decisions are up to host controller implementers and beyond the scope of this ECN. The host will need to (at least) expose a control for compliance testing in order to allow testing of the L1 and new resume timings.

Compliance Testing Implications
Compliance to this feature is quite easy to test. The L1 is an explicit transaction, which is easily captured via a bus analyzer. In addition, the resume signaling is easily captured and measured using mostly existing test procedures (modified only to accommodate the new timings).

Specification Addendum
The following is the text for the addendum to the USB 2.0 specification.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

1 Link Power Management (LPM)

This addendum for USB defines power management states (LPM states) and mechanisms to affect state changes that are used by hosts and hubs to efficiently manage bus and system power. The 'Link' in this context is intended to encapsulate the downstream facing port of a host or hub, the upstream facing port of a device and the data lines connecting the ports. The LPM states are defined in detail in Table 1-1. Note that the L0, L2 and L3 states are formal names for conditions already defined, but unnamed in the USB 2.0 specification. There are no changes suggested or implied to device, hub or device behavior under L0, L2 or L3 states. Section 1.1 details the requirements of hosts/hub and devices related to the L1 state.

Port commands are used to manage the transition of a link from L0 and L1 (see Section 4), regardless of whether the link to be transitioned is a root port or a downstream port of a discrete hub. This is similar to how all other link-specific states are managed in USB 2.0. The role and responsibilities are the same for either the hub or host that owns the port to which the port command is addressed. Therefore, in this addendum, with respect to LPM activities, the host and hub are referred to interchangeably.

Table 1-1. USB Link Power Management (Lx) States

LPM State	Description
L0 (On)	In this state, the port is enabled for propagation of transaction signaling traffic. A port in L0 is either actively transmitting or receiving data (L0-Active) or able to do so but not currently transmitting or receiving information (L0-Idle). While in this state Start-of-Frame (SOF) packets are issued by the host at a rate corresponding to the speed of the client device (1ms for full-speed, 125□s for high-speed). Note, the host transmits keep alives or the USB 2.0 hub translates some SOFs into keep alives if the device connected on the downstream port is a Low-speed device. The line state during L0-Active and L0-Idle vary according link speed as does the mechanism by which the host detects device hot-removal (e.g. during SOF phase for high-speed devices). Entry to L0 is via reset or resume signaling (either from L1 or L2).
L1 (Sleep)	L1 is similar to L2 (below) but supports finer granularity in use. When in L1, the line state is identical to L2 (see below). Entry to L1 is started by a request to a hub or host port to transition to L1. A LPM transaction is sent to the downstream device. The requested transition can only occur if the device response with an ACK handshake (see Section 2.2)). Exit from L1 is via remote wake, resume signaling, reset signaling or disconnect. L1 does not impose any specific power draw requirements (from VBUS) on the attached device as L2 does. Either the host or device can initiate resume signaling when in L1. Although the signaling levels of resume are the same as L2, the duration of the signaling and transitional latencies associated with the L1 to L0 transition are much shorter (see Section 1.1).
L2 (Suspend)	This is the formalized name for USB 2.0 Suspend, see Section 7.1.7.6 in the USB 2.0 specification. Entry to L2 is nominally triggered by a command to a hub or host port to transition to suspend, at which point the port ceases repeating signaling down the port (and may transition the port out of high-speed mode). The device discovers the suspend condition via observing 3ms of inactivity. The resultant line state is either Low or Full-speed idle (see Section 7.1.7.6). L2 also imposes power draw requirements (from VBUS) on the attached device (see Section 7.1.7.6). Exit from this state is via remote wake, resume signaling, reset signaling or disconnect.
L3 (Off)	In this state, the port is not capable of performing any data signaling. It corresponds to the *powered-off*, *disconnected*, and *disabled* states.

Figure 1-1 illustrates the allowed LPM state transitions. For example, a host must transition a link from L1 (Sleep) to L0 before transitioning it to L2 (Suspend), and similarly when transitioning from L2 to L1.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

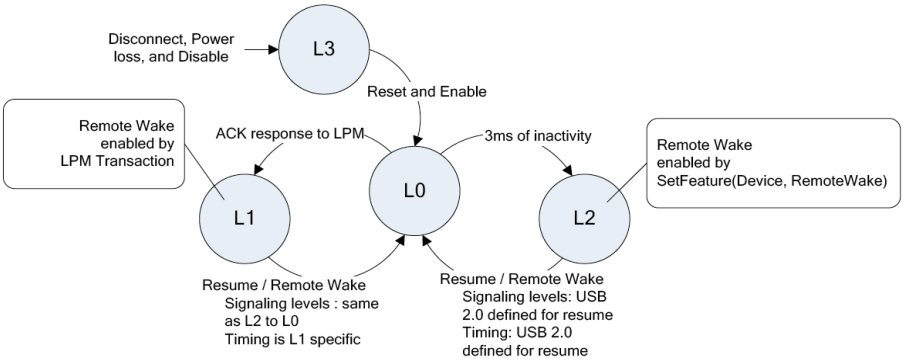


Figure 1-1. LPM State Transition Diagram

1.1 LPM State: L1 (Sleep)

The L1 (Sleep) LPM state differs from the L2 (USB 2.0 Suspend) state in two ways: the transitional latencies into and out of the state are much shorter (than L2) and there are no explicit power draw requirements on the attached device (L2 requires a suspended device to reduce its power draw from VBUS). Although there are no specification-mandated power requirements, the device won't be monitoring bus activity when its upstream port is in L1 and should take the opportunity to conserve power whenever possible. The host will provide an indication to the device on the minimum time it will be driving resume if it (the host) initiates L1 exit. The indicator is set by the host based on its current workload policies and will be in the range 50 to 1200 µs. The device can use the value of this indicator to deploy its own power efficiency optimizations, based on the responsiveness indicated by the host. L1 supplements the existing L2 state by utilizing most of the existing suspend/resume infrastructure (signaling levels, line states while in the state, etc.) but, provides much faster transitional latencies between L1 and L0 (On). Note a host must not initiate an action that results in the transmittal of an LPM transaction to a device unless it (the addressed device) is connected directly to a port downstream of the host or hub. The host must not initiate a command to issue an LPM transaction to a host or hub port when a split transaction is in flight to the addressed device.

Table 1-2 includes comparisons between the L1 and L2 LPM states.

Table 1-2. Summary Similarities/Differences Between L1 and L2

	L1 (Sleep)	L2 (Suspend)
Entry	Explicitly entered via LPM extended transaction	Implicitly entered via 3ms of link inactivity
Exit	Device or host-initiated via resume signaling; Remote-wake can be (optionally) enabled/disabled via the LPM transaction.	Device- or host-initiated via resume signaling; device-initiated resumes can be (optionally) enabled/disabled by software
Signaling	Low and Full-speed idle	Low and Full-speed idle
Latencies	*Entry*: ~10µs *Exit*: ~70µs s to 1ms (host-specific)	*Entry*: ~3ms *Exit*: >30ms (OS-dependent)
Link Power Consumption	~0.6mW (data line- pull-ups)	~0.6mW (data line- pull-ups)
Device Power Consumption	Device power consumption level is application/implementation specific	Device consumption is limited to: ≤500 µA or ≤2.5mA
Hot Removal	Natively detected per USB2 mechanisms	Natively detected per USB 2.0 mechanisms

Note, the time from the link returning to L0 (from L1) and the first transaction addressed to the device is host controller implementation specific. It is recommended that hosts service endpoints on the device as quickly as possible (given current host controller state transfer type of endpoints on the device and service interval). For example, quickly servicing reduces or avoids unnecessary latency in the data stream and may also provide an early opportunity to transition the link back to L1.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

2. Protocol

A transaction is used to request a device connected to a port to transition to L1. USB 2.0 had only one 'reserved' PID value, so an extension transaction is defined to allow future protocol extensibility. This section is organized into two parts: Section 2.1 defines the extension (EXT) transaction framework and Section 2.2 defines the LPM extended transaction.

2.1 USB 2.0 Extension Transaction

The extension transaction uses the remaining USB 2.0 reserved PID value (0000B), which is in the Special PID type group. This addendum names this PID value the EXT PID, as illustrated in Table 2-1. Note, this table overrides Table 8-1 PID Types in the USB specification, Revision 2.0.

Table 2-1 PID Types

PID Type	PID Name	PID<3:0>*	Description
Token	OUT	0001B	Address + endpoint number in host-to-function transaction
	IN	1001B	Address + endpoint number in function-to-host transaction
	SOF	0101B	Start-of-Frame marker and frame number
	SETUP	1101B	Address + endpoint number in host-to-function transaction for SETUP to a control pipe
Data	DATA0	0011B	Data packet PID even
	DATA1	1011B	Data packet PID odd
	DATA2	0111B	Data packet PID high-speed, high bandwidth isochronous transaction in a microframe (see Section 5.9.2 for more information)
	MDATA	1111B	Data packet PID high-speed for split and high bandwidth isochronous transactions (see Sections 5.9.2, 11.20, and 11.21 for more information)
Handshake	ACK	0010B	Receiver accepts error-free data packet
	NAK	1010B	Receiving device cannot accept data or transmitting device cannot send data
	STALL	1110B	Endpoint is halted or a control pipe request is not supported
	NYET	0110B	No response yet from receiver (see Sections 8.5.1 and 11.17-11.21)
Special	PRE	1100B	(Token) Host-issued preamble. Enables downstream bus traffic to low-speed devices.
	ERR	1100B	(Handshake) Split Transaction Error Handshake (reuses PRE value)
	SPLIT	1000B	(Token) High-speed Split Transaction Token (see Section 8.4.2)
	PING	0100B	(Token) High-speed flow control probe for a bulk/control endpoint (see Section 8.5.1)
	EXT	0000B	(Token) Protocol extension token (see Section 2.1.1)

*Note: PID bits are shown in MSb order. When sent on the USB, the rightmost bit (bit 0) will be sent first.

2.1.1 Protocol Extension Token

This addendum defines an extended token phase used to extend the USB transaction protocol. The token phase for an EXTended transaction is comprised of two token packets (each a standard 3 byte token packet), see Figure 2-1.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

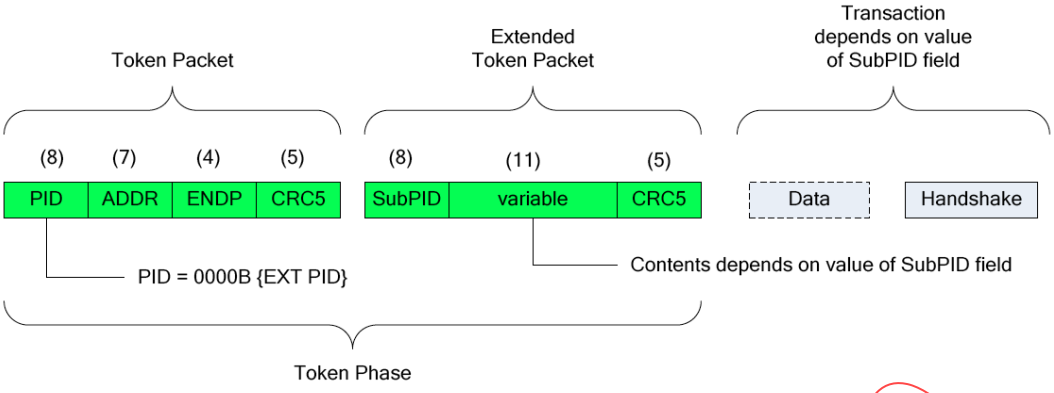


Figure 2-1. Packets in an Extension Token Transaction

The first packet of an extension transaction is a standard token packet with EXT in the PID field, including: ADDR, ENDP and CRC5 fields. The EXT PID indicates that the next packet (with standard, back-to-back inter-packet timing) is an extended token packet.

The second packet of an extension transaction is the Extended Token packet. The basic structure of this packet is identical to a standard token packet in that the first field is a PID (subPID) field and the end of the packet contains a CRC5, which is used to check the 11 bits between the SubPID field and the CRC5. The format of the variable field in the extension token packet depends on the value of the SubPID field. Table 2-2 lists the defined SubPID codes. As with existing token packets, the SubPID value indicates the direction and format of the remainder of the transaction. This allows definition of transactions with data phases with in-bound or out-bound data, and inclusion or omission of a handshake phase. Note that an extended transaction has an effective 6 bytes of token information.

Table 2-2. SubPID Types

SubPID Name	PID<3:0>	Description
N/A	0000B	Reserved for future use
Reserved	0001B 1001B 0101B 1101B 1000B 0100B 1100B	These PID values must not be used for Extended Token packet PIDs. These values are used for the Tokens in the base PID table (see Table 2-1 PID Types, above) and cannot be reused for SubPID types.
LPM	0011B	LPM token

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 2-2. SubPID Types (cont.)

SubPID Name	PID<3:0>	Description
	1011B 0111B 1111B 0010B 1010B 1110B 0110B	Reserved for future use

2.2 USB 2.0 LPM Extended Transaction

This section defines the format of the LPM extended transaction. The first sub-section 2.2.1 defines the LPM token and sub-section 2.2.2 defines the semantics of the LPM transaction.

2.2.1 LPM Token

The LPM token packet is only valid in an Extended transaction format, as illustrated in Figure 2-2.

	Token Packet					Extended Token Packet		
	(lsb)			(msb)	Inter-packet Gap	(lsb)		(msb)
Field	EXT PID	ADDR	ENDP	CRC5		LPM SubPID	bmAttributes	CRC5
Bits	8	7	4	5		8	11	5

Figure 2-2. LPM Extended Token

The format of the *bmAttributes* field of the LPM token is illustrated in Table 2-3.

Table 2-3. LPM Token *bmAttributes* Field Definition

Bits	Field	Description
10:9	*Reserved*	Reserved for future use
8	*bRemoteWake*	A value of one (1B) in this field enables the addressed device to wake the host upon any meaningful application-specific event (e.g. an interrupt for a device with one or more interrupt endpoints). A value of zero (0B) disables the device from initiating remote wake.
7:4	*HIRD*	Host Initiated Resume Duration. See Section 4.1 for value definition.
3:0	*bLinkState*	The link state (Lx) the addressed device must transition to after responding to this transaction with an ACK. Valid values for this field are: **Value** — **Description** 0000B — Reserved for future use 0001B — L1 (Sleep) 0010B – 1111B — Reserved for future use

Refer to Section 2.2.2 for a full definition for the LPM transaction and Section 1 for a full overview of Link Power Management.

2.2.2 LPM Transaction

LPM transactions are used by a host to transition a host to device link to a specific link power management state (see Section 1). An LPM transaction is a two phase extended USB transaction, consisting of an extended token phase (two token packets) and a handshake

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

phase (handshake packet) as shown in Figure 2-3. Note that the EXT token must be transmitted before the LPM token.

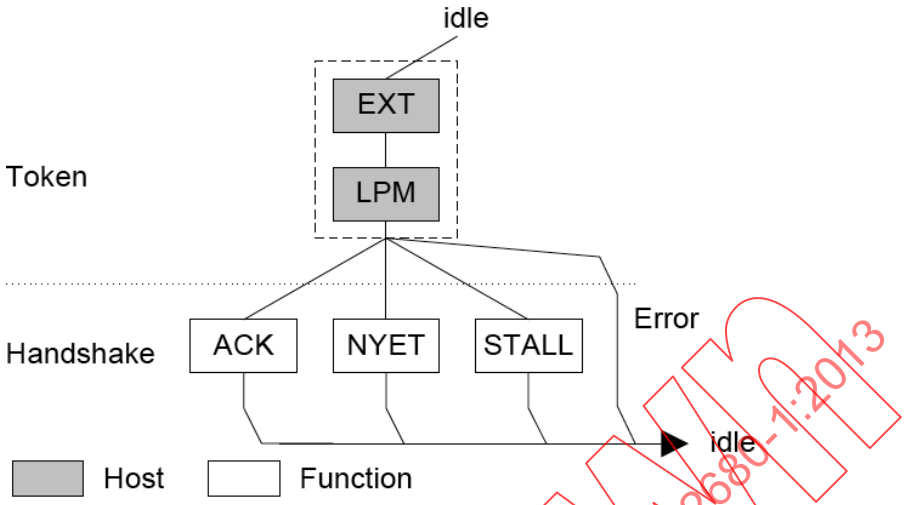


Figure 2-3. LPM Transaction Format

When a host is ready to transition a port from L0 to a deeper power savings state, it issues a command to the port, which results in an LPM transaction. The device function responds with an ACK if it is ready to make the transition or a NYET if it is not currently ready to make the transition usually because it has data pending for the host. A device must transmit a STALL handshake if it does not support the requested link state (Lx, see the bmAttributes field of the extended LPM transaction). If the device detects errors in either of the token packets or does not understand the protocol extension transaction, no handshake is returned.

3. Framework: USB Device Capabilities – USB 2.0 Extension

This section defines an optional device-level, capabilities descriptor for wired USB devices, and provides an indicator in the existing USB 2.0 framework that system software can use to determine that the device supports the new reporting descriptor type. A unique descriptor is used to report additional USB 2.0 device-level capabilities that are not reported via the Device descriptor. The USB Device Capabilities – USB 2.0 Extension descriptor is only available with a BOS descriptor set. The BOS descriptor is defined in Section 7.4.1 of the Certified Wireless USB Specification, Revision 1.0. This descriptor cannot be accessed directly by a GetDescriptor() or SetDescriptor() request.

Table 3-1. USB Device Capabilities – USB 2.0 Extension Descriptor

Offset	Field	Size	Value	Description
0	*bLength*	1	Number	Size of this descriptor in bytes
1	*bDescriptorType*	1	Constant	Descriptor type: DEVICE CAPABILITY Type.
2	*bDevCapabilityType*	1	Constant	Capability type: USB 2.0 EXTENSION (002H)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 3-1. USB Device Capabilities – USB 2.0 Extension Descriptor (cont.)

Offset	Field	Size	Value	Description
3	*bmAttributes*	4	Bitmap	Bitmap encoding of supported device level features. A value of one in a bit location indicates a feature is supported; a value of zero indicates it is not supported. Encodings are: **Bit** **Encoding** 0 **Reserved**. Must be set to zero. 1 **LPM**. A value of one in this bit location indicates that this device supports the Link Power Management protocol. 31:2 **Reserved**. Must be set to zero.

The value of the *bcdUSB* field in the standard USB 2.0 Device Descriptor is used to indicate that the device supports the request to read the BOS Descriptor (i.e. GetDescriptor(BOS)). Devices that support the BOS descriptor must have a *bcdUSB* value of 0201H or larger..

4. Hub

This section is a supplement to the Hub device definition chapter in the USB 2.0 specification. Some portions of this section over-ride or supplement information in the USB 2.0 specification. These are called out where applicable. The mechanisms in USB 2.0 to manage port states are via Get/Clear/Set Port Feature commands. Most hosts provide equivalent controls to system software for the downstream ports. LPM adds extensions to the port management command set for managing the L1 link state. It is recommended that hosts provide equivalent mechanisms to those defined below. Managing the L1 state of a port is intended to be very similar to managing a port's L2 (Suspend) state, including: a request to transition into the state, a request to explicitly transition the port out of the state, a request to inquire about the current state and support for remote wake. This addendum defines extensions to the USB 2.0 Hub port feature management commands and also defines a new management command for transitioning a port into the L1 state. The resume timing requirements (L1 to L0) apply equally to both hubs and hosts unless explicitly noted otherwise.

4.1 L1 Entry – Hub Operational Model

The operational control model for placing a port into L1 is similar to placing a port into L2 (Suspend), i.e. a command is sent via the Default Control Pipe to the hub, requesting it to transition a specific port to L1. A new SET and TEST control read transfer is used for this command. The operational command is fully contained in the SETUP stage and the DATA stage is used to return a completion code for the command. On receipt of the SETUP stage of the request to transition a port to L1, the hub begins NAKing the data stage transaction and quickly services the command. When the command is complete, it returns a completion code in the DATA stage and the host will complete the transfer with the STATUS stage. The details of this command are provided in Section 4.10.1. The port control model is illustrated in Figure 4-1.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

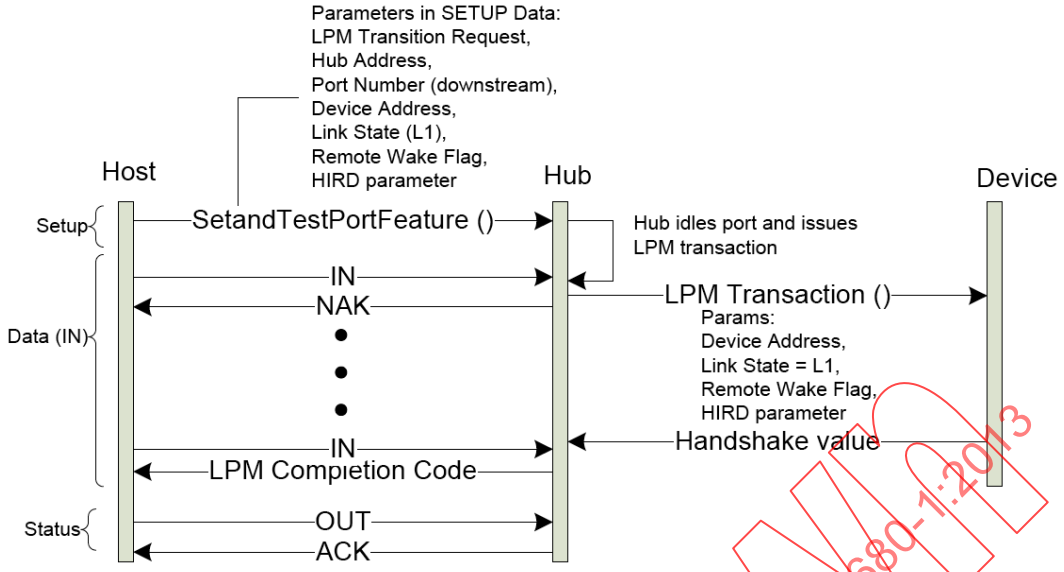


Figure 4-1. Port Control Model for Transitioning a Port to L1

If the designated port is in the proper state (i.e. enabled) when the request to transition to L1 is received, the hub first stops repeating traffic down the port by the next micro-frame boundary (after the SETUP stage was received) and issues an LPM transaction to the device (independent of the traffic which may be occurring on the other ports). When the LPM transaction completes the hub evaluates the results of the transaction to determine what to do with the port and what completion code to return to the host. If the device responds with an ACK, the port is placed into the L1 state and a SUCCESS completion code is sent back to the host. If the device responds with a NYET, the port is returned to the enabled state (restore normal operation) and a completion code of NYET is sent back to the host. If the hub gets a transaction error (including timeout), it must retry the LPM transaction two more times. If there are three consecutive errors then the port is returned to the enabled state and an error completion code (see Section 4.10.3) is sent back to the host.

The hub must respond to the data stage of this control transfer within 200 µs. This allows for an arrival of the SETUP at the beginning of one micro-frame and for completion of a low-speed transaction in the following micro-frame, with some additional margin.

The host system sets the Remote Wake Flag parameter in this request to enable or disable the addressed device for remote wake from L1. The value of this flag will temporarily (while in L1) override the current setting of the *Remote Wake* feature settable by the standard Set/ClearFeature() commands defined in Universal Serial Bus Specification, revision 2.0, Chapter 9.

The host system communicates to the device the duration of how long the host will drive resume when it (the host) initiates exit from L1 via the HIRD (Host Initiated Resume Duration) parameter. It sets the HIRD value based on its requirements for responsiveness. For example, it will use short HIRD encodings if it requires low latency to return the device to full operational state. It will use longer HIRD encodings if the latency to full operational state are less stringent. The device can use the HIRD value to help determine what power optimization features it should employ in response. For example, the device could use long HIRD encodings as a flag to idle significantly more logic (clocks, logic, etc.) than it would with short HIRD encodings. The power optimizations and commensurate power reduction in power consumption are device implementation specific.

The HIRD value is a 4-bit encoded value. The value of 0000B is interpreted as 50 µs. Each value up adds 75 µs to the previous value. For example, 0001B is 125 µs, 0010B is 200 µs and so on. Based on this rule, the maximum value resume drive time is at encoding value 1111B which represents 1.2ms.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4.1.1 LPM Transaction and Transition to L1

The transition from L0 to L1 occurs only when a device responds with an ACK handshake to a successful LPM transaction. After transmission of the ACK handshake, a device must not start transition to L1 before $T_{L1TOKENRETRY(MIN)}$ but must have started transition to L1 by $T_{L1TOKENRETRY(MAX)}$ (t1 to t2, see Figure 4-2). If the device is in high-speed mode in L0, then the ports (downstream facing port on a host or hub and upstream facing port on a device) must transition to Full-speed mode as the link transitions to L1 (this is the same process and line state as for L2 (Suspend)). The speed mode transition must be complete before the minimum L1 residency time elapses ($T_{L1Residency(min)}$) in order to support resume signaling initiated from either the host/hub or device. If in Full or Low-speed mode in L0, then the link is in the same mode in L1 (again, same as for L2 (Suspend), see Section 7.1.7.6 in the USB 2.0 specification).

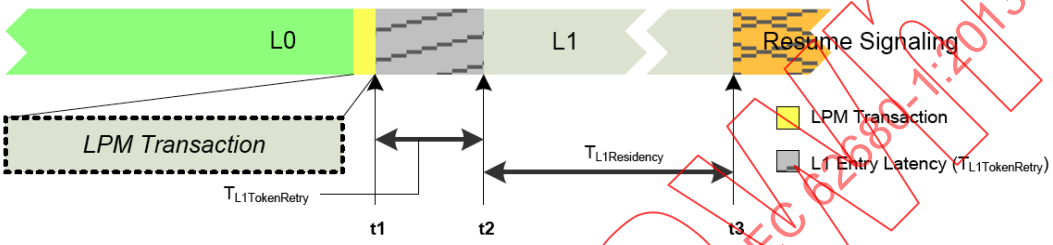


Figure 4-2. LPM Transaction and Transition Timing to L1

A device can reject transition to L1 by responding with a NYET response to the LPM token. If a device does not support extended transactions, it must not respond to the token. As noted above, the host/hub must apply standard transaction retry rules when experiencing either a corrupted handshake or a timeout (i.e. three transaction attempts). A host/hub must transmit a retry of the token (for lost ACK) before the minimum $T_{L1TokenRetry}$ expires.

4.2 L1 Exit (Resume)– Hub Operational Model

This section details the operational model for transitioning a link out of L1. It covers both device initiated 'remote wake' model and a host-initiated commanded L1 Exit. The following subsections are organized by: (1) detail functional requirements on how resume signaling is propagated and how those requirements co-exist with existing L2 rules, (2) describe device-initiated L1 exit and then (3) host-initiated L1-exit.

Either a host or device can initiate a transition from L1 to L0. The signaling methods are exactly the same as defined for the L2 (Suspend) to L0 transition (i.e. resume signaling) with the exception of the timing constraints. The mode transition from Full-speed to High-speed required for High-speed devices (for suspend mode) apply here also (see Section 7.1.7.6 in the USB 2.0 specification).

An important basic concept is a port in L1 or L2 does nothing more than listen for resume signaling. Upon receipt of resume signaling, the downstream port transitions to L0 (so the device can participate in the resume signaling sequence). This concept is important to understand the behavior in Table 4-1, below (and borrows all the functionality and simplicity of implementation of a hub from USB 2.0).

There are two rules for hub with respect to propagating resume signaling. These rules are consistent and backward compatible with requirements on USB 2.0 hubs and propagation of resume signaling (from Suspend).

Rule1: A hub propagates an L1 exit resume upstream when the upstream port is in L1 or L2. Note, when the upstream port is in L2, the hub turns the device initiated L1 exit into a normal L2 exit (obeying all of its rules). This effectively causes the hub to transition its upstream facing port to L0.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Rule2: When a hub whose upstream facing port is in L1 or L2 begins to transmit resume signaling, it will 'broadcast' that signaling out all enabled (L0) ports (from Rule1 above, this includes the upstream facing port). The 'link' does not return to L0 until the upstream facing port of the device matches the state of the downstream facing port. This does not occur 'officially' until the end of resume (EOR) is received by the device. This rule meets the intent of the USB 2.0 specification. Note that this addendum introduces new combinations of upstream and downstream port power states where some combinations may not result in predictable behavior. In order to simplify the hub implementation (by not forcing it to track combinations of upstream and downstream port states), system software is expected to implement the policy described in Section 4.2.1.

4.2.1 Device Initiated L1 Exit

The timing events related to a device initiated L1 exit are illustrated in Figure 4-3. After placing a link into L1, a host or hub may decide to idle other logic associated with the port, including shutting down high-speed clocks, etc. The host or hub must be capable of detecting and responding to resume signaling with the timings specified here, by whatever mechanisms that are appropriate. Figure 4-3 shows two device initiated resume scenarios which demonstrate the host/hub requirements with respect to detecting and responding to resume signaling, depending on whether its signaling clock is running at the time the remote wake signaling is detected.

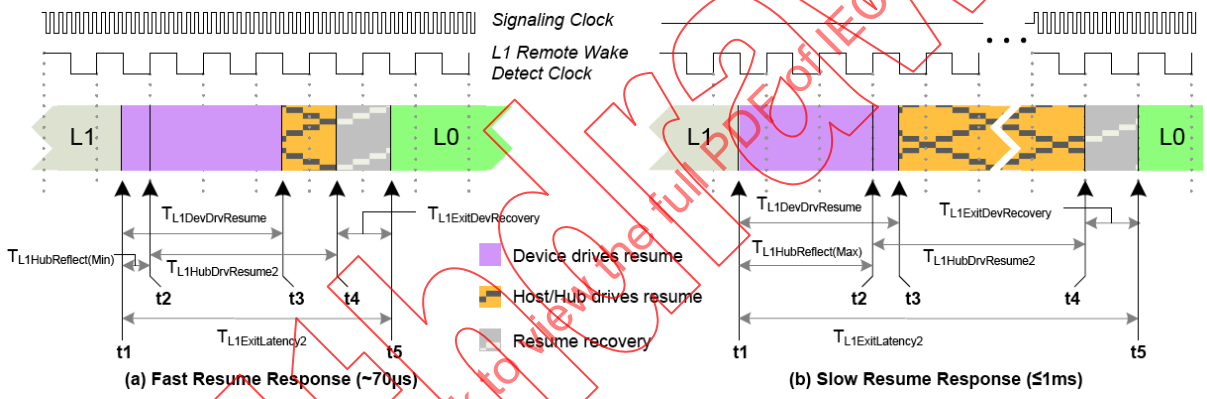


Figure 4-3. Device Initiated L1 to L0 Transition (Remote Wake)

Figure 4-3 illustrates a device initiated resume (also called Remote Wake). A device initiating L1 exit must drive resume signaling for nominally $T_{L1DevDrvResume}$ [t1->t3]. The host must detect the resume signaling and reflect it (i.e. drive resume signaling on its corresponding port) within $T_{L1HubReflect}$ [t1->t2]. The host must drive resume signaling for a minimum of $T_{L1HubDrvResume2}$ [t2->t4]. This ensures that it drives resume signaling long enough for the device to stop driving resume, turn its receiver on and be ready to detect the end of resume signaling (i.e. speed appropriate EOP) [t3->t4]. From here on the process is identical to the host-initiated L1 to L0 described above.

A device is not allowed to assert remote wake from the L1 state unless it has been explicitly enabled to via a flag in the LPM transaction (see Section 2.2.1). The L1 to L0 transitional latency is comprised of several discrete signaling events. The duration of these discrete events are constrained to ensure the maximum L1 to L0 transitional latency is less than or equal to $T_{L1ExitLatency2}$ [t1->t5]. The maximum constraint is defined to allow aggressive L1 use.

Figure 4-3 illustrates two general scenarios that depict the timings based on the state of the host/hub's signaling clock (i.e. active or not active) at the time the device initiates remote wake. In Figure 4-3(a), the host/hub's signaling clock is active at the time it receives the remote wake signaling on the port in L1. In this case, the total time from remote wake to L0 ($T_{L1ExitLatency2}$) [t1->t5] is very fast because the host/hub can detect the resume very quickly ($T_{L1HubReflect(Min)}$) and is not constrained by the need to restart its signaling clock before

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

transmitting end of resume. In Figure 4-3(b), the host/hub's signaling clock is not running at the time it receives the remote wake, so the host/hub uses a much slower clock to detect remote wake and to reflect (begin driving) resume signaling down the port ($T_{L1HubReflect(Max)}$). Note that the host/hub must begin driving resume before the device stops driving remote wake (these are the same line state). This means the host/hub must detect and drive resume within $T_{L1HubReflect(Max)}$ of when the device asserted remote wake (this ensures the host is driving resume before the device releases remote wake, with a 1µs guard band). The host/hub needs to have its signaling clock stable in order to properly signal end-of-resume, and it may take some time to get it restarted and stable. The host/hub must complete resume signaling [t2->t4] before $T_{L1HubDrvResume2(Max)}$ elapses.

The host must provide at least $T_{L1ExitDevRecovery}$ before addressing any transactions to the device.

Figure 4-4 illustrates the timing relationships for an example remote wake L1 exit sequence. This example is almost identical to a remote wake from suspend (L2) except the signaling event durations are much shorter.

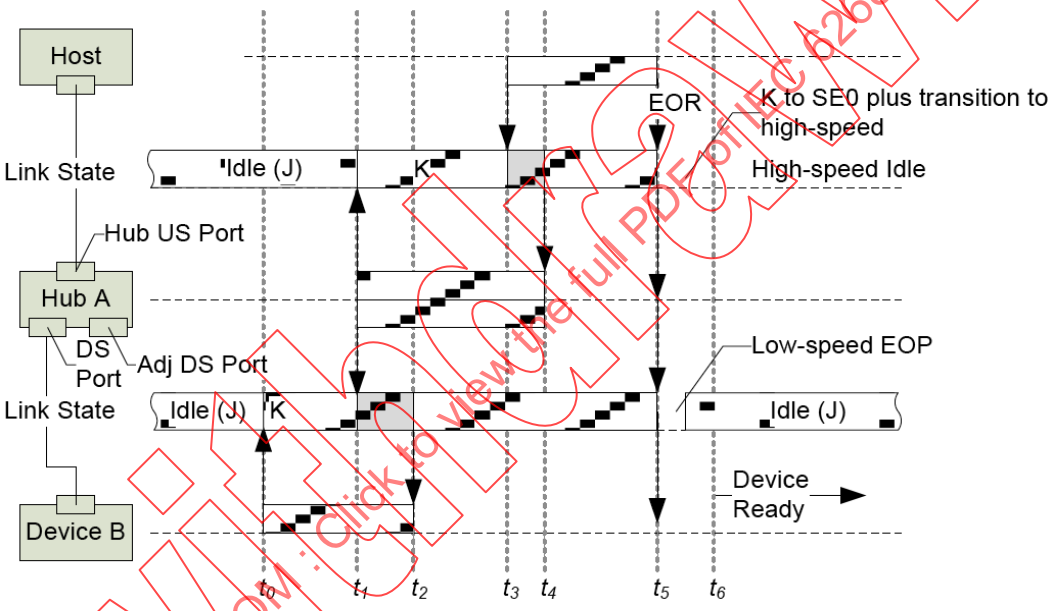


Figure 4-4. Example Remote-wakeup L1 Exit with Full-speed Device Under Connected Hub

At [t0] the full-speed device drives a K on its upstream port, initiating the L1 Exit (remote wake). The duration of the device driving K is [t0 to t2] ($T_{L1DevDrvResume}$). At t1 the hub upstream of the device begins driving K on the link, so between t1 and t2 both the device and the hub are driving on the link between Device B and Hub A. At the same time [t1] the Hub also begins driving a K on its upstream link and continues to drive K (upstream and downstream) until t4 (duration is also $T_{L1DevDrvResume}$). At t3 the host upstream of the hub begins driving K on the link, so between t3 and t4 both the hub and the host are driving K on the link between Hub A and the Host. The Host drives K downstream until t5 at which time it drives a speed-appropriate EOR (end of resume) on the link between the Host and Hub A (which includes a transition to high-speed mode). Hub A repeats the K to SE0 transition (at t5) and times two Low-speed bit times of SE0 and then drives a J on the link between Hub A and Device B. The Host waits until after t6 to issue a transaction to Device B.

Table 4-1 is a summary analysis of a remote wake transitioning a downstream from (L1/L2) to L0 and its effects on other ports on the Hub (both upstream and downstream) for all combinations of the L-state of those other ports. This table reflects a policy that is

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

recommended for the host system for managing the relationship between the power state of a hub's upstream facing port and the power states of its downstream facing ports. The general policy is that downstream facing ports power state level should be the same as, or below the power savings state level of the upstream facing port. The behavior of the system is undefined if this policy is not implemented by the host.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 4-1. Device Initiated Resume Propagation and Adjacent Port Side-effects

Port States			
Hub DS Port (Remote Wake)	**Hub US Port**	**Adj. DS Port**	**Description**
L1->L0	L0	L0	**General:** The DS link where the device is connected is the only port that makes a transition. When L1 exit is complete, a port change bit is set (to a 1b) which results in a subsequent port change notification. No effect on adjacent ports.
		L1	
		L2	
	L1	L0	**This combination should not be allowed by the host because the resume signaling will not recover the devices attached to adjacent L0 ports (see policy statement above).**
		L1	No effect on adjacent ports that are not in the enabled state.
		L2	
	L2	L0	**This combination should not be allowed by the host (see policy statement above).**
		L1	Resume signaling is only transmitted through (downstream) ports that are in L0.
		L2	
L2->L0	L0	L0	**General:** USB2 defined resume signaling (remote wake) on a suspended port. Hub upstream facing port is in L0, so only suspended port transitions; a port change bit is set (to a 1b) which results in a subsequent port change notification. No effect on adjacent ports.
		L1	
		L2	
	L1	L0	**This combination should not be allowed by the host (see policy statement above).**
		L1	
		L2	
	L2	L0	USB2 defined resume. Hub reflects resume signaling upstream (transitions upstream port out of L2) and down all enabled (L0) ports.
		L1	Resume signaling is only transmitted through (downstream) ports that are in L0.
		L2	

DS – Downstream Port
US – Upstream Port

4.2.2 Host Initiated L1 Exit

The operational model for host initiated transition out of L1 is the same as a host initiated transition out of L2 (USB2 Suspend). The control method is a ClearPortFeature (PORT_L1, HIRD, PortNumber) command which is a control write command (no data phase) addressed to a hub port. The port control method is illustrated in Figure 4-5.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

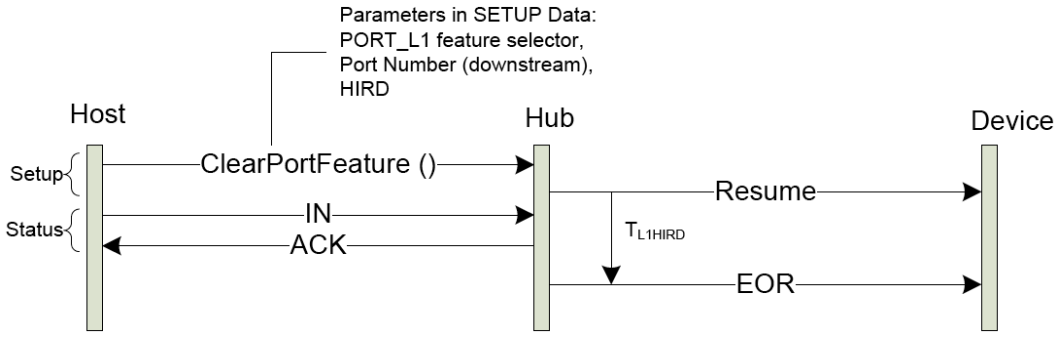


Figure 4-5. Basic Port Control Model for Transitioning a Port out of L1

Host initiated resume signaling from L1 to L0 is illustrated in Figure 4-6.

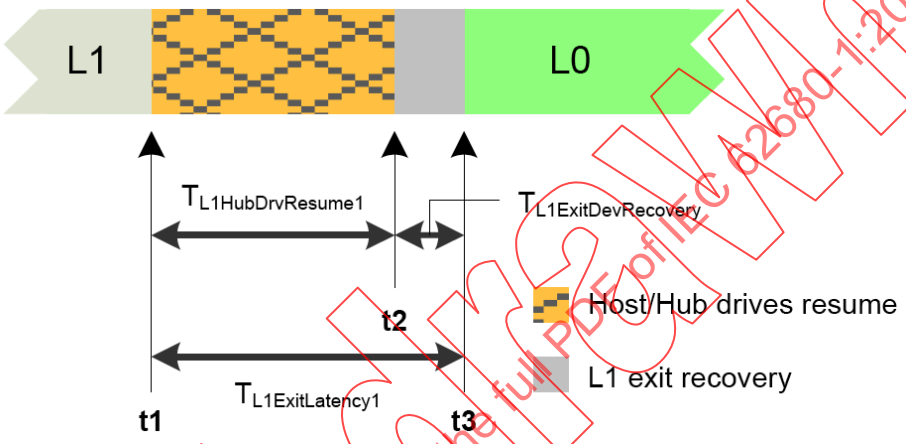


Figure 4-6. Host Initiated L1 to L0 Transition (L1 Exit)

A host initiating L1 exit drives resume signaling for a minimum of $T_{L1ExitHostDrvResume1(min)}$ [t1->t2] and provides at least $T_{L1ExitDevRecovery}$ [t2->t3] of resume recovery time between the end of resume signaling (EOP) and transmittal of the first transaction that targets the resumed device. The duration of the resume signaling is specified by the host in a parameter called the HIRD (Host Initiated Resume Duration) and is in the range between $T_{L1HubDrvResume1(min)}$ and $T_{L1HubDriveResume1(max)}$. The total L1 exit latency expected from a host initiated L1 exit is the range of $T_{L1ExitLantency1}$, [t1->t3]. As with suspend, in order to restore a communication path between the host and a device in L1, each link in the hierarchy must be explicitly transitioned to the L0 state. This means one ClearPortFeature(L1) for each link in the path.

4.3 Hub Port State Machines with L1 Additions

The following sections detail additions to the USB 2.0 port state machine sections required to support the L1 link state. These sections are addendums, not replacements of the referenced USB 2.0 specification Hub Chapter sections.

There are three reference ports (control points) in a hub which together with the Hub Repeater and Hub Controller comprise the functional operation of a hub. Figure 4-7 illustrates the documented inter-state machine signals between the different reference port state machines that exist inside of a USB 2.0 Hub, with the necessary extensions required to support L1. The inter-state machine signals used in USB 2.0 are documented. The USB2.0 suspend/resume relevant signals are ‘highlighted’ bold line. The colored lines and boxes illustrate the extensions necessary to support L1.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

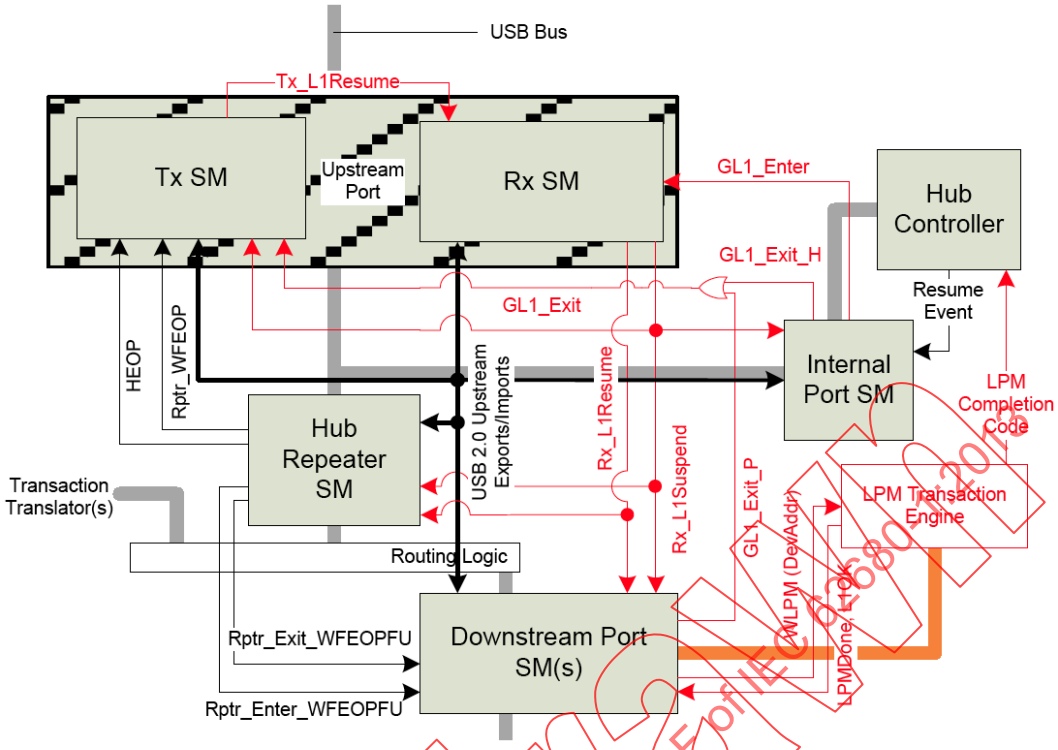


Figure 4-7. USB 2.0 Hub Reference Port State Machine Relationships with L1 Additions

A new element for L1 support is for each port there is an LPM Transaction Engine (LPM TE). The purpose of this new element is described in Section 4.8.1. The following sections detail the internal state machine changes required to support L1. Each section focuses on the additions to the existing state machines to support L1. Intersection with existing USB 2.0 defined states are illustrated via simple gray states. The new states and transitions required to support L1 functionality are highlighted (in red for color display or light states for grayscale). All transitions illustrated are new transitions for L1 support. These diagrams do not alter or change the existing USB 2.0 defined transitions between existing USB 2.0 defined states.

4.4 Upstream Facing Port Receiver State Machine – w/L1 Additions

The upstream port receiver (Rx) state machine is responsible for detecting long-term signaling events, including reset, resume and suspend. Refer to the USB 2.0 specification for details. L1 entry only occurs relative to the completion status of an LPM transaction addressed to the hub (as a device). The transaction is handled by the hub controller which is also responsible for exporting a control delivered by the internal port state machine (see Section 4.7) notifying that the L0 to L1 transition must occur on the upstream port.

Figure 4-8 illustrates the additions to the upstream port receiver state machine to support L1.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

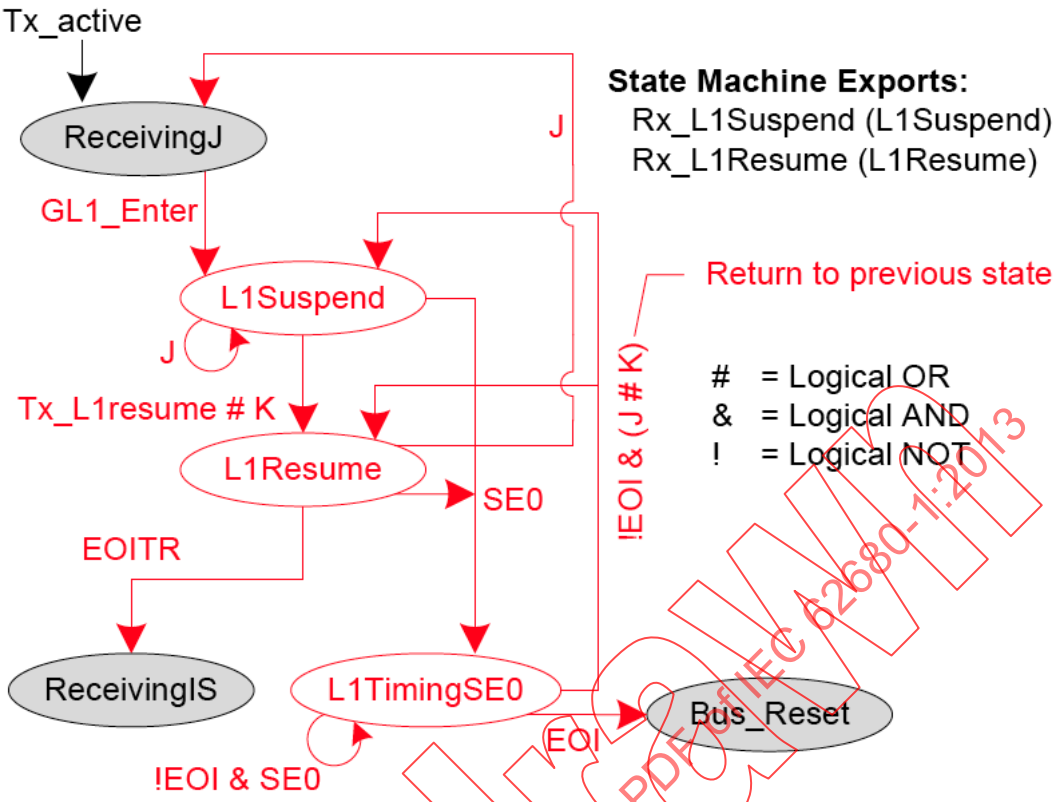


Figure 4-8. L1 Addendum to the Upstream Facing Port Receiver State Machine

Table 4-2. Upstream Facing Port Receiver Signal/Event Definitions (Addendum)

Signal/Event Name	Event/Signal Source	Description
J	Internal	Receiving a 'J' (IDLE) or an 'SE1' on the upstream facing port
K	Internal	Receiving a 'K' on the upstream facing port
EOI	Internal	End of timed interval
EOITR	Internal	Generated 24 full-speed bit times after the K->SE0 transition at the end of L1 exit
GL1_Enter	Internal Port	Hub has received and positively acknowledged an LPM request to transition to L1
Tx_L1resume	Transmitter	Transmitter is in the L1_SResume state

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4.4.1 L1Suspend

This state is entered:

- From the ReceivingJ state after the hub controller has positively acknowledged an LPM transaction and asserted GL1_Enter (via the internal port, see Section 4.7).
- From the L1ReceivingSE0 state as a result of a transient SE0 that was detected in this state (previously).

On entry to this state, the Hub Controller starts a 50 µs timer (TL1Residency), see Section 4.7.2. The Hub Controller cannot generate resume signaling until after the timer expires. The repeater remains in this state while there is a J state (full-speed idle) on the upstream facing port.

4.4.2 L1Resume

This state is entered:

- From the L1Suspend state when the Transmitter is in the L1Sresume state or if there is a transition to the K state on the upstream facing port
- From the L1ReceivingSE0 state as a result of a transient SE0 that was detected in this state (previously).

This is a timed state. The intent is that the repeater remains in this state until it recognizes a valid End of Resume event (EOR) which would be an EOITR for connected at high-speed or if connected at full-speed, when the J-state is detected after a low-speed EOP is detected (two low-speed bit times of SE0 followed by a J). A hub must keep a clock of at least 64 KHz available in order to be able to accurately time resume signaling. If this state is being held pending stabilization of the hub's normal signaling clock, the Receiver must provide a K to the repeater for propagation to the downstream facing ports. When normal signaling clocks are stable, the Receiver must repeat the incoming signals.

4.4.3 L1ReceivingSE0

This state is entered:

- From the L1Suspend state when there is a transition to the SE0 state on the upstream facing port.
- From the L1Resume state when the SE0 state on the upstream port persists beyond a valid EOR.

The hub must save the previous state in order to know where to transition to in the case where the SE0 is a transient. This is a timed state were the behavioral requirements are the same as for the ReceivingSE0 state in the USB 2.0 specification (see Section 11.6.3.8). The timer is reset each time this state starts. If the upstream port transitions to a J or K state before the timer expires, the receiver returns to the previous state.

4.5 Upstream Facing Port Transmitter State Machine – w/L1 Additions

In USB 2.0, this state machine is used to monitor the upstream facing port while the Repeater has connectivity in the upstream direction. The purpose of this monitoring is to prevent erroneous events from propagating from downstream ports to the upstream port. This state machine also plays a role in the remote wake (resume) process. For example, when the hub is suspended and there is a remote wake event (either from a downstream port or from the Hub Controller), it is from this state that the Hub drives a K state on the upstream port. Figure 4-9 illustrates the new states and transitions required for the upstream facing port transmitter to support L1.

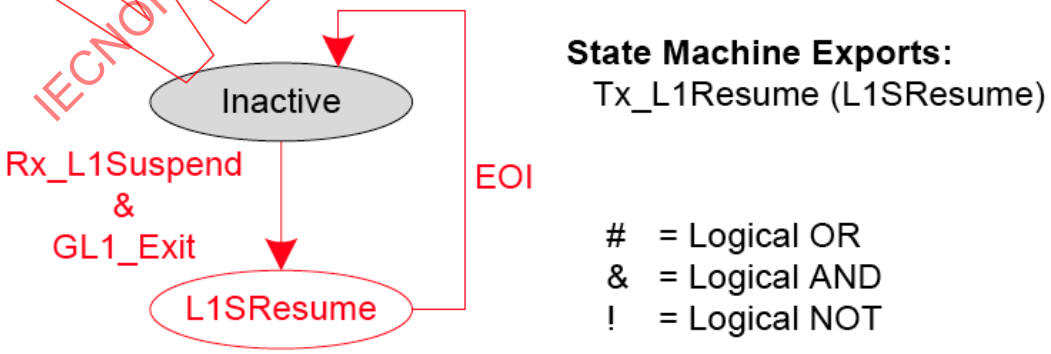


Figure 4-9. Addendum to the Upstream Facing Port Transmitter State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 4-3. Upstream Facing Port Transmitter Signal/Event Definitions (Addendum)

Signal/Event Name	Event/Signal Source	Description
Rx_L1Suspend	Receiver	The upstream facing port receiver state machine indicates that it is in the L1Suspend state
GL1_Exit	Internal Port & Downstream facing Port	This is a logical OR of a the hub controller signaling that it needs to generate a remote wake signal due to a hub event and the detection of a remote wake event from a downstream facing port that is in the L1 state
EOI	Internal	End of timed interval

4.5.1 L1SResume

The port enters this state from the Inactive state when the receiver is in the L1Suspend state and there is a remote wake event coming from either the hub controller or a downstream port. This is a timed state with duration of $T_{L1DevDrvResume}$.

4.6 Hub Repeater State Machine – with L1 Additions

The Hub Repeater state machine has only one addition to support L1. The WFEOPFU state is also entered from the WFSOPFU, WFSOP or WFEOP states when Rx_L1Resume is valid.

4.7 Internal Port – with L1 Additions

The internal port is the connection between the Hub Controller and the Hub Repeater. Besides conveying the serial data to/from the Hub Controller, the internal port is the source of certain resume signals. Figure 4-10 illustrates the internal port state machine addendum to support L1 entry and exit events.

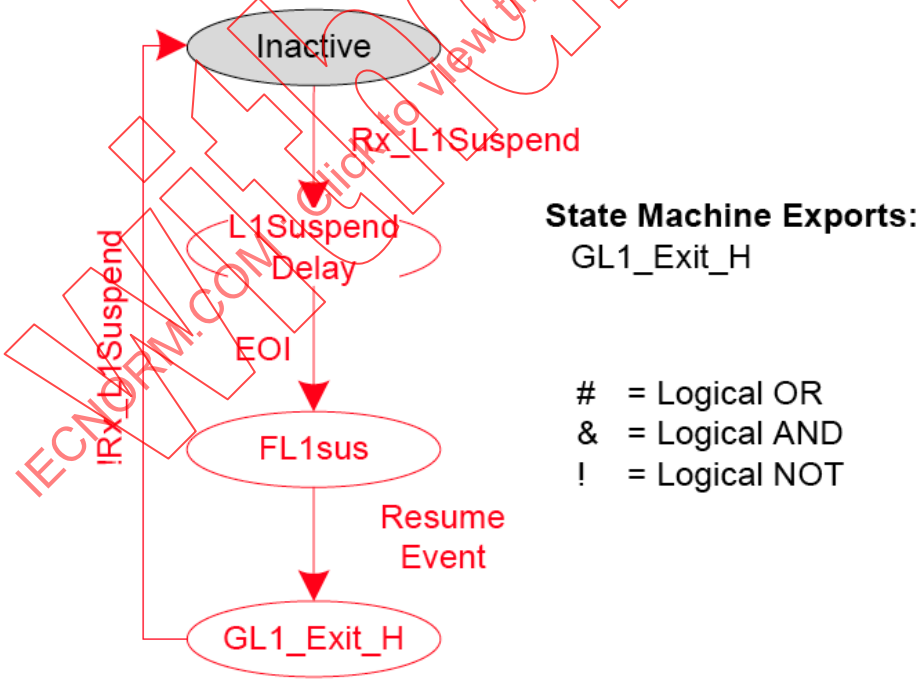


Figure 4-10. Addendum to the Internal Port State Machine

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Table 4-4. Internal Port Signal/Event Definitions (Addendum)

Signal/Event Name	Event/Signal Source	Description
Rx_L1Suspend	Receiver	The upstream facing port receiver state machine indicates that it is in the L1Suspend state
Resume Event	Hub Controller	A resume condition exists in the Hub Controller
EOI	Internal	End of timed interval

4.7.1 Inactive

This state is the same as documented in the USB 2.0 specification (Section 11.4.1), with the additional exception that this state is entered whenever the Receiver is not in the Suspend or L1Suspend states.

4.7.2 L1Suspend Delay

This state is entered from the Inactive state when the Receiver transitions to the L1Suspend state. This is a timed state with duration of TL1Residency.

4.7.3 FL1sus

This state is entered when the L1Suspend Delay interval expires. This state indicates that the hub is in a full L1 Suspend state.

4.7.4 GL1_Exit_H

This state is entered from the FL1sus state when a resume condition exists in the Hub Controller. The resume conditions are the same as those documented in the USB 2.0 specification (Section 11.4.4) for the GResume state. Resume condition also includes: a C_PORT_L1 bit is set in any port. In this state, the internal port generates signaling to emulate an SOP_FD to the Hub Repeater.

4.8 Downstream Facing Port – with L1 Additions

The following sections describe the L1 functional additions to the reference state machine for a downstream facing port. This description is organized with a complete overview state machine diagram, which includes the necessary functional additions to support L1. Then, only the new states are described. For a full description of the existing port states, please refer to Section 11.5 of the USB 2.0 Specification.

The downstream port state machine controls the behavior of each downstream port. There reference model provides for one downstream port state machine instance for each port. Figure 4-11 illustrates the additional states in downstream port state machine to support L1.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

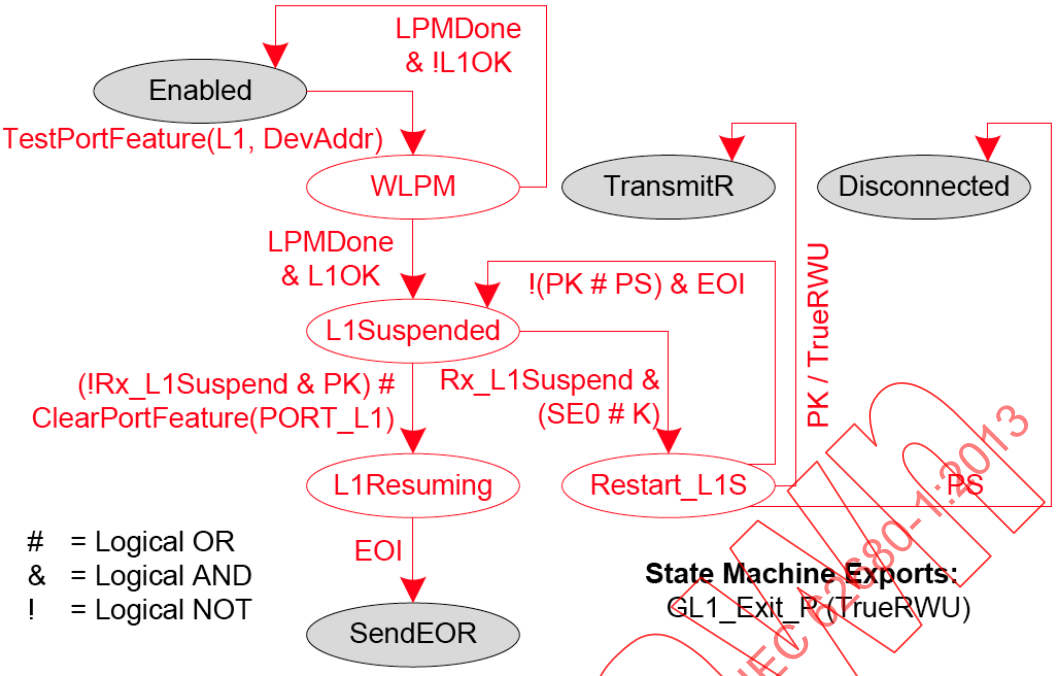


Figure 4-11. Addendum to Downstream Facing Hub Port State Machine

Table 4-5. Downstream Port Signal/Event Definitions (Addendum)

Signal/Event Name	Event/Signal Source	Description
PK	Internal	K lasting for at least TDDIS (see Table 7-13 in USB 2.0 Specification)
PS	Internal	SE0 lasting for at least TDDIS
EOI	Internal	End of timed interval or sequence
SE0	Internal	SE0 received on Port
Rx_L1Suspend	Receiver	Upstream port Receiver is in the L1Suspend state
TrueRWU	Internal	K lasting for at least TDDIS
LPMDone	LPM TE	Response signal from the LPM Transaction Engine that the LPM transaction has completed
L1OK	LPM TE	Indicates that the device acknowledged the LPM request

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4.8.1 WLPM

The port enters this state:

- From the Enabled state when it receives a SetandTestPortFeature(PORT_L1, DevAddr) request

In this reference model, each port state machine has access to an LPM Transaction Engine (LPM TE, see Figure 4-7). The LPM TE is used to run the LPM transaction to the device connected to the port in response to a SetandTestPortFeature() request. On entry to this state, the LPM TE is requested to issue an LPM transaction to the device at the device address DevAddr. The LPM TE will complete the transaction, including any retries that may be necessary. The transaction is conducted at the connected signaling speed rate indicated in the port.

The port will remain in this state until the LPM TE signals the requested LPM is completed (LPMDone). Note that the LPM TE provides a complete code back to the Hub Controller which it uses to respond to the data phase of the SetandTestPortFeature(PORT_L1) request.

4.8.2 L1Suspended

The port enters this state:

- From the WLPM state when the LPM TE signals that the LPM transaction is complete and that the response is L1OK. LPM TE will only return L1OK when the device responded with a NYET handshake to the LPM transaction.
- From the Restart_L1S state when a persistent K or persistent SE0 has not been seen within 9 µs of entering that state.

While a port is in the L1Suspended state, the port's differential transmitter is disabled. A high-speed port reverts from high-speed to full-speed terminations but its speed status continues to be high-speed. The port must not perform normal disconnect detection until at least 4ms after entering this state. An implementation must have a K/SE0 'noise' filter for a port that is in this state. This filter can time the length of K/SE0 and, if the length of the K/SE0 is shorter than TDDIS, the port must remain in this state.

4.8.3 L1Resuming

A port enters this state from the L1Suspended state in either of the following scenarios:

- If a 'K' is detected on the port and persists for at least 2.5 µs and the Receiver is not in the L1Suspended state.
- When a ClearPortFeature (PORT_L1) request is received.

This is a timed state. The duration of the timer must be set to TL1HubDrvResume1 if the entry is due to the ClearPortFeature (PORT_L1) request. The duration of the timer must be set to the TL1HubDrvResume2 if entry is due to detection of a 'K' in the L1Suspended state. While in this state, the hub drives a 'K' on the port.

4.8.4 Restart_L1S

A port enters the Restart_L1S state from the L1Suspended state when an SE0 or 'K' is seen at the port and the Receiver is in the L1Suspend state. In this state, the port continually monitors the bus state. If the bus is in the K state for at least TDDIS, the port sets the C_PORT_L1 bit, exits to TransmitR, and generates a signal called TrueRWU (exported as GL1_Exit_P). If the bus is in the SE0 state for at least TDDIS, the port exits to the Disconnected state. Either of these transitions must happen within 9 µs after entering the Restart_L1S state; otherwise, the port must transition back to the L1Suspended state.

4.8.5 SendEOR

This is the same state as defined in the USB 2.0 specification (Section 11.5.1.11), except that it has an additional entry path. This state is entered from the L1Resuming state after the interval timer expires. The remainder of the state definition is unchanged to support L1.

4.9 LPM Timing Characteristics

This section defines the timing parameters and their values that are used in Sections 4.2.1and 4.2.2.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 4-6. Summary LPM Timing Characteristics

Parameter	Symbol	Conditions	Min	Max	Units
L1 Residency	$T_{L1Residency}$	Figure 4-2	50	>50	µs
Device delay before transitioning to L1 after transmitting ACK	$T_{L1TokenRetry}$	Figure 4-2	8	10	µs
Host initiated L1 Exit Host drives resume time	$T_{L1HubDrvResume1}$	Figure 4-6 Note 1	50 ±1	1200 ±1	µs
Device initiated L1 Exit Device drives resume time	$T_{L1DevDrvResume}$	Figure 4-3	50 ±1	n/a	µs
L1 Exit device recovery time	$T_{L1ExitDevRecovery}$	Figure 4-3 Figure 4-6	10	n/a	µs
L1 Exit Latency (Host initiated)	$T_{L1ExitLatency1}$	Figure 4-6	60	1210	µs
L1 Exit Latency (Device initiated)	$T_{L1ExitLatency2}$	Figure 4-3	70	1000	µs
Host/Hub drive resume in response to Device remote wake	$T_{L1HubDrvResume2}$	Figure 4-3	60 ±1	990 ±1	µs
Host/Hub detect and begin reflecting resume	$T_{L1HubReflect}$	Figure 4-3	Implementation dependent	48	µs

Note 1. The range of $T_{L1HubDrvResume1}$ is not continuous. See Section 4.1 for details.

4.10 Hub Framework Extensions for LPM

This section details the extensions to the Hub framework necessary to support LPM.

Table 4-7 lists the feature selectors for the hub class and includes new codes for PORT_L1 and C_PORT_L1. This version of the table overrides the version presented in the USB 2.0 specification (Table 11-17).

Table 4-7. Hub Class Feature Selectors

	Recipient	Value
C_HUB_LOCAL_POWER	Hub	0
C_HUB_OVER_CURRENT	Hub	1
PORT_CONNECTION	Port	0
PORT_ENABLE	Port	1
PORT_SUSPEND (L2)	Port	2
PORT_OVER_CURRENT	Port	3
PORT_RESET	Port	4
PORT_POWER	Port	8
PORT_LOW_SPEED	Port	9

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 4-7. Hub Class Feature Selectors (cont.)

	Recipient	Value
PORT_L1	Port	10
C_PORT_CONNECTION	Port	16
C_PORT_ENABLE	Port	17
C_PORT_SUSPEND	Port	18
C_PORT_OVER_CURRENT	Port	19
C_PORT_RESET	Port	20
PORT_TEST	Port	21
PORT_INDICATOR	Port	22
C_PORT_L1	Port	23

4.10.1 Clear Port Feature

This request works as specified in the USB 2.0 specification, with the following additions. This request is used to clear the additional following features:

- PORT_L1
- C_PORT_L1

Clearing the PORT_L1 feature causes a host-initiated L1 exit (resume) on the specified port. If the port is not in the L1 state, the hub must treat this request as a functional no-operation.

When the hub supports LPM, the *wIndex* field is actually formatted into three fields as illustrated in Table 4-8.

Table 4-8. *wIndex* Definition for Clear Port Feature on an LPM Enabled Hub

Bits	Field Name	Description
15..8	Selector	Identifies the port indicator selector when clearing a port indicator.
7..4	HIRD	Host Initiated Resume Duration. The encoded value of the minimum time the hub must drive resume on the designated port. The encodings for this field are defined in Section 4.1. NOTE: the host must ensure the HIRD values used in SetAndTestPortFeature and ClearPortFeature (to the same device) are consistent.
3..0	Port	The downstream port number for the L1 feature.

4.10.2 Get Port Status

This request works as specified in the USB 2.0 specification, with the following additions.

4.10.2.1 Port Status Bits with L1 Additions

The definition of Port Status bits is unchanged for the USB 2.0 defined bit fields. This addendum defines bit 5 as the PORT_L1 status indicator bit field, as illustrated in Table 4-9. Refer to the USB 2.0 specification for full bit functionality descriptions not explicitly described in this addendum.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 4-9. Port Status Bits with L1 Additions

Bit	Description
0	**Current Connect Status:** (PORT_CONNECTION) This field reflects whether or not a device is currently connected to this port. 0 = No device is present. 1 = A device is present on this port.
1	**Port Enabled/Disabled:** (PORT_ENABLE) Ports can be enabled by the USB System Software only. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by the USB System Software. 0 = Port is disabled. 1 = Port is enabled.
2	**Suspend:** (PORT_SUSPEND) This field indicates whether or not the device on this port is suspended. Setting this field causes the device to suspend by not propagating bus traffic downstream. This field may be reset by a request or by resume signaling from the device attached to the port. 0 = Not suspended. 1 = Suspended or resuming.
3	**Over-current:** (PORT_OVER_CURRENT) If the hub reports over-current conditions on a per-port basis, this field will indicate that the current drain on the port exceeds the specified maximum. For more details, see Section 7.2.1.2.1. 0 = All no over-current condition exists on this port. 1 = An over-current condition exists on this port.
4	**Reset:** (PORT_RESET) This field is set when the host wishes to reset the attached device. It remains set until the reset signaling is turned off by the hub. 0 = Reset signaling not asserted. 1 = Reset signaling asserted.
5	**L1:** (PORT_L1) This field indicates whether or not the link state of this port is in L1. This field gets set as a side-effect of a successful SetandTestPortFeature(PORT_L1). This field may be reset by a request or by resume signaling from the device attached to the port. 0 = Not in the L1 link state. 1 = L1 state or resuming.
6-7	**Reserved** These bits return 0 when read.
8	**Port Power:** (PORT_POWER) This field reflects a port's logical, power control state. Because hubs can implement different methods of port power switching, this field may or may not represent whether power is applied to the port. The device descriptor reports the type of power switching implemented by the hub. 0 = This port is in the Powered-off state. 1 = This port is not in the Powered-off state.
9	**Low-Speed Device Attached:** (PORT_LOW_SPEED) This is relevant only if a device is attached. 0 = Full-speed or High-speed device attached to this port (determined by bit 10). 1 = Low-speed device attached to this port.
10	**High-speed Device Attached:** (PORT_HIGH_SPEED) This is relevant only if a device is attached. 0 = Full-speed device attached to this port. 1 = High-speed device attached to this port.
11	**Port Test Mode:** (PORT_TEST) This field reflects the status of the port's test mode. Software uses the SetPortFeature() and ClearPortFeature() requests to manipulate the port test mode. 0 = This port is not in the Port Test Mode. 1 = This port is in Port Test Mode.
12	**Port Indicator Control:** (PORT_INDICATOR) This field is set to reflect software control of the port indicator. For more details see Sections 11.5.3, 11.24.2.7.1, and 11.24.2.13. 0 = Port indicator displays default colors. 1 = Port indicator displays software controlled color.
13-15	**Reserved** These bits return 0 when read.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4.10.2.1.1 PORT_SUSPEND

This bit is set to one when the port is selectively transitioned to the L1 state by the USB System Software. While this bit is set, the hub does not propagate downstream-directed traffic to this port, but the hub will respond to resume signaling from the port. This bit can be set

only if the port's PORT_ENABLE bit is set, the hub receives a SetandTestPortFeature(PORT_L1) request and the downstream device responds with an ACK handshake to the LPM transaction. This bit is cleared to zero on the transition from the SendEOP state to the Enabled state, or on the transition from the Restart_S state to the Transmit state, or on any event that causes the PORT_ENABLE bit to be cleared while the PORT_L1 bit is set.

The SetandTestPortFeature(PORT_L1) request may be issued by the USB System Software at any time but will have an effect only as specified in Section 4.1.

4.10.2.2 Port Change Bits with L1 Additions

The definition of Port Change bits is unchanged for the USB 2.0 defined bit fields. This addendum defines bit 5 as the C_PORT_L1 status indicator bit field as illustrated in Table 4-10.

Table 4-10. Port Change Bits with L1 Additions

Bit	Description
0	**Connect Status Change:** (C_PORT_CONNECTION) Indicates a change has occurred in the port's Current Connect Status. The hub device sets this field as described in Section 11.24.2.7.2.1. 0 = No change has occurred to Current Connect status. 1 = Current Connect status has changed.
1	**Port Enable/Disable Change:** (C_PORT_ENABLE) This field is set to one when a port is disabled because of a Port_Error condition (see Section 11.8.1).
2	**Suspend Change:** (C_PORT_SUSPEND) This field indicates a change in the host-visible suspend state of the attached device. It indicates the device has transitioned out of the Suspend state. This field is set only when the entire resume process has completed. That is, the hub has ceased signaling resume on this port. 0 = No change. 1 = Resume complete.
3	**Over-Current Indicator Change:** (C_PORT_OVER_CURRENT) This field applies only to hubs that report over-current conditions on a per-port basis (as reported in the hub descriptor). 0 = No change has occurred to Over-Current Indicator. 1 = Over-Current Indicator has changed. If the hub does not report over-current on a per-port basis, then this field is always zero.
4	**Reset Change:** (C_PORT_RESET) This field is set when reset processing on this port is complete. 0 = No change. 1 = Reset complete.
5	**L1 Change:** (C_PORT_L1) This field indicates a change in the host-visible link state of the associated port. It indicates that the link has transitioned out of the L1 state. This field is set only when the entire L1 exit process has completed, that is, the hub has ceased signaling resume on this port. 0 = No change 1 = L1 Exit complete
6-15	**Reserved.** These bits return 0 when read.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4.10.2.3 C_PORT_L1

This bit is set on the following transitions:

- On transition from the Resuming state to the SendEOR state
- On transition from the Restart_L1S state to the TransmitR state

This bit will be cleared by a ClearPortFeature(C_PORT_L1) request, or while the port is in the Powered-off state.

4.10.3 Set and Test Port Feature

This request sets port feature that is specified in the setup data and returns a completion code indicating the result of setting the feature.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
10100011B	SET_AND_TEST	Feature Selector	Selector-specific	Selector-specific	Selector-specific

PORT_L1 is the only feature selector (*wValue*) that is used with the SET_AND_TEST command. Table 4-11 summarizes the allowed feature selectors and selector-specific field settings for this command.

Table 4-11. Set and Test Port Feature Details

Feature	wIndex	wLength	Data
PORT_L1	**Bits** **Value** 3..0 Port Number 7..4 HIRD 14..8 Device Address 15 Remote Wake Enable	One	Completion Code. Value encodings are: **Value** **Meaning** 00H Success 01H Reserved 10H NYET 11H Timeout

The hub must meet the following requirements:

- If the port is in the L1 state, the hub must treat a SetandTestPortFeature(PORT_L1) request as a functional no-operation and report a Success completion code.
- If the port is not in the Enabled, Transmitting, or L1 state, the hub must treat a SetandTestPortFeature(PORT_L1) request as a functional no-operation.
- If the port is not in the Powered-off state, the hub must treat a SetPortFeature(PORT_POWER) request as a functional no-operation.

It is a Request Error if *wValue* is not a feature selector in the set {PORT_L1}, if *wIndex*[7..0] specifies a port that does not exist, or if *wLength* is not as specified above in Table 4-11. If the hub is not configured, the hub's response to this request is undefined. The host should ensure that no active split transactions are in flight (not completed) to the device at Device Address. The behavior of the system if an LPM transaction occurs to a device that is in the midst of a split transaction is undefined.

The value of the HIRD field (Host Initiated Resume Duration) field is passed directly through to the <<tbd>> field of the LPM transaction. System software sets this field to indicate to the recipient device how long the host will drive resume if it (the host) initiates and exit from L1 via ClearPortFeature(). Refer to Section 4.1 for the definition of how values in this field are encoded.

4.10.3.1 Port_L1

This commands a host or hub port to transition into the L1 link state. The host/hub receiving this command will cease repeating traffic down the addressed port (*wIndex*) and will transmit an LPM token to the device connected downstream of the port. It is the responsibility of the system to ensure the host/hub, port number and device address are consistent. The hub NAKs the data stage IN transaction until the LPM transaction has completed. Upon completion of the LPM transaction, the hub will complete the data stage of this command by returning a one byte completion code. The encoding of the completion code is listed in Table 4-11. The host/hub must provide immediate retries in the event that it detects an error during

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

the LPM transaction. If it cannot complete the LPM transaction after three attempts, it will restore the port to the enabled state and returns a completion code indicating that the LPM request encountered a timeout. If the device responds to the LPM transaction with a NYET, the host/hub restores the port to the enabled state and returns a completion code indicating NYET. If the device responds to the LPM transaction with an ACK, the host/hub places the downstream facing port into the L1 state, sets the PORT_L1 status bit to a one and returns a completion code of Success.

4.10.4 Set Port Feature

This command cannot be used to set the PORT_L1 or the C_PORT_L1 features and will report a Request Error if attempted.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

USB ENGINEERING CHANGE NOTICE

Title: Suspend Current Limit Changes
Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN:

This ECN raises the maximum suspend current to 2.5 mA for all devices except InterChip USB (ICUSB) devices. ICUSBdevice suspend current limit remains at 150 µA.

For the purposes of establishing suspend current limits, there is no difference between a highpower device and a low-power device, nor is there a difference between devices with DEVICE_REMOTE_WAKEUP feature enabled and those without DEVICE_REMOTE_WAKEUP feature enabled. All devices (except ICUSB) have a maximum suspend current limit of 2.5 mA.

NOTE: This ECN affects the core USB 2.0 specification. Previous ECNs and specifications (except ICUSB) that make reference to the former 500 µA limit should be read according to this ECN to permit the new 2.5 mA suspend current limit.

Reasons for ECN:

Simplification of hardware designs, reduced development effort and cost and time wasted waiting for automatic suspend current waivers. Furthermore, the USB Battery Charging Specification presumes that 2.5 mA suspend current is available to all devices.

Impact on existing peripherals and systems:

No impact on currently certified devices. Hosts with very tight suspend current budgets may be affected by new bus-powered hubs drawing higher suspend current prior to configuration, or by new low-power devices using the new limit.

Hardware Implications:

Devices that formerly required waivers for (2.5 mA > ICCSL > 500 µA) no longer require waivers. Bus-powered hubs may require up to 12.5 mA prior to configuration, 2.5 mA for the hub electronics and 2.5 mA for each port with an attached device.

Software Implications:

Host power-management software will need to be aware of the higher potential suspend current when budgeting low-power states and when configuring bus-powered hubs.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Compliance Testing Implications:

All tests that verify 500 µA suspend current are replaced by 2.5 mA suspend current tests. A new test for configured bus-powered hubs permits 12.5 mA suspend current.

Time-averaging schemes for determining suspend current compliance (Section 7.2.3) are unchanged by this ECN.

Changes to USB-IF Full and Low Speed Compliance Test Procedure Section D.6, Step 2

- Pass
  1) Measured current is 2.5 mA or less
  OR
  2) Device is a configured bus-powered hub AND measured current is 12.5 mA or less
- Fail
  1) Measured current exceeds 12.5 mA AND device is a configured bus-powered hub
  OR
  2) Measured current exceeds 2.5 mA

Specification Changes:

All changes (except repagination and indexing) occur in Chapter 7.

Chapter 7 changes:

Replace Section 7.2.3 Paragraph 1 with the following:

> All USB Devices (except bus - powered hubs) may draw up to 2.5 mA during suspend. Configured bus - powered hubs and configured bus - powered compound devices may consume a maximum suspend current of 12.5 mA. This limit includes the 2.5 mA suspend current that must be available for each externally - available downstream port on the hub. Unconfigured bus - powered hubs and compound devices operate as a low - power devices and must limit suspend current to 2.5 mA. When computing suspend current, the current from VBUS through the bus pull - up and pull - down resistors must be included.

Replace Section 7.2.3 Paragraph 2 Sentence 4 with the following:

> The average current cannot exceed the average suspend current limit (ICCS or ICCSHUB, see Table 7-7) during any 1.0-second interval (TSUSAVG1).

Change reference to "ICCS(H/L)" from I_{CCS} in Figure 7 - 48 to "ICCS"

Change Table 7 - 7 ICCSH and ICCSL as follows:

Suspended Device (except Configured Bus - Powered Hub)	I_{CCS}	Section 7.2		2.5	mA
Suspended Configured Bus - Powered Hub	I_{CCSHUB}	Section 7.2.3		12.5	mA

Remove Table 7 - 7 Note 15.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

USB ENGINEERING CHANGE NOTICE

Title: 5V Short Circuit Withstand Requirement Change
Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN:

This ECN proposes to remove the 5V short circuit withstand requirement for the USB D+ and/or D- bus traces.

NOTE: This ECN affects the core USB 2.0 specification.

Reasons for ECN:

1. The 5V short circuit requirement limits the scalability of silicon fabrication processes moving forward. The advanced processes do not easily support this voltage level, which is 5X or more of the native process voltage.

2. The proposed change also provides an opportunity in reducing the silicon area and average power. Analysis shows a ~16% transceiver silicon area reduction and a ~3x standby idle power saving can be achieved on a silicon process which does not support a 3.3 volt tolerant transistor.

3. The original intent of this requirement was to protect the ports against degradation if the USB cable was damaged resulting in a VBUS short to only to one or both of the data signal lines (Dp or Dm). Research into the likelihood of this condition occurring has resulted in zero reports of this fail condition. (Source: USB IF database)

Lab tests were also conducted to comprehend the probability of a VBUS short to Dp or Dm when the cables were crushed or cut. No failures were seen under “normal,” unintentional, damage circumstance like rolling a chair across a cable, shutting it in a metal cabinet, or a cable cut with scissors. In order to see what shorts could occur, more extreme (though improbable) tests were conducted. Fifty High Speed cables were crushed in a vise until a short occurred. In every case, VBUS shorted to ground first. This is protected by the over-current mechanism. Any short that might have occurred to a data signal line would be masked by a short to ground. Another extreme test condition used diagonal cutters or a utility knife to cut the cable. In this case, when a short was observed, 713 out of 713 High Speed cable cut shorted VBUS to ground.

Low Speed cable almost exhibits similar characteristics when going through the cable cutting and crushing experiment. 0.4% probability of getting data wire shorted to VBUS was observed when a particular type of Low Speed mouse cable was crushed at the exact same location by > 260 lbs force, which was exerted by a galvanized pipe against the concrete floor. However the shorts were intermittent. There was no continuous short that may cause long term reliability issue (more detailed experiment results can be found in the attachment, Cable Test Report.doc). In summary, the probability of a damaged cable which causes VBUS shorted to data bus is very low.

4. Some devices may have connected their pull-up resistor to VBUS, as described in USB ECN titled “Pull-up/pull-down resistors”. The voltage divider formed by the device pull-up and host pull-down will guarantee the data wire park at a safe voltage level, which is below the VBUS value. In summary, the host/hub will not see 5V at the wire when interoperating with these devices.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5. A final argument for removing the 5 volt tolerance requirement from USB is that other external interconnect standards like 1394 do not protect against the power short to the data wires.

Impact on existing peripherals and systems:

None

Hardware Implications:

OEMs that require short circuit withstand to VBUS can implement the requirement with an external hub or other external protection circuitry.

Software Implications:

None

Compliance Testing Implications:

Compliance to this requirement has never been tested so removal of the VBUS short condition will have no effect on compliance testing.

Specification Changes:

Change section 7.1.1 USB Driver Characteristic, 6th paragraph as follows:

A USB transceiver is required to withstand a continuous short circuit of D+ and/or D- to ~~VBUS,~~ GND, other data line, or the cable shield at the connector, for a minimum of 24 hours without degradation. It is recommended that transceivers be designed so as to withstand such short circuits indefinitely. The device must not be damaged under this short circuit condition when transmitting 50% of the time and receiving 50% of the time (in all supported speeds). The transmit phase consists of a symmetrical signal that toggles between drive high and drive low. ~~This requirement must be met for max value of VBUS (5.25 V).~~ A USB transceiver isrecommended, but not required, to withstand a continuous short of D+ and/or D- to VBUS for a minimum of 24 hours without degradation.

It is recommended that these AC and short circuit stresses be used as qualification criteria against which the long-term reliability of each device is evaluated.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

USB ENGINEERING CHANGE NOTICE

Title: Device Capacitance
Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN

This ECN ensures that all USB peripherals have a detectable change in capacitance on VBus when they are attached. This ensures that OTG devices and Embedded Hosts can detect attachment of peripherals in the absence of VBus.

In addition the ECN clarifies an inconsistency in the USB 2.0 specification.

Reasons for ECN

OTG devices and Embedded Hosts are becoming more prevalent in the eco-system and it would improve the interoperability and end user ease of use if OTG devices and hosts can properly detect the attachment of devices when VBus is turned-off (typically to conserve power).

Impact on Existing Peripherals and Systems:

It is estimated that around 10% of USB devices going into compliance have less than 1µF capacitance on VBUS as measured by the inrush test. All of these are self-powered devices.

With the increased uptake of USB charging and a desire to be detectable via the Attach Detection Protocol (ADP), it is anticipated that the number of USB devices with less than 1µF capacitance will reduce.

Hardware Implications:

A small number of USB devices will need to add a 1µF capacitance to their VBUS line in order to be detectable via ADP. The majority of USB devices already have a capacitance larger than 1µF on VBUS.

Software Implications:

Large improvement in usability since USB device attachment/detachment can now be detected reliably.

Compliance Testing Implications:

Method of testing this is inrush. The compliance tests will need to be verified to ensure that the limits on the test are still correct.

Specification Changes

The last sentence in Section 7.2.4.2 to be modified from:

> "A minimum of 1.0uF is recommended for bypass across Vbus."

To the following text:

> "A USB device is required to expose a capacitance on the VBUS pin of its connector of CRPB. This capacitance shall be greater than CRPB min for voltages on the VBUS pin from 0V to 5.25V, regardless of whether the USB device is powered or unpowered."

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Table 7-7 entry for CRPB to be modified as follows:

Upstream Facing Port Bypass Capacitance	CRPB	VBUS to GND across the whole VBUS voltage range (0 - 5.25V); Note 9, Section 7.2.4.1	1.0	10.0	µF

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

USB ENGINEERING CHANGE NOTICE

Title: Material Change
Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN

Change several of the materials in the USB 2.0 specification and allow optional alternatives that are either better materials or equivalent substitutes that ultimately enhance the products. The substitute materials must meet all performance requirements as defined in the USB 2.0 document and Compliance Specification.

Reasons for ECN

By allowing alternate materials, designers will be able to make better USB 2.0 products. The optional materials lend themselves to either better performance and/or better costs to manufacture. One of the changes permits a broader range of materials, including halogen-free grades of plastics.

Impact on Existing Peripherals and Systems:

These changes are optional and they have no impact on existing products. The proposed substitutes will give the manufacturer the option to choose the most cost effective materials while maintaining the overall performance of the products.

Hardware Implications:

None

Software Implications:

None

Compliance Testing Implications:

If a product is submitted as RoHS compliant then the Independent Test Labs (ITLs) must verify that the product is lead free and verify that it meets Solderability requirements. This has already been communicated to the ITLs.

Specification Changes

Changes to Section 6.5.3.1, 6.5.4.1 and on all cable and connector drawings:

Remove all references to UL 94-V-0 rating and remove all references to use of PVC (Polyvinyl Chloride Jacket).

Changes to Figure 6-11 (Figure 6-14 in MiniB ECN) in Section 6.6:

Allow equivalent materials as braid material.

Outer shield interwoven >= 65% Tinned Copper Braid or equivalent braid material.

Changes to Section 6.5.3.2 Receptacle Shell Materials from:

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Add stainless steel as an optional shell material:

Substrate Material: 0.30 + 0.05 mm phosphor bronze, nickel silver, or other copper based high strengthmaterials.

Plating:

1. Underplate: Optional. Minimum 1.00 micrometers (40 microinches) nickel. In addition, manufacturer may use a copper underplate beneath the nickel.

2. Outside: Minimum 2.5 micrometers (100 microinches) bright tin or bright tin-lead.

3. Optionally, the shell material may simply be unplated stainless steel

Changes to Section 6.5.3.3 Receptacle Contact Materials (Section 6.5.3.3.1 Series “A” and Series “B” in MiniB ECN)

Allow matte tin and matte nickel plating on solder tails. Add test method for tin based coatings.

Substrate Material: 0.30 + 0.05 mm minimum half-hard phosphor bronze or other high strength copper based material.

Plating: Contacts are to be selectively plated.

A. Option I

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.

2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.70 micrometers (28 microinches) palladium.

3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead, matte tin plating or matte nickel plating over the underplate

4. Tin based coatings may be susceptible to tin whiskering and shall be tested according to JEDEC Standard JESD201, Environmental Acceptance Requirements for Tin Whisker Susceptibility of Tin and Tin Alloy Surface Finishes, Class 2. The supplier should use a suitable tin whisker mitigation practice (see IPC/JEDEC JP002) and provide data demonstrating compliance with the Class 2 level tin whisker acceptance testing requirements of JEDEC standard JESD-201.

B. Option II

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.

2. Mating Area: Minimum 0.05 micrometers (2 microinches) gold over a minimum of 0.75 micrometers (30 microinches) palladium-nickel.

3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead, matte tin plating or matte nickel plating over the underplate

4. Tin based coatings may be susceptible to tin whiskering and shall be tested according to JEDEC Standard JESD201, Environmental Acceptance Requirements for Tin Whisker Susceptibility of Tin and Tin Alloy Surface Finishes, Class 2. The supplier should use a suitable tin whisker mitigation practice (see IPC/JEDEC JP002) and

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

provide data demonstrating compliance with the Class 2 level tin whisker acceptance testing requirements of JEDEC standard JESD-201.

C. Option III

1. Underplate: Minimum 1.25 micrometers (50 microinches) nickel. Copper over base material is optional.

2. Mating Area: Minimum 0.75 micrometers (30 microinches) gold.

3. Solder Tails: Minimum 3.8 micrometers (150 microinches) bright tin-lead, matte tin plating or matte nickel plating over the underplate

4. Tin based coatings may be susceptible to tin whiskering and shall be tested according to JEDEC Standard JESD201, Environmental Acceptance Requirements for Tin Whisker Susceptibility of Tin and Tin Alloy Surface Finishes, Class 2. The supplier should use a suitable tin whisker mitigation practice (see IPC/JEDEC JP002) and provide data demonstrating compliance with the Class 2 level tin whisker acceptance testing requirements of JEDEC standard JESD-201.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

USB ENGINEERING CHANGE NOTICE

Title: USB 2.0 Phase-locked SOFs
Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN

The current USB2.0 spec has no requirement that the SOFs sent before and after a selective suspend condition need to be phase-locked. Also the LPM ECN has no requirements on this when going in and out of level 1 power management. This ECN proposes to add a requirement in the spec that host controllers must guarantee that the SOFs sent before and after a selective suspend or L1 condition must be in phase-lock.

Reasons for ECN

The benefit of this ECN is that isochronous applications (like audio speakers) that use the SOFs to synchronize internal clocks, can now also do aggressive power management. If they provide enough buffering on the device side, it is possible that they transfer lots of data over the USB bus and then go into suspend or L1 while consuming the data on the application side.

Impact on Existing Peripherals and Systems:

As this ECN proposes to add something in the spec that was not explicitly mentioned before, it could be that existing systems today already support this, but it is not guaranteed. To make sure that isochronous devices can make optimal use of power savings on the USB, this requirement is being added to the USB specification.

Hardware Implications:

No changes to a host controller implementation are required as this ECN does not require host controllers to maintain phase-lock between SOF tokens across a global/system suspend.

Software Implications:

No changes to software are required.

Compliance Testing Implications:

The EHCI Test Suite must be eventually updated to add a test that checks that the SOFs on the bus are in phase-lock before and after a selective suspend.

Specification Changes

This ECN covers one addition to the USB2.0 spec to guarantee that isochronous USB devices can make optimal use of the power management features defined in the USB2.0 spec and the LPM ECN.

Section 7.1.7.7 describes the general behavior in case of suspend/resume. This section only states that a host must start sending traffic within 3ms after completion of the resume signaling to prevent the devicefrom going back into suspend. It doesn't mention anything on the phase-lock of the SOF tokens that are sent.

Following text must be added at the end of paragraph 3 of section 7.1.7.7:

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

“The SOF tokens sent by a host after resuming a selective suspended port must be still phase-locked with respect to the SOF tokens that were sent before the suspend/resume signaling.”

Section 10.2.3 describes the required behavior of a USB2 host controller for Frame and micro Frame generation. This section doesn’t describe any requirements on the Frame or micro Frame generation when coming out of suspend.

Following text must be added at the end of section 10.2.3:

“When a downstream port of the host is put in selective suspend, the host must keep the frame timer running. A host is required to start sending SOF tokens after the suspend/resume signaling that are still in phase-lock with the SOF tokens that were sent before the suspend/resume. If the host controller is put in global suspend, it is not required to keep phase-lock between the SOF tokens sent before and after the global suspend.”

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

USB ENGINEERING CHANGE NOTICE

Title: USB 2.0 Connect Timing
Applies to: Universal Serial Bus Specification, Revision 2.0

Summary of ECN

This ECN addresses the following three issues:

• dead battery pre-connect current

• post-connect current timing

• VBUS valid to connect timing

Dead battery pre-connect current

The Battery Charging spec contains the Dead Battery Provision, which allows devices with weak or dead batteries to draw up to 100mA for the time required to power the device up, and bring it to a state where it can connect.

The USB 2.0 spec currently allows devices to draw 100mA for only 100ms before connecting. This time is not sufficient to for most portable devices with dead batteries to power up.

The USB 2.0 spec would be modified such that a device with a dead battery could draw 100mA for the time required to power up the device, and bring it to a state where it can connect.

Post-connect current timing

The Battery Charging spec allows devices to draw 100mA for 1 second after the connect event, regardless of what is happening on the bus.

The USB 2.0 spec currently requires devices to drop down to suspend current within 10ms of no bus activity, during both the attach debounce time (TATTDB) and during the reset time (TDRST). In practice, no devices do this. During USB compliance testing, suspend current is not tested until several seconds after the connect event.

The USB 2.0 spec would be modified such that a device is not required to drop down to the suspend current during the 1 second immediately following the connect event.

VBUS session valid to connect time

During compliance testing, a device that is already powered up must connect within 1 second of VBUS valid. The reason for this test is to provide a user experience where something happens within a short time of a device being attached to a host.

The USB 2.0 spec is not explicit in this issue. Figure 7-29 can be interpreted to imply a VBUS valid to connect time of 100ms. But Figure 7-29 does not distinguish between the VBUS valid event and the time at which the device draws current.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

The USB spec would be modified such that the VBUS valid to connect time is explicitly called out. As well, the USB spec would differentiate between the VBUS valid event and the time at which the device starts to draw current from VBUS.

Reasons for ECN

The benefits of this ECN are outlined below.

Dead battery pre-connect current

The USB 2.0 spec will align with the Dead Battery Provision, which allows a device with a dead battery to power up when attached to a USB host.

Post-connect current timing

The USB 2.0 spec will align with the fact that devices don't drop to suspend current during the attach debounce time, TATTDB. The spec will also align with the USB compliance program which doesn't check for suspend current until at least one second after the connect event.

VBUS session valid to connect time

The USB 2.0 spec will align with the USB compliance program, which requires a powered on device to connect within 1 second of VBUS valid.

Impact on Existing Peripherals and Systems:

There is no impact to existing peripherals and systems.

Hardware Implications:

No change is required of existing peripheral hardware.

Host and hub hardware must now be able to provide a unit load current for at least 1 sec (TCON_ISUSP) after a peripheral connects.

Previously, peripherals would have to drop down to suspend current several seconds after the connect event if a host suspended the bus. This several second time came from compliance testing and not from the USB 2.0 spec. The USB 2.0 spec was more even constraining, and required a peripheral to drop down to suspend current immediately after the connect event, and before start of reset. However, this was not realistic for connect to reset times of 100ms or less, and was universally ignored.

Software Implications:

There are no software changes required for either the host or peripheral. The host is still allowed to suspend the bus anytime after connect.

Compliance Testing Implications:

Existing compliance tests are not affected for self powered devices or devices with good batteries. Devices with weak or dead batteries should also be tested against the Portable Device Compliance Tests associated with the Battery Charging 1.1 specification.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Specification Changes

INSTRUCTIONS: In this section of the ECN, any paragraph starting with the word "INSTRUCTIONS" contains instructions for modifying the text of the USB specification. Any other paragraphs contain new text that should be inserted in the USB specification.

INSTRUCTIONS: In Section 7.1.7.3, replace Figure 7-29 as well as the previous one paragraph and following six paragraphs with the following.

Figure 7-29 shows the timing associated with a peripheral connecting to a host or hub port.

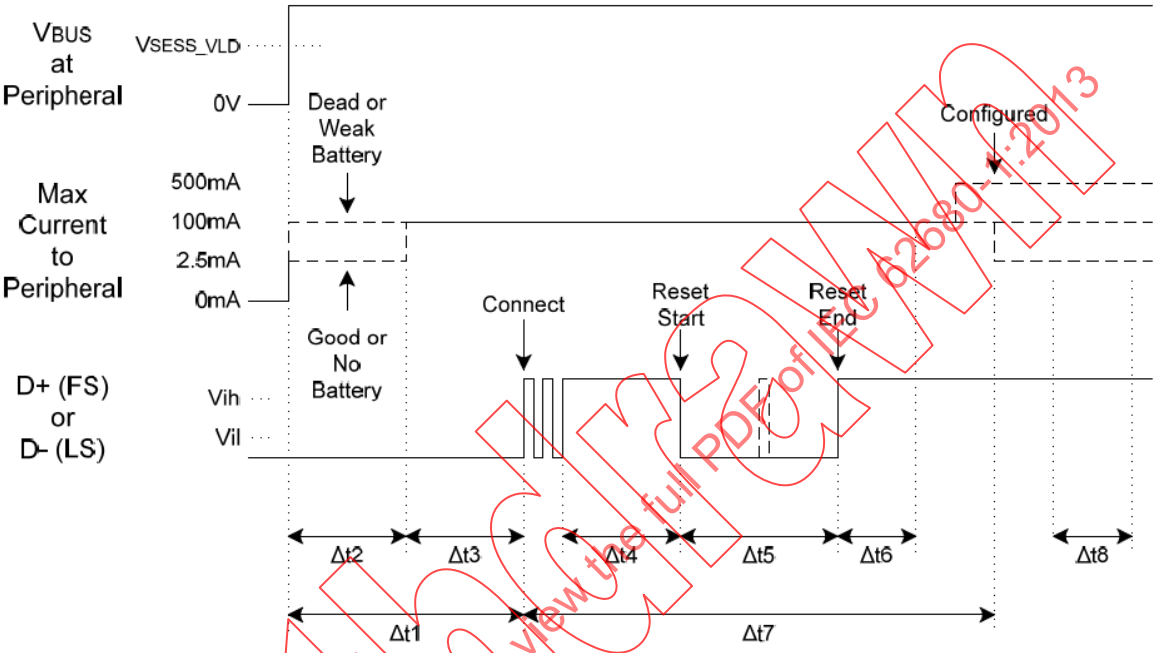


Figure 7-29 Connect Event Timing

Δt1 (TSVLD_CON_PWD, TSVLD_CON_WKB) This is the maximum time from when VBUS crosses VSESS_VLD to when the peripheral is required to connect (pull D+ or D- high) and be ready for enumeration. For bus powered peripherals and peripherals that are already powered up, this time is TSVLD_CON_PWD. For portable devices with dead or weak batteries, this time is TSVLD_CON_WKB. During TSVLD_CON_WKB, a portable device with a dead or weak battery that is drawing up to IUNIT is required to signal the upstream port by driving D+ to VDP_SRC. If a portable device with a dead or weak battery cannot connect and be enumerated after drawing IUNIT for TSVLD_CON_WKB, then it shall reduce its current draw to ISUSP.

Δt2 (TSVLD_UNIT) This time starts when VBUS crosses the VSESS_VLD threshold, and ends 100ms before the connect event. During this time, portable devices with dead or weak batteries are allowed to draw up to IUNIT from VBUS. All other peripherals are only allowed to draw up to ISUSP during this time.

Δt3 (TUNIT_CON) This is the maximum time that a bus powered peripheral or a portable device with a good battery can draw up to IUNIT from VBUS before connecting.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Δt4 (TCON_RST) This is a debounce interval with a minimum duration of 100 ms that is provided by USB System Software. It ensures that the electrical and mechanical connection is stable before software attempts to reset the attached device. The interval starts when the USB System Software is notified of a connect event (D+ or D- pulled high). The interval restarts if there is a disconnect event. The debounce interval ensures that power is stable at the device for at least 100 ms before resetting the peripheral.

Δt5 (TDRST) This is the period of time hubs drive reset to a peripheral. Refer to Section 7.1.7.5 and Section 11.5.1.5 for details.

Δt6 (TRSTRCY) The USB System Software guarantees a minimum of 10 ms for reset recovery. Peripheral response to any bus transactions addressed to the default device address during the reset recovery time is undefined.

Δt7 (TCON_ISUSP) After connecting, a peripheral is allowed to draw up to one unit load current for a time of TCON_ISUSP, before having to obey the rules of dropping down to suspend current. If, during this time, a peripheral becomes configured, it is then allowed to draw its configured current.

Δt8 (T2SUSP) After a time of TCON_ISUSP from the connect event, a peripheral must obey the rules of going into suspend (see Section 7.1.7.6).

Figure 7-29 shows the connect event timing from the perspective of peripheral. From a hub perspective, there is delay between the time that the hub is requested to switch on a port, to the time that the hub outputs a valid voltage on VBUS. This delay is a function of the type of hub port switch. Hubs report this time in the hub descriptor (see Section 11.15.2.1), which can be read via a request to the Hub Controller (see Section 11.16.2.4).

INSTRUCTIONS: Globally replace TCCS with TSUSP.

INSTRUCTIONS: In Table 7-7 below the row for Low-power port, add a row for the parameter VSESS_VLD. In the Parameter box, add the text “Peripheral session valid threshold”. The min and max values are 0.8V and 4.0V respectively.

INSTRUCTIONS: In Table 7-14, replace TATTDB with TCON_RST. In the Parameter box, replace the existing text with “Debounce interval provided by USB System Software after peripheral connects.”

INSTRUCTIONS: In Table 7-14, replace TSIGATT with TSVLD_CON_PWD. In the Parameter box, replace the existing text with “Time from VBUS crossing VSESS_VLD to a powered up peripheral connecting (pull D+ or D- above VIH).” Replace the max value of 100ms with 1sec.

INSTRUCTIONS: In Table 7-14, add the parameter TSVLD_CON_WKB, with a maximum value of 45 minutes. In the Conditions box put “Section 7.1.7.3”. In the Parameter box put “Time from VBUS crossing VSESS_VLD to a peripheral with a dead or weak battery connecting and being ready to enumerate.”

INSTRUCTIONS: In Table 7-14, add the parameter TUNIT_CON, with a maximum value of 100 msec. In the Conditions box put “Section 7.1.7.3”. In the Parameter box put “Time a peripheral can draw unit load from VBUS before connecting.”

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

INSTRUCTIONS: In Table 7-14, add the parameter T_{CON_ISUSP}, with a maximum value of 1 sec. In the Conditions box put “Section 7.1.7.3”. In the parameter box put “Time after connect when peripheral must obey rules of suspend.”

INSTRUCTIONS: In Table 7-7, under the heading “Output levels for low/full-speed”, add the parameter V_{DP_SRC} with a minimum value of 0.5V and a maximum value of 0.7V.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

USB ENGINEERING CHANGE NOTICE

Title: USB TEST_MODE selector values
Applies to: USB Specification Revision 2.0

Summary of ECN:

Add new TEST_MODE selector values 0x06 and 0x07 from the reserved list in Table 9-7 of the USB Revision 2.0 Specification.

Reasons for ECN:

The use of these reserved selector values is necessary to allow automated compliance testing for On-The-Go and Embedded Host units.

Impact on Existing Peripherals and Systems:

No impact, as these values are currently reserved.

Hardware Implications:

There are no hardware implications.

Software Implications:

There are no software implications.

Compliance Testing Implications:

Compliance testing is covered in the OTG and EH Supplements. This proposal to flag the use of the TEST_MODE selector values has no additional impact.

Specification Changes:

Replace Following Part of Section 9.4.9:

Table 9-7. Test Mode Selectors

Value	Description
00H	Reserved
01H	Test_J
02H	Test_K
03H	Test_SE0_NAK
04H	Test_Packet
05H	Test_Force_Enable
06H-3FH	Reserved for standard test selectors
3FH-BFH	Reserved
C0H-FFH	Reserved for vendor-specific test modes.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

If the feature selector is *TEST_MODE*, then the most significant byte of *wIndex* is used to specify the specific test mode. The recipient of a SetFeature(TEST_MODE…) must be the device; i.e., the lower byte of *wIndex* must be zero and the *bmRequestType* must be set to zero. The device must have its power cycled to exit test mode. The valid test mode selectors are listed in Table 9-7. See Section 7.1.20 for more information about the specific test modes.

If *wLength* is non-zero, then the behavior of the device is not specified.

If an endpoint or interface is specified that does not exist, then the device responds with a Request Error.

Default state: A device must be able to accept a SetFeature(TEST_MODE, TEST_SELECTOR) request when in the Default State. Device behavior for other SetFeature requests while the device is in the Default state is not specified.

Address state: If an interface or an endpoint other than endpoint zero is specified, then the device responds with a Request Error.

Configured state: This is a valid request when the device is in the Configured state.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

With:

Table 9-7. Test Mode Selectors

Value	Description
00H	Reserved
01H	Test_J
02H	Test_K
03H	Test_SE0_NAK
04H	Test_Packet
05H	Test_Force_Enable
06H-07H	On-The-Go and Embedded Host Test Functions
08H-3FH	Reserved for standard test selectors
3FH-BFH	Reserved
C0H-FFH	Reserved for vendor-specific test modes.

If the feature selector is *TEST_MODE*, then the most significant byte of *wIndex* is used to specify the specific test mode. The recipient of a SetFeature(TEST_MODE…) must be the device; i.e., the lower byte of *wIndex* must be zero and the *bmRequestType* must be set to zero. The device must have its power cycled to exit test mode. The valid test mode selectors are listed in Table 9-7. See Section 7.1.20 for more information about the specific test modes.

The functionality and means of exiting the On-The-Go and Embedded Host Test Functions is specified in the 'On-The-Go and Embedded Host Supplement to the USB Revision 2.0 Specification'.

If *wLength* is non-zero, then the behavior of the device is not specified.

If an endpoint or interface is specified that does not exist, then the device responds with a Request Error.

Default state: A device must be able to accept a SetFeature(TEST_MODE, TEST_SELECTOR) request when in the Default State. Device behavior for other SetFeature requests while the device is in the Default state is not specified.

Address state: If an interface or an endpoint other than endpoint zero is specified, then the device responds with a Request Error.

Configured state: This is a valid request when the device is in the Configured state.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

INTERFACES DE BUS UNIVERSEL EN SÉRIE POUR LES DONNÉES ET L'ALIMENTATION ÉLECTRIQUE –

Partie 1: Spécification du bus universel en série, révision 2.0

AVANT-PROPOS

1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI - entre autres activités - publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) La CEI ne fournit elle-même aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains domaines, l'accès à des marques de conformité CEI. La CEI n'est responsable d'aucun des services proposés par les organismes de certification indépendants.
6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de l'identification de l'un quelconque ou de la totalité de ces droits de propriété industrielle.

La Norme internationale CEI 62680-1 a été établie par le domaine technique 14: Interfaces et méthodes de mesure pour les équipements d'ordinateur personnel, du comité d'études 100 de la CEI: Systèmes et équippements audio, vidéo et services de données.

Le texte de la présente norme est issu des documents établis par l'USB Implementers Forum (USB-IF). La structure et les règles rédactionnelles utilisées dans la présente publication reflètent la pratique de l'organisation qui l'a soumise.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
100/2156/FDIS	100/2188/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Afin d'éviter toute corruption des Figures, les originaux de la version anglaise ont été inclus dans la version française de la présente norme. Toutefois, pour les besoins des utilisateurs de la version française, une traduction du texte anglais a été ajoutée sous forme d'un tableau sous chaque Figure. A noter que ces "Légendes" ne contiennent aucune information supplémentaire par rapport à l'original anglais.

Une liste de toutes les parties de la série des CEI 62680, publiées sous le titre général *Interfaces de bus universel en série pour les données et l'alimentation électrique* est disponible sur le site internet de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "http://webstore.iec.ch" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT - Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

INTRODUCTION

La série des CEI 62680 repose sur une série de spécifications qui ont été développées à l'origine par l'USB Implementers Forum (USB-IF). Ces spécifications ont été soumises à la CEI sous l'égide d'un accord particulier entre la CEI et l'USB IF.

L'USB Implementers Forum, Inc.(USB-IF) est un organisme à but non lucratif fondé par le groupe de sociétés qui a développé la spécification du bus universel en série. L'USB-IF a été créé pour fournir une plate-forme de soutien et de forum pour le progrès et l'adoption de la technologie du bus universel en série. Le forum facilite le développement de périphériques (appareils) USB compatibles et de haute qualité et promeut les avantages de la technologie USB et la qualité des produits qui ont été validés par des essais de conformité.

TOUTES LES SPÉCIFICATIONS USB VOUS SONT FOURNIES "TELLES QUELLES" SANS AUCUNE GARANTIE DE QUELQUE SORTE QUE CE SOIT, Y COMPRIS LES GARANTIES DE QUALITÉ MARCHANDE, DE NON-VIOLATION OU D'ADÉQUATION À UN USAGE PARTICULIER. L'USB IMPLEMENTERS FORUM ET LES AUTEURS DES SPÉCIFICATIONS USB DÉCLINENT TOUTE RESPONSABILITÉ, Y COMPRIS LA VIOLATION DES DROITS D'AUTEUR, CONCERNANT L'UTILISATION, LA MISE EN APPLICATION OU LES INFORMATIONS DE LA PRÉSENTE SPÉCIFICATION.

LA DISPOSITION DE SPÉCIFICATIONS USB NE SAURAIT VOUS ACCORDER UNE LICENCE, EXPLICITE OU IMPLICITE, PAR ESTOPPEL OU DE TOUTE AUTRE FAÇON, À DES DROITS DE PROPRIÉTE INTELLECTUELLE.

La conclusion des accords des adoptants de l'USB peut toutefois permettre à une société signataire de participer à un accord de licence réciproque et libre de droits pour les produits conformes. Pour plus d'informations, voir:
http://www.usb.org/developers/docs/
http://www.usb.org/developers/devclass_docs#approved

LA CEI NE PREND AUCUNE POSITION QUANT À SAVOIR S'IL VOUS EST RECOMMANDÉ DE CONCLURE TOUT ACCORD DES ADOPTANTS DE L'USB OU DE PARTICIPER À L'USB IMPLEMENTERS FORUM."

La présente série traite des interfaces de bus universel en série pour les données et l'alimentation électrique et est composée des parties suivantes:

CEI 62680-1, Interfaces de bus universel en série pour les données et l'alimentation électrique – Partie 1: Spécification du bus universel en série, révision 2.0

CEI 62680-2, Interfaces de bus universel en série pour les données et l'alimentation électrique – Partie 2: Bus universel en série – Spécification des câbles et des connecteurs Micro-USB, révision 1.01

CEI 62680-3, Interfaces de bus universel en série pour les données et l'alimentation électrique – Partie 3: spécification de chargement des batteries USB, révision 1.2

CEI 62680-4, Interfaces de bus universel en série pour les données et l'alimentation électrique – Partie 4: document des classes des câbles et des connecteurs de bus universel en série, rév. 2.0

La présente partie de la CEI 62680 est constituée de plusieurs parties distinctes:

- le corps du texte, qui correspond à la spécification initiale et l'ensemble des ECN et des errata développés par l'USB-IF;

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

SOMMAIRE

1 Chapitre 1 Introduction 782
1.1 Motivation 782
1.2 Objectif de la spécification 783
1.3 Domaine d'application du document 783
1.4 Conformité des produits à l'USB 783
1.5 Organisation du document 783
2 Chapitre 2 Termes et abréviations 784
3 Chapitre 3 Contexte 790
3.1 Objectifs du bus universel en série 790
3.2 Taxinomie de l'espace d'application 790
3.3 Liste des caractéristiques 791
4 Chapitre 4 Vue d'ensemble architecturale 794
4.1 Description du système USB 794
4.1.1 Topologie en bus 794
4.2 Interface physique 796
4.2.1 Domaine électrique 796
4.2.2 Mécanique 796
4.3 Alimentation 797
4.3.1 Distribution de l'alimentation 797
4.3.2 Gestion de l'énergie 797
4.4 Protocole de bus 797
4.5 Robustesse 798
4.5.1 Détection des erreurs 798
4.5.2 Gestion des erreurs 798
4.6 Configuration du système 798
4.6.1 Connexion des appareils USB 798
4.6.2 Déconnexion des appareils USB 799
4.6.3 Enumération de bus 799
4.7 Types de flux de données 799
4.7.1 Transferts de contrôle 800
4.7.2 Transferts en bloc 800
4.7.3 Transferts d'interruption 800
4.7.4 Transferts isochrones 800
4.7.5 Allocation de bande passante USB 800
4.8 Appareils USB 801
4.8.1 Caractérisation de l'appareil 801
4.8.2 Description de l'appareil 801
4.9 Hôte USB: matériel et logiciel 804
4.10 Extensions architecturales 804
5 Chapitre 5 Modèle de flux de données USB 805
5.1 Points de vue du concepteur 805
5.2 Topologie en bus 807
5.2.1 Hôte USB 807
5.2.2 Appareils USB 808
5.2.3 Topologie de bus physique 809

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.2.4 Topologie de bus logique 812
5.2.5 Relation logiciel client-fonction 812
5.3 Flux de communication USB 813
5.3.1 Points d'extrémité de l'appareil 815
5.3.2 Canalisations 816
5.3.3 Trames et microtrames 819
5.4 Types de transfert 819
5.4.1 Exemples de calcul de tableau 820
5.5 Transferts de contrôle 821
5.5.1 Format de données des transferts de contrôle 822
5.5.2 Direction du transfert de contrôle 822
5.5.3 Contraintes liées à la taille des paquets des transferts de contrôle 822
5.5.4 Contraintes liées à l'accès au bus des transferts de contrôle 823
5.5.5 Séquences de données des transferts de contrôle 826
5.6 Transferts isochrones 827
5.6.1 Format de données des transferts isochrones 827
5.6.2 Direction des transferts isochrones 827
5.6.3 Contraintes liées à la taille des paquets des transferts isochrones 827
5.6.4 Contraintes liées à l'accès au bus des transferts isochrones 829
5.6.5 Séquences de données des transferts isochrones 830
5.7 Transferts d'interruption 830
5.7.1 Format de données des transferts d'interruption 830
5.7.2 Direction des transferts d'interruption 831
5.7.3 Contraintes liées à la taille des paquets des transferts d'interruption 831
5.7.4 Contraintes liées à l'accès au bus des transferts d'interruption 832
5.7.5 Séquences de données des transferts d'interruption 834
5.8 Transferts en bloc 834
5.8.1 Format de données des transferts en bloc 835
5.8.2 Direction des transferts en bloc 835
5.8.3 Contraintes liées à la taille des paquets des transferts en bloc 835
5.8.4 Contraintes liées à l'accès au bus des transferts en bloc 836
5.8.5 Séquences de données des transferts en bloc 837
5.9 Points d'extrémité à haute vitesse et à large bande passante 837
5.9.1 Points d'extrémité d'interruption à large bande passante 838
5.9.2 Points d'extrémité isochrones à large bande passante 839
5.10 Transactions de scission 840
5.11 Accès au bus des transferts 841
5.11.1 Gestion du transfert 841
5.11.2 Suivi des transactions 844
5.11.3 Calcul des temps de transaction de bus 847
5.11.4 Calcul des tailles de mémoire tampon dans les fonctions et le logiciel 849
5.11.5 Réclamation de bande passante de bus 849
5.12 Considérations spéciales pour les transferts isochrones 849
5.12.1 Exemple d'application isochrone non-USB 851
5.12.2 Modèle d'horloge USB 854
5.12.3 Synchronisation d'horloge 856
5.12.4 Appareils isochrones 857

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.12.5 Mise en mémoire tampon préalable des données 866
5.12.6 Suivi SOF 867
5.12.7 Gestion des erreurs 867
5.12.8 Mise en mémoire tampon pour la mise en correspondance du débit 868
6 Chapitre 6 Domaine mécanique 871
6.1 Vue d'ensemble sur l'architecture 871
6.2 Protocole de connecteur claveté 871
6.3 Câble 873
6.4 Ensemble de câbles 873
6.4.1 Ensembles de câbles amovibles normalisés 873
6.4.2 Ensembles de câbles captifs à haute/vitesse intégrale 875
6.4.3 Ensembles de câbles captifs à faible vitesse 878
6.4.4 Ensembles de câbles interdits 880
6.5 Configuration mécanique des connecteurs et exigences en termes de matériel 880
6.5.1 Emplacement de l'icône USB 881
6.5.2 Données de terminaison des connecteurs USB 882
6.5.3 Réceptacles des séries "A" et "B" 882
6.5.4 Fiches des séries "A" et "B" 887
6.6 Configuration mécanique des câbles et exigences en termes de matériel 892
6.6.1 Description 892
6.6.2 Construction 893
6.6.3 Caractéristiques électriques 895
6.6.4 Caractéristiques environnementales des câbles 895
6.6.5 Référence 895
6.7 Normes de conformité électrique, mécanique et environnementale 896
6.7.1 Documents applicables 901
6.8 Mise à la terre USB 901
6.9 Schémas de référence PCB 901
7 Chapitre 7 Domaine électrique 908
7.1 Signalisation 908
7.1.1 Caractéristiques du Pilote USB 912
7.1.2 Montée et chute du signal de données, diagrammes en œil 919
7.1.3 Dérivation de câble 932
7.1.4 Caractéristiques des récepteurs 932
7.1.5 Identification de la vitesse de l'appareil 933
7.1.6 Caractéristiques d'entrée 935
7.1.7 Niveaux de signalisation 938
7.1.8 Codage/décodage des données 952
7.1.9 Insertion de bits 952
7.1.10 Modèle Sync 955
7.1.11 Débit de signalisation des données 955
7.1.12 Intervalle de trame 956
7.1.13 Signalisation de la source de données 956
7.1.14 Synchronisations de signalisation d'un hub 959
7.1.15 Gigue des données du récepteur 961
7.1.16 Délai du câble 963

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.1.17 Atténuation de câble 964
7.1.18 Temps d'inversion du bus et délai interpaquet 965
7.1.19 Délai maximum du signal de bout en bout 966
7.1.20 Prise en charge du mode d'essai 967
7.2 Distribution de l'alimentation 968
7.2.1 Classes de appareils 968
7.2.2 Budget de chute de tension 974
7.2.3 Régulation de puissance au cours d'une interruption/reprise 974
7.2.4 Connexion et déconnexion dynamiques 975
7.3 Couche physique 977
7.3.1 Exigences réglementaires 977
7.3.2 Synchronisation du bus/caractéristiques électriques 977
7.3.3 Formes d'ondes de synchronisation 987
8 Chapitre 8 Couche de protocole 992
8.1 Ordonnancement des octets/bits 992
8.2 Champ SYNC 992
8.3 Formats des champs de paquets 992
8.3.1 Champ de l'identificateur de paquets 992
8.3.2 Champs d'adresse 994
8.3.3 Champ du nombre de trames 994
8.3.4 Champs de données 995
8.3.5 Contrôles de redondance cyclique 995
8.4 Formats du paquet 996
8.4.1 Paquets de jetons 996
8.4.2 Paquet de jetons spécial de transaction divisée 997
8.4.3 Paquets de début de trame 1002
8.4.4 Paquets de données 1003
8.4.5 Paquets d'établissement de liaison 1004
8.4.6 Réponses d'établissement de liaison 1005
8.5 Séquences des paquets de transactions 1006
8.5.1 Limitation de NAK par un contrôle de débit Ping 1015
8.5.2 Transactions en masse 1020
8.5.3 Transferts de contrôle 1026
8.5.4 Transactions d'interruption 1029
8.5.5 Transactions isochrones 1030
8.6 Synchronisation et relance de basculement des données 1035
8.6.1 Initialisation par jeton SETUP 1036
8.6.2 Transactions de données réussies 1036
8.6.3 Données corrompues ou non acceptées 1037
8.6.4 Etablissement de liaison ACK corrompu 1037
8.6.5 Transactions à basse vitesse 1038
8.7 Détection et récupération des erreurs 1040
8.7.1 Catégories de taux d'erreur de paquets 1040
8.7.2 Temps d'exécution de bus 1040
8.7.3 Fin de paquet incorrecte 1041
8.7.4 Récupération suite à un murmure et une perte d'activité 1042
9 Chapitre 9 Cadre d'appareils 1043

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9.1 Etats des appareils USB 1043
9.1.1 Etats des appareils visibles 1043
9.1.2 Enumération de bus 1047
9.2 Opérations génériques de appareils USB 1048
9.2.1 Connexion et déconnexion dynamiques 1048
9.2.2 Attribution d'adresses 1048
9.2.3 Configuration 1048
9.2.4 Transfert de données 1049
9.2.5 Gestion de l'énergie 1049
9.2.6 Traitement des requêtes 1049
9.2.7 Erreur de requête 1051
9.3 Requêtes de appareils USB 1052
9.3.1 Type bmRequest 1052
9.3.2 bRequest 1052
9.3.3 wValue 1052
9.3.4 wIndex 1053
9.3.5 wLength 1053
9.4 Requêtes normalisées de l'appareil 1053
9.4.1 ClearFeature 1055
9.4.2 Get Configuration 1056
9.4.3 Get Descriptor 1056
9.4.4 Get Interface 1057
9.4.5 Get Status 1057
9.4.6 Set Address 1059
9.4.7 Set Configuration 1060
9.4.8 Set Descriptor 1060
9.4.9 Set Feature 1061
9.4.10 Set Interface 1062
9.4.11 Synch Frame 1062
9.5 Descripteurs 1063
9.6 Définitions de descripteur USB normalisé 1063
9.6.1 Appareil 1064
9.6.2 Device_Qualifier 1066
9.6.3 Configuration 1066
9.6.4 Other_Speed_Configuration 1068
9.6.5 Interface 1068
9.6.6 Point d'extrémité 1070
9.6.7 Chaîne 1074
9.7 Définitions de classe d'appareil 1075
9.7.1 Descripteurs 1075
9.7.2 Interface(s) et utilisation des points d'extrémité 1075
9.7.3 Requêtes 1075
10 Chapitre 10 Hôte USB: Matériel et logiciel 1076
10.1 Présentation de l'hôte USB 1076
10.1.1 Présentation 1076
10.1.2 Mécanismes de contrôle 1079
10.1.3 Flux de données 1080

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.1.4 Collecte des statistiques d'état et d'activité 1080
10.1.5 Considérations vis-à-vis de l'interface électrique 1080
10.2 Exigences vis-à-vis du contrôleur hôte 1080
10.2.1 Traitement des états 1081
10.2.2 Sérialiseur/désérialiseur 1081
10.2.3 Génération de trames et de microtrames 1081
10.2.4 Traitement des données 1082
10.2.5 Moteur de protocole 1083
10.2.6 Traitement des erreurs de transmission 1083
10.2.7 Réveil à distance 1083
10.2.8 Hub racine 1083
10.2.9 Interface du système hôte 1084
10.3 Présentation des mécanismes logiciels 1084
10.3.1 Configuration des appareils 1084
10.3.2 Gestion des ressources 1086
10.3.3 Transferts de données 1087
10.3.4 Définitions de données communes 1088
10.4 Pilote de contrôleur hôte 1088
10.5 Pilote de bus série universel 1089
10.5.1 Présentation de l'USBD 1089
10.5.2 Exigences vis-à-vis des mécanismes de commande USBD 1091
10.5.3 Mécanismes du canal USBD 1093
10.5.4 Gestion de l'USB via les mécanismes USBD 1095
10.5.5 Transfert d'une commande de pré-amorçage USB au système d'exploitation 1097
10.6 Guides d'environnement du système d'exploitation 1097
11 Chapitre 11 Spécification relative au hub 1099
11.1 Présentation 1099
11.1.1 Architecture d'un hub 1099
11.1.2 Connectivité du hub 1100
11.2 Minuteur de trame/microtrame de hub 1102
11.2.1 Plage de minuteur de microtrame à grande vitesse 1102
11.2.2 Plage de minuteur de trame à vitesse intégrale 1103
11.2.3 Synchronisation du minuteur de trame/microtrame 1103
11.2.4 Gigue de microtrame liée à la gigue de trame 1106
11.2.5 Points de synchronisation EOF1 et EOF2 1106
11.3 Comportement de l'hôte en fin de trame 1109
11.3.1 Dernier paquet hôte à vitesse intégrale/basse vitesse 1110
11.3.2 Annulation du paquet à vitesse intégrale/basse vitesse 1110
11.3.3 Prédiction d'exécution de transaction à vitesse intégrale/basse vitesse 1110
11.4 Port interne 1111
11.4.1 Inactif 1112
11.4.2 Retard Suspension 1112
11.4.3 Suspension Totale (Fsus) 1112
11.4.4 Générer Reprise (GResume) 1112
11.5 Ports orientés en aval 1113

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.5.1 Descriptions d'état du port orienté en aval 1115
11.5.2 Minuteur de détection de déconnexion 1119
11.5.3 Indicateur de port 1120
11.6 Port orienté en amont 1122
11.6.1 Vitesse intégrale 1122
11.6.2 Grande vitesse 1122
11.6.3 Récepteur 1122
11.6.4 Emetteur 1125
11.7 Répéteur de hub 1127
11.7.1 Connectivité de paquet à grande vitesse 1128
11.7.2 Diagramme d'états du répéteur de hub 1130
11.7.3 Attente de début de paquet du port en amont (WFSOPFU) 1132
11.7.4 Attente de fin de paquet du port en amont (WFEOPFU) 1132
11.7.5 Attente de début de paquet (WFSOP) 1132
11.7.6 Attente de fin de paquet (WFEOP) 1132
11.8 Evaluation d'état de bus 1133
11.8.1 Erreur de port 1133
11.8.2 Détection de vitesse 1133
11.8.3 Collision 1134
11.8.4 Comportement de port à basse vitesse 1134
11.9 Suspension et Reprise 1135
11.10 Comportement de réinitialisation du hub 1138
11.11 Contrôle d'alimentation de port de hub 1139
11.11.1 Associations multiples 1139
11.12 Contrôleur de hub 1140
11.12.1 Organisation des points d'extrémité 1140
11.12.2 Architecture et utilisation des informations sur le hub 1141
11.12.3 Traitement d'informations de changement de port 1141
11.12.4 Topogramme binaire de changement de hub et de port 1143
11.12.5 Reporting et rétablissement de surintensité 1144
11.12.6 Traitement des énumérations 1145
11.13 Configuration de hub 1145
11.14 Traducteur de transactions 1146
11.14.1 Présentation 1146
11.14.2 Programmation du traducteur de transactions 1149
11.15 Informations sur la notation des transactions fractionnées 1152
11.16 Diagrammes d'états communs aux transactions fractionnées 1156
11.16.1 Diagramme d'états du contrôleur hôte 1158
11.16.2 Diagramme d'états du traducteur de transactions 1162
11.17 Présentation de la traduction de transactions en masse/de contrôle 1170
11.17.1 Séquences de transactions fractionnées en masse/de contrôle 1171
11.17.2 Diagrammes d'état de transactions fractionnées en masse/de contrôle 1179
11.17.3 Séquençage en masse/de contrôle 1187
11.17.4 Exigences vis-à-vis de la mise en tampon en masse/de contrôle 1187
11.17.5 Autres détails en masse/de contrôle 1188

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.18 Mise en pipeline des transactions fractionnées périodiques et gestion des tampons 1188
11.18.1 Meilleur budget à vitesse intégrale 1188
11.18.2 Pipeline de microtrame TT 1189
11.18.3 Génération de trames à vitesse intégrale 1190
11.18.4 Exigences vis-à-vis de la programmation de transactions programmées de l'hôte 1191
11.18.5 Génération de réponse TT 1194
11.18.6 Exigences vis-à-vis du traitement des transactions périodiques TT 1196
11.18.7 Suivi des transactions TT 1197
11.18.8 Recherche d'état de transaction complete-split TT 1198
11.19 Espace tampon TT approximatif requis 1200
11.20 Présentation de la traduction de transactions d'interruption 1200
11.20.1 Séquences de transactions fractionnées d'interruption 1200
11.20.2 Diagrammes d'états des transactions fractionnées d'interruption 1205
11.20.3 Séquençage OUT d'interruption 1211
11.20.4 Séquençage IN d'interruption 1212
11.21 Présentation de la traduction de transactions isochrones 1214
11.21.1 Séquences de transactions fractionnées isochrones 1215
11.21.2 Diagrammes d'états de transaction fractionnée isochrone 1218
11.21.3 Séquençage OUT isochrone 1223
11.21.4 Séquençage IN isochrone 1224
11.22 Traitement des erreurs TT 1225
11.22.1 Perte de synchronisation TT avec SOF HS 1225
11.22.2 Exigences vis-à-vis de la synchronisation du minuteur de trame et de microtrame TT 1226
11.23 Descripteurs 1228
11.23.1 Descripteurs normalisés pour classe de hub 1228
11.23.2 Descripteurs spécifiques à la classe 1236
11.24 Demandes 1238
11.24.1 Demandes normalisées 1238
11.24.2 Demandes spécifiques à la classe 1238
Annexe A Exemples de transactions 1255
Annexe B Exemples de déclarations pour diagrammes d'états 1410
Annexe C Diagrammes d'états de protocole de réinitialisation 1419
Index 1429

Note: L'ensemble des notices de modification technique (Engineering Change Notice, ECN) et des documents d'errata en date du 1er septembre 2012 qui appartiennent à cette spécification principale suivent la dernière page de la spécification à partir de la page 1507.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Bus universel en série
Spécification

Compaq
Hewlett-Packard
Intel
Lucent
Microsoft
NEC
Philips

Révision 2.0
27 avril 2000

Domaine d'application de la présente révision

La révision 2.0 de la spécification est destinée à la conception des produits. Chaque tentative a été réalisée dans le but de garantir une spécification cohérente et réalisable. Il convient que les mises en application assurent la conformité à la présente révision.

Historique des révisions

Révision	Date d'édition	Commentaires
0.7	11 novembre 1994	Remplace la révision 0.6e.
0.8	30 décembre 1994	Révisions apportées aux Chapitres 3-8, 10 et 11. Ajout des annexes.
0.9	13 avril 1995	Révisions apportées à tous les chapitres.
0.99	25 août 1995	Révisions apportées à tous les chapitres.
1.0 FDR	13 novembre 1995	Révisions apportées aux Chapitres 1, 2, 5-11.
1.0	15 janvier 1996	Modifications des Chapitres 5, 6, 7, 8, 9, 10 et 11 par souci de cohérence.
1.1	23 septembre 1998	Mise à jour de tous les chapitres pour corriger les problèmes identifiés.
2.0 (version 0.79)	5 octobre 1999	Révisions apportées aux Chapitres 5, 7, 8, 9, 11 pour ajouter la haute vitesse.
2.0 (version 0.9)	21 décembre 1999	Révisions apportées à tous les chapitres pour ajouter la haute vitesse.
2.0	27 avril 2000	Révisions pour le mode haute vitesse.

Spécification du bus universel en série
Copyright © 2000, Compaq Computer Corporation,
Hewlett-Packard Company,
Intel Corporation,
Lucent Technologies Inc,
Microsoft Corporation,
NEC Corporation,
Koninklijke Philips Electronics N.V.
Tous droits réservés.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

DENI DE RESPONSABILITÉ CONCERNANT LA PROPRIÉTE INTELLECTUELLE

LA PRÉSENTE SPÉCIFICATION VOUS EST FOURNIE "TELLE QUELLE" SANS AUCUNE GARANTIE DE QUELQUE SORTE QUE CE SOIT, Y COMPRIS LES GARANTIES DE QUALITÉ MARCHANDE, DE NON-VIOLATION OU D'ADÉQUATION À UN USAGE PARTICULIER. LES AUTEURS DE LA PRÉSENTE SPÉCIFICATION USB DECLINENT TOUTE RESPONSABILITÉ, Y COMPRIS LA VIOLATION DES DROITS D'AUTEUR, CONCERNANT L'UTILISATION OU LA MISE EN APPLICATION DES INFORMATIONS DE LA PRÉSENTE SPÉCIFICATION. LA DISPOSITION DE LA PRÉSENTE SPÉCIFICATION NE SAURAIT VOUS ACCORDER UNE LICENCE, EXPLICITE OU IMPLICITE, PAR ESTOPPEL OU DE TOUTE AUTRE FAÇON, À DES DROITS DE PROPRIÉTÉ INTELLECTUELLE.

Tous les noms de produit sont les marques commerciales, les marques déposées ou les marques de service de leurs détenteurs respectifs.

Envoyer les commentaires par courrier électronique à echsup@usb.org

Pour les informations sur le secteur d'activité, voir la page web de l'USB Implementers Forum à l'adresse http://www.usb.org

Reconnaissance de la contribution technique à la norme USB 2.0

Les auteurs de la présente spécification souhaiteraient remercier les personnes suivantes pour leur participation aux groupes de travail techniques du groupe de promoteurs de la norme USB 2.0. Ils souhaiteraient également remercier les personnes des sociétés promotrices de la norme USB 2.0 et de tout le secteur d'activité qui ont contribué au développement de la présente spécification.

Groupe de travail sur le hub

John Garney	Intel Corporation (président/éditeur)
Ken Stufflebeam	Compaq Computer Corporation
David Wooten	Compaq Computer Corporation
Matt Nieberger	Hewlett-Packard Company
John Howard	Intel Corporation
Venkat Iyer	Intel Corporation
Steve McGowan	Intel Corporation
Geert Knapen	Royal Philips Electronics
Zong Liang Wu	Royal Philips Electronics
Jim Clee	Lucent Technologies Inc
Jim Guziak	Lucent Technologies Inc
Dave Thompson	Lucent Technologies Inc
John Fuller	Microsoft Corporation
Nathan Sherman	Microsoft Corporation
Mark Williams	Microsoft Corporation
Nobuo Furuya	NEC Corporation
Toshimi Sakurai	NEC Corporation
Moto Sato	NEC Corporation
Katsuya Suzuki	NEC Corporation

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Groupe de travail sur la partie électrique

Jon Lueker	Intel Corporation (président/éditeur)
David Wooten	Compaq Computer Corporation
Matt Nieberger	Hewlett-Packard Company
Larry Taugher	Hewlett-Packard Company
Venkat Iyer	Intel Corporation
Steve McGowan	Intel Corporation
Mike Pennell	Intel Corporation
Todd West	Intel Corporation
Gerrit den Besten	Royal Philips Electronics
Marq Kole	Royal Philips Electronics
Zong Liang Wu	Royal Philips Electronics
Jim Clee	Lucent Technologies Inc
Jim Guziak	Lucent Technologies Inc
Par Parikh	Lucent Technologies Inc
Dave Thompson	Lucent Technologies Inc
Ed Giaimo	Microsoft Corporation
Mark Williams	Microsoft Corporation
Toshihiko Ohtani	NEC Corporation
Kugao Ouchi	NEC Corporation
Katsuya Suzuki	NEC Corporation
Toshio Tasaki	NEC Corporation

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

1 Chapitre 1 Introduction

1.1 Motivation

La motivation première du bus universel en série (USB) provient de trois considérations corrélées:

- Connexion du PC au téléphone

Il est tout à fait clair que la fusion du calcul informatisé et de la communication représente la base de la génération suivante d'applications de productivité. Le déplacement des types de données orientés machine et humain d'un emplacement ou d'un environnement à un autre dépend d'une connectivité omniprésente et bon marché. Malheureusement, les domaines du calcul informatisé et de la communication ont évolué de manière indépendante. L'USB fournit un lien omniprésent qui peut être utilisé pour de nombreuses interconnexions ordinateur avec téléphone.

- Facilité d'utilisation

Le manque de flexibilité de la reconfiguration de l'ordinateur est reconnu comme étant le talon d'Achille pour son déploiement futur. La combinaison des interfaces graphiques conviviales et des mécanismes matériels et logiciels associés à la nouvelle génération d'architectures de bus ont rendu les ordinateurs moins conflictuels et plus faciles à reconfigurer. Cependant, du point de vue de l'utilisateur final, les interfaces E/S des ordinateurs, tels que les ports série/parallèles, les interfaces de clavier/souris/joystick, etc., ne disposent pas des attributs du Plug and Play.

- Extension des ports

L'ajout de périphériques externes est toujours restreint par la disponibilité des ports. L'absence d'un bus périphérique bidirectionnel, de faible coût et de faible à moyenne vitesse a freiné la prolifération créative de périphériques tels que les adaptateurs de téléphone/télécopie/modem, les répondeurs téléphoniques, les numériseurs, les assistants numériques personnels, les claviers, les souris, etc. Les interconnexions existantes sont optimisées pour un ou deux produits ponctuels. Pour chaque nouvelle fonction ou fonctionnalité ajoutée à l'ordinateur, une nouvelle interface a été définie pour répondre à ce besoin.

La motivation la plus récente pour la norme USB 2.0 découle du fait que les ordinateurs ont des performances de plus en plus élevées et peuvent traiter de larges quantités de données. Parallèlement, les périphériques des ordinateurs ont apporté plus de performance et de fonctionnalité. Les applications destinées aux utilisateurs, telles que l'imagerie numérique, demandent une connexion à hautes performances entre l'ordinateur et ces périphériques de plus en plus sophistiqués. La norme USB 2.0 répond à ce besoin en ajoutant un troisième débit de transfert de 480 Mo/s à ceux de 12 Mo/s et 1,5 Mo/s définis à l'origine pour l'USB. La norme USB 2.0 est une évolution naturelle de l'USB qui fournit l'augmentation de bande passante souhaitée tout en préservant les motivations d'origine de l'USB et en maintenant la pleine compatibilité avec les périphériques existants.

Par conséquent, l'USB reste la réponse à la connectivité de l'architecture des ordinateurs. Il est rapide, bidirectionnel, isochrone, faible coût, avec une interface série dynamiquement raccordable qui est cohérente avec les exigences de la plateforme PC d'aujourd'hui et de demain.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

1.2 Objectif de la spécification

Le présent document définit un USB de norme industrielle. La spécification décrit les attributs du bus, la définition du protocole, les types de transaction, la gestion du bus et l'interface de programmation requise pour la conception et la création des systèmes et des périphériques conformes à cette norme.

L'objectif est de permettre aux appareils de différents vendeurs de fonctionner ensemble dans une architecture ouverte. La spécification se veut une amélioration de l'architecture des ordinateurs, couvrant les environnements portables, de bureau et domestiques. Il est prévu que la spécification accorde suffisamment de liberté aux OEM (fabricant d'équipement d'origine) de systèmes et aux développeurs de périphériques pour une polyvalence des produits et leur différenciation sur le marché sans avoir à prendre en charge des interfaces obsolètes ou à perdre en termes de compatibilité.

1.3 Domaine d'application du document

La spécification s'adresse en premier lieu aux développeurs de périphériques et aux OEM de systèmes, mais elle fournit des informations précieuses aux fournisseurs de systèmes d'exploitation de plateforme/BIOS/pilotes d'appareil, aux éditeurs de logiciels indépendants/fournisseurs indépendants de matériel d'adaptateurs et aux vendeurs de contrôleurs de plateforme/adaptateur. La présente spécification peut être utilisée pour développer de nouveaux produits et le logiciel associé.

1.4 Conformité des produits à l'USB

Les adoptants de la spécification de l'USB 2.0 ont signé l'accord des adoptants de l'USB 2.0, ce qui leur donne accès à une licence réciproque et libre de droits fournie par les promoteurs et d'autres adoptants à certaines propriétés intellectuelles contenues dans des produits conformes à la spécification USB 2.0. Les adoptants peuvent démontrer la conformité à la spécification à l'aide du programme d'essais défini par l'USB Implementers Forum. Les produits qui démontrent leur conformité à la spécification se voient accorder certains droits pour l'utilisation du logo de l'USB Implementers Forum selon la licence de logo.

1.5 Organisation du document

Les Chapitres 1 à 5 fournissent une vue d'ensemble à tous les lecteurs, tandis que les Chapitres 6 à 11 contiennent des informations techniques détaillées permettant de définir l'USB.

- Il convient que les responsables de la mise en application lisent plus particulièrement les Chapitres 5 à 11.
- Il convient que les responsables de la mise en application des contrôleurs hôtes USB lisent plus particulièrement les Chapitres 5 à 8, 10 et 11.
- Il convient que les responsables de la mise en application des pilotes d'appareil USB lisent plus particulièrement les Chapitres 5, 9 et 10.

Le présent document est complété et référencé par les spécifications des classes d'appareil de bus universel en série. Les spécifications des classes d'appareil existent pour une large variété d'appareils. Contacter l'USB Implementers Forum pour de plus amples détails.

Les lecteurs sont également priés de contacter les vendeurs de systèmes d'exploitation pour les liaisons des systèmes d'exploitation spécifiques à l'USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

2 Chapitre 2
Termes et abréviations

Le présent chapitre décrit et définit les termes et les abréviations utilisés dans la présente spécification.

ACK	Paquet poignée de mains indiquant un accusé de réception positif.
Appareil actif	Appareil qui est alimenté et qui ne se trouve pas dans l'état Interrompu.
Données asynchrones	Données transférées à intervalles irréguliers avec des exigences de latence moins rigoureuses.
RA asynchrone	Le débit de données entrant, Fs_i, et le débit de données sortant, Fs_o, du processus RA sont indépendants (c'est-à-dire, aucune horloge maître n'est partagée). Voir également l'adaptation de débit.
SRC asynchrone	Le débit d'échantillons entrant, Fs_i, et le débit d'échantillons sortant, Fs_o, du processus SRC sont indépendants (c'est-à-dire, aucune horloge maître n'est partagée). Voir également la conversion du débit d'échantillons.
Appareil audio	Appareil source ou destinataire de données analogiques échantillonnées.
AWG#	Mesure du diamètre d'un câble définie par la norme American Wire Gauge.
Babillage	Activité inattendue du bus qui persiste au-delà d'un point spécifié de la (micro)trame.
Bande passante	Quantité de données transmise par unité de temps, généralement en bits par seconde (b/s) ou en octets par secondes (B/s).
Gros-boutiste	Méthode de stockage des données qui enregistre l'octet de poids fort des valeurs à octets multiples à l'adresse mémoire la plus petite. Par exemple, un entier sur 16 bits stocké sous le format Gros-boutiste format enregistre l'octet de poids faible à l'adresse la plus grande et l'octet de poids fort à l'adresse la plus petite. Voir également petit-boutiste.
Bit	Unité d'information utilisée par les ordinateurs numériques. Représente le plus petit élément de mémoire adressable dans un ordinateur. Un bit exprime le choix entre deux possibilités et est généralement représenté par le chiffre un (1) ou le chiffre zéro (0) logique.
Insertion de bits	Insertion d'un bit "0" dans un flux de données pour déclencher une transition électrique dans les câbles électriques, permettant à une PPL de rester verrouillée.
b/s	Débit de transmission exprimé en bits par seconde.
B/s	Débit de transmission exprimé en octets par seconde.
Mémoire tampon	Stockage utilisé pour compenser toute différence des débits de données ou des moments de survenance des événements pendant la transmission des données d'un appareil à un autre.
Transfert en bloc	Un des quatre types de transfert USB. Les transferts en bloc représentent de grosses communications non périodiques et par salves qui sont utilisées pour un transfert pouvant utiliser toute bande passante disponible et également être retardé jusqu'à ce que la bande passante soit disponible. Voir également le type de transfert.
Enumération de bus	Détection et identification des appareils USB.
Octet	Elément de données dont la taille est égale à huit bits.
Fonctionnalité	Attributs d'un appareil USB qui sont gérés par l'hôte.
Caractéristiques	Qualités immuables d'un appareil USB; par exemple, la classe d'appareil est une caractéristique de l'appareil.
Client	Logiciel résidant sur l'hôte qui interagit avec le logiciel système USB pour organiser le transfert de données entre une fonction et l'hôte. Le client est souvent le fournisseur et le consommateur des données transférées.
Configuration du logiciel	Logiciel résidant sur l'hôte qui est responsable de la configuration d'un appareil USB. Cela peut être un configurateur de système ou un logiciel spécifique à l'appareil.
Point d'extrémité de contrôle	Paire de points d'extrémité d'appareil ayant le même numéro de point d'extrémité qui sont utilisés par un canal de contrôle. Les points d'extrémité de contrôle transfèrent les données dans les deux directions et, par conséquent, utilisent les deux directions de point d'extrémité d'une adresse d'appareil et d'une combinaison de numéros de point d'extrémité. Chaque point d'extrémité de contrôle consomme donc deux adresses de point d'extrémité.
Canal de contrôle	Identique à un canal de messages.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Transfert de contrôle	Un des quatre types de transfert USB. Les transferts de contrôle prennent en charge les communications de type configuration/commande/état entre le client et la fonction. Voir également le type de transfert.
CRC	Voir Contrôle de redondance cyclique.
IRC	Couplage téléphonie informatique.
Contrôle de redondance cyclique (CRC)	Contrôle réalisé sur les données visant à détecter si une erreur s'est produite pendant la transmission, la lecture ou l'écriture des données. Le résultat d'un CRC est généralement enregistré ou transmis avec les données contrôlées. Le résultat enregistré ou transmis est comparé au CRC calculé pour les données afin de déterminer si une erreur s'est produite.
Adresse par défaut	Adresse définie par la spécification USB et utilisée par un appareil USB lors de sa première mise en marche ou de sa réinitialisation. L'adresse par défaut est 00H.
Canal par défaut	Canal de messages créé par le logiciel système USB pour passer les informations de contrôle et d'état entre l'hôte et le point d'extrémité zéro d'un appareil USB.
Appareil	Entité logique ou physique qui réalise une fonction. L'entité actuellement décrite dépend du contexte de référence. Au niveau le plus bas, l'appareil peut faire référence à une seule composante matérielle, comme un appareil de mémoire. A un niveau plus élevé, il peut faire référence à une collection de composantes matérielles qui réalisent une fonction particulière, tel qu'un appareil d'interface USB. A un niveau encore plus élevé, l'appareil peut faire référence à la fonction réalisée par une entité connectée à l'USB; par exemple, un appareil de modem de données/télécopie. Les appareils peuvent être physiques, électriques, adressables et logiques. Lorsqu'il est utilisé comme une référence non spécifique, un appareil USB est soit un hub, soit une fonction.
Adresse d'un appareil	Valeur sur sept bits représentant l'adresse d'un appareil sur l'USB. L'adresse de l'appareil est l'adresse par défaut (00H) lorsque l'appareil USB est mis en marche pour la première fois ou réinitialisé. Une adresse unique est attribuée aux appareils par le logiciel système USB.
Point d'extrémité de l'appareil	Portion uniquement adressable d'un appareil USB qui est la source ou le destinataire des informations contenues dans un flux de communication entre l'hôte et l'appareil. Voir également l'adresse de point d'extrémité.
Ressources de l'appareil	Ressources fournies par les appareils USB, tels que l'espace de mémoire tampon et les points d'extrémité. Voir également les ressources d'hôte et les ressources de bus universel en série.
Logiciel de l'appareil	Logiciel responsable de l'utilisation d'un appareil USB. Cet appareil peut ou non être responsable de la configuration de l'appareil à utiliser.
Aval	Direction du flux de données à partir de l'hôte ou à distance de l'hôte. Un port en aval est le port du hub le plus éloigné de l'hôte du point de vue électrique qui génère le trafic de données en aval à partir du hub. Les ports en aval reçoivent des données de trafic en amont.
Pilote	Lorsqu'il est fait référence au matériel, un assembleur-désassembleur de paquets d'E/S qui dirige une charge externe. Lorsqu'il est fait référence au logiciel, un programme responsable de l'interfaçage vers un appareil matériel, c'est-à-dire, un pilote d'appareil.
DWORD	Mot double. Un élément de données dont la taille est égale à deux mots (c'est-à-dire, quatre octets ou 32 bits).
Branchement et débranchement dynamiques	Capacité à connecter et déconnecter des appareils lorsque l'hôte est en fonctionnement.
E^2PROM	Voir Mémoire morte effaçable électriquement et programmable.
EEPROM	Voir Mémoire morte effaçable électriquement et programmable.
Mémoire morte effaçable électriquement et programmable (EEPROM)	Technologie de stockage en mémoire non volatile et réinscriptible.
Utilisateur final	L'utilisateur d'un hôte.
Point d'extrémité	Voir le point d'extrémité de l'appareil.
Adresse de point d'extrémité	Combinaison d'un numéro de point d'extrémité et d'une direction de point d'extrémité sur un appareil USB. Chaque adresse de point d'extrémité prend en charge le transfert des données dans une direction.
Direction de point d'extrémité	Direction du transfert de données sur l'USB. La direction peut être soit IN, soit OUT. IN se réfère aux transferts vers l'hôte; OUT se réfère aux transferts à partir de l'hôte.
Numéro de point d'extrémité	Une valeur sur quatre bits entre les valeurs 0H et FH incluse qui est associée à un point d'extrémité sur un appareil USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Détecteur d'enveloppe	Circuit électronique situé dans un appareil USB qui surveille les canaux de données USB et détecte les caractéristiques de signal liées à une certaine tension.
EOF	Fin de (micro)trame
EOP	Fin de paquet.
Port externe	Voir Port.
Modèle d'œil	Représentation du signal USB qui fournit des niveaux minimum et maximum de tension, ainsi que la gigue de signal.
Fausse EOP	Evènement parasite, généralement causé par le bruit, qui est interprété par un récepteur de paquets comme étant une EOP.
Trame	Base de temps égale à 1 milliseconde établie sur des bus à vitesse intégrale/faible vitesse.
Modèle de trame	Séquence de trames qui présentent un modèle répétitif dans le nombre d'échantillons transmis par trame. Pour un transfert audio de 44,1 kHz, le modèle de trame pourrait être composé de neuf trames contenant 44 échantillons suivis d'une trame contenant 45 échantillons.
Fs	Voir débit d'échantillons.
Duplex intégral	Transmission de données d'ordinateur ayant lieu dans les deux directions de façon simultanée.
Vitesse intégrale	Fonctionnement USB à 12 Mb/s. Voir aussi faible vitesse et haute vitesse.
Fonction	Appareil USB qui fournit une fonctionnalité à l'hôte, telle qu'une connexion ISDN, un microphone numérique ou des haut-parleurs.
Paquet poignée de mains	Paquet qui accuse réception ou rejette une condition spécifique. Par exemple, voir ACK et NAK.
Point d'extrémité à grande bande passante	Point d'extrémité d'appareil à haute vitesse qui transfère plus de 1 024 octets et moins de 3 073 octets par microtrame.
Haute vitesse	Fonctionnement USB à 480 Mb/s. Voir également faible vitesse et vitesse intégrale.
Hôte	Système informatique hôte sur lequel le contrôleur hôte USB est installé. Il comprend la plateforme matérielle hôte (processeur, bus, etc.) et le système d'exploitation utilisé.
Contrôleur hôte	Interface USB de l'hôte.
Pilote de contrôleur hôte (HCD)	Couche logicielle USB qui abstractise le matériel du contrôleur hôte. Le pilote du contrôleur hôte fournit une SPI (interface périphérique série) pour l'interaction avec un contrôleur hôte. Le pilote du contrôleur hôte masque les spécificités de la mise en application matérielle du contrôleur hôte.
Ressources de l'hôte	Ressources fournies par l'hôte, telles que l'espace de mémoire tampon et les interruptions. Voir également les ressources de l'appareil et les ressources de bus universel en série.
Hub	Appareil USB qui fournit des connexions supplémentaires à l'USB.
Etage de hub	Un plus le nombre de liens USB sur un chemin de communication entre l'hôte et une fonction. Voir Figure 4-1.
Demande d'interruption (IRQ)	Signal matériel qui permet à un appareil d'attirer l'attention d'un hôte. L'hôte invoque généralement une routine d'interruption de service pour traiter la condition qui a entraîné de la requête.
Transfert d'interruption	Un des quatre types de transfert USB. Les caractéristiques du transfert d'interruption sont les suivantes: petite quantité de données, non périodique, basse fréquence et latence bornée. Les transferts d'interruption sont généralement utilisés pour traiter les besoins de service. Voir également le type de transfert.
Paquet de requêtes E/S	Requête identifiable par un client logiciel pour le déplacement des données entre lui-même (sur l'hôte) et un point d'extrémité d'un appareil dans une direction appropriée.
IRP	Voir Paquet de requêtes E/S.
IRQ	Voir Demande d'interruption.
Données isochrones	Flux de données dont la synchronisation dépend de son débit de livraison.
Appareil isochrone	Entité pourvue de points d'extrémité isochrones, telle que définie dans la spécification USB, qui est la source ou le destinataire de flux analogiques échantillonnés ou de flux de données synchrones.
Point d'extrémité destinataire isochrone	Point d'extrémité ayant la capacité de consommer un flux de données isochrone qui est envoyé par l'hôte.
Point d'extrémité source isochrone	Point d'extrémité ayant la capacité de produire un flux de données isochrone et de l'envoyer à l'hôte.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Transfert isochrone	Un des quatre types de transfert USB. Les transferts isochrones sont utilisés pour le travail avec des données isochrones. Les transferts isochrones fournissent une communication périodique et continue entre l'hôte et l'appareil. Voir également le type de transfert.
Gigue	Tendance à un manque de synchronisation causé par des changements mécaniques ou électriques. Plus précisément, le déphasage des impulsions numériques sur un moyen de transmission.
kb/s	Débit de transmission exprimé en kilobits par seconde.
kB/s	Débit de transmission exprimé en kilooctets par seconde.
Petit-boutiste	Méthode de stockage des données qui enregistre l'octet de poids faible des valeurs à octets multiples à l'adresse mémoire la plus petite. Par exemple, un entier sur 16 bits stocké sous le format petit-boutiste enregistre l'octet de poids faible à l'adresse la plus petite et l'octet de poids fort à l'adresse suivante. Voir également gros-boutiste.
LOA	Perte d'activité de bus caractérisée par un SOP sans une EOP correspondante.
Faible vitesse	Fonctionnement USB à 1,5 Mb/s. Voir aussi vitesse intégrale et haute vitesse.
LSb	Bit de poids faible.
LSB	Octet de poids faible.
Mb/s	Débit de transmission exprimé en mégabits par seconde.
MB/s	Débit de transmission exprimé en mégaoctets par seconde.
Canal de messages	Canal bidirectionnel qui transfère les données à l'aide d'un paradigme requête/données/état. Les données ont une structure imposée qui permet une identification et une communication fiables des requêtes.
Microtrame	Base de temps de 125 microsecondes établie sur des bus à haute vitesse.
MSb	Bit de poids fort.
MSB	Octet de poids fort.
NAK	Paquet poignée de mains indiquant un accusé de réception négatif.
Non-retour à zéro inversé (NRZI)	Méthode d'encodage des données série dont les uns et les zéros sont représentés par des tensions faibles et élevées opposées et alternatives pour lesquelles il existe une tension de non-retour à zéro (référence) entre les bits codés. Elimine le besoin des impulsions d'horloge.
NRZI	Voir non-retour à zéro inversé.
Objet	Logiciel hôte ou structure de données représentant une entité USB.
Paquet	Ensemble de données organisé dans un groupe de transmission. Les paquets contiennent généralement trois éléments: les informations de contrôle (par exemple, source, destination et longueur), les données à transférer et les bits de détection et de correction des erreurs.
Mémoire tampon de paquet	Mémoire tampon logique utilisée par un appareil USB pour envoyer ou recevoir un seul paquet. Elle détermine la taille maximum du paquet que l'appareil peut envoyer ou recevoir.
ID de paquet (PID)	Champ d'un paquet USB qui indique le type du paquet et, par extension, le format du paquet et le type de détection des erreurs appliqué au paquet.
Phase	Paquet de jetons, de données ou poignée de mains. Une transaction a trois phases.
Boucle numérique à verrouillage de phase (PLL)	Circuit qui joue le rôle d'un détecteur de phase afin de maintenir un oscillateur en phase avec une fréquence entrante.
Appareil physique	Appareil qui a une mise en application physique; par exemple, haut-parleurs, microphones et lecteurs CD.
PID	Voir ID de paquet.
Tuyau	Abstraction logique représentant l'association entre un point d'extrémité sur un appareil et le logiciel de l'hôte. Un canal dispose de plusieurs attributs; par exemple, un canal peut transférer des données sous forme de flux (canal de flux) ou de messages (canal de messages). Voir également canal de flux et canal de messages.
PLL	Voir Boucle à verrouillage de phase.
scrutation	Demande auprès de plusieurs appareils, un à la fois, s'ils ont des données à transmettre.
POR	Voir Réinitialisation à la mise sous tension.
Port	Point d'accès à ou en provenance d'un système ou d'un circuit. Pour l'USB, le point auquel un appareil USB est branché.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Réinitialisation à la mise sous tension (POR)	Restauration d'un appareil de stockage, d'un registre ou d'une mémoire vers un état prédéfini lors de la mise sous tension.
Débit de données programmable	Soit un débit de données fixe (points de terminaison sur une fréquence), un nombre limité de débits de données (32 kHz, 44,1 kHz, 48 kHz, ...) ou un débit de données programmable en continu. Les fonctionnalités exactes de programmation d'un point d'extrémité doivent être consignées dans le descripteur approprié de points d'extrémité spécifique à la classe.
Protocole	Ensemble spécifique de règles, procédures ou conventions relatives au format et à la synchronisation de transmission des données entre deux appareils.
RA	Voir l'adaptation de débit.
Adaptation du débit	Processus par lequel un flux de données entrant, échantillonné au niveau de Fs_i, est converti en flux de données sortant, échantillonné au niveau de Fs_o, avec une certaine perte de qualité déterminée par l'algorithme d'adaptation de débit. Des mécanismes de contrôle des erreurs sont requis pour le processus. Fs_i et Fs_o peuvent être différents et asynchrones. Fs_i est le débit de données entrant de la RA; Fs_o est le débit de données sortant de la RA.
Requête	Requête effectuée vers un appareil USB contenue dans la portion de données d'un paquet SETUP.
Retrait	Action d'achèvement du service d'un transfert et de notification au client logiciel approprié de l'achèvement.
Hub racine	Hub USB directement connecté au contrôleur hôte. Ce hub (étage 1) est connecté à l'hôte.
Port racine	Port en aval d'un hub racine.
Echantillon	La plus petite unité de données sur laquelle un point d'extrémité fonctionne; propriété d'un point d'extrémité.
Débit d'échantillons (Fs)	Nombre d'échantillons par seconde, exprimé en Hertz (Hz).
Conversion de débit d'échantillons (SRC)	Mise en application dédiée du processus RA pour l'utilisation sur les flux de données analogiques échantillonnés. Le mécanisme de contrôle des erreurs est remplacé par des techniques d'interpolation.
Service	Procédure fournie par une interface de programmation système (SPI).
Intervalle de service	Période entre des requêtes consécutives vers un point d'extrémité USB pour l'envoi ou la réception des données.
Gigue de service	Ecart de la fourniture de service par rapport à son temps de remise programmé.
Débit de service	Nombre de services vers un point d'extrémité donné par unité de temps.
SOF	Voir Début de trame.
SOP	Début de paquet.
IPS	Voir Interface de programmation système.
Transaction de scission	Type de transaction pris en charge par les contrôleurs hôtes et les hubs. Ce type de transaction permet le raccordement des appareils à vitesse intégrale ou faible vitesse à des hubs fonctionnant à haute vitesse.
SRC	Voir Conversion de débit d'échantillons.
O	Partie de la séquence composant un transfert de contrôle; les niveaux comprennent le niveau Configuration, le niveau Données et le niveau Etat.
Début de trame (SOF)	Première transaction de chaque (micro)trame. Un SOF permet aux points d'extrémité d'identifier le début de la (micro)trame et de synchroniser les horloges internes des points d'extrémité vers l'hôte.
Canal de flux	Canal qui transfère les données sous forme de flux d'échantillons sans aucune structure USB définie.
Type de synchronisation	Classification qui caractérise la fonctionnalité d'un point d'extrémité isochrone à se connecter à d'autres points d'extrémité isochrones.
RA synchrone	Le débit de données entrant, Fs_i, et le débit de données sortant, Fs_o, du processus RA dérivent de la même horloge maître. Il existe une relation fixe entre Fs_i et Fs_o.
SRC synchrone	Le débit d'échantillons entrant, Fs_i, et le débit d'échantillons sortant, Fs_o, du processus SRC dérivent de la même horloge maître. Il existe une relation fixe entre Fs_i et Fs_o.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Interface de programmation système (SPI)	Interface définie pour les services fournis au logiciel système.
TDM	Voir Multiplexage par répartition dans le temps.
TDR	Voir Réflectomètre de domaine temporel.
Terminaison	Composantes passives raccordées à l'extrémité des câbles pour éviter que les signaux soient réfléchis ou renvoyés en écho.
Multiplexage par répartition dans le temps (TDM)	Méthode de transmission simultanée de signaux multiples (données, voix et/ou vidéo) sur un moyen de communication en entrelaçant chaque partie d'un signal l'une après l'autre.
Réflectomètre de domaine temporel (TDR)	Instrument capable de mesurer les caractéristiques d'impédance des lignes de signaux USB.
Expiration	Détection de l'absence d'activité d'un bus pendant un intervalle prédéterminé.
Paquet de jetons	Type de paquet identifiant la transaction qui doit être effectuée sur le bus.
Transaction	Fourniture d'un service à un point d'extrémité; elle est composée d'un paquet de jetons, d'un paquet de données optionnel et d'un paquet poignée de mains optionnel. Des paquets spécifiques sont permis/requis en se basant sur le type de transaction.
Traducteur de transaction	Composante fonctionnelle d'un hub USB. Le traducteur de transaction répond à des transactions à haute vitesse spéciales et les convertit en transactions à vitesse intégrale/faible vitesse avec des appareils à vitesse intégrale/faible vitesse raccordés aux ports dirigés en aval.
Transfert	Une ou plusieurs transactions de bus pour déplacer les informations entre un client logiciel et sa fonction.
Type de transfert	Détermine les caractéristiques du flux de données entre un client logiciel et sa fonction. Quatre types de transfert normalisés sont définis: contrôle, interruption, en bloc et isochrone.
Temps de retournement	Durée pendant laquelle un appareil doit attendre avant de commencer à transmettre un paquet après la réception d'un paquet afin d'éviter les collisions sur l'USB. Ce temps repose sur les caractéristiques de longueur et de délai de propagation du câble et de l'emplacement de l'appareil transmetteur par rapport aux autres appareils sur l'USB.
Pilote de bus universel en série (USBD)	L'entité logicielle résidant sur l'hôte qui est responsable de la fourniture des services communs aux clients qui manipulent une ou plusieurs fonctions sur un ou plusieurs contrôleurs hôtes.
Ressources de bus universel en série	Ressources fournies par l'USB, telles que la bande passante et la puissance. Voir également Ressources de l'appareil et Ressources de l'hôte.
Amont	Direction du transfert de données vers l'hôte. Un port en amont est le port de l'appareil le plus proche de l'hôte du point de vue électrique qui génère le trafic de données en amont à partir du hub. Les ports en amont reçoivent des données de trafic en aval.
USBD	Voir Pilote de bus universel en série.
USB-IF	USB Implementers Forum, Inc. est une organisation sans but lucratif créée pour faciliter le développement des produits conformes à la norme USB et promouvoir la technologie.
Appareil virtuel	Appareil représenté par une couche d'interface logicielle. Un exemple d'appareil virtuel est un disque dur avec le pilote d'appareil associé et le logiciel client qui lui permet de reproduire un fichier audio .WAV.
Mot	Elément de données dont la taille est égale à deux octets (16 bits).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

3 Chapitre 3
Contexte

Le présent chapitre présente une brève description du contexte du bus universel en série (USB), y compris les objectifs de conception, les caractéristiques du bus et les technologies existantes.

3.1 Objectifs du bus universel en série

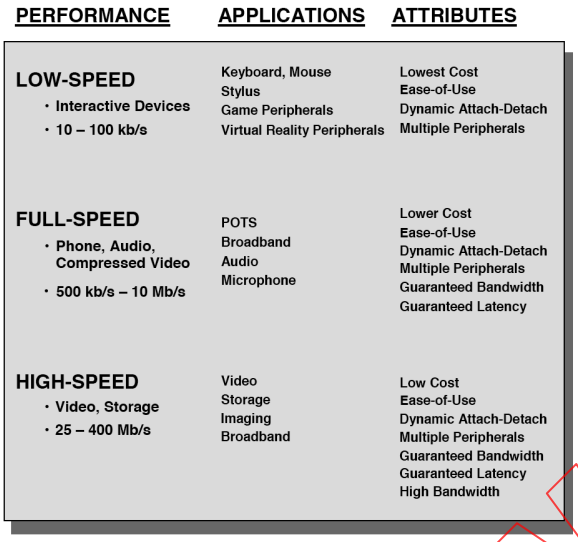
L'USB est conçu pour être une extension de norme industrielle de l'architecture des ordinateurs avec un accent sur les périphériques des ordinateurs qui activent les applications consommateur et métier. Les critères suivants ont été appliqués pour la définition de l'architecture de l'USB:

- Facilité d'utilisation de l'extension de périphérique d'ordinateur
- Solution de faible coût qui prend en charge les débits de transfert pouvant atteindre 480 Mb/s
- Prise en charge complète des données en temps réel pour la voix, l'audio et la vidéo
- Flexibilité du protocole pour les transferts de données isochrones en mode mixte et la messagerie asynchrone
- Intégration à la technologie des appareils de commodité
- Compréhension de plusieurs configurations d'ordinateur et de facteurs de forme
- Fourniture d'une interface normalisée capable d'une diffusion rapide dans le produit
- Activation de nouvelles classes d'appareil qui augmentent la fonctionnalité des ordinateurs
- Compatibilité ascendante complète de l'USB 2.0 pour les appareils construits sous les versions précédentes de la spécification

3.2 Taxinomie de l'espace d'application

La Figure 3-1 décrit la taxinomie pour la gamme des charges de travail de trafic de données qui peuvent être desservies sur un USB. Comme on peut le voir, un bus de 480 Mb/s comprend les gammes de données à haute vitesse, vitesse intégrale et faible vitesse. Généralement, des types de données à haute et vitesse intégrale peuvent être isochrones, tandis que les données à faible vitesse proviennent des appareils interactifs. L'USB est tout d'abord un bus d'ordinateur, mais il peut être facilement appliqué à d'autres appareils de calcul basés sur l'hôte. L'architecture logicielle permet une extension ultérieure de l'USB en fournissant la prise en charge de plusieurs contrôleurs hôtes USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
PERFORMANCE	PERFORMANCE
APPLICATIONS	APPLICATIONS
ATTRIBUTES	ATTRIBUTS
LOW-SPEED	FAIBLE VITESSE
Interactive devices	Appareils interactifs
Keyboard, Mouse	Clavier, souris
Stylus	Stylet
Game Peripherals	Périphériques de jeu
Virtual Reality Peripherals	Périphériques de réalité virtuelle
Lowest Cost	Coût le plus faible
Ease-of-Use	Facilité d'utilisation
Dynamic Attach-Detach	Connexion-déconnexion dynamique
Multiple Peripherals	Périphériques multiples
FULL-SPEED	VITESSE INTEGRALE
Phone, Audio	Téléphone, audio
Compressed Video	Vidéo comprimée
POTS	POTS
Broadband	Bande passante
Audio	Audio
Microphone	Microphone
Lower Cost	Coût plus faible
Guaranteed Bandwidth	Bande passante garantie
Guaranteed Latency	Latence garantie
HIGH-SPEED	HAUTE VITESSE
Video, Storage	Vidéo, stockage
Video	Vidéo
Storage	Stockage
Imaging	Imagerie
Low Cost	Faible coût
High Bandwidth	Bande passante élevée

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Figure 3-1. Taxinomie de l'espace d'application

3.3 Liste des caractéristiques

La spécification USB fournit une sélection d'attributs qui peuvent atteindre plusieurs points d'intégration de prix/performance et activer des fonctions permettant la différenciation au niveau du système et des composantes. Les caractéristiques sont classées en catégories selon les bénéfices suivants:

Facilité d'utilisation pour l'utilisateur final

- Modèle unique pour le câblage et les connecteurs
- Détails électriques isolés de l'utilisateur final (par exemple, terminaisons de bus)

- Périphériques d'auto-identification, mise en correspondance des fonctions au pilote et configuration automatiques
- Périphériques raccordables et reconfigurables de manière dynamique

Large gamme de charges de travail et d'applications

- Adapté aux largeurs de bande d'appareil de quelques kb/s à plusieurs centaines de Mb/s
- Prend en charge les types de transfert isochrones et asynchrones sur le même ensemble de câbles
- Prend en charge le fonctionnement concurrent de plusieurs appareils (connexions multiples)
- Prend en charge jusqu'à 127 appareils physiques
- Prend en charge le transfert de plusieurs flux de données et de messages entre l'hôte et les appareils
- Autorise les appareils composés (c'est-à-dire, les périphériques composés de plusieurs fonctions)
- Surcharge moins importante du protocole, donnant lieu à une forte utilisation du bus

Bande passante isochrone

- Bande passante garantie et faibles latences appropriées à la téléphonie, l'audio, la vidéo, etc.

Flexibilité

- Prend en charge une large gamme de tailles de paquet, ce qui permet d'avoir de nombreuses options de mise en mémoire tampon d'appareil
- Permet une large gamme de débits de données d'appareil en prenant en compte la taille de mémoire tampon et les latences des paquets
- Le contrôle de débit pour la gestion de la mémoire tampon est intégré au protocole

Robustesse

- Le mécanisme de gestion des erreurs/rétablissement en cas de panne est intégré au protocole
- L'insertion et la suppression dynamiques des appareils sont identifiées en temps réel perçu par les utilisateurs
- Prend en charge l'identification des appareils défectueux

Synergie avec l'industrie des ordinateurs

- Le protocole est simple à mettre en application et à intégrer
- Cohérent avec l'architecture prête à l'emploi des ordinateurs
- Tire profit des interfaces existantes de système d'exploitation

Mise en application de faible coût

- Sous-canal de faible coût à 1,5 Mb/s
- Optimisé pour l'intégration au matériel périphérique et hôte
- Adapté pour le développement des périphériques de faible coût
- Câbles et connecteurs de faible coût
- Utilise des technologies de commodité

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Procédure de mise à niveau

- Architecture évolutive pour la prise en charge de multiples contrôleurs hôtes USB dans un système

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

4 Chapitre 4
Vue d'ensemble architecturale

Le présent chapitre présente une vue d'ensemble sur l'architecture et les concepts clés du bus universel en série (USB). L'USB est un bus par câble qui prend en charge l'échange des données entre un ordinateur hôte et un large éventail de périphériques accessibles simultanément. Les périphériques connectés se partagent la bande passante de l'USB à travers un protocole de jetons programmé par l'hôte. Le bus permet aux périphériques d'être connectés, configurés, utilisés, et déconnectés pendant que l'hôte et les autres périphériques sont en fonctionnement.

Les chapitres suivants décrivent les différentes composantes de l'USB de manière détaillée.

4.1 Description du système USB

Un système USB est décrit selon trois zones de définition:

- Interconnexion USB
- Appareils USB
- Hôte USB

L'interconnexion USB est la méthode selon laquelle les appareils USB sont connectés et communiquent avec l'hôte. Cela comprend les éléments suivants:

- Topologie en bus: modèle de connexion entre les appareils USB et l'hôte.
- Relations inter-couches: en termes de pile de fonctionnalité, les tâches USB effectuées au niveau de chaque couche du système.
- Modèles de flux de données: manière selon laquelle les données se déplacent dans le système sur l'USB entre les producteurs et les consommateurs.
- Programme USB: l'USB fournit une interconnexion partagée. L'accès à l'interconnexion est programmé afin de prendre en charge les transferts de données isochrones et d'éliminer la surcharge de l'arbitrage.

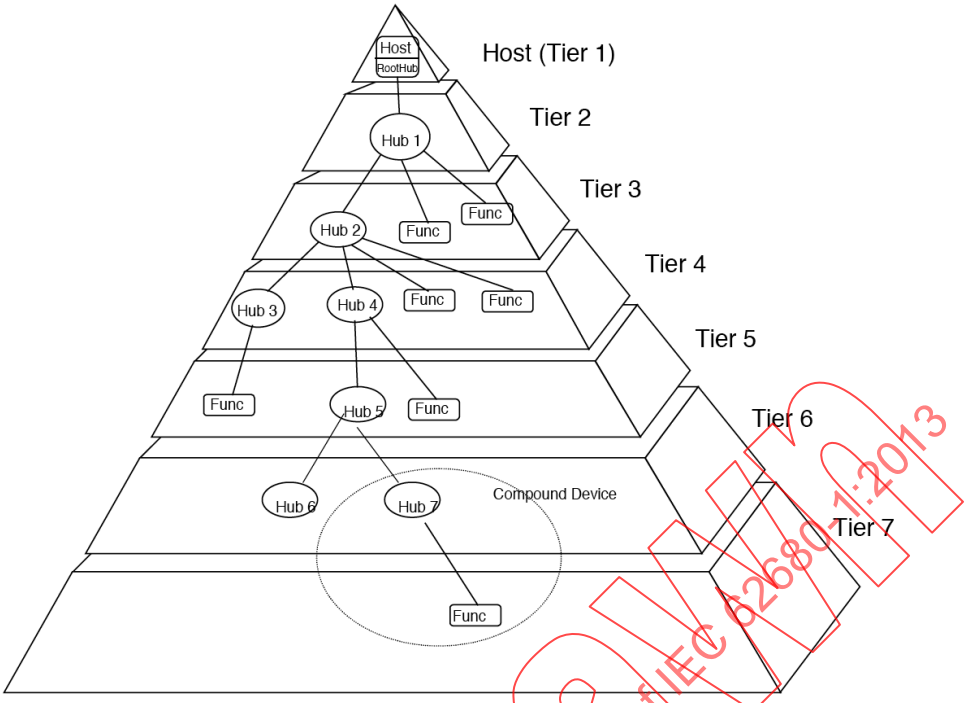
Les appareils USB et l'hôte USB sont décrits en détail dans les sections ultérieures.

4.1.1 Topologie en bus

L'USB connecte les appareils USB à l'hôte USB. L'interconnexion physique USB est une topologie en étoile à plusieurs niveaux. Un hub se trouve au centre de chaque étoile. Chaque segment de câble est une connexion point à point entre l'hôte et un hub ou une fonction, ou un hub connecté à un autre hub ou une fonction. La Figure 4-1 illustre la topologie de l'USB.

A cause des contraintes de synchronisation admises pour le hub et les durées de propagations des câbles, le nombre maximum de niveaux autorisés est égal à sept (y compris le niveau de base). Il faut noter que parmi les sept niveaux, cinq hubs, autres que la base, peuvent être pris en charge au maximum sur un chemin de communication entre l'hôte et un appareil. Un appareil composé (voir Figure 4-1) occupe deux niveaux; par conséquent, il ne peut pas être activé s'il est connecté au septième niveau. Seules les fonctions peuvent être activées au septième niveau.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Host (Tier 1)	Hôte (Niveau 1)
Tier 2	Niveau 2
Tier 3	Niveau 3
Tier 4	Niveau 4
Tier 5	Niveau 5
Tier 6	Niveau 6
Tier 7	Niveau 7
Host	Hôte
RootHub	Hub racine
Compound Device	Appareil composé

Figure 4-1. Topologie en bus

4.1.1.1 Hôte USB

Il ne peut y avoir qu'un seul hôte dans un système USB. L'interface USB du système informatique hôte est appelée le contrôleur hôte. Le contrôleur hôte peut être mis en application sous forme d'une combinaison de matériel, micrologiciel ou logiciel. Un hub racine est intégré au système hôte pour fournir un ou plusieurs points de connexion.

Des informations supplémentaires concernant l'hôte peuvent être trouvées à la Section 4.9 et au Chapitre 10.

4.1.1.2 Appareils USB

Les appareils USB se trouvent parmi les suivants:

- Hubs, qui fournissent des points de connexion supplémentaires à l'USB
- Fonctions, qui fournissent des fonctionnalités au système, telles qu'une connexion ISDN, un joystick numérique ou des haut-parleurs

Les appareils USB présentent une interface USB normalisée selon les termes suivants:

- Leur compréhension du protocole USB
- Leur réponse aux opérations USB normalisées, telles que la configuration et la réinitialisation
- Les informations descriptives de leurs fonctionnalités normalisées

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Des informations supplémentaires concernant les appareils USB peuvent être trouvées à la Section 4.8 et au Chapitre 9.

4.2 Interface physique

L'interface physique de l'USB est décrite dans les spécifications électriques (Chapitre 7) et mécaniques (Chapitre 6) du bus.

4.2.1 Domaine électrique

L'USB transfère les signaux et l'alimentation électrique sur un câble composé de quatre fils illustré à la Figure 4-2. La signalisation se produit sur deux fils d'un segment point à point.

Il existe trois débits de données:

- Le débit binaire de signalisation USB à haute vitesse est égal à 480 Mb/s.
- Le débit binaire de signalisation USB à vitesse intégrale est égal à 12 Mb/s.
- Un mode de signalisation à faible vitesse et à fonctionnalité limitée est également défini à 1,5 Mb/s.

Les contrôleurs hôtes et les hubs de l'USB 2.0 fournissent des fonctionnalités afin que des données puissent être transmises à haute et faible vitesse entre le contrôleur hôte et le hub, mais aussi une transmission à vitesse intégrale ou faible vitesse entre le hub et l'appareil. Cette fonctionnalité minimise l'impact que les appareils à vitesse intégrale et faible vitesse ont sur la bande passante disponible pour les appareils à haute vitesse.

Le mode à faible vitesse est défini pour prendre en charge un nombre limité d'appareils à faible bande passante, tels que les souris, parce qu'une utilisation plus générale dégraderait l'utilisation du bus.

L'horloge est transmise de façon codée avec les données différentielles. Le schéma d'encodage de l'horloge est le NRZI avec insertion de bits pour assurer des transitions adéquates. Un champ SYNC précède chaque paquet pour permettre au(x) récepteur(s) de synchroniser ses(leurs) horloges de récupération de bits.

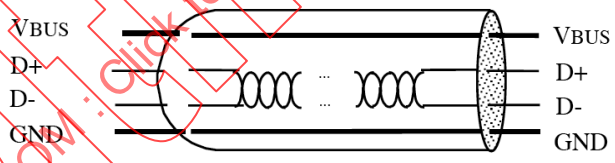


Figure 4-2. Câble USB

Chaque segment de câble contient également des fils VBUS et GND pour alimenter les appareils en électricité. VBUS a une valeur nominale de +5 V au niveau de la source. L'USB autorise des segments de câble de longueurs variables, pouvant atteindre plusieurs mètres, en choisissant le calibre de conducteur approprié correspondant à la chute de tension ohmique et à d'autres attributs, tels que le budget d'alimentation électrique des appareils et la flexibilité des câbles. Afin de fournir les niveaux de tension d'entrée garantis et l'impédance de terminaison correcte, les terminaisons faussées sont utilisées à chaque extrémité du câble. Les terminaisons permettent également la détection des connexions et déconnexions au niveau de chaque port et de différencier les appareils à haute/vitesse intégrale des appareils à faible vitesse.

4.2.2 Mécanique

Les spécifications mécaniques des câbles et des connecteurs sont fournies au Chapitre 6. Tous les appareils disposent d'une connexion en amont. Les connecteurs en amont et en aval ne sont pas mécaniquement interchangeables, éliminant ainsi les connexions illégales de boucle avec retour au niveau des hubs. Le câble comporte quatre conducteurs: une paire torsadée de signal de calibre normalisé et une paire d'alimentation dans une gamme de

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

calibres autorisés. Les caractéristiques du connecteur à quatre positions sont: logement blindé, robustesse spécifiée et facilité de connexion-déconnexion.

4.3 Alimentation

La spécification traite deux aspects de l'alimentation électrique:

- La distribution de l'alimentation sur l'USB aborde la question de la manière dont les appareils USB consomment l'électricité fournie par l'hôte sur l'USB.
- La gestion de l'alimentation électrique porte sur la manière dont le logiciel système USB et les appareils trouvent leur place dans le système de gestion de l'alimentation électrique basée sur l'hôte.

4.3.1 Distribution de l'alimentation

Chaque segment USB fournit une quantité limitée d'alimentation sur le câble. L'hôte fournit l'alimentation utilisée par les appareils USB qui sont directement connectés. En outre, un appareil USB peut disposer de sa propre alimentation. Les appareils USB qui dépendent totalement de l'alimentation provenant du câble sont appelés des appareils alimentés par bus. En revanche, ceux qui disposent d'une source d'alimentation alternative sont appelés des appareils autoalimentés. Un hub fournit également l'alimentation aux appareils USB qui y sont connectés. L'architecture autorise les hubs alimentés par bus dans le cadre de certaines contraintes de topologie qui sont abordées ultérieurement au Chapitre 11.

4.3.2 Gestion de l'énergie

Un hôte USB peut avoir un système de gestion de l'alimentation qui est indépendant de l'USB. Le logiciel système USB fonctionne avec le système de gestion de l'alimentation de l'hôte pour traiter les évènements d'alimentation du système, tels que l'interruption ou la reprise. De plus, les appareils USB mettent généralement en application des caractéristiques supplémentaires de gestion de l'alimentation qui leur permettent de gérer leur alimentation par un logiciel système.

Les caractéristiques de distribution et de gestion de l'alimentation de l'USB autorisent son intégration aux systèmes critiques en termes de consommation, tels que les ordinateurs bloc-notes à accumulateurs.

4.4 Protocole de bus

L'USB est un bus interrogé. Le contrôleur hôte lance tous les transferts de données.

La plupart des transactions de bus impliquent la transmission avec un maximum de trois paquets. Chaque transaction commence lorsque le contrôleur hôte, sur une base horaire, envoie un paquet USB décrivant le type et la direction de la transaction, les adresses des appareils USB et le nombre de points d'extrémité. Ce paquet est appelé "paquet de jetons". L'appareil USB destinataire effectue la sélection lui-même en décodant les champs d'adresse appropriés. Dans une transaction donnée, les données sont transférées soit de l'hôte vers un appareil, soit d'un appareil vers l'hôte. La direction du transfert de données est spécifiée dans le paquet de jetons. Ensuite, la source de la transaction envoie un paquet de données ou indique qu'elle n'a aucune donnée à transférer. En général, la destination répond avec un paquet poignée de mains indiquant si le transfert s'est déroulé avec succès.

Certaines transactions de bus entre les contrôleurs hôtes et les hubs impliquent la transmission de quatre paquets. Ces types de transactions sont utilisés pour gérer les transferts de données entre l'hôte et les appareils à vitesse intégrale/faible vitesse.

Le modèle USB de transfert de données entre une source ou une destination de l'hôte et un point d'extrémité d'un appareil est appelé un canal. Il existe deux types de canaux: le flux et le message. Les données de flux n'ont aucune structure définie par USB, contrairement aux données de message. De plus, les canaux ont des associations de bande passante des données, un type de service de transfert et des caractéristiques de point d'extrémité comme la directivité et les tailles de mémoire tampon. La plupart des canaux sont créés lorsqu'un

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

appareil USB est configuré. Un canal de messages, le canal de contrôle par défaut, existe toujours après la mise sous tension d'un appareil afin de fournir l'accès à la configuration, à l'état et aux informations de contrôle de l'appareil.

Le programme des transactions permet le contrôle du flux de certains canaux de flux. Au niveau matériel, cela permet d'éviter les situations de sous-exploitation ou de dépassement des mémoires tampons à l'aide d'une poignée de mains NAK pour accélérer le débit de données. Lorsqu'une poignée de mains NAK a été utilisée pour une transaction, celle-ci est tentée de nouveau lorsque le temps de bus est disponible. Le mécanisme de contrôle du flux permet l'élaboration de programmes flexibles qui gèrent le service concurrent d'un mélange hétérogène de canaux de flux. Par conséquent, plusieurs canaux de flux peuvent être desservis à différents intervalles et avec des paquets de différentes tailles.

4.5 Robustesse

Plusieurs attributs de l'USB contribuent à sa robustesse:

- Intégrité du signal à l'aide des pilotes différentiels, des récepteurs et du blindage
- Protection CRC des champs de contrôle et des données
- Détection des connexions et déconnexions et de la configuration des ressources au niveau système
- Rétablissement automatique du protocole à l'aide de temporisations pour les paquets perdus ou corrompus
- Contrôle du flux des données de transmission en continu pour garantir l'isochronie et la gestion de la mémoire tampon matérielle
- Constructions de données et de canaux de contrôle pour assurer l'indépendance vis-à-vis des interactions négatives entre les fonctions

4.5.1 Détection des erreurs

Le taux d'erreurs sur les bits de base du support USB est censé être proche de celui d'un fond de panier et la nature des pointes de tension est probablement transitoire. Pour fournir une protection contre ces transitoires, chaque paquet inclut des champs de détection d'erreurs. Lorsque l'intégrité des données est requise, comme dans le cas des appareils sans perte de données, une procédure de reprise sur erreur peut être invoquée du point de vue matériel ou logiciel.

Le protocole comprend des CRC séparés pour les champs de contrôle et de données de chaque paquet. Un CRC qui a échoué est considéré comme indiquant un paquet corrompu. Le CRC indique une couverture de 100 % pour les erreurs à bit unique ou à deux bits.

4.5.2 Gestion des erreurs

Le protocole permet la gestion des erreurs du point de vue matériel ou logiciel. La gestion des erreurs matérielles comprend l'établissement de rapports et la relance des transferts échoués. Un contrôleur hôte USB tente une transmission qui rencontre des erreurs trois fois au maximum avant d'informer le logiciel client de l'échec. Le logiciel client peut récupérer d'une manière spécifique à la mise en application.

4.6 Configuration du système

L'USB prend en charge la connexion ou la déconnexion de l'USB à tout moment des appareils USB. Par conséquent, le logiciel système doit prendre en compte les changements dynamiques de la topologie de bus physique.

4.6.1 Connexion des appareils USB

Tous les appareils USB se connectent à l'USB par l'intermédiaire de ports situés sur des appareils USB spécialisés appelés hubs. Les hubs ont des bits d'état qui sont utilisés pour signaler la connexion ou la déconnexion d'un appareil USB d'un de ses ports. L'hôte interroge

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

le hub pour retrouver ces bits. Dans le cas d'une connexion, l'hôte active le port et s'adresse à l'appareil USB par l'intermédiaire du canal de contrôle de l'appareil à l'adresse par défaut.

L'hôte attribue une adresse USB unique à l'appareil, puis détermine si l'appareil USB nouvellement connecté est un hub ou une fonction. L'hôte détermine son extrémité du canal de contrôle pour l'appareil USB à l'aide de l'adresse USB attribuée et du point d'extrémité numéro zéro.

Si l'appareil USB connecté est un hub et que des appareils USB sont connectés à ces ports, alors la procédure ci-dessus s'applique à chaque appareil USB connecté.

Si l'appareil USB connecté est une fonction, les notifications de connexion sont traitées par le logiciel hôte adapté à la fonction.

4.6.2 Déconnexion des appareils USB

Lorsqu'un appareil USB a été déconnecté d'un des ports du hub, celui-ci désactive le port et fournit une indication de déconnexion d'appareil à l'hôte. L'indication de déconnexion est ensuite traitée par le logiciel système USB approprié. Si l'appareil USB déconnecté est un hub, le logiciel système USB doit traiter la déconnexion du hub et de tous les appareils USB qui étaient auparavant connectés au système par l'intermédiaire du hub.

4.6.3 Enumération de bus

L'énumération de bus est l'activité qui identifie et attribue des adresses uniques aux appareils connectés à un bus. Etant donné que l'USB autorise la connexion et la déconnexion des appareils USB de l'USB à tout moment, l'énumération de bus est une activité continue du logiciel système USB. En outre, l'énumération de bus de l'USB comprend également la détection et le traitement des déconnexions.

4.7 Types de flux de données

L'USB prend en charge les données fonctionnelles et l'échange de contrôle entre l'hôte USB et un appareil USB comme un ensemble de canaux unidirectionnels ou bidirectionnels. Les transferts de données USB ont lieu entre le logiciel hôte et un point d'extrémité particulier d'un appareil USB. De telles associations entre le logiciel hôte et un point d'extrémité particulier d'un appareil USB sont appelées canaux. En général, le mouvement des données sur un canal est indépendant du flux de données sur un autre canal. Un appareil USB donné peut avoir plusieurs canaux. Par exemple, un appareil USB donné pourrait avoir un point d'extrémité qui prenne en charge un canal pour le transport des données vers l'appareil USB et un autre point d'extrémité qui prenne en charge un canal pour le transport des données à partir d'un appareil USB.

L'architecture USB englobe quatre types de transfert de données:

- Transferts de contrôle: utilisés pour la configuration d'un appareil au moment de la connexion et pouvant être utilisés à d'autres fins spécifiques à l'appareil, y compris le contrôle d'autres canaux de l'appareil.
- Transferts de données en bloc: générés ou consommés en quantités relativement grandes et par salves et qui ont une grande latitude dynamique en termes de contraintes de transmission.
- Transferts d'interruption de données: utilisés pour la livraison en temps utile, mais fiable des données, par exemple des caractères ou des coordonnées avec un écho perceptible par l'homme ou des caractéristiques de réponse de rétroaction.
- Transferts isochrones de données: occupent une quantité négociée à l'avance de la bande passante USB avec une latence de livraison négociée à l'avance. (Egalement appelés transferts de flux continu en temps réel).

Un canal prend en charge uniquement un des types de transferts décrits ci-dessus pour la configuration d'un appareil donné. Le modèle USB de flux de données est décrit de manière plus détaillée au Chapitre 5.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

4.7.1 Transferts de contrôle

Les données de contrôle sont utilisées par le logiciel système USB pour configurer les appareils lors de leur première connexion. Un autre logiciel de pilote peut choisir d'utiliser des transferts de contrôle d'une manière spécifique à la mise en application. La livraison des données s'effectue sans perte.

4.7.2 Transferts en bloc

Les données en bloc sont généralement constituées de grandes quantités de données, telles que celles utilisées pour les imprimantes ou les numériseurs. Les données en bloc sont séquentielles. Un échange fiable des données est assuré au niveau matériel en utilisant la détection des erreurs du matériel et en invoquant un nombre limité de répétitions de tentative du matériel. De même, la bande passante utilisée par les données en bloc peut varier en fonction des autres activités de bus.

4.7.3 Transferts d'interruption

Un transfert à latence limitée vers ou en provenance d'un appareil est qualifié de données d'interruption. Ces données peuvent être présentées pour un transfert par un appareil à tout moment et sont délivrées par l'USB à un débit qui ne soit pas inférieur à celui spécifié par l'appareil.

Les données d'interruption sont généralement constituées d'une notification d'évènements, de caractères ou de coordonnées qui sont organisés sur un ou plusieurs octets. Un exemple de données d'interruption concerne les coordonnées provenant d'un appareil de pointage. Bien qu'aucun débit de synchronisation ne soit requis de manière spécifique, les données interactives peuvent avoir des limites de temps de réponse que l'USB doit prendre en charge.

4.7.4 Transferts isochrones

Les données isochrones sont continues et en temps réel du point de vue de la création, la livraison et la consommation. Les informations de synchronisation sont déterminées par le débit constant avec lequel les données isochrones sont reçues et transférées. Les données isochrones doivent être délivrées au débit reçu pour maintenir sa synchronisation. En plus du débit de livraison, les données isochrones peuvent également être sensibles aux délais de livraison. Pour les canaux isochrones, la bande passante requise repose généralement sur les caractéristiques d'échantillonnage de la fonction associée. La latence requise est liée à la mémoire tampon disponible au niveau de chaque point d'extrémité.

Un exemple typique de données isochrones est la voix. Si le débit de livraison de ces flux de données n'est pas maintenu, des chutes apparaissent dans le flux de données à cause de la sous-exploitation ou des dépassements de la mémoire tampon ou de la trame. Même si les données sont livrées au débit approprié par le matériel USB, les délais de livraison introduits par le logiciel peuvent dégrader les applications nécessitant un retournement en temps réel, telles que la conférence audio basée sur la téléphonie.

La livraison en temps utile des données isochrones est assurée au prix de pertes transitoires potentielles dans le flux de données. En d'autres termes, toute erreur survenue dans la transmission électrique n'est pas corrigée par des mécanismes matériels tels que les répétitions de tentative. En pratique, le taux d'erreurs sur les bits de base de l'USB est censé être suffisamment bas pour ne pas représenter un problème. Une portion dédiée de bande passante est allouée aux flux de données isochrones USB afin de s'assurer que les données puissent être livrées au débit souhaité. L'USB est également conçu pour un délai minimum des transferts de données isochrones.

4.7.5 Allocation de bande passante USB

La bande passante USB est distribuée entre les canaux. L'USB alloue la bande passante à certains canaux lors de la création d'un canal. Les appareils USB doivent proposer une mise en mémoire tampon des données. Il est supposé que les appareils USB nécessitant plus de bande passante sont en mesure de fournir des mémoires tampons plus étendues. L'objectif

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

de l'architecture USB est de s'assurer que le délai matériel induit par la mise en mémoire tampon est limité à quelques millisecondes.

La capacité de bande passante de l'USB peut être distribuée entre plusieurs flux de données différents. Cela permet la connexion d'un large éventail d'appareils à l'USB. Qui plus est, différents débits binaires d'un appareil, avec une large plage dynamique, peuvent être pris en charge simultanément.

La spécification USB définit les règles selon lesquelles il est admis que chaque type de transfert à accéder au bus.

4.8 Appareils USB

Les appareils USB sont divisés en classes d'appareils tels que hub, interface humaine, imprimante, appareil d'imagerie ou appareil de stockage en masse. La classe d'appareils hub désigne un appareil spécialement conçu pour l'USB qui fournit des points de connexion USB supplémentaires (voir Chapitre 11). Les appareils USB doivent transporter des informations d'auto-identification et relatives à la configuration générique. Ils doivent également avoir en permanence un comportement conforme aux états d'appareil USB définis.

4.8.1 Caractérisation de l'appareil

L'accès à tous les appareils USB se réalise au moyen d'une adresse USB qui est attribuée lorsque l'appareil est connecté et énuméré. Chaque appareil USB prend également en charge un ou plusieurs canaux par l'intermédiaire desquels l'hôte peut communiquer avec l'appareil. Tous les appareils USB doivent prendre en charge un canal spécialement conçu au niveau du point d'extrémité zéro auquel le canal de contrôle USB de l'appareil USB est connecté. Tous les appareils USB prennent en charge un mécanisme d'accès commun pour l'accès aux informations via ce canal de contrôle.

Les informations requises pour décrire totalement l'appareil USB sont associées au canal de contrôle au niveau du point d'extrémité zéro. Ces informations relèvent des catégories suivantes:

- Norme: cette information a une définition commune à tous les appareils USB et comprend des éléments tels que l'identification du vendeur, la classe de l'appareil et la fonctionnalité de gestion de l'alimentation. Les descriptions de l'appareil, de la configuration, de l'interface et des points d'extrémité comportent les informations liées à la configuration concernant l'appareil. Des informations détaillées concernant ces descripteurs peuvent être trouvées au Chapitre 9.
- Classe: la définition de cette information varie selon la classe de l'appareil USB.
- Vendeur USB: le vendeur de l'appareil USB est libre d'introduire toutes les informations souhaitées. Cependant, le format n'est pas déterminé par cette spécification.

De plus, chaque appareil USB comporte des informations concernant le contrôle USB et l'état.

4.8.2 Description de l'appareil

Il existe deux grandes unités de classes d'appareil: les hubs et les fonctions. Seuls les hubs ont la capacité de fournir des points de connexion USB supplémentaires. Les fonctions fournissent des fonctionnalités supplémentaires à l'hôte.

4.8.2.1 Hubs

Les hubs représentent un élément clé dans l'architecture prête à l'emploi de l'USB. La Figure 4-3 illustre un hub typique. Les hubs permettent de simplifier la connectivité USB du point de vue de l'utilisateur et offrent la robustesse à un coût et une complexité relativement bas.

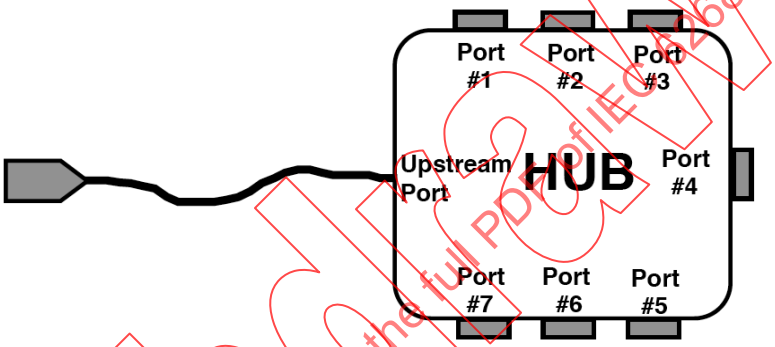
Les hubs sont des hubs de câblage et activent les caractéristiques de connexion multiple de l'USB. Les points de connexion sont appelés ports. Chaque hub convertit un point de

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

connexion unique en des points de connexion multiples. L'architecture prend en charge la concaténation de multiples hubs.

Le port en amont d'un hub connecte le hub vers l'hôte. Chaque port en an aval d'un hub permet la connexion d'un autre hub ou d'une autre fonction. Les hubs peuvent détecter la connexion et la déconnexion au niveau de chaque port en aval et activer la distribution de l'alimentation des appareils en aval. Chaque port en aval peut être activé de manière individuelle et connecté à des appareils à grande vitesse, vitesse intégrale ou faible vitesse.

Un hub de la norme USB 2.0 est formé de trois parties: le contrôleur du hub, le répéteur du hub et le traducteur de transaction. Le répéteur du hub est un commutateur contrôlé par un contrôleur situé entre le port en amont et les ports en aval. Il dispose également d'une prise en charge matérielle pour la réinitialisation et la signalisation d'interruption/reprise. Le contrôleur hôte fournit la communication vers/en provenance de l'hôte. L'état spécifique au hub et les commandes de contrôle permettent à l'hôte de configurer un hub et de surveiller et contrôler ses ports. Le traducteur de transaction fournit les mécanismes pour la prise en charge des appareils à vitesse intégrale/faible vitesse situés derrière le hub, tout en transmettant les données de tous les appareils entre l'hôte et le hub à haute vitesse.



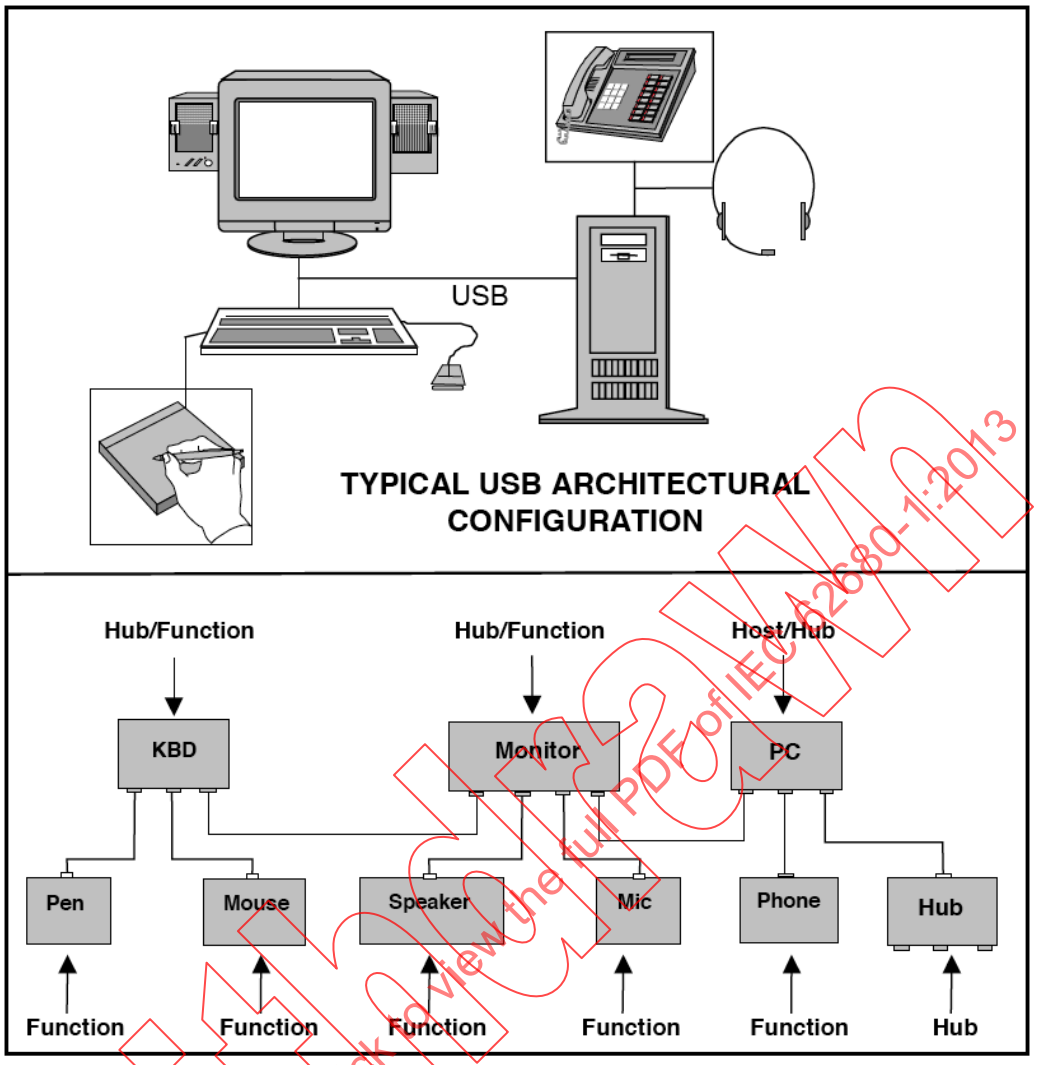
Légende

Anglais	Français
Upstream Port	Port en amont

Figure 4-3. Un hub typique

La Figure 4-4 illustre la manière dont les hubs fournissent la connectivité dans un environnement typique d'ordinateur de bureau.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
TYPICAL USB ARCHITECTURAL CONFIGURATION	CONFIGURATION ARCHITECTURALE USB TYPIQUE
Hub/Function	Hub/Fonction
Host/Hub	Hôte/Hub
KBD	KBD
Monitor	Moniteur
PC	Ordinateur
Pen	Stylo
Mouse	Souris
Speaker	Haut-parleur
Mic	Micro
Phone	Téléphone
Hub	Hub
Function	Fonction

Figure 4-4. Hubs dans un environnement d'ordinateur de bureau

4.8.2.2 Fonctions

Une fonction est un appareil USB qui peut transmettre ou recevoir des données ou des informations de contrôle sur le bus. Une fonction est généralement mise en application comme un appareil périphérique séparé avec un câble qui se branche dans le port d'un hub. Toutefois, un paquetage physique peut mettre en application plusieurs fonctions et un hub intégré avec un seul câble USB. Cela s'appelle un appareil composé. Un appareil composé apparaît à l'hôte comme un hub avec un ou plusieurs appareils USB non amovibles.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Chaque fonction contient des informations de configuration qui décrivent ses fonctionnalités et les exigences en termes de ressources. Avant de pouvoir utiliser une fonction, celle-ci doit être configurée par l'hôte. Cette configuration comprend l'allocation de bande passante USB et la sélection des options de configuration spécifiques à la fonction.

Des exemples de fonctions sont les suivants:

- Un appareil d'interface humaine tel qu'une souris, un clavier, une tablette ou une manette de jeu
- Un appareil d'imagerie tel qu'un numériseur, une imprimante ou une caméra
- Un appareil de stockage en masse tel qu'un lecteur de CD-ROM, un lecteur de disquette ou un lecteur de DVD

4.9 Hôte USB: matériel et logiciel

L'hôte USB interagit avec les appareils USB par l'intermédiaire du contrôleur hôte. L'hôte est responsable des tâches suivantes:

- Détection de la connexion et de la déconnexion des appareils USB
- Gestion du flux de contrôle entre l'hôte et les appareils USB
- Gestion du flux de données entre l'hôte et les appareils USB
- Collecte des statistiques d'état et d'activité
- Fourniture de l'alimentation aux appareils USB connectés

Le logiciel système USB de l'hôte gère les interactions entre les appareils USB et le logiciel d'appareil basé sur l'hôte. Il existe cinq domaines d'interaction entre le logiciel système USB et le logiciel d'appareil:

- Enumération et configuration des appareils
- Transferts isochrones de données
- Transferts asynchrones de données
- Gestion de l'alimentation
- Informations relatives aux appareils et à la gestion du bus

4.10 Extensions architecturales

L'architecture USB englobe l'extensibilité au niveau de l'interface entre le pilote du contrôleur hôte et le pilote USB. Les mises en application avec de multiples contrôleurs hôtes et les pilotes hôtes associés sont possibles.

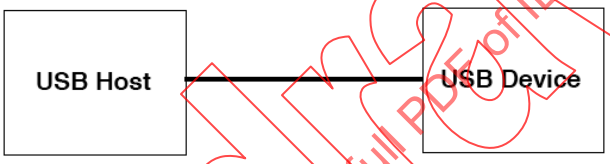
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5 Chapitre 5
Modèle de flux de données USB

Le présent chapitre apporte des informations sur la manière dont les données se déplacent sur l'USB. Les informations contenues dans le présent chapitre affectent tous les concepteurs. Les informations présentées se trouvent à un niveau supérieur à celui des définitions de la signalisation et des protocoles du système. Voir les Chapitres 7 et 8 pour plus de détails sur les parties respectives du système USB. Le présent chapitre fournit un cadre d'informations qui est davantage développé aux Chapitres 9 à 11. Il convient que tous les concepteurs lisent le présent chapitre afin de comprend les concepts clés de l'USB.

5.1 Points de vue du concepteur

L'USB fournit des services de communication entre un hôte et les appareils USB connectés. Toutefois, la simple vue qu'un utilisateur final voit lorsqu'il connecte un ou plusieurs appareils USB à un hôte, comme illustré sur la Figure 5-1, est en fait plus compliquée à mettre en application que celle indiquée sur la figure. Différentes vues du système sont requises pour expliquer les exigences spécifiques à l'USB du point de vue de différents concepteurs. Plusieurs caractéristiques et concepts importants doivent être pris en charge afin de fournir à l'utilisateur final le fonctionnement fiable exigé par les ordinateurs personnels d'aujourd'hui. L'USB est représenté par des couches pour en faciliter l'explication et permettre aux concepteurs de produits USB particuliers de se concentrer sur les détails liés à leur produit.



Légende

Anglais	Français
USB Host	Hôte USB
USB Device	Appareil USB

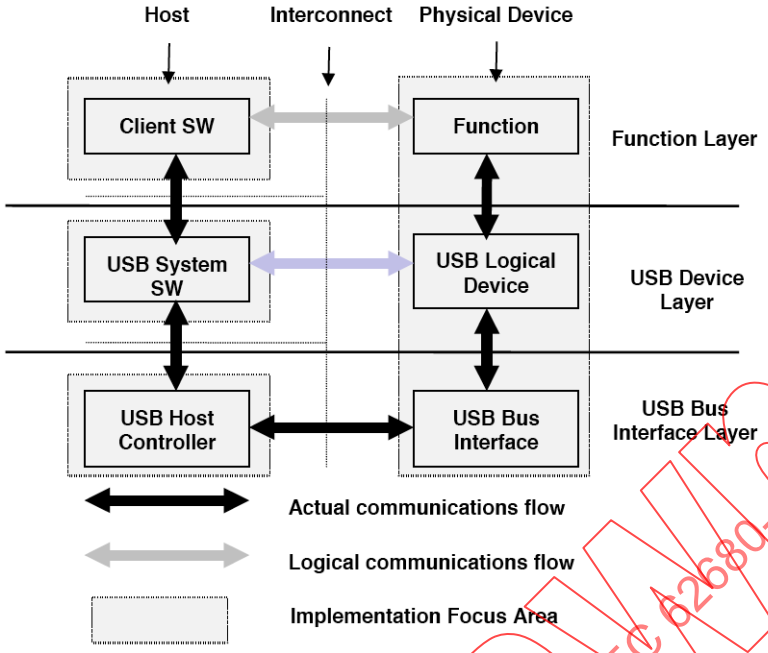
Figure 5-1. Simple vue hôte/appareil USB

La Figure 5-2 montre une vue d'ensemble approfondie de l'USB en identifiant les différentes couches du système qui sont décrites de façon plus détaillée dans la suite de la spécification. En particulier, il y a quatre zones prioritaires pour la mise en application:

- Appareil physique USB: composante matérielle située à l'extrémité d'un câble USB qui effectue une fonction utile pour l'utilisateur final.
- Logiciel client: logiciel qui s'exécute sur l'hôte, correspondant à un appareil USB. Ce logiciel client est généralement inclus dans le système d'exploitation ou fourni avec l'appareil USB.
- Logiciel système USB: logiciel qui prend en charge l'USB dans un système d'exploitation particulier. Le logiciel système USB est généralement fourni avec le système d'exploitation, indépendamment des appareils USB particuliers ou du logiciel client.
- Contrôleur hôte USB (interface de bus côté hôte): matériel et logiciel qui permettent aux appareils USB de se connecter à un hôte.

Il existe des droits et des responsabilités partagés entre les quatre composants systèmes USB. La suite de la présente spécification décrit les détails requis pour prendre en charge les flux de communication robustes et fiables entre une fonction et son client.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Host	Hôte
Interconnect	Interconnexion
Physical Device	Appareil physique
Client SW	Logiciel client
Function	Fonction
Function Layer	Couche Fonction
USB System SW	Logiciel système USB
USB Logical Device	Appareil logique USB
USB Device Layer	Couche appareil USB
USB Host Controller	Contrôleur hôte USB
USB Bus Interface	Interface de bus USB
USB Bus Interface Layer	Couche Interface de bus USB
Actual communications flow	Flux de communication actuel
Logical communications flow	Flux de communication logique
Implementation Focus Area	Zone prioritaire pour la mise en application

Figure 5-2. Zones d'application USB

Tel qu'illustré à la Figure 5-2, la simple connexion d'un hôte à un appareil nécessite une interaction entre plusieurs couches et entités. La couche Interface de bus USB fournit la connectivité physique/de signalisation/de paquets entre l'hôte et un appareil. La couche Appareil USB représente la vue dont le logiciel système USB dispose pour effectuer des opérations USB génériques avec un appareil. La couche Fonction fournit des fonctionnalités supplémentaires à l'hôte par l'intermédiaire d'une couche Client logiciel appropriée mise en correspondance. Les couches Appareil USB et Fonction ont chacune une vue de la communication logique dans le cadre de leur couche qui utilise la couche Interface de bus USB pour accomplir le transfert des données.

La vue physique de la communication USB, décrite aux Chapitres 6, 7 et 8, est liée à la vue logique de la communication présentée aux Chapitres 9 et 10. Le présent chapitre décrit des concepts clés qui affectent les concepteurs USB et il convient qu'il soit lu par tous avant de continuer avec la suite de la spécification pour trouver les détails les plus pertinents pour leur produit.

Pour décrire et gérer la communication USB, les concepts suivants sont importants:

- Topologie en bus: la Section 5.2 présente les principales composantes physiques et logiques de l'USB et les relations qui les unissent.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- Modèles de flux de communication: les Sections 5.3 à 5.8 décrivent le flux de communication entre l'hôte et les appareils par l'intermédiaire de l'USB et définissent les quatre types de transfert USB.
- Gestion de l'accès au bus: la Section 5.11 décrit la gestion de l'accès au bus dans le cadre de l'hôte pour la prise en charge d'une large gamme de débits de communication par les appareils USB.
- Considération spéciale pour les transferts isochrones: la Section 5.12 présente les caractéristiques de l'USB spécifiques aux appareils nécessitant des transferts de données isochrones. Il n'est pas nécessaire que les concepteurs d'appareils non isochrones lisent la Section 5.12.

5.2 Topologie en bus

La topologie USB comporte quatre parties principales:

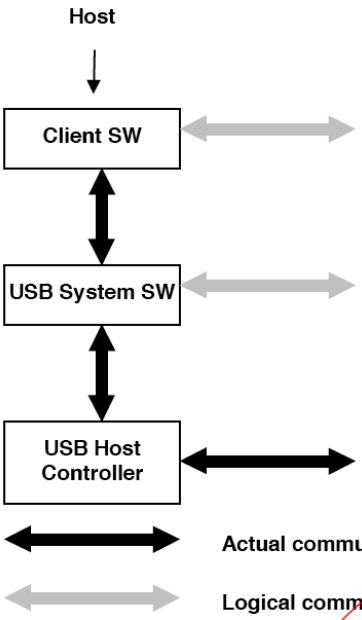
- Hôte et appareils: les composantes principales d'un système USB
- Topologie physique: la méthode de connexion des éléments USB
- Topologie logique: les rôles et responsabilités des différents éléments USB et la manière dont l'USB apparaît du point de vue de l'hôte ou d'un appareil
- Relations logiciel client-fonction: la manière dont le logiciel client et ses interfaces de fonction associées sur un appareil USB se voient les uns les autres

5.2.1 Hôte USB

La composition logique de l'hôte est représentée à la Figure 5-3 et comprend les éléments suivants:

- Contrôleur hôte USB
- Logiciel système USB agrégé (pilote USB, pilote de contrôleur hôte et logiciel hôte)
- Client

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Host	Hôte
Client SW	Logiciel client
USB System SW	Logiciel système USB
USB Host Controller	Contrôleur hôte USB
Actual communications flow	Flux de communication actuel
Logical communications flow	Flux de communication logique

Figure 5-3. Composition de l'hôte

L'hôte USB occupe une position unique en tant qu'unité de coordination de l'USB. En plus de sa position physique spéciale, l'hôte a des responsabilités spécifiques par rapport à l'USB et les appareils qui y sont connectés. L'hôte contrôle tous les accès à l'USB. Un appareil USB obtient l'accès au bus uniquement lorsque l'accès a été autorisé par l'hôte. L'hôte est également responsable de la surveillance de la topologie de l'USB.

Pour une discussion complète sur l'hôte et ses tâches, voir le Chapitre 10.

5.2.2 Appareils USB

La composition logique d'un appareil USB physique est représentée à la Figure 5-4 et comprend les éléments suivants:

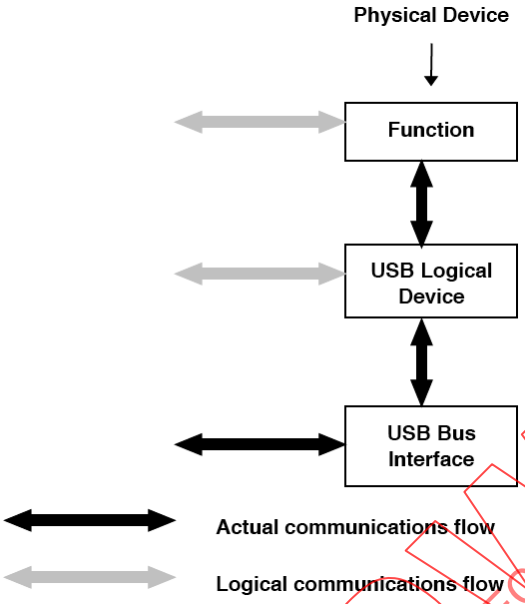
- Interface de bus USB
- Appareil logique USB
- Fonction

Les appareils physiques USB fournissent une fonctionnalité supplémentaire à l'hôte. Les types de fonctionnalité fournis par les appareils USB varient largement. Toutefois, tous les appareils logiques USB présentent la même interface de base à l'hôte. Cela permet à l'hôte de gérer les aspects concernant l'USB de différents appareils USB de la même manière.

Pour aider l'hôte à identifier et configurer les appareils USB, chaque appareil transporte et enregistre les informations relatives à la configuration. Certaines informations enregistrées sont communes à tous les appareils logiques. D'autres informations sont spécifiques à la fonctionnalité fournie par l'appareil. Le format détaillé de ces informations varie selon la classe de l'appareil.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Pour une discussion complète sur les appareils USB, voir le Chapitre 9.



Légende

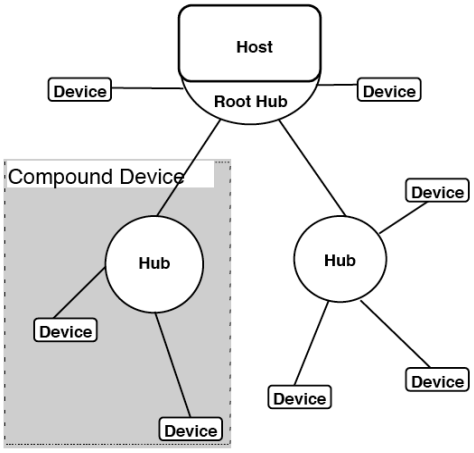
Anglais	Français
Physical Device	Appareil physique
Function	Fonction
USB Logical Device	Appareil logique USB
USB Bus Interface	Interface de bus USB
Actual communications flow	Flux de communication actuel
Logical communications flow	Flux de communication logique

Figure 5-4. Composition de l'appareil physique

5.2.3 Topologie de bus physique

Les appareils USB sont physiquement connectés à l'hôte par l'intermédiaire d'une topologie en étoile à plusieurs niveaux, telle qu'illustrée à la Figure 5-5. Les points de connexion USB sont fournis par une classe spéciale d'appareil USB appelée un hub. Les points de connexion supplémentaires fournis par un hub sont appelés des ports. Un hôte dispose d'un hub intégré appelé le hub racine. L'hôte fournit un ou plusieurs points de connexion par l'intermédiaire du hub racine. Les appareils USB qui fournissent une fonctionnalité supplémentaire à l'hôte sont appelés des fonctions. Pour éviter les connexions circulaires, un classement à plusieurs niveaux est imposé à la topologie en étoile de l'USB. Cela résulte en une configuration arborescente illustrée à la Figure 5-5.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



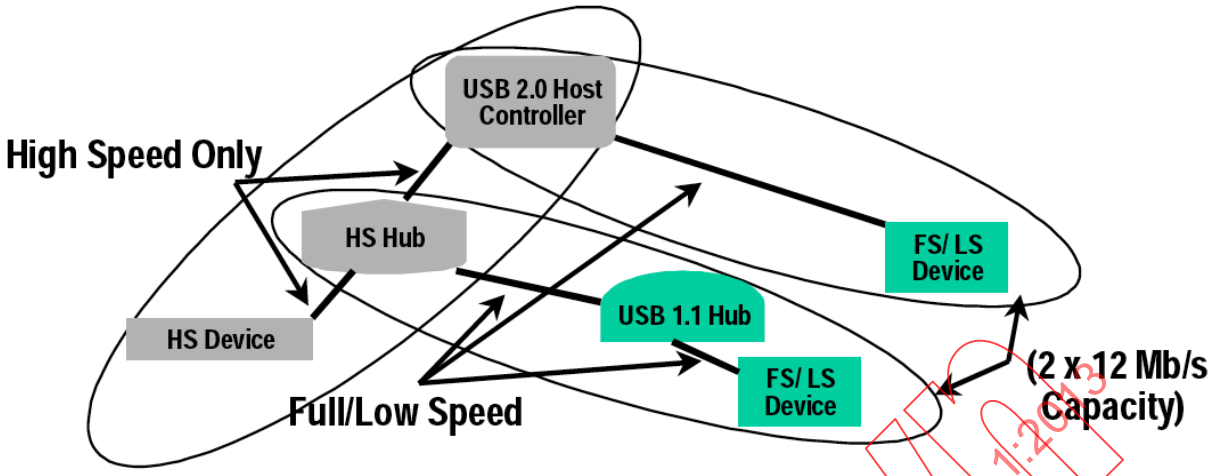
Légende

Anglais	Français
Device	Appareil
Host	Hôte
Root Hub	Hub racine
Compound Device	Appareil composé

Figure 5-5. Topologie de bus physique USB

Plusieurs fonctions peuvent être empaquetées ensemble dans ce qui semble être un appareil physique unique. Par exemple, un clavier et une boule de traçage peuvent être combinés dans un seul paquetage. A l'intérieur du paquetage, les fonctions individuelles sont connectées en permanence à un hub et c'est le hub interne qui est connecté à l'USB. Lorsque plusieurs fonctions sont combinées avec un hub dans un seul paquetage, elles sont désignées sous le terme d'appareil composé. Le hub et chaque fonction connectée au hub dans le cadre d'un appareil composé se voient attribuer leur propre adresse d'appareil. Un appareil qui dispose de plusieurs interfaces contrôlées de façon indépendante les unes des autres s'appelle un appareil composite. Un appareil composite n'a qu'une seule adresse d'appareil. Du point de vue de l'hôte, un appareil composé est le même qu'un hub séparé avec plusieurs fonctions qui y sont connectées. La Figure 5-5 illustre un appareil composé.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
High Speed Only	Grande vitesse uniquement
USB 2.0 Host Controller	Contrôleur hôte USB 2.0
HS Hub	Hub HS
HS Device	Appareil HS
USB 1.1 Hub	Hub USB 1.1
FS/LS Device	Appareil FS/LS
Full/Low Speed	Vitesse intégrale/faible vitesse
FS/LS Device	Appareil FS/LS
(2 x 12 Mb/s Capacity)	(Capacité de 2 x 12 Mb/s)

Figure 5-6. Plusieurs bus à vitesse intégrale dans un système à haute vitesse

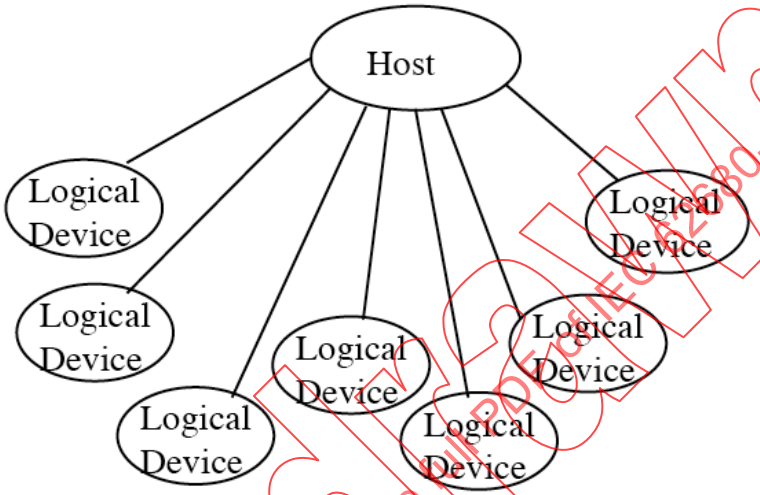
Le hub joue un rôle spécial dans un système à haute vitesse. Le hub isole l'environnement de signalisation à vitesse intégrale/faible vitesse de l'environnement de signalisation à haute vitesse. La Figure 5-6 montre un hub fonctionnant à haute vitesse et prenant en charge un appareil à haute vitesse qui y est connecté. Le hub autorise également des hubs USB1.1 à se connecter et à fonctionner à vitesse intégrale/faible vitesse avec des appareils qui fonctionnent à vitesse intégrale/faible vitesse uniquement. Le contrôleur hôte prend également en charge directe la connexion d'appareils qui fonctionnent à vitesse intégrale/faible vitesse uniquement. Le Chapitre 11 décrit comment le hub accomplit l'isolation des deux environnements de signalisation.

Chaque hub fonctionnant à haute vitesse ajoute essentiellement un (ou plusieurs) bus à vitesse intégrale/faible vitesse supplémentaire(s); c'est-à-dire, chaque hub prend en charge 12 Mb/s (plusieurs en option) supplémentaires de bande passante à vitesse intégrale/faible vitesse. Cela permet de connecter plusieurs bus à vitesse intégrale/faible vitesse sans nécessiter l'ajout de contrôleurs hôtes supplémentaires dans un système. Bien qu'il puisse y avoir plusieurs bus à vitesse intégrale/faible vitesse de 12 Mb/s, seuls 127 appareils USB peuvent être connectés tout au plus à un seul contrôleur hôte.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.2.4 Topologie de bus logique

Tandis que les appareils se connectent physiquement à l'USB dans une topologie en étoile à plusieurs niveaux, l'hôte communique avec chaque appareil logique comme s'il était directement connecté au port racine. Cela crée la vue logique illustrée à la Figure 5-7 qui correspond à la topologie physique présentée à la Figure 5-5. Les hubs sont des appareils logiques, bien qu'ils ne soient pas présentés à la Figure 5-7 afin de simplifier l'image. Bien que la plupart des activités d'appareil hôte/logique utilisent cette perspective logique, l'hôte demeure conscient de la topologie physique afin de prendre en charge le traitement de la déconnexion des hubs. Lorsqu'un hub est déconnecté, tous les appareils connectés au hub doivent être supprimés de la vue de l'hôte sur la topologie logique. Une discussion plus complète sur les hubs peut être consultée au Chapitre 11.



Légende

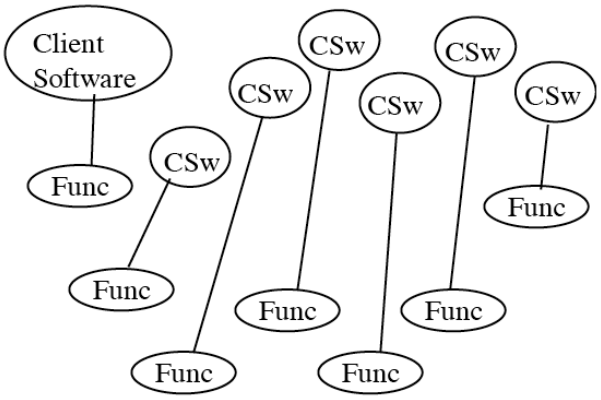
Anglais	Français
Host	Hôte
Logical Device	Appareil logique

Figure 5-7. Topologie de bus logique USB

5.2.5 Relation logiciel client-fonction

Bien que la topologie physique et logique de l'USB reflète la nature partagée du bus, le logiciel client (CSw) manipulant une interface de fonction USB se voir présenter la vue qu'il gère uniquement avec sa(ses) interfaces(s) d'intérêt. Le logiciel client des fonctions USB doit utiliser les interfaces de programmation logicielle USB pour manipuler ses fonctions au lieu de les manipuler directement par l'intermédiaire de la mémoire ou des accès E/S comme les autres bus (par exemple, PCI, EISA, PCMCIA, etc.). Pendant le fonctionnement, il convient que le logiciel client soit indépendant des autres appareils qui peuvent se connecter à l'USB. Cela permet au concepteur de l'appareil et du logiciel client de se concentrer sur les détails relatifs à la conception de l'interaction matérielle/logicielle. La Figure 5-8 illustre le point de vue d'un concepteur de appareils sur les relations entre le logiciel client et les fonctions USB en matière de topologie logique USB de la Figure 5-7.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Client Software	Logiciel client
Func	Fonction

Figure 5-8. Relations logiciel client-fonction

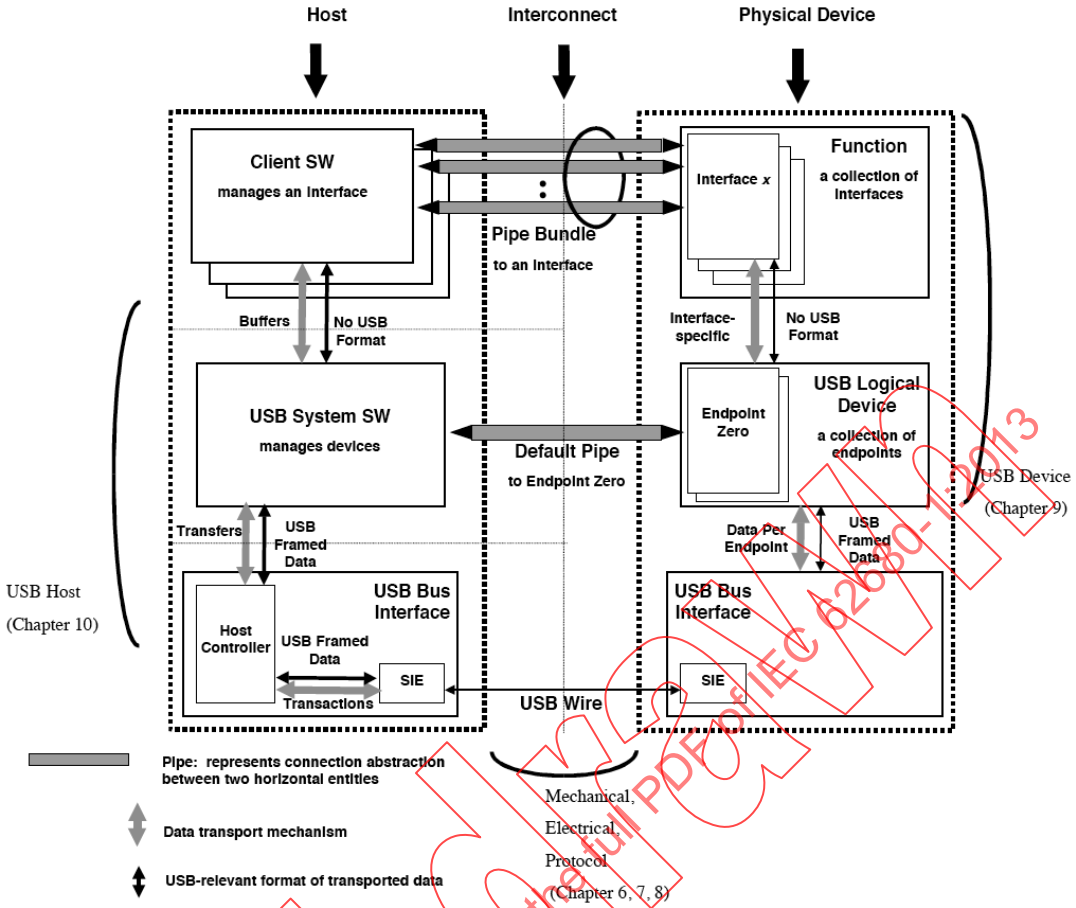
5.3 Flux de communication USB

L'USB fournit un service de communication entre le logiciel de l'hôte et sa fonction USB. Les fonctions peuvent avoir différentes exigences de débit de communication pour les interactions client-fonction. L'USB fournit une meilleure utilisation globale du bus en permettant la séparation des différents flux de communication vers une fonction USB. Chaque flux de communication utilise l'accès au bus pour réaliser la communication entre le client et la fonction. Chaque flux de communication se termine au niveau d'un point d'extrémité d'un appareil. Les points d'extrémité d'un appareil sont utilisés pour identifier les aspects de chaque flux de communication.

La Figure 5-9 montre une vue plus détaillée de la Figure 5-2. La définition complète des flux de communication actuels de la Figure 5-2 prend en charge les flux de communication des couches appareil logique et fonction. Ces flux de communication traversent plusieurs limites d'interface. Les Chapitres 6 à 8 décrivent les définitions mécaniques, électriques et d'interface de protocole du "câble" USB Le Chapitre 9 décrit l'interface de programmation d'appareil USB qui permet la manipulation d'un appareil USB à partir du côté hôte du câble. Le Chapitre 10 décrit deux interfaces logicielles du côté hôte:

- Pilote de contrôleur hôte (HCD): interface logicielle entre le contrôleur hôte USB et le logiciel système USB. Cette interface permet plusieurs mises en application de contrôleur hôte sans que les logiciels hôtes doivent dépendre d'une mise en application particulière. Un pilote USB peut prendre en charge différents contrôleurs hôtes sans aucune connaissance particulière d'une mise en application d'un contrôleur hôte. Le concepteur d'un contrôleur hôte fournit une mise en application HCD qui prend en charge le contrôleur hôte.
- Pilote USB (USBD): interface entre le logiciel système USB et le logiciel client. Cette interface fournit aux clients des fonctions utiles pour la manipulation des appareils USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

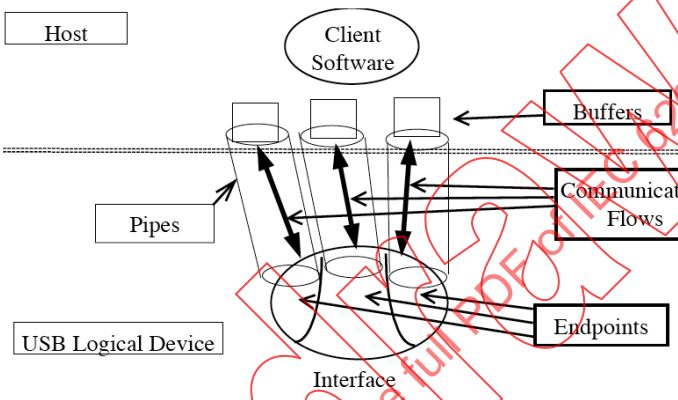
Anglais	Français
Host	Hôte
Interconnect	Interconnexion
Physical Device	Appareil physique
Client SW manages an interface	Le logiciel client gère une interface
Pipe Bundle to an interface	Faisceau de canaux vers une interface
Interface x	Interface x
Function a collection of interfaces	Fonction une collection d'interfaces
Buffers	Mémoires tampons
No USB Format	Aucun format USB
Interface specific	Spécifique à l'interface
USB System SW manages devices	Le logiciel système USB gère les appareils
Default Pipe to Endpoint zero	Canal par défaut vers point d'extrémité zéro
Endpoint Zero	Point d'extrémité zéro
USB Logical Device a collection of endpoints	Appareil logique USB une collection de points d'extrémité
USB Device (Chapter 9)	Appareil USB (Chapitre 9)
Transfers	Transferts
USB Framed Data	Données USB mises en trame
Data per Endpoint	Données par point d'extrémité
USB Host (Chapter 10)	Hôte USB (Chapitre 10)
USB Bus Interface	Interface de bus USB
Host Controller	Contrôleur hôte
Transactions	Transactions
SIE	SIE
USB Wire	Câble USB
Pipe: represents connection abstraction between two horizontal entities	Canal: représente une abstraction de connexion entre deux entités horizontales
Data transport mechanism	Mécanisme de transport des données
USB-relevant format or transported data	Format relatif à l'USB ou données transportées
Mechanical, Electrical, Protocol (Chapter 6, 7, 8)	Domaine mécanique, domaine électrique, protocole (Chapitres 6, 7, 8)

Figure 5-9. Vue détaillée de l'hôte/appareil USB

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Un appareil logique USB se présente au système USB sous la forme d'une collection de points d'extrémité. Les points d'extrémité sont regroupés dans des ensembles d'extrémité qui mettent en application une interface. Les interfaces sont des vues de la fonction. Le logiciel système USB gère l'appareil à l'aide du canal de contrôle par défaut. Le logiciel client gère une interface à l'aide de faisceaux de canaux (associé à un ensemble de points d'extrémité). Le logiciel client exige que les données se déplacent sur l'USB entre une mémoire tampon de l'hôte et un point d'extrémité de l'appareil USB. Le contrôleur hôte (ou l'appareil USB, en fonction de la direction de transfert) emballe les données dans un paquet pour les déplacer sur l'USB. Le contrôleur hôte coordonne également le moment auquel l'accès au bus est utilisé pour le déplacement du paquet de données sur l'USB.

La Figure 5-10 illustre la manière dont les flux de communications sont transportés sur les canaux entre les points d'extrémité et les mémoires tampons du côté hôte. Les sections suivantes décrivent les points d'extrémité, les canaux et les flux de communication de façon plus détaillée.



Légende

Anglais	Français
Host	Hôte
Client Software	Logiciel client
Buffers	Mémoires tampons
Pipes	Canaux
Communication Flows	Flux de communication
USB Logical Device	Appareil logique USB
Interface	Interface
Endpoints	Points d'extrémité

Figure 5-10. Flux de communication USB

Le logiciel de l'hôte communique avec un appareil logique par l'intermédiaire d'un ensemble de flux de communication. L'ensemble de flux de communication est sélectionné par les concepteurs logiciels/matériels de l'appareil pour faire correspondre efficacement les exigences de communication de l'appareil aux caractéristiques de transfert USB.

5.3.1 Points d'extrémité de l'appareil

Un point d'extrémité est une portion identifiable de façon unique d'un appareil USB qui est le terminus d'un flux de communication entre l'hôte et l'appareil. Chaque appareil logique USB est composé d'une collection de points d'extrémité indépendants. Chaque appareil logique a une adresse unique attribuée par le système au moment de la connexion de l'appareil. Au moment de la conception, chaque point d'extrémité d'un appareil se voit attribuer un identifiant unique déterminé par l'appareil appelé le numéro de point d'extrémité. Chaque point d'extrémité a une direction déterminée par l'appareil du flux de données. La combinaison de l'adresse de l'appareil, du numéro de point d'extrémité et de la direction permet un référencement unique de chaque point d'extrémité. Chaque point d'extrémité est une connexion simple qui prend en charge le flux de données dans une direction: soit l'entrée (de l'appareil vers l'hôte), soit la sortie (de l'hôte vers l'appareil).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Un point d'extrémité a des caractéristiques qui permettent de déterminer le type de service de transfert requis entre le point d'extrémité et le logiciel client. Un point d'extrémité se décrit par:

- Les exigences de fréquence/latence d'accès au bus
- Les exigences de bande passante
- Le numéro de point d'extrémité
- Les exigences de comportement de la gestion des erreurs
- La taille maximum des paquets que le point d'extrémité est en mesure d'envoyer ou de recevoir
- Le type de transfert du point d'extrémité (voir la Section 5.4 pour les détails)
- La direction dans laquelle les données sont transférées entre le point d'extrémité et l'hôte

Les points d'extrémité autres que ceux avec un numéro zéro se trouvent dans un état inconnu et ne peuvent pas être accessibles à l'hôte avant d'être configurés.

5.3.1.1 Exigences des points d'extrémité zéro

Tous les appareils USB doivent mettre en application une méthode de contrôle par défaut qui utilise les points d'extrémité d'entrée et de sortie avec un point d'extrémité numéro zéro. Le logiciel système USB utilise cette méthode de contrôle par défaut pour initialiser et manipuler génériquement l'appareil logique (par exemple, pour configurer l'appareil logique) comme étant le canal de contrôle par défaut (voir Section 5.3.2). Le canal de contrôle par défaut fournit l'accès aux informations de configuration de l'appareil et autorise un accès générique à l'état USB et au contrôle. Le canal de contrôle par défaut prend en charge les transferts, tel que défini à la Section 5.5. Les points d'extrémité avec le numéro zéro de point d'extrémité sont toujours accessibles lorsqu'un appareil a été connecté, mis sous tension et reçu une réinitialisation de bus.

Un appareil USB capable de fonctionner à haute vitesse doit avoir un niveau minimum de prise en charge du fonctionnement à vitesse intégrale. Lorsque l'appareil est connecté à un hub fonctionnant à vitesse intégrale, l'appareil doit:

- Être en mesure de se réinitialiser avec succès à vitesse intégrale
- Répondre avec succès aux requêtes normalisées: set_address, set_configuration, get_descriptor de l'appareil et des descripteurs de configuration, et retourner les informations appropriées

L'appareil à haute vitesse peut ou peut ne pas être en mesure de prendre en charge sa fonctionnalité prévue pendant le fonctionnement à vitesse intégrale.

5.3.1.2 Exigences des points d'extrémité autres que zéro

Les fonctions peuvent avoir des points d'extrémité supplémentaires selon les besoins de leur mise en application. Les fonctions à faible vitesse sont limitées à deux points d'extrémité optionnels en plus des deux requis pour la mise en application du canal de contrôle par défaut. Les appareils à vitesse intégrale peuvent avoir des points d'extrémité supplémentaires uniquement limités par la définition du protocole (c'est-à-dire, un nombre maximum de 15 points d'extrémité d'entrée supplémentaires et de 15 points d'extrémité de sortie supplémentaires).

Les points d'extrémité autres que ceux du canal de contrôle par défaut ne peuvent pas être utilisés avant la configuration de l'appareil comme faisant partie du processus de configuration de l'appareil (voir Chapitre 9).

5.3.2 Canalisations

Un canal USB est une association entre un point d'extrémité d'un appareil et le logiciel de l'hôte. Les canaux représentent la capacité de déplacement des données entre le logiciel de

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

l'hôte par l'intermédiaire d'une mémoire tampon et un point d'extrémité d'un appareil. Il existe deux modes de communication de canal mutuellement exclusifs:

- Flux: les données qui se déplacent dans un canal n'ont aucune structure définie pour l'USB
- Message: les données qui se déplacent dans un canal ont une structure définie pour l'USB

L'USB n'interprète pas le contenu des données qu'il livre par l'intermédiaire d'un canal. Même si un canal de messages exige que les données soient structurées conformément aux définitions USB, le contenu des données n'est pas interprété par l'USB.

En outre, les canaux sont associés aux éléments suivants:

- Une demande d'accès au bus USB et d'utilisation de la bande passante.
- Un type de transfert.
- Les caractéristiques du point d'extrémité associé, comme la directivité et les tailles maximales de charge de travail des données. La charge de travail des données représente les données qui sont transportées dans le champ de données d'un paquet de données dans le cadre d'une transaction de bus (tel que défini au Chapitre 8).

Le canal qui est composé de deux points d'extrémité avec un numéro zéro de point d'extrémité est appelé le canal de contrôle par défaut. Ce canal existe toujours une fois qu'un appareil a été mis sous tension et a reçu une réinitialisation de bus. Les autres canaux sont créés lorsqu'un appareil USB est configuré. Le canal de contrôle par défaut est utilisé par le logiciel système USB pour déterminer les exigences d'identification et de configuration de l'appareil et pour configurer l'appareil. Le canal de contrôle par défaut peut également être utilisé par un logiciel spécifique à l'appareil après la configuration de l'appareil. Le logiciel système USB conserve la "possession" du canal de contrôle par défaut et supervise l'utilisation du canal par un autre logiciel client.

Normalement, un client logiciel demande des transferts de données par l'intermédiaire de paquets de requêtes E/S (IRP) vers un canal, puis attend ou est averti une fois que les transferts ont été effectués. Les détails concernant les IRP sont définis de manière spécifique au système d'exploitation. La présente spécification utilise le terme pour faire référence à une requête identifiable par un client logiciel pour le déplacement des données entre lui-même (sur l'hôte) et un point d'extrémité d'un appareil dans une direction appropriée. Un client logiciel peut pousser un canal à retourner tous les IRP en cours s'il le souhaite. Le client logiciel est notifié qu'un IRP a été envoyé lorsque les transactions de bus qui y sont associées se sont terminées avec succès ou en raison d'erreurs.

Si aucun IRP n'est en attente ni en cours sur le canal, ce dernier est au repos et le contrôleur hôte ne prend aucune mesure concernant le canal; c'est-à-dire, le point d'extrémité de ce canal ne détecte aucune transaction de bus dans sa direction. La seule activité de bus présente sur un canal est celle des IRP en attente pour ce canal.

Si un canal non isochrone rencontre un problème qui le contraint à envoyer un paquet STALL à l'hôte (voir Chapitre 8) ou que trois erreurs de bus sont rencontrées sur l'un des paquets d'un IRP, ce dernier est interrompu/retiré, tous les IRP en cours sont également retirés et plus aucun IRP n'est accepté avant que le client logiciel ne récupère du problème (de manière dépendante de la mise en application) et accuse réception de la condition d'arrêt ou d'erreur par l'intermédiaire d'un appel USBD. Un état approprié informe le client logiciel de l'IRP spécifique du résultat d'erreur ou d'arrêt (voir Chapitre 10). Le comportement du canal isochrone est décrit à la Section 5.6.

Il peut être nécessaire qu'un IRP ait besoin de plusieurs charges utiles de données pour déplacer les données du client sur le bus. Les charges utiles de données de cet IRP à multiples charges de travail de données sont censées avoir la taille de paquet maximum jusqu'à la dernière charge utile de données qui contient la partie restante de l'IRP global. Voir la description de chaque type de transfert pour de plus amples détails. Pour un tel IRP, les petits paquets (c'est-à-dire, dont la taille est inférieure à la taille maximale des charges utiles

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

de données) en entrée qui ne remplissent pas complètement la mémoire tampon d'un IRP peuvent avoir une ou plusieurs significations possibles, en fonction des attentes d'un client:

- Un client peut s'attendre à trouver une quantité variable de données dans un IRP. Dans ce cas, un petit paquet qui ne remplit pas la mémoire tampon d'un IRP peut être utilisé tout simplement comme un délimiteur à l'intérieur de la bande pour indiquer la "fin de l'unité de données". Il convient que l'IRP soit retiré sans erreur et que le contrôleur hôte avance à l'IPR suivant.
- Un client peut s'attendre à trouver une quantité spécifique de données. Dans ce cas, un petit paquet qui ne remplit pas la mémoire tampon d'un IRP est l'indication d'une erreur. Il convient que l'IRP soit retiré, que le canal soit bloqué et que tous les IRP en attente associés au canal soient également retirés.

Etant donné que le contrôleur hôte doit avoir un comportement différent dans les deux cas et qu'il ne peut pas savoir de lui-même quel comportement adopter pour un IRP donné, le comportement souhaité peut être indiqué par le client pour chaque IRP.

Un point d'extrémité peut informer l'hôte qu'il est occupé en répondant par un NAK. Les NAK ne sont pas utilisés comme condition de retrait pour retourner un IRP au client logiciel. Un nombre indifférent de NAK peut être rencontré durant le traitement d'un IRP donné. Une réponse NAK à une transaction ne constitue pas une erreur et n'est pas considérée comme l'une des trois erreurs décrites ci-dessus.

5.3.2.1 Canaux de flux

Les canaux de flux délivrent des données dans la portion de paquet de données des transactions de bus sans qu'aucune structure requise pour l'USB ne soit imposée au contenu de données. Les données entrent par une extrémité d'un canal de flux et sortent par l'autre extrémité dans le même ordre. Les canaux de flux sont toujours unidirectionnels dans leur flux de communication.

Les données traversant un canal de flux sont censées interagir avec ce que l'USB croit être un simple client. Le logiciel système USB n'est pas tenu de réaliser la synchronisation entre plusieurs clients qui peuvent utiliser le même canal de flux. Les données présentées à un canal de flux se déplacent dans le canal dans un ordre séquentiel: premier entré, premier sorti.

Un canal de flux menant à un appareil est lié un seul point d'extrémité de l'appareil dans la direction appropriée (c'est-à-dire, correspondant à un jeton IN ou OUT tel que défini par la couche protocole). Le numéro du point d'extrémité de l'appareil situé dans la direction opposée peut être utilisé pour un autre canal de flux de l'appareil.

Les canaux de flux prennent en charge les types de transfert en bloc, isochrones et d'interruption, lesquels sont expliqués dans les sections ultérieures.

5.3.2.2 Canaux de messages

Les canaux de messages interagissent avec le point d'extrémité d'une manière différente de celle des canaux de flux. Tout d'abord, une requête est envoyée à l'appareil USB à partir de l'hôte. Cette requête est suivie du(des) transfert(s) de données dans la direction appropriée. Finalement, une étape Etat suit à un moment ultérieur. Afin de satisfaire au paradigme requête/données/état, les canaux de messages imposent une structure au flux de communication qui permet une identification et une communication fiables des commandes. Les canaux de messages permettent les flux de communication dans les deux directions, bien que le flux de communication puisse se faire principalement dans un sens. Le canal de contrôle par défaut est toujours un canal de messages.

Le logiciel système USB s'assure que plusieurs requêtes ne sont pas envoyées simultanément à un canal de messages. Un appareil doit servir une seule requête de message à la fois par canal de messages. Plusieurs clients logiciels situés sur l'hôte peuvent effectuer des requêtes par l'intermédiaire du canal de contrôle par défaut, mais elles sont

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

envoyées à l'appareil dans l'ordre premier venu, premier sorti. Un appareil peut contrôler le flux des informations pendant les étapes Données et Etat sur la base de sa capacité à répondre aux transactions de l'hôte (voir le Chapitre 8 pour de plus amples détails).

Un canal de messages ne reçoit généralement pas le message suivant en provenance de l'hôte avant que le traitement du message actuel n'ait été terminé au niveau de l'appareil. Toutefois, il existe des conditions d'erreur dans lesquelles un transfert de messages peut être interrompu par l'hôte et le canal de messages peut recevoir un nouveau transfert de message de façon prématurée (du point de vue de l'appareil). Du point de vue du logiciel manipulant un canal de messages, une erreur survenue sur une partie d'un IRP retire l'IRP actuel et tous les IRP mis en file d'attente. Le client logiciel qui a demandé l'IRP est notifié de l'achèvement de l'IRP avec une indication d'erreur appropriée.

Un canal de messages menant à un appareil nécessite un seul numéro de point d'extrémité dans les deux directions (jetons IN et OUT). L'USB n'autorise pas l'association d'un canal de messages à différents numéros de point d'extrémité pour chaque direction.

Les canaux de messages prennent en charge le type de transfert de contrôle qui est expliqué à la Section 5.5.

5.3.3 Trames et microtrames

L'USB établit une base de temps égale à 1 milliseconde appelée une trame sur un bus à vitesse intégrale/faible vitesse et une base de temps de 125 µs appelée une microtrame sur un bus à haute vitesse. Une (micro)trame peut contenir plusieurs transactions. Chaque type de transfert définit les transactions qui sont admises dans une (micro)trame pour un point d'extrémité. Les points d'extrémité isochrones et d'interruption reçoivent des opportunités d'accès au bus toutes les N (micro)trames. Les valeurs de N et d'autres détails concernant les transferts isochrones et d'interruption sont décrites aux Sections 5.6 et 5.7.

5.4 Types de transfert

L'USB transporte les données de transport dans un canal entre une mémoire tampon associée à un client logiciel de l'hôte et un point d'extrémité de l'appareil USB. Les données transportées par les canaux de messages sont acheminées sous la forme d'une structure définie par l'USB, mais l'USB autorise le transport des données de structure spécifique à l'appareil dans le cadre d'une charge utile des données de message définie par l'USB. L'USB définit également la mise en paquets des données déplacées sur le bus pour tous les canaux (flux ou messages), mais le formatage et l'interprétation des données transportées dans la charge utile de données d'une transaction de bus demeurent la responsabilité du logiciel client et de la fonction utilisant le canal. Toutefois, l'USB propose différents types de transfert qui sont optimisés pour assurer une meilleure correspondance avec les exigences de service du logiciel client et de la fonction utilisant le canal. Un IRP utilise une ou plusieurs transactions de bus pour déplacer les informations entre un client logiciel et sa fonction.

Chaque type de transfert détermine diverses caractéristiques du flux de communication, y compris les suivantes:

- Format de données imposé par l'USB
- Direction du flux de communication
- Contraintes de taille de paquet
- Contraintes d'accès au bus
- Contraintes de latence
- Séquences de données requises
- Traitement des erreurs

Les concepteurs d'un appareil USB choisissent les fonctionnalités des points d'extrémité de l'appareil. Lorsqu'un canal est établi pour un point d'extrémité, la plupart des caractéristiques de transfert du canal sont déterminées et demeurent identiques pendant toute la durée de vie

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

du canal. Les caractéristiques de transfert qui peuvent être modifiées sont décrites pour chaque type de transfert.

L'USB définit quatre types de transfert:

- Transferts de contrôle: communication requête/réponse initiée par le logiciel de l'hôte, par salves et non périodique, généralement utilisée pour les opérations de commande/état.
- Transferts isochrones: communication périodique et continue entre l'hôte et l'appareil, généralement utilisée pour les informations relatives à la durée. Ce type de transfert préserve également le concept de durée encapsulée dans les données. Cela n'implique pas, toutefois, le fait que la livraison qui requiert ces données ait toujours une criticité temporelle.
- Transferts d'interruption: communication à basse fréquence et à latence bornée.
- Transferts en bloc: communication non périodique et par salves de paquets de grande taille, généralement utilisée pour les données pouvant utiliser toute la bande passante disponible et également être retardées jusqu'à ce que la bande passante soit disponible.

Chaque type de transfert est décrit en détail dans les quatre sections majeures suivantes. Les données d'un IRP sont transportées par le champ de données du paquet de données, tel que décrit à la Section 8.3.4. Le Chapitre 8 décrit également les détails du protocole qui sont affectés par l'utilisation de chaque type de transfert particulier.

5.4.1 Exemples de calcul de tableau

Les sections suivantes décrivent chaque type de transfert USB. Ces sections contiennent des tableaux qui illustrent le nombre maximum de transactions qu'une (micro)trame peut contenir. Ces tableaux peuvent être utilisés pour déterminer le comportement de performance maximum possible pour un type de transfert spécifique. La performance réelle peut varier avec les détails de mise en application du système spécifique.

Chaque tableau indique:

- La surcharge de protocole requise pour le type de transfert spécifique (et la vitesse)
- Pour certaines tailles de charge utile de données échantillons:
  - La bande passante maximum possible soutenue dans ce cas
  - Le pourcentage d'une (micro)trame requis par chaque transaction
  - Le nombre maximum de transactions dans une (micro)trame pour le cas spécifique
  - Les octets restants dans une (micro)trame pas nécessaires pour le cas spécifique
  - Le nombre total d'octets de données transportés dans une seule (micro)trame pour le cas spécifique

Une transaction d'un type de transfert particulier nécessite généralement plusieurs paquets. La surcharge de protocole pour chaque transaction comprend:

- Un champ SYNC pour chaque paquet: 8 bits (vitesse intégrale/faible vitesse) ou 32 bits (haute vitesse)
- Un octet PID pour chaque paquet: inclut des bits (de contrôle) PID et PID inversé
- Une EOP pour chaque paquet: 3 bits (vitesse intégrale/faible vitesse) ou 8 bits (haute vitesse)
- Dans un paquet à jetons, le numéro de point d'extrémité, l'adresse de l'appareil et les champs CRC5 (16 bits au total).
- Dans un paquet de données, les champs CRC16 (16 bits au total)
- Dans un paquet de données, tous les champs de données (8 bits par octet)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- Pour les transactions à paquets multiples, l'intervalle inter-paquets ou le temps de retournement de bus requis.

Pour ces calculs, il est supposé qu'aucune insertion binaire n'est requise.

En utilisant le jeton OUT d'interruption à faible vitesse, il y a 5 paquets dans la transaction:

- Un paquet spécial PRE
- Un paquet à jetons
- Un paquet spécial PRE
- Un paquet de données
- Un paquet poignée de mains

Il existe un temps de retournement de bus entre les paquets de données et poignée de mains. La surcharge de protocole est donc:

5 SYNC, 5 PID, Point d'extrémité + CRC5, CRC16, 5 EOP et délai inter-paquets (un retournement de bus, 1 délai entre paquets et 2 temps de configuration de hub).

5.5 Transferts de contrôle

Les transferts de contrôle autorisent l'accès à différentes parties d'un appareil. Les transferts de contrôle sont conçus pour prendre en charge les flux de communication de type configuration/commande/état entre le logiciel client et sa fonction. Un transfert de contrôle est composé d'une transaction de bus Configuration qui déplace les informations de requête de l'hôte à la fonction, de zéro ou plusieurs transactions qui transmettent les données dans la direction indiquée par la transaction Configuration et d'une transaction Etat qui retourne les informations d'état de la fonction à l'hôte. La transaction Etat retourne "succès" lorsque le point d'extrémité a terminé avec succès le traitement de l'opération demandée. La Section 8.5.3 décrit les détails concernant les paquets, les transactions de bus et les séquences de transaction qui sont utilisés pour accomplir le transfert de contrôle. Le Chapitre 9 décrit les détails des codes de commandes USB définis.

Chaque appareil USB doit mettre en application le canal de contrôle par défaut en tant que canal de messages. Ce canal est utilisé par le logiciel système USB. Le canal de contrôle par défaut fournit l'accès aux informations de configuration, d'état et de contrôle de l'appareil USB. Une fonction peut, mais ne doit pas, fournir des points d'extrémité aux canaux de contrôle supplémentaires pour ses propres besoins de mise en application.

L'infrastructure de l'appareil USB (voir Chapitre 9) définit les requêtes normalisées, de classe d'appareil ou spécifiques au vendeur qui peuvent être utilisées pour manipuler l'état d'un appareil. Les descripteurs sont également définis et peuvent être utilisés pour contenir différentes informations sur l'appareil. Les transferts de contrôle fournissent un mécanisme de transport permettant d'accéder aux descripteurs de l'appareil et envoient des requêtes à un appareil pour manipuler son comportement.

Les transferts de contrôle effectuent des transports uniquement par l'intermédiaire des canaux de messages. Par conséquent, les flux de données qui utilisent les transferts de contrôle doivent satisfaire aux définitions des structures de données USB, telles que décrites à la Section 5.5.1.

Le système USB fait "de son mieux" pour prendre en charge la livraison de transferts de contrôle entre l'hôte et les appareils. Une fonction et son logiciel client ne peuvent pas demander une fréquence d'accès au bus ou une bande passante spécifique aux transferts de contrôle. Le logiciel système USB peut limiter l'accès au bus et la bande passante qu'un appareil pourrait souhaiter pour les transferts de contrôle. Ces limites sont définies aux Sections 5.5.3 et 5.5.4.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.5.1 Format de données des transferts de contrôle

Le paquet Configuration a une structure définie par l'USB qui correspond à l'ensemble minimum de commandes requis pour permettre la communication entre l'hôte et un appareil. La définition de la structure autorise les extensions spécifiques au vendeur pour les commandes spécifiques à l'appareil. Les transactions Données qui suivent la Configuration ont une structure définie par l'USB, à l'exception des cas où elles transportent des informations spécifiques au vendeur. La transaction Etat a également une structure définie par l'USB. Les définitions Configuration/Données spécifiques au transfert de contrôle sont décrites à la Section 8.5.3 et au Chapitre 9.

5.5.2 Direction du transfert de contrôle

Les transferts de contrôle sont pris en charge par l'intermédiaire du flux de communication bidirectionnel sur les canaux de messages. En conséquence, lorsqu'un canal de contrôle est configuré, il utilise le point d'extrémité d'entrée et de sortie avec le numéro de point d'extrémité spécifié.

5.5.3 Contraintes liées à la taille des paquets des transferts de contrôle

Un point d'extrémité des transferts de contrôle spécifie la taille maximum de la charge utile des données que le point d'extrémité peut accepter du bus ou lui transmettre. Les tailles maximum permises de la charge utile de données des transferts de contrôle pour les appareils à vitesse intégrale s'élèvent à 8, 16, 32 ou 64 octets; pour les appareils à haute vitesse, elle s'élève à 64 octets et pour les appareils à faible vitesse à 8 octets. Cette taille maximum s'applique aux charges utiles de données des paquets Données suivis d'un paquet Configuration; c'est-à-dire, la taille spécifiée est celle du champ de données du paquet, tel que défini au Chapitre 8, sans inclure les autres informations requises par le protocole. Un paquet Configuration a toujours une taille de huit octets. Un canal de contrôle (y compris le canal de contrôle par défaut) utilise toujours sa valeur *wMaxPacketSize* pour les charges utiles de données.

Un point d'extrémité indique la valeur de la taille maximum de ses charges utiles de données dans les informations de sa configuration. L'USB n'exige pas que les charges utiles de données transmises aient exactement la taille maximum; c'est-à-dire, si une charge utile de données est inférieure à la taille maximum, il n'est pas nécessaire qu'elle soit complétée pour atteindre la taille maximum.

Tous les contrôleurs hôtes doivent prendre en charge les tailles maximales de charge utile de données de 8, 16, 32 et 64 octets pour les points d'extrémité de contrôle à vitesse intégrale, uniquement les tailles maximales de charge utile de données de 8 octets pour les points d'extrémité de contrôle à faible vitesse et uniquement les tailles maximales de charge utile de données de 64 octets pour les points d'extrémité de contrôle à haute vitesse. Aucun contrôleur hôte n'est tenu de prendre en charge des tailles maximales de charge utile de données plus grandes ou plus petites.

Afin de déterminer la taille maximum des paquets pour le canal de contrôle par défaut, le logiciel système USB lit le descripteur d'appareil. L'hôte lit les huit premiers octets du descripteur d'appareil. L'appareil répond toujours avec au moins ces octets initiaux dans un seul paquet. Une fois que l'hôte a lu la partie initiale du descripteur d'appareil, il est sûr d'avoir lu le champ *wMaxPacketSize* de ce canal par défaut (7ème octet du descripteur d'appareil). Il permet ensuite d'allouer la taille correcte à toutes les transactions ultérieures. Pour tous les autres points d'extrémité de contrôle, la taille maximum de charge utile de données est connue après la configuration afin que le logiciel système USB puisse s'assurer qu'aucune charge utile de données ne soit envoyée au point d'extrémité dont la taille est supérieure à la taille prise en charge.

Un point d'extrémité doit toujours transmettre les charges utiles de données avec un champ de données inférieur ou égal au champ *w*MaxPacketSize du point d'extrémité (voir Chapitre 9). Lorsqu'un transfert de contrôle implique une quantité de données plus importante que la taille maximum pouvant être acceptée pour une charge utile de données, toutes les charges utiles

IECNORM.COM : Click to view the full PDF of iec 62680-1:2013
Withdrawn

de données doivent avoir la taille maximum, à l'exception de la dernière charge utile de données qui contient les données restantes.

L'étape Données d'un transfert de contrôle à partir d'un point d'extrémité vers l'hôte s'achève lorsque le point d'extrémité effectue l'une des actions suivantes:

- Il a transféré la quantité exacte de données spécifiée pendant l'étape Configuration
- Il transfère un paquet avec une taille de charge utile inférieure à la valeur de *wMaxPacketSize* ou transfère un paquet de longueur nulle

Une fois que l'étape Données est terminée, le contrôleur hôte passe à l'étape Etat plutôt que de continuer avec une autre transaction de données. Si le contrôleur hôte ne passe pas à l'étape Etat une fois que l'étape Données est terminée, le point d'extrémité arrête le canal comme souligné à la Section 5.3.2. Si une charge utile de données d'une taille plus grande que prévu provient du point d'extrémité, l'IRP du transfert de contrôle est interrompu/retiré.

L'étape Données d'un transfert de contrôle allant de l'hôte vers un point d'extrémité est achevée lorsque toutes les données ont été transférées. Si le point d'extrémité reçoit une charge utile de données d'une taille plus grande que prévu provenant de l'hôte, il arrête le canal.**Contraintes liées à l'accès au bus des transferts de contrôle**

Les transferts de contrôle peuvent être utilisés par des appareils USB à haute, vitesse intégrale et faible vitesse.

Un point d'extrémité n'a aucun moyen d'indiquer la fréquence souhaitée d'accès au bus pour un canal de contrôle. L'USB équilibre les exigences d'accès au bus de tous les canaux de contrôle et des IRP spécifiques qui sont en attente pour fournir la meilleure livraison possible entre le logiciel client et les fonctions.

L'USB exige qu'une partie de chaque (micro)frame soit réservée à l'accès des transferts de contrôle pour l'utilisation suivante:

- Si les transferts de contrôle tentés (de manière dépendante de la mise en application) consomment moins de 10 % du temps de trames pour les points d'extrémité à vitesse intégrale/faible vitesse ou moins de 20 % d'une microtrame pour des points d'extrémité à haute vitesse, le temps restant peut être utilisé pour prendre en charge les transferts en bloc (voir Section 5.8).
- Un transfert de contrôle qui a été tenté et doit être tenté de nouveau peut être tenté dans la (micro)trame actuelle ou ultérieure; c'est-à-dire, il n'est pas nécessaire qu'il soit tenté de nouveau dans la même (micro)trame.
- S'il existe un nombre de transferts de contrôle plus important que le temps réservé, mais qu'il existe une (micro)trame supplémentaire qui n'est pas utilisée par des transferts isochrones ou d'interruption, un contrôleur hôte peut déplacer les transferts de contrôle supplémentaires dès qu'ils sont disponibles.
- S'il y a un nombre trop important de transferts de contrôle en attente pour le temps de (micro)trame disponible, les transferts de contrôle sont sélectionnés pour être déplacés sur le bus de manière appropriée.
- S'il existe des transferts de contrôle en attente pour des points d'extrémité multiples, les transferts de contrôle des différents points d'extrémité sont sélectionnés selon la politique d'accès équitable dépendante de la mise en application du contrôleur hôte.
- Il convient qu'une transaction de transfert de contrôle qui est fréquemment tentée ne soit pas censée consommer une part excessive du temps de bus.

Les points de contrôle à haute vitesse doivent prendre en charge le protocole de contrôle des flux PING pour les transactions OUT. Les détails de ce protocole sont décrits à la Section 8.5.1.

Ces exigences autorisent que les transferts de contrôle entre l'hôte et les appareils soient régulièrement déplacés sur le bus "du mieux possible".

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Le logiciel système USB peut, à sa discrétion, varier le débit des transferts de contrôle vers un point d'extrémité particulier. Un point d'extrémité et son logiciel client ne peuvent pas préciser un débit spécifique de service pour les transferts de contrôle. Un point d'extrémité de contrôle peut distinguer zéro ou plusieurs transferts dans une seule (micro)trame. Le temps de bus mis à la disposition d'un client logiciel et de son point d'extrémité peut être changé au fur et à mesure que d'autres appareils sont insérés ou supprimés du système ou également au fur et à mesure que des transferts de contrôle sont requis par d'autres points d'extrémité d'appareil.

La fréquence de bus et la synchronisation des (micro)trames limitent le nombre maximum de transferts de contrôle réussis dans le cadre d'une (micro)trame d'un système USB. Pour les bus à vitesse intégrale/faible vitesse, le nombre de transferts de contrôle réussis par trame est limité à moins de 29 charges utiles de données à vitesse intégrale de huit octets ou à moins de quatre charges utiles de données à faible vitesse de huit octets. Pour les bus à haute vitesse, le nombre de transferts de contrôle est limité à moins de 32 charges utiles de données à haute vitesse de 64 octets par microtrame.

Le Tableau 5-1 affiche les informations concernant les paquets à faible vitesse de différentes tailles et le nombre maximum de paquets possibles dans une trame. Le tableau ne contient pas la surcharge associée à l'insertion binaire.

Tableau 5-1. Limites des transferts de contrôle à faible vitesse

	Surcharge de protocole (63 octets)		(15 octets SYNC, 15 octets PID, 6 points d'extrémité + octets CRC, 6 octets CRC, 8 octets de données Configuration et un délai inter-paquets de 13 octets (EOP, etc.))			
	Charge utile de données	Bande passante max. (octets/seconde)	Bande passante de trame par transfert	Transferts max.	Octets restants	Données utiles en octets/trame
	1	3 000	26 %	3	40	3
	2	6 000	27 %	3	37	6
	4	12 000	28 %	3	31	12
	8	24 000	30 %	3	19	24
Max		187 500				187

Pour toutes les vitesses, à cause d'un transfert de contrôle composé de plusieurs paquets, les paquets peuvent être répartis sur plusieurs (micro)trames afin de répartir le temps de bus requis sur plusieurs (micro)trames.

La réservation de 10 % des trames pour les transferts non périodiques à vitesse intégrale/faible vitesse signifie que, dans un système avec un temps de bus entièrement alloué, tous les transferts de contrôle à vitesse intégrale du système se disputent une valeur nominale de trois transferts de contrôle par trame. Etant donné que le système USB utilise des transferts de contrôle à des fins de configuration en plus des autres transferts de contrôle pouvant être requis par un autre logiciel client, il convient qu'un client logiciel donné et sa fonction ne s'attendent pas à pouvoir utiliser la bande passante tout entière pour leurs propres besoins de contrôle. Les contrôleurs hôtes sont également libres de déterminer la manière dont les transactions de bus individuelles des transferts de contrôle spécifiques sont déplacées sur le bus dans le cadre et sur les trames. Un point d'extrémité pourrait voir toutes les transactions de bus d'un transfert de contrôle dans la même trame ou réparties sur plusieurs trames non contiguës. Pour diverses raisons de mise en application, un contrôleur hôte peut ne pas être en mesure de fournir le nombre maximum théorique de transferts de contrôle par trame.

Pour les points d'extrémité à haute vitesse, la réservation de 20 % de microtrames pour des transferts non périodiques signifie que tous les transferts de contrôle à haute vitesse se disputent un nombre nominal de six transferts de contrôle par microtrame. Les transferts de

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

contrôle à haute vitesse se disputent le temps de microtrame avec des transactions divisées (voir les Sections 11.15-11.21 pour plus d'informations concernant les transactions divisées) pour les transferts de contrôle à vitesse intégrale et faible vitesse. Les transferts de contrôle à vitesse intégrale et faible vitesse se disputent le même temps de trame disponible. Toutefois, les transferts de contrôle à haute vitesse de certains points d'extrémité peuvent se produire simultanément avec les transferts de contrôle à vitesse intégrale et faible vitesse d'autres points d'extrémité. Les transferts de contrôle à faible vitesse prennent tout simplement plus de temps.

Le Tableau 5-2 affiche les informations concernant les transferts de contrôle à vitesse intégrale de différentes tailles et le nombre maximum de transferts possibles dans une trame. Ce tableau a été généré en émettant l'hypothèse qu'il existe une transaction de l'étape Données et que l'étape Données a une phase d'état de longueur nulle. Le tableau illustre la puissance possible de deux charges utiles de données dont la taille est inférieure ou égale à la taille maximum permise pour les charges utiles de données. Le tableau ne contient pas la surcharge associée à l'insertion binaire.

Tableau 5-2. Limites des transferts de contrôle à vitesse intégrale

	Surcharge de protocole (45 octets)		**(9 octets SYNC, 9 octets PID, 6 points d'extrémité + octets CRC, 6 octets CRC, 8 octets de données Configuration et un délai inter-paquets de 7 octets (EOP, etc.))**			
	Charge utile de données	**Bande passante max. (octets/seconde)**	**Bande passante de trame par transfert**	**Transferts max.**	**Octets restants**	**Données utiles en octets/trame**
	1	32 000	3 %	32	23	32
	2	62 000	3 %	31	43	62
	4	120 000	3 %	30	30	120
	8	224 000	4 %	28	16	224
	16	384 000	4 %	24	36	384
	32	608 000	5 %	19	37	608
	64	832 000	7 %	13	83	832
Max		1 500 000				1 500

Le Tableau 5-3 affiche les informations concernant les transferts de contrôle à haute vitesse de différentes tailles et le nombre maximum de transferts possibles dans une microtrame. Ce tableau a été généré en émettant l'hypothèse qu'il existe une transaction de l'étape Données et que l'étape Données a une étape d'état de longueur nulle. Le tableau illustre la puissance possible de deux charges utiles de données dont la taille est inférieure ou égale à la taille maximum permise pour les charges utiles de données. Le tableau ne contient pas la surcharge associée à l'insertion binaire.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 5-3. Limites des transferts de contrôle à haute vitesse

	Surcharge de protocole (173 octets)		(sur la base de 480 Mb/s et un intervalle inter-paquets de 8 bits, retournement de bus de 88 bits min., sync de 32 bits, EOP de 8 bits: (9x4 octets SYNC, 9 octets PID, 6 EP/ADDR+CRC, 6 CRC16, 8 données Configuration, délai inter-paquets de 9x(1+11) octets (EOP, etc.))			
	Charge utile de données	Bande passante max. (octets/seconde)	Bande passante de trame par transfert	Transferts max.	Octets restants	Données utiles en octets/microtrame
	1	344 000	2 %	43	18	43
	2	672 000	2 %	42	150	84
	4	1 344 000	2 %	42	66	168
	8	2 624 000	2 %	41	79	328
	16	4 992 000	3 %	39	129	624
	32	9 216 000	3 %	36	120	1 152
	64	15 872 000	3 %	31	153	1 984
Max		60 000 000				7 500

5.5.5 Séquences de données des transferts de contrôle

Les transferts de contrôle exigent qu'une transaction de bus Configuration soit envoyée de l'hôte à l'appareil pour décrire le type d'accès de contrôle qu'il convient que l'appareil effectue. La transaction Configuration est suivie de zéro ou plusieurs transactions Données de contrôle qui transportent les informations spécifiques de l'accès requis. Finalement, une transaction Etat termine le transfert de contrôle et autorise le point d'extrémité à retourner l'état du transfert de contrôle au logiciel client. Une fois que la transaction Etat du transfert de contrôle est terminée, l'hôte peut avancer jusqu'au transfert de contrôle suivant du point d'extrémité. Tel que décrit à la Section 5.5.4, chaque transaction de contrôle et le transfert de contrôle suivant sont déplacés sur le bus à un certain moment défini par la mise en application du contrôleur hôte.

Le point d'extrémité peut être occupé pour une durée spécifique à l'appareil pendant les transactions Données et Etat du transfert de contrôle. Durant les moments où le point d'extrémité indique qu'il est occupé (voir les Chapitres 8 et 9 pour les détails), l'hôte tente de nouveau la transaction à un moment ultérieur.

Si une transaction Configuration est reçue par un point d'extrémité avant qu'un transfert de contrôle initié précédemment soit terminé, l'appareil doit interrompre le transfert/l'opération actuel(le) et traiter la nouvelle transaction Configuration de contrôle. Il convient normalement qu'une transaction Configuration ne soit pas envoyée avant l'achèvement d'un transfert de contrôle antérieur. Toutefois, si un transfert est interrompu, par exemple à cause d'erreurs sur le bus, l'hôte peut envoyer la transaction Configuration suivante de façon prématurée du point de vue du point d'extrémité.

Une fois qu'une condition d'arrêt est constatée ou qu'une erreur est détectée par l'hôte, un point de contrôle peut récupérer en acceptant le PID Configuration suivant; c'est-à-dire, les actions de récupération par l'intermédiaire d'un autre canal ne sont pas requises pour les points d'extrémité de contrôle. Pour le canal de contrôle par défaut, une réinitialisation d'appareil est nécessaire en dernier recours pour supprimer la condition d'arrêt ou d'erreur si le PID Configuration suivant n'est pas accepté.

L'USB fournit une détection robuste des erreurs et la récupération/retransmission des erreurs qui se sont produites pendant les transferts de contrôle. Les transmetteurs et les récepteurs peuvent rester synchronisés en ce qui concerne leur emplacement dans un transfert de contrôle et récupèrent avec un minimum d'effort. La retransmission des paquets Données et Etat peut être détectée par un récepteur par l'intermédiaire des indicateurs de nouvelle tentative de données du paquet. Un transmetteur peut déterminer de façon fiable que son récepteur correspondant a accepté avec succès un paquet transmis par les informations retournées suite à une poignée de mains avec le paquet. Le protocole autorise la distinction entre un paquet retransmis de son paquet d'origine, excepté pour un paquet Configuration de

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

contrôle. Les paquets Configuration peuvent être retransmis à cause d'une erreur de transmission; toutefois, les paquets Configuration ne peuvent pas indiquer si un paquet représente une transmission d'origine ou tentée de nouveau.

5.6 Transferts isochrones

Dans les environnements non-USB, les transferts isochrones ont l'implication générale des transferts à débit constant et tolérants aux erreurs. Dans l'environnement USB, la demande d'un type de transfert isochrone fournit au demandeur les éléments suivants:

- Un accès garanti à la bande passante avec une latence limitée
- Un débit de données constant garanti par l'intermédiaire d'un canal tant que les données sont fournies au canal
- Dans le cas d'une défaillance de livraison due aux erreurs, aucune nouvelle tentative pour livrer les données

Bien que le type de transfert isochrone soit conçu pour prendre en charge les sources et les destinations isochrones, le logiciel qui utilise ce type de transfert ne doit pas être isochrone afin d'utiliser le type de transfert. La Section 5.12 présente plus de détails sur les considérations spéciales pour le traitement des données isochrones de l'USB.

5.6.1 Format de données des transferts isochrones

L'USB n'impose aucune structure au contenu de données des flux de communication pour les canaux isochrones.

5.6.2 Direction des transferts isochrones

Un canal isochrone est un canal de flux qui est, par conséquent, toujours unidirectionnel. La description d'un point d'extrémité identifie si le débit de communication d'un canal isochrone donné se dirige à l'intérieur ou à l'extérieur de l'hôte. Si un appareil requiert un débit de communication isochrone bidirectionnel, deux canaux isochrones doivent être utilisés, un dans chaque direction.

5.6.3 Contraintes liées à la taille des paquets des transferts isochrones

Un point d'extrémité d'une configuration donnée d'un canal isochrone spécifie la taille maximale de charge utile de données qu'il peut transmettre ou recevoir. Le logiciel système USB utilise ces informations pendant la configuration pour s'assurer qu'il y a un temps de bus suffisant pour s'adapter à la charge utile de données maximum de chaque (micro)trame. Si le temps de bus est suffisant pour la charge utile de données maximum, la configuration est réalisée; dans le cas contraire, la configuration n'est pas réalisée.

L'USB limite la taille maximum de charge utile de données à 1 023 octets pour chaque point v isochrone à vitesse intégrale. Les points d'extrémité à haute vitesse peuvent atteindre des charges utiles de données jusqu'à 1 024 octets. Un point d'extrémité à haute vitesse et avec une large bande passante spécifie s'il nécessite deux ou trois transactions par microtrame. Le Tableau 5-4 affiche les informations concernant les transactions isochrones à vitesse intégrale de différentes tailles et le nombre maximum de transactions possibles dans une trame. Le tableau est grisé pour indiquer qu'un point d'extrémité isochrone à vitesse intégrale (avec une taille *wMaxpacket* différente de zéro) ne doit pas faire partie d'un paramètre d'interface par défaut. Le tableau ne contient pas la surcharge associée à l'insertion binaire.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 5-4. Limites des transactions isochrones à vitesse intégrale

	Surcharge de protocole (9 octets)		(2 octets SYNC, 2 octets PID, 2 octets Point d'extrémité + CRC, 2 octets CRC et un délai inter-paquets de 1 octet)			
	Charge utile de données	Bande passante max. (octets/seconde)	Bande passante de trame par transfert	Transferts max.	Octets restants	Données utiles en octets/trame
	1	150 000	1 %	150	0	150
	2	272 000	1 %	136	4	272
	4	460 000	1 %	115	5	460
	8	704 000	1 %	88	4	704
	16	960 000	2 %	60	0	960
	32	1 152 000	3 %	36	24	1 152
	64	1 280 000	5 %	20	40	1 280
	128	1 280 000	9 %	10	130	1 280
	256	1 280 000	18 %	5	175	1 280
	512	1 024 000	35 %	2	458	1 024
	1 023	1 023 000	69 %	1	468	1 023
Max		1 500 000				1 500

Le Tableau 5-5 affiche les informations concernant les transactions isochrones à haute vitesse de différentes tailles et le nombre maximum de transactions possibles dans une microtrame. Le tableau est grisé pour indiquer qu'un point d'extrémité isochrone à haute vitesse ne doit pas faire partie d'un paramètre d'interface par défaut. Le tableau ne contient pas la surcharge associée à l'insertion binaire.

Tout point d'extrémité d'un canal isochrone peut ne pas avoir exactement la taille maximum spécifiée pour le point d'extrémité. La taille d'une charge utile de données est déterminée par le transmetteur (logiciel client ou fonction) et peut varier selon les besoins d'une transaction à l'autre. L'USB s'assure que la taille, quelle qu'elle soit, qui est présentée au contrôleur hôte est délivrée sur le bus. La taille actuelle d'une charge utile de données est déterminée par le transmetteur de données et peut être inférieure à la taille maximum négociée à l'avance. Les erreurs de bus peuvent changer la taille actuelle des paquets détectée par le récepteur. Toutefois, ces erreurs peuvent être détectées par le CRC des données ou en sachant que le récepteur a une taille approximativement égale à celle prévue pour chaque transaction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 5-5. Limites des transactions isochrones à haute vitesse

	Surcharge de protocole		(sur la base de 480 Mb/s et un intervalle inter-paquets de 8 bits, retournement de bus de 88 bits min., sync de 32 bits, EOP de 8 bits: (2x4 octets SYNC, 2 octets PID , 2 EP/ADDR+addr+CRC5, 2 CRC16 et délai inter-paquets de 2x(1+11) octets (EOP, etc.))			
	Charge utile de données	Bande passante max. (octets/seconde)	Bande passante de trame par transfert	Transferts max.	Octets restants	Données utiles en octets/microtrame
	1	1 536 000	1 %	192	12	192
	2	2 992 000	1 %	187	20	374
	4	5 696 000	1 %	178	24	712
	8	10 432 000	1 %	163	2	1 304
	16	17 664 000	1 %	138	48	2 208
	32	27 392 000	1 %	107	10	3 424
	64	37 376 000	1 %	73	54	4 672
	128	46 080 000	2 %	45	30	5 760
	256	51 200 000	4 %	25	150	6 400
	512	53 248 000	7 %	13	350	6 656
	1 024	57 344 000	14 %	7	66	7 168
	2 048	49 152 000	28 %	3	1 242	6 144
	3 072	49 152 000	41 %	2	1 280	6 144
Max		60 000 000				7 500

Tous les paramètres d'interface par défaut de l'appareil ne doivent inclure aucun point d'extrémité isochrone avec des tailles de charge utile de données non nulles (spécifiées par l'intermédiaire du champ *wMaxPacketSize* dans le descripteur de point d'extrémité). Les paramètres d'interface alternatifs peuvent spécifier des tailles de charge utile de données non nulles pour les points d'extrémité isochrones. Si les points de terminaison isochrones ont une grande taille de charge utile de données, il est recommandé que des configurations alternatives supplémentaires ou des paramètres d'interface soient utilisés pour spécifier un intervalle de tailles de charge utile de données. Cela augmente les chances que l'appareil puisse être utilisé avec succès en combinaison avec d'autres appareils USB.

5.6.4 Contraintes liées à l'accès au bus des transferts isochrones

Les transferts isochrones peuvent être utilisés uniquement par les appareils à vitesse intégrale et haute vitesse.

L'USB exige que 90 % max. de chaque trame au plus soient alloués aux transferts périodiques (isochrones et d'interruption) des points d'extrémité à vitesse intégrale. Les points d'extrémité à haute vitesse peuvent allouer 80 % max. d'une microtrame pour les transferts périodiques.

Un point d'extrémité isochrone doit spécifier la période d'accès au bus dont il a besoin. Les points d'extrémité à vitesse intégrale/haute vitesse doivent spécifier une période désirée équivalente à ($2^{bInterval-1}$) x F, où bInterval représente l'intervalle de 1 à (et y compris) 16 et F représente 125 µs pour la haute vitesse et 1 ms pour la vitesse intégrale. Cela permet aux transferts isochrones à vitesse intégrale/haute vitesse d'avoir des débits inférieurs à une transaction par (micro)trame. Cependant, un point d'extrémité isochrone doit être préparé à traiter des débits d'interrogation plus rapides que celui spécifié. Un hôte ne doit pas émettre plus d'une transaction dans une (micro)trame pour un point d'extrémité isochrone, à moins que le point d'extrémité soit à haute vitesse et qu'il ait une large bande passante (voir ci-dessous). Un point d'extrémité IN isochrone doit retourner un paquet de longueur nulle à chaque fois que des données sont demandées dans un intervalle plus rapide que celui spécifié et que les données sont indisponibles.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Un point d'extrémité à haute vitesse peut déplacer jusqu'à 3 072 octets par microtrame (ou 192 Mb/s). Un point d'extrémité isochrone à haute vitesse qui nécessite plus de 1 024 octets par seconde est appelé un point d'extrémité à large bande passante. Un point d'extrémité à large bande passante utilise plusieurs transactions par microtrame. Un point d'extrémité à large bande passante doit spécifier une période de 1x125 µs (c'est-à-dire, une valeur *bInterval* égale à 1). Voir la Section 5.9 pour plus d'informations concernant les détails sur les transactions multiples par microtrame pour les points d'extrémité à haute vitesse et à large bande passante.

Les erreurs du bus ou les délais de programmation du système d'exploitation du logiciel client peuvent aboutir à un transfert nul de paquets dans une (micro)trame. Dans ce cas, il convient qu'une indication d'erreur soit retournée comme état au logiciel client. Un appareil peut également détecter cette situation en suivant les jetons SOF et en signalant toute perturbation du modèle spécifié de période d'accès au bus.

La fréquence de bus et la synchronisation de (micro)trame limitent le nombre maximum de transactions isochrones réussies dans le cadre d'une (micro)trame pour un système USB à moins de 151 charges utiles de données à vitesse intégrale d'un octet ou à moins de 193 charges utiles de données à haute vitesse d'un octet. Pour diverses raisons de mise en application, un contrôleur hôte peut ne pas être en mesure de fournir le nombre maximum théorique de transactions isochrones par (micro)trame.

5.6.5 Séquences de données des transferts isochrones

Les transferts isochrones ne prennent pas en charge la retransmission des données en réponse aux erreurs du bus. Un récepteur peut déterminer si une erreur de transmission s'est produite. Le protocole USB à deux niveaux n'autorise pas que les poignées de mains soient retournées au transmetteur d'un canal isochrone. Généralement, les poignées de mains devraient être retournées pour indiquer au transmetteur si un paquet a été reçu avec succès ou non. Pour les transferts isochrones, l'opportunité est plus importante que l'exactitude/retransmission et, selon les faibles taux d'erreur escomptés sur le bus, le protocole est optimisé avec l'hypothèse que les transferts ont abouti avec succès. Les récepteurs isochrones peuvent déterminer s'ils ont omis des données pendant une (micro)trame. De même, un récepteur peut déterminer la quantité de données manquante. La Section 5.12 décrit ces mécanismes USB dans de plus amples détails.

Un point d'extrémité de transferts isochrones ne s'arrête jamais car il n'existe aucune poignée de mains pour signaler une condition d'arrêt. Les erreurs sont indiquées comme l'état associé à l'IRP d'un transfert isochrone, mais le canal isochrone n'est pas arrêté en cas d'erreur. Si aucune erreur n'est détectée, l'hôte continue le traitement des données associées à la (micro)trame suivante du transfert. Seule une détection limitée des erreurs est possible car le protocole des transactions isochrones n'autorise pas les poignées de mains par transaction.

5.7 Transferts d'interruption

Le type de transfert d'interruption est conçu pour prendre en charge les appareils qui doivent rarement envoyer ou recevoir des données, mais qui ont des périodes de service limitées. La requête d'un canal avec un type de transfert d'interruption fournit au demandeur les éléments suivants:

- Période de service maximum garantie pour le canal
- Nouvelles tentatives de transfert à la période suivante, dans le cas d'une défaillance de livraison occasionnelle due aux erreurs sur le bus

5.7.1 Format de données des transferts d'interruption

L'USB n'impose aucune structure au contenu de données des flux de communication pour les canaux d'interruption.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.7.2 Direction des transferts d'interruption

Un canal d'interruption est un canal de flux qui est, par conséquent, toujours unidirectionnel. La description d'un point d'extrémité identifie si le débit de communication d'un canal d'interruption donné se dirige à l'intérieur ou à l'extérieur de l'hôte.

5.7.3 Contraintes liées à la taille des paquets des transferts d'interruption

Un point d'extrémité d'un canal d'interruption spécifie la taille maximale de charge utile de données qu'il peut transmettre ou recevoir. La taille maximum permise pour les charges utiles de données d'interruption s'élève à 64 octets ou est inférieure à celle de la vitesse intégrale. Les points d'extrémité à haute vitesse peuvent avoir une taille maximum permise pour les charges utiles de données jusqu'à 1 024 octets. Un point d'extrémité à haute vitesse et avec une large bande passante spécifie s'il nécessite deux ou trois transactions par microtrame. La taille maximum permise pour les charges utiles de données des appareils à faible vitesse a une limite inférieure ou égale à huit octets. Cette taille maximum s'applique aux charges utiles de données des paquets de données; c'est-à-dire, la taille spécifiée est celle du champ de données du paquet, tel que défini au Chapitre 8, sans inclure d'autres informations requises par le protocole. L'USB n'exige pas que les paquets de données transmises aient exactement la taille maximum; c'est-à-dire, si un paquet de données est inférieur à la taille maximum, il n'est pas nécessaire de le compléter pour atteindre la taille maximum.

Tous les contrôleurs hôtes doivent prendre en charge les tailles maximales de charge utile de données allant de 0 à 64 octets pour les points d'extrémité d'interruption à vitesse intégrale, de 0 à 8 octets pour les points d'extrémité d'interruption à faible vitesse et de 0 à 1 024 octets pour les points d'extrémité d'interruption à haute vitesse. Voir la Section 5.9 pour plus d'informations concernant les détails sur les transactions multiples par microtrame pour les points d'extrémité à haute vitesse et à large bande passante. Aucun contrôleur hôte n'est tenu de prendre en charge les tailles maximales de charge utile de données plus grandes.

Le logiciel système USB détermine la taille maximum de charge utile de données utilisée pour un canal d'interruption pendant la configuration de l'appareil. Cette taille demeure constante durant la durée de vie d'une configuration de l'appareil. Le logiciel système USB utilise la taille maximum de charge utile de données déterminée pendant la configuration pour s'assurer qu'il y a un temps de bus suffisant pour s'adapter cette taille maximum de charge utile de données durant sa période d'affectation. Si le temps de bus est suffisant, le canal est établi; dans le cas contraire, le canal n'est pas établi. Toutefois, la taille actuelle d'une charge utile de données est encore déterminée par le transmetteur de données et peut être inférieure à la taille maximum.

Un point d'extrémité doit toujours transmettre les charges utiles de données avec un champ de données inférieur ou égal au champ *w*MaxPacketSize du point d'extrémité. Un appareil peut déplacer les données par l'intermédiaire d'un canal d'interruption plus large que *wMaxPacketSize*. Un client logiciel peut accepter ces données par l'intermédiaire d'un IRP du transfert d'interruption qui nécessite plusieurs transactions de bus sans requérir une notification d'achèvement de l'IRP par transaction. Cela peut se faire en spécifiant une mémoire tampon qui puisse contenir la taille de données souhaitée. La taille de la mémoire tampon est un multiple de *w*MaxPacketSize avec un reste. Le point d'extrémité doit transférer chaque transaction, à l'exception de la dernière en tant que *w*MaxPacketSize, et la dernière transaction représente le reste. Les transactions de données multiples sont déplacées sur le bus pendant la période établie pour le canal.

Lorsqu'un transfert d'interruption implique une quantité de données plus importante que la taille maximum actuellement établie qui peut être acceptée pour une charge utile de données, toutes les charges utiles de données doivent avoir la taille maximum, à l'exception de la dernière charge utile de données qui contient les données restantes. Un transfert d'interruption s'achève lorsque le point d'extrémité effectue une des actions suivantes:

- Il a transféré la quantité exacte de données prévues.
- Il transfère un paquet avec une taille de charge utile inférieure à la valeur de *wMaxPacketSize* ou transfère un paquet de longueur nulle.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Lorsqu'un transfert d'interruption est terminé, le contrôleur hôte retire l'IRP actuel et avance à l'IRP suivant. Si une charge utile de données reçue est plus grande que prévue, l'IRP d'interruption est interrompu/retiré et le canal bloque les IRP antérieurs jusqu'à ce que l'état soit corrigé ou reconnu.

Tous les paramètres d'interface par défaut de l'appareil à haute vitesse ne doivent inclure aucun point d'extrémité d'interruption avec une taille de charge utile de données (spécifiée par l'intermédiaire du champ *wMaxPacketSize*) supérieure à 64 octets. Les paramètres d'interface alternatifs peuvent spécifier des tailles de charge utile de données plus grandes pour les points d'extrémité d'interruption. Si les points d'extrémité d'interruption ont une grande taille de charge utile de données, il est recommandé que des configurations supplémentaires ou des paramètres d'interface alternatifs soient utilisés pour spécifier un intervalle de tailles de charge utile de données. Cela augmente les chances que l'appareil puisse être utilisé avec succès en combinaison avec d'autres appareils USB.

5.7.4 Contraintes liées à l'accès au bus des transferts d'interruption

Les transferts d'interruption peuvent être utilisés par des appareils à faible, vitesse intégrale et haute vitesse. Les points d'extrémité à haute vitesse peut allouer tout au plus 80 % d'une microtrame pour les transferts périodiques. L'USB exige que 90 % de chaque trame tout au plus soient alloués aux transferts périodiques (isochrones et d'interruption) à vitesse intégrale/faible vitesse.

La fréquence de bus et la synchronisation de (micro)trame limitent le nombre maximum de transactions d'interruption réussies dans le cadre d'une (micro)trame pour un système USB à moins de 108 charges utiles de données à vitesse intégrale d'un octet, à moins de 10 charges utiles de données à faible vitesse d'un octet ou à moins de 134 charges utiles de données à haute vitesse d'un octet. Pour diverses raisons de mise en application, un contrôleur hôte peut ne pas être en mesure de fournir le nombre maximum ci-dessus de transactions d'interruption par (micro)trame.

Le Tableau 5-6 affiche les informations concernant différentes transactions d'interruption à faible vitesse et le nombre maximum de transactions possibles dans une trame. Le Tableau 5-7 affiche des informations similaires pour les transactions d'interruption à vitesse intégrale. Le Tableau 5-8 affiche des informations similaires pour les transactions d'interruption à haute vitesse. La portion grisée du Tableau 5-8 indique les tailles de charge utile de données d'un point d'extrémité d'interruption à haute vitesse qui ne doivent pas faire partie d'un paramètre d'interface par défaut. Les tableaux ne contiennent pas la surcharge associée à l'insertion binaire.

Tableau 5-6. Limites des transactions d'interruption à faible vitesse

	Surcharge de protocole (19 octets)	(5 octets SYNC, 5 octets PID, 2 octets Point d'extrémité + CRC, 2 octets CRC et un délai inter-paquets de 5 octets)				
	Charge utile de données	**Bande passante max. (octets/seconde)**	**Bande passante de trame par transfert**	**Transferts max.**	**Octets restants**	**Données utiles en octets/trame**
	1	9 000	11 %	9	7	9
	2	16 000	11 %	8	19	16
	4	32 000	12 %	8	3	32
	8	48 000	14 %	6	25	48
Max		187 500				187

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 5-7. Limites des transactions d'interruption à vitesse intégrale

	Surcharge de protocole (13 octets)		(3 octets SYNC, 3 octets PID, 2 octets Point d'extrémité + CRC, 2 octets CRC et un délai inter-paquets de 3 octets)			
	Charge utile de données	Bande passante max. (octets/seconde)	Bande passante de trame par transfert	Transferts max.	Octets restants	Données utiles en octets/trame
	1	107 000	1 %	107	2	107
	2	200 000	1 %	100	0	200
	4	352 000	1 %	88	4	352
	8	568 000	1 %	71	9	568
	16	816 000	2 %	51	21	816
	32	1 056 000	3 %	33	15	1 056
	64	1 216 000	5 %	19	37	1 216
Max		1 500 000				1 500

Tableau 5-8. Limites des transactions d'interruption à haute vitesse

	Surcharge de protocole		(sur la base de 480 Mb/s et un intervalle inter-paquets de 8 bits, retournement de bus de 88 bits min., sync de 32 bits, EOP de 8 bits: (3x4 octets SYNC, 3 octets PID , 2 octets EP/ADDR+CRC, 2 CRC16 et délai inter-paquets de 3x(1+11) octets (EOP, etc.))			
	Charge utile de données	Bande passante max. (octets/seconde)	Bande passante de trame par transfert	Transferts max.	Octets restants	Données utiles en octets/microtrame
	1	1 064 000	1 %	133	52	133
	2	2 096 000	1 %	131	33	262
	4	4 064 000	1 %	127	7	508
	8	7 616 000	1 %	119	3	952
	16	13 440 000	1 %	105	45	1 680
	32	22 016 000	1 %	86	18	2 752
	64	32 256 000	2 %	63	3	4 032
	128	40 960 000	2 %	40	180	5 120
	256	49 152 000	4 %	24	36	6 144
	512	53 248 000	8 %	13	129	6 656
	1 024	49 152 000	14 %	6	1 026	6 144
	2 048	49 152 000	28 %	3	1 191	6 144
	3 072	49 152 000	42 %	2	1 246	6 144
Max		60 000 000				7 500

Un point d'extrémité d'un canal d'interruption spécifie la période d'accès au bus souhaitée. Un point d'extrémité à vitesse intégrale peut spécifier une période souhaitée entre 1 ms et 255 ms. Les points d'extrémité à faible vitesse sont limités à une spécification allant de 10 ms à 255 ms uniquement. Les points d'extrémité à haute vitesse peuvent spécifier une période souhaitée $(2^{bInterval-1})$x125 µs, où *bInterval* représente l'intervalle de 1 à 16 (inclus). Le logiciel système USB utilise ces informations pendant la configuration pour déterminer une période pouvant être soutenue. La période fournie par le système peut être plus courte que celle souhaitée par l'appareil ou égale à la période la plus courte définie par l'USB (microtrame de 125 µs ou trame de 1 ms). Le logiciel client et l'appareil peuvent dépendre uniquement du fait que l'hôte s'assure que la durée écoulée entre deux tentatives de transaction avec le point d'extrémité n'est pas plus longue que la période souhaitée. Il faut noter que les erreurs du bus peuvent empêcher la livraison réussie d'une transaction d'interruption sur le bus et, par conséquent, entraîner le dépassement de la période souhaitée. De même, le point d'extrémité est interrogé uniquement lorsque le client logiciel dispose d'un IRP pour un transfert d'interruption en attente. Si le temps de bus pour la réalisation d'un transfert d'interruption est alloué et qu'il n'y a aucun IRP en attente, le point d'extrémité n'a pas l'opportunité de transférer les données à ce moment précis. Une fois qu'un IRP est disponible, ses données sont transférées au cours de la prochaine période allouée.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Un point d'extrémité à haute vitesse peut déplacer jusqu'à 3 072 octets par microtrame (ou 192 Mb/s). Un point d'extrémité d'interruption à haute vitesse qui nécessite plus de 1 024 octets par seconde est appelé un point d'extrémité à large bande passante. Un point d'extrémité à large bande passante utilise plusieurs transactions par microtrame. Un point d'extrémité à large bande passante doit spécifier une période de 1x125 µs (c'est-à-dire, une valeur *bInterval* égale à 1). Voir la Section 5.9 pour plus d'informations concernant les détails sur les transactions multiples par microtrame pour les points d'extrémité à haute vitesse et à large bande passante.

Les transferts d'interruption sont déplacés sur l'USB en accédant à un point d'extrémité d'interruption durant chaque période spécifiée. Pour les points d'extrémité d'interruption d'entrée, l'hôte ne dispose d'aucun moyen pour déterminer si un point d'extrémité est la source d'une interruption sans accéder au point d'extrémité ni demander un transfert d'interruption. Si le point d'extrémité ne dispose d'aucune donnée d'interruption à transmettre lorsque l'hôte y accède, il répond avec un NAK. Il convient qu'un point d'extrémité fournisse uniquement des données d'interruption lorsqu'il a une interruption en attente pour éviter de notifier un client logiciel de l'achèvement d'un IRP de façon erronée. Une charge utile de données de longueur nulle est un transfert valide et peut être utile pour certaines mises en application.

5.7.5 Séquences de données des transferts d'interruption

Les transactions d'interruption peuvent utiliser soit des bits de bascule de données alternatifs, tels que les bits qui basculent uniquement après l'achèvement réussi du transfert, soit un basculement continu des bits de bascule de données. Dans tous les cas, l'hôte doit supposer que l'appareil obéit entièrement aux règles de poignée de mains/nouvelle tentative, tel que défini au Chapitre 8. Un appareil peut choisir de basculer à chaque fois les PID DATA0/DATA1 de manière à ignorer les poignées de mains provenant de l'hôte. Toutefois, dans ce cas, le logiciel client peut omettre certains paquets de données lorsqu'une erreur se produit parce que le contrôleur hôte interprète le paquet suivant comme étant une nouvelle tentative pour un paquet omis.

Si une condition d'arrêt est détectée sur un canal d'interruption à cause des erreurs de transmission ou d'une poignée de mains STALL retournée par le point d'extrémité, tous les IRP en attente sont retirés. La suppression de la condition d'arrêt se réalise par l'intermédiaire d'une intervention logicielle par un canal de contrôle séparé. Cette récupération réinitialise le bit de bascule de données à DATA0 pour le point d'extrémité de l'hôte et de l'appareil. Les transactions d'interruption sont tentées de nouveau à cause des erreurs détectées sur le bus qui affectent un transfert donné.

5.8 Transferts en bloc

Le type de transfert en bloc est conçu pour prendre en charge les appareils qui doivent communiquer des quantités relativement importantes de données à des moments extrêmement variables lorsque le transfert peut utiliser toute la bande passante disponible. La requête d'un canal avec un type de transfert en bloc fournit au demandeur les éléments suivants:

- Accès à l'USB sur une base de bande passante disponible
- Nouvelles tentatives de transfert, dans le cas d'une défaillance de livraison occasionnelle due aux erreurs sur le bus
- Livraison garantie des données, mais aucune garantie de bande passante ou de latence

Les transferts en bloc se produisent uniquement sur une base de bande passante disponible. Pour un USB avec des quantités importantes de bande passante libre, les transferts en bloc peuvent se produire relativement vite; pour un USB avec peu de bande passante disponible, les transferts en bloc peuvent se produire sur une période de temps relativement longue.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

5.8.1 Format de données des transferts en bloc

L'USB n'impose aucune structure au contenu de données des flux de communication pour les canaux en bloc.

5.8.2 Direction des transferts en bloc

Un canal de bloc est un canal de flux et, par conséquent, les flux de communication entrent ou sortent de l'hôte de manière continue pour un canal donné. Si un appareil requiert un débit de communication en bloc bidirectionnel, deux canaux de bloc doivent être utilisés, un dans chaque direction.

5.8.3 Contraintes liées à la taille des paquets des transferts en bloc

Un point d'extrémité pour transferts en bloc spécifie la taille maximum des charges utiles de données que le point d'extrémité peut accepter en provenance du bus ou transmettre au bus. L'USB limite les tailles maximum autorisées des charges utiles de données en bloc à 8, 16, 32 ou 64 octets pour les points d'extrémité à vitesse intégrale et à 512 octets pour les points d'extrémité à haute vitesse. Un appareil à faible vitesse ne doit pas avoir de points d'extrémité en bloc. Cette taille maximum s'applique aux charges utiles de données des paquets de données; c'est-à-dire, la taille spécifiée est celle du champ de données du paquet, tel que défini au Chapitre 8, sans inclure d'autres informations requises par le protocole.

Un point d'extrémité en bloc est conçu pour prendre en charge une taille maximum de charge utile de données. Un point d'extrémité en bloc indique la valeur de la taille maximum de ses charges utiles de données dans les informations de sa configuration. L'USB n'exige pas que les charges utiles de données transmises aient exactement la taille maximum; c'est-à-dire, si une charge utile de données est inférieure à la taille maximum, il n'est pas nécessaire qu'elle soit complétée pour atteindre la taille maximum.

Tous les contrôleurs hôtes doivent prendre en charge des tailles maximales de paquet de 8, 16, 32 et 64 octets pour les points d'extrémité en bloc à vitesse intégrale et de 512 octets pour les points d'extrémité en bloc à haute vitesse. Aucun contrôleur hôte ne doit prendre en charge des tailles maximales de paquet plus grandes ou plus petites.

Pendant la configuration, le logiciel système USB lit la taille maximum de charge utile de données du point d'extrémité et s'assure qu'aucune charge utile de données dont la taille est plus grande que celle prise en charge n'est envoyée au point d'extrémité.

Un point d'extrémité doit toujours transmettre les charges utiles de données avec un champ de données inférieur ou égal à la valeur *w*MaxPacketSize indiquée du point d'extrémité. Lorsqu'un IRP en bloc implique une quantité de données plus importante que la taille maximum qui peut être acceptée pour une charge utile de données, toutes les charges utiles de données doivent avoir la taille maximum, à l'exception de la dernière charge utile de données qui contient les données restantes. Un transfert en bloc s'achève lorsque le point d'extrémité effectue une des actions suivantes:

- Il a transféré la quantité exacte de données prévues
- Il transfère un paquet avec une taille de charge utile inférieure à la valeur de *wMaxPacketSize* ou transfère un paquet de longueur nulle

Lorsqu'un transfert en bloc est terminé, le contrôleur hôte retire l'IRP actuel et avance à l'IRP suivant. Si une charge utile de données reçue est plus grande que prévue, tous les IRP en bloc qui sont en attente pour ce point d'extrémité sont interrompus/retirés.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

5.8.4 Contraintes liées à l'accès au bus des transferts en bloc

Seuls les appareils à vitesse intégrale et à haute vitesse peuvent utiliser des transferts en bloc.

Un point d'extrémité n'a aucun moyen d'indiquer la fréquence souhaitée d'accès au bus pour un canal en bloc. L'USB équilibre les exigences d'accès au bus de tous les canaux en bloc et des IRP spécifiques qui sont en attente pour effectuer la meilleure livraison possible entre le logiciel client et les fonctions. Les déplacements des transferts de contrôle sur le bus sont prioritaires par rapport aux transferts en bloc.

Aucune disponibilité de temps n'est garantie pour les transferts en bloc, contrairement aux transferts de contrôle. Les transferts en bloc se déplacent sur le bus uniquement sur une base de bande passante disponible. Si une partie du temps de bus n'est pas utilisée à d'autres fins, les transferts en bloc sont déplacés sur le bus. Si des transferts de bus sont en attente pour plusieurs points de terminaison, les transferts en bloc des différents points d'extrémité sont sélectionnés selon la politique d'accès équitable dépendante de la mise en application du contrôleur hôte.

Tous les transferts en bloc qui sont en attente dans un système se disputent le même temps de bus disponible. Pour cette raison, le logiciel système USB peut, à sa discrétion, varier le temps de bus mis à la disposition des transferts en bloc vers un point d'extrémité particulier. Un point d'extrémité et son logiciel client ne peuvent pas préciser un débit spécifique de service pour les transferts en bloc. Le temps de bus mis à la disposition d'un client logiciel et de son point d'extrémité peut être changé au fur et à mesure que d'autres appareils sont insérés ou supprimés du système ou également au fur et à mesure que des transferts en bloc sont requis par d'autres points d'extrémité d'appareil. Le logiciel client ne peut pas déterminer l'ordre entre les transferts en bloc et de contrôle; c'est-à-dire, dans certaines situations, les transferts en bloc peuvent être délivrés avant les transferts de contrôle.

Les points d'extrémité OUT en bloc à haute vitesse doivent prendre en charge le protocole de contrôle de débit PING. Les détails de ce protocole sont décrits à la Section 8.5.1.

La fréquence de bus et la synchronisation de (micro)trame limitent le nombre maximum de transactions en bloc réussies dans le cadre d'une (micro)trame pour un système USB à moins de 72 charges utiles de données à vitesse intégrale de huit octets ou à moins de 14 charges utiles de données à haute vitesse de 512 octets. Le Tableau 5-9 affiche les informations concernant les transactions en bloc à vitesse intégrale de différentes tailles et le nombre maximum de transactions possibles dans une trame. Le tableau ne contient pas la surcharge associée à l'insertion binaire. Le Tableau 5-10 affiche des informations similaires pour les transactions en bloc à haute vitesse.

Tableau 5-9. Limites des transactions en bloc à vitesse intégrale

	Surcharge de protocole (13 octets)		(3 octets SYNC, 3 octets PID, 2 octets Point d'extrémité + CRC, 2 octets CRC et un délai inter-paquets de 3 octets)			
	Charge utile de données	Bande passante max. (octets/seconde)	Bande passante de trame par transfert	Transferts max.	Octets restants	Données utiles en octets/trame
	1	107 000	1 %	107	2	107
	2	200 000	1 %	100	0	200
	4	352 000	1 %	88	4	352
	8	568 000	1 %	71	9	568
	16	816 000	2 %	51	21	816
	32	1 056 000	3 %	33	15	1 056
	64	1 216 000	5 %	19	37	1 216
Max		1 500 000				1 500

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 5-10. Limites des transactions en bloc à haute vitesse

	Surcharge de protocole (55 octets)		(3x4 octets SYNC, 3 octets PID , 2 octets EP/ADDR+CRC, 2 CRC16 et délai inter-paquets de 3x(1+11) octets (EOP, etc.))			
	Charge utile de données	Bande passante max. (octets/seconde)	Bande passante de trame par transfert	Transferts max.	Octets restants	Données utiles en octets/microtrame
	1	1 064 000	1 %	133	52	133
	2	2 096 000	1 %	131	33	262
	4	4 064 000	1 %	127	7	508
	8	7 616 000	1 %	119	3	952
	16	13 440 000	1 %	105	45	1 680
	32	22 016 000	1 %	86	18	2 752
	64	32 256 000	2 %	63	3	4 032
	128	40 960 000	2 %	40	160	5 120
	256	49 152 000	4 %	24	36	6 144
	512	53 248 000	8 %	13	129	6 656
Max		60 000 000				7 500

Les contrôleurs hôtes sont libres de déterminer la manière dont les transactions de bus individuelles des transferts en bloc spécifiques sont déplacées sur le bus dans le cadre des (micro)trames et sur les (micro)trames. Un point d'extrémité peut voir toutes les transactions de bus d'un transfert en bloc dans la même (micro)trame ou réparties sur plusieurs (micro)trames non contiguës. Pour diverses raisons de mise en application, un contrôleur hôte peut ne pas être en mesure de fournir le nombre maximum de transactions par (micro)trame énoncé ci-dessus.

5.8.5 Séquences de données des transferts en bloc

Les transactions en bloc utilisent des bits de bascule de données qui basculent uniquement après l'achèvement réussi de la transaction afin de conserver la synchronisation entre le transmetteur et le récepteur lorsque les transactions sont tentées de nouveau à cause des erreurs. Les transactions en bloc sont initialisées à DATA0 lorsque le point d'extrémité est configuré par un transfert de contrôle approprié. L'hôte démarre également la première transaction en bloc avec DATA0. Si une condition d'arrêt est détectée sur un canal en bloc à cause des erreurs de transmission ou d'une poignée de mains STALL retournée par le point d'extrémité, tous les IRP en attente sont retirés. La suppression de la condition d'arrêt se réalise par l'intermédiaire d'une intervention logicielle par un canal de contrôle séparé. Cette récupération réinitialise le bit de bascule de données à DATA0 pour le point d'extrémité de l'hôte et de l'appareil.

Les transactions en bloc sont tentées de nouveau à cause des erreurs détectées sur le bus qui affectent une transaction donnée.

5.9 Points d'extrémité à haute vitesse et à large bande passante

L'USB prend en charge les points d'extrémité d'interruption ou isochrones individuels à haute vitesse qui nécessitent des débits de données pouvant atteindre 192 Mb/s (c'est-à-dire, 3 072 octets de données par microtrame). Une, deux ou trois transactions à haute vitesse sont permises dans une seule microtrame pour la prise en charge de points d'extrémité à large bande passante.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Un point d'extrémité d'interruption ou isochrone à haute vitesse indique qu'il a besoin de plus de 1 024 octets par microtrame lorsque 12..11 bits du champ *wMaxPacketSize* du descripteur de point d'extrémité (voir Tableau 5-11) ne sont pas nuls. La valeur inférieure de 11 bits du champ *wMaxPacketSize* indique la taille de la charge utile de données pour chaque transaction individuelle tandis que la valeur de 12..11 indique le nombre maximum de transactions nécessaires possibles. Voir la Section 9.6.6 pour les restrictions relatives aux valeurs de 12..11 bits et 10..0 bits.

Tableau 5-11. Champ *wMaxPacketSize* du descripteur de point d'extrémité

Bits	15..13	12..11	10..0
Champ	Réservé, doit être égal à zéro	Nombre de transactions par microtrame	Taille maximum de charge utile de données en octets

Note: cette représentation signifie que les points d'extrémité nécessitant deux transactions par microtrame spécifient une taille totale de charge utile de données dans la microtrame qui est un multiple de deux octets. De même, les points d'extrémité nécessitant trois transactions par microtrame spécifient une taille totale de charge utile de données qui est un multiple de trois octets. Dans tous les cas, n'importe quel nombre d'octets peut être transféré dans une microtrame.

Le contrôleur hôte doit émettre un nombre approprié de transactions à haute vitesse par microtrame. Les erreurs de l'hôte ou du bus peuvent amener le contrôleur hôte à émettre un nombre de transactions inférieur au nombre requis pour le point d'extrémité. La(Les) première(s) transaction(s) doi(ven)t avoir la(les) charge(s) utile(s) spécifiée(s) par la valeur inférieure de 11 bits du champ *wMaxPacketSize* si une quantité suffisante de données est disponible, tandis que la dernière transaction contient la quantité de données restantes qui est inférieure ou égale à la taille maximum spécifiée. Le contrôleur hôte peut émettre des transactions pour le même point d'extrémité l'une après l'autre (selon les données actuellement fournies) ou des transactions pour d'autres points d'extrémité entre les transactions d'un point d'extrémité à large bande passante.

5.9.1 Points d'extrémité d'interruption à large bande passante

Pour les transactions d'interruption, si le point d'extrémité répond à une transaction avec un NAK pendant une microtrame, le contrôleur hôte ne doit plus émettre de transactions supplémentaires pour ce point d'extrémité jusqu'à la période suivante.

Si le point d'extrémité ne répond pas dans la période impartie pour une transaction, le contrôleur doit de nouveau tenter la transaction. Le point d'extrémité spécifie le nombre maximum de transactions souhaitées par microtrame. Si le nombre maximum de transactions par microtrame n'a pas été atteint, le contrôleur hôte peut immédiatement tenter de nouveau la transaction pendant la microtrame actuelle. Il est recommandé aux contrôleurs hôtes d'effectuer immédiatement une nouvelle tentative étant donné que cela minimise l'impact sur les appareils qui sont sensibles à la bande passante. Si le nombre maximum de transactions par microtrame a été atteint, le contrôleur hôte doit de nouveau tenter la transaction durant la période suivante allouée au point d'extrémité.

Un contrôleur hôte peut émettre un nombre de transactions inférieur au nombre maximum pour un point d'extrémité par microtrame uniquement si plus d'une seule mémoire tampon est requise pour les transactions dans le cadre de la microtrame.

Le séquencement normal de bits de bascule de données DATA0/DATA1 est utilisé pour chaque transaction d'interruption durant une microtrame.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

5.9.2 Points d'extrémité isochrones à large bande passante

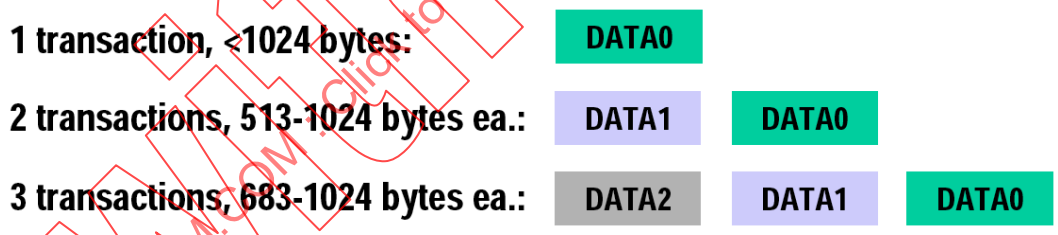
Pour les transactions isochrones, si un point d'extrémité IN fournit une quantité de charge utile de données inférieure à la quantité maximum spécifiée par son descripteur de point d'extrémité, l'hôte ne doit émettre aucune transaction supplémentaire pour ce point d'extrémité dans le cadre de la même microtrame.

Pour un point d'extrémité OUT isochrone, le contrôleur hôte doit émettre le nombre de transactions requis pour les données actuellement fournies, sans dépasser le nombre maximum spécifié par le descripteur de point d'extrémité. Les transactions émises doivent respecter les tailles maximales de charge utile spécifiées dans le descripteur de point d'extrémité.

Une nouvelle tentative ne doit jamais être effectuée pour les points d'extrémité isochrones.

Les points d'extrémité (IN et OUT) isochrones à large bande passante doivent prendre en charge le séquencement PID de données. Le séquencement PID de données fournit la prise en charge requise pour que le récepteur de données puisse détecter un ou plusieurs paquets perdus/endommagés par microtrame.

Le séquencement PID de données pour un point d'extrémité IN isochrone à haute vitesse et à large bande passante utilise une séquence répétitive de PID DATA2, DATA1, DATA0 pour le paquet de données de chaque transaction dans le cadre d'une microtrame. S'il existe uniquement une seule transaction dans la microtrame, seul un PID de paquet de données DATA0 est utilisé. S'il existe deux transactions par microtrame, DATA1 est utilisé pour le paquet de données de la première transaction et DATA0 est utilisé pour le paquet de données de la deuxième transaction. S'il existe trois transactions par microtrame, DATA2 est utilisé pour le paquet de données de la première transaction, DATA1 est utilisé pour le paquet de données de la deuxième et DATA0 est utilisé pour le paquet de données de la troisième. Dans tous les cas, la séquence de PID de données se répète à la microtrame suivante. La Figure 5-11 affiche l'ordre des PID de paquet de données qui sont utilisés dans les transactions ultérieures dans le cadre d'une microtrame pour des points d'extrémité IN isochrones à large bande passante.



Légende

Anglais	Français
1 transaction, < 1024 bytes:	1 transaction, < 1 024 octets:
2 transactions, 513-1024 bytes ea.:	2 transactions, 513 - 1 024 octets ea.:
3 transactions, 683-1024 bytes ea.:	3 transactions, 683 - 1 024 octets ea.:

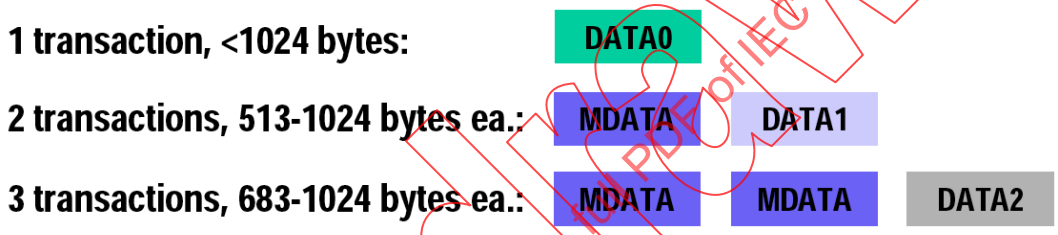
Figure 5-11. Séquence des PID de la phase de données des points d'extrémité IN isochrones à large bande passante

Un point d'extrémité doit répondre à un jeton IN de la première transaction avec DATA2 lorsqu'il nécessite le déplacement de trois transactions de données. Il doit répondre avec DATA1 pour la première transaction lorsqu'il requiert deux transactions et avec DATA0 lorsqu'il requiert une seule transaction. Après la première transaction, le point d'extrémité suit la séquence des PID de données décrite ci-dessus.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

L'hôte connaît le nombre maximum de transactions permises par microtrame pour le point d'extrémité IN. L'hôte s'attend à ce que la réponse à la première transaction code (par l'intermédiaire du PID de paquet de données) le nombre de transactions requises par le point d'extrémité dans le cadre de la microtrame actuelle. Si l'hôte ne reçoit aucune réponse appropriée et exempte d'erreur à une transaction, il ne doit plus émettre de transaction supplémentaire vers le point d'extrémité pour la microtrame actuelle. Lorsque l'hôte reçoit un paquet de données DATA0 en provenance du point d'extrémité, il ne doit plus émettre de transaction supplémentaire vers le point d'extrémité de la microtrame actuelle.

Le séquencement des PID de données pour un point d'extrémité IN isochrone à haute vitesse et à large bande passante utilise une séquence différence de celle utilisée pour un point d'extrémité IN. L'hôte doit émettre un paquet de données DATA0 lorsqu'il existe une seule transaction. L'hôte doit émettre un MDATA pour la première transaction et un DATA1 pour la deuxième transaction lorsqu'il existe deux transactions par microtrame. L'hôte doit émettre deux transactions MDATA et un DATA1 pour la troisième transaction lorsqu'il existe trois transactions par microtrame. Ces séquences permettent au point d'extrémité de détecter si une microtrame a été perdue/endommagée durant une microtrame. La Figure 5-12 affiche l'ordre des PID de paquet de données qui sont utilisés dans les transactions ultérieures dans le cadre d'une microtrame pour des points d'extrémité OUT isochrones à large bande passante.



Légende

Anglais	Français
1 transaction, < 1024 bytes:	1 transaction, < 1 024 octets:
2 transactions, 513-1024 bytes ea.:	2 transactions, 513 - 1 024 octets ea.:
3 transactions, 683-1024 bytes ea.:	3 transactions, 683 - 1 024 octets ea.:

Figure 5-12. Séquence des PID de la phase de données des points d'extrémité OUT isochrones à large bande passante

Si les mauvaises transactions OUT sont détectées par le point d'extrémité, toutes les données transférées durant la microtrame doivent être traitées comme si elles avaient rencontré une erreur. Il faut noter que dans le cas des trois transactions par microtrame avec une transaction MDATA manquante, l'USB ne fournit aucun moyen au point d'extrémité de déterminer laquelle des deux transactions MDATA a été perdue. Il peut exister des méthodes spécifiques à l'application pour déterminer les données perdues de manière précise, mais l'USB ne fournit aucune méthode pour ce faire au niveau du bus.

5.10 Transactions de scission

Les contrôleurs hôtes et les hubs prennent en charge un type de transaction supplémentaire appelé transaction de scission. Ce type de transaction permet le raccordement des appareils à vitesse intégrale ou faible vitesse à des hubs fonctionnant à haute vitesse. Ces transactions impliquent uniquement les contrôleurs hôtes et les hubs et ne sont pas visibles aux appareils. Les transactions de scission à haute vitesse pour les transferts d'interruption et isochrones doivent être allouées par l'hôte à partir de 80 % de la portion périodique d'une microtrame. De plus amples informations sur les transactions de scission peuvent être trouvées aux Chapitres 8 et 11.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.11 Accès au bus des transferts

La réalisation d'un transfert de données entre l'hôte et un appareil USB nécessite l'utilisation de la bande passante USB. La prise en charge d'une large variété d'appareils isochrones et asynchrones nécessite que les exigences de transfert de chaque appareil soient satisfaites. Le processus d'attribution de la bande passante de bus aux appareils est appelé la gestion du transfert. Plusieurs entités de l'hôte coordonnent les informations transmises sur l'USB: le logiciel client, le pilote USB (USBD) et le pilote de contrôleur hôte (HCD). Les concepteurs de ces entités doivent connaître les concepts clés liés à l'accès au bus:

- Gestion du transfert: les entités et les objets qui prennent en charge le flux de communication sur l'USB
- Suivi de transactions: mécanismes USB utilisés pour suivre le mouvement des transactions dans le système
- Temps de bus: durée de déplacement d'un paquet d'informations sur le bus
- Taille de mémoire tampon d'appareil/logiciel: espace requis pour la prise en charge d'une transaction de bus
- Réclamation de bande passante de bus: conditions dans lesquelles la bande passante a été allouée à d'autres transferts, mais n'a pas été utilisée et qui peut maintenant être éventuellement utilisée par les transferts de contrôle et en bloc

Les sections précédentes se sont concentrées sur la liaison qui existe entre le logiciel client et une fonction et la signification des flux logiques sur un canal entre deux entités. La présente section se concentre sur les différentes parties de l'hôte et sur la manière dont elles doivent interagir pour la prise en charge du déplacement des données sur l'USB. Ces informations peuvent également représenter un intérêt pour les concepteurs d'appareils afin qu'ils puissent comprendre les aspects concernant le rôle de l'hôte lorsqu'un client demande un transfert et la manière dont le transfert est présenté à l'appareil.

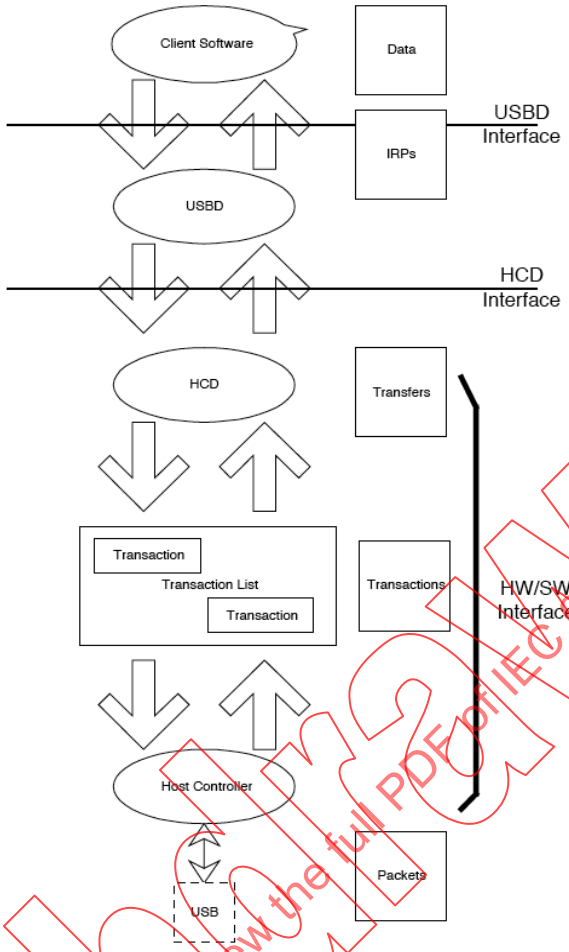
5.11.1 Gestion du transfert

La gestion du transfert implique plusieurs entités qui agissent sur différentes entités afin de déplacer les transactions sur le bus:

- Logiciel client: consomme/génère des données spécifiques à la fonction en provenance/à destination d'un point d'extrémité de fonction par l'intermédiaire d'appels et de procédures de rappel nécessitant des IRP avec l'interface USBD.
- Pilote USB (USBD): convertit les données des IRP clients à destination/en provenance du point d'extrémité de l'appareil par l'intermédiaire d'appels/procédures de rappel avec le HCD approprié. Un seul IRP client peut impliquer un ou plusieurs transferts.
- Pilote de contrôleur hôte (HCD): convertit les IRP à destination/en provenance des transactions (comme l'exige la mise en application d'un contrôleur hôte) et les organise pour la manipulation par le contrôleur hôte. Les interactions entre le HCD et son matériel dépendent de la mise en application et ne relèvent pas du domaine d'application de la spécification USB.
- Contrôleur hôte: prend les transactions et génère l'activité de bus par l'intermédiaire de paquets pour déplacer les données spécifiques à la fonction sur le bus pour chaque transaction.

La Figure 5-13 illustre la manière dont les entités sont organisées sous forme de flux d'informations entre le logiciel client et l'USB. Les objets d'intérêt principal de chaque entité sont indiqués au niveau des interfaces entre entités.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Client Software	Logiciel client
Data	Données
USBD Interface	Interface USBD
IRPs	IRP
HCD Interface	Interface HCD
Transfers	Transferts
Transaction	Transaction
Transaction List	Liste de transactions
Transactions	Transactions
HW/SW Interface	Interface matérielle/logicielle
Host Controller	Contrôleur hôte
Packets	Paquets

Figure 5-13. Conversion des informations USB allant du client logiciel au bus

5.11.1.1 Logiciel client

Le logiciel client détermine les transferts qui doivent être effectués avec une fonction. Il utilise les interfaces appropriées spécifiques au système d'exploitation pour demander les IRP. Le logiciel client ne connaît que l'ensemble de canaux (c'est-à-dire, l'interface) dont il doit manipuler la fonction. Le client connaît et satisfait à toutes les contraintes relatives à l'accès et à la bande passante de bus décrites précédemment pour chaque type de transfert. Les requêtes effectuées par le logiciel client sont présentées par l'intermédiaire de l'interface USBD.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Certains clients peuvent manipuler les fonctions USB par l'intermédiaire d'autres interfaces de classe d'appareil définies par le système d'exploitation et peuvent eux-mêmes ne pas être en mesure d'effectuer des appels directs USBD. Toutefois, il existe toujours un client de niveau inférieur qui effectue les appels USBD pour passer les IRP à l'USBD. Tous les IRP doivent satisfaire à l'ensemble de contraintes négociées à l'avance relatives à la bande passe lorsque le canal a été établi. Si une fonction est déplacée d'un environnement différent de l'USB à l'USB, le pilote qui aurait directement manipulé le matériel de la fonction par l'intermédiaire de la mémoire ou des accès E/S est le logiciel client de niveau le plus bas dans l'environnement USB qui interagit maintenant avec l'USBD pour manipuler la fonction USB du pilote.

Une fois que le logiciel client a demandé un transfert de sa fonction et que la requête a été satisfaite, le client logiciel reçoit une notification de l'état d'achèvement de l'IRP. Si le transfert a impliqué un transfert de données de la fonction à l'hôte, le logiciel client peut accéder aux données de la mémoire tampon de données associée avec l'IRP terminé.

L'interface USBD est définie au Chapitre 10.

5.11.1.2 Pilote USB

Le pilote de bus universel en série (USBD) est généralement impliqué dans la supervision de l'accès au bus à deux moments:

- Lorsqu'un appareil est connecté au bus pendant la configuration
- Pendant les transferts normaux

Lorsqu'un appareil est connecté et configuré, l'USBD est impliqué afin de s'assurer que la configuration souhaitée de l'appareil peut être prise en charge sur le bus. L'USBD reçoit les requêtes de configuration provenant du logiciel de configuration qui décrivent la configuration souhaitée de l'appareil: point(s) d'extrémité, type(s) de transfert, période(s) de transfert, taille(s) de données, etc. L'USBD accepte ou rejette une requête de configuration sur la base de la disponibilité de la bande passante et de la capacité à prendre en charge le type de requête sur le bus. S'il accepte la requête, l'USBD crée un canal pour le demandeur du type souhaité, avec les contraintes définies pour le type de transfert. L'attribution de la bande passante aux points d'extrémité périodiques ne doit pas se faire après la configuration de l'appareil et, une fois que celle-ci est effectuée, une attribution de bande passante peut être réalisée sans changer la configuration de l'appareil.

Les aspects de la configuration de l'USBD sont généralement spécifiques au système d'exploitation et utilisent lourdement les caractéristiques de configuration du système d'exploitation pour éviter la définition d'interfaces supplémentaires (redondantes).

Une fois qu'un appareil est configuré, le client logiciel peut demander aux IRP de déplacer les données entre les points d'extrémité de sa fonction et lui-même.

5.11.1.3 Pilote de contrôleur hôte

Le pilote de contrôleur hôte (HCD) est chargé de suivre les IRP en cours et de s'assurer que la bande passante USB et les temps maximums des (micro)trames ne sont jamais dépassés. Lorsque les IRP sont prévus pour un canal, le HCD les ajoute à la liste des transactions. Lorsqu'un IRP est terminé, le HCD notifie le client logiciel demandeur de l'état d'achèvement de l'IRP. Si l'IRP a impliqué un transfert de données de la fonction vers le client logiciel, les données ont été placées dans la mémoire tampon de données indiquée par le client.

Les IRP sont définis en fonction du système d'exploitation.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5.11.1.4 Liste des transactions

La liste des transactions est une description dépendante de la mise en application du contrôleur hôte de l'ensemble actuel des transactions de bus qui doivent être exécutées sur le bus. Seuls le HCD et son contrôleur hôte ont accès à la représentation spécifique. Chaque description contient des descriptions de transaction dans le cadre desquelles les paramètres, tels que la taille des données en octets, l'adresse de l'appareil et le numéro du point d'extrémité ainsi que la zone de mémoire vers laquelle les données doivent être envoyées ou de laquelle elles proviennent, sont identifiés.

Une liste de transactions et l'interface entre le HCD et son contrôleur hôte sont généralement représentées en fonction d'une mise en application et ne sont pas définies de manière explicite comme faisant partie intégrante de la spécification USB.

5.11.1.5 Contrôleur hôte

Le contrôleur hôte a accès à la liste des transactions et la traduit dans l'activité de bus. En outre, le contrôleur hôte fournit un mécanisme d'établissement de rapports à l'aide duquel l'état d'une transaction (terminée, en attente, interrompue, etc.) peut être obtenu. Le contrôleur hôte convertit les transactions dans des activités appropriées dépendantes de la mise en application qui peuvent mener les paquets USB à se déplacer sur la topologie de bus issue du hub racine.

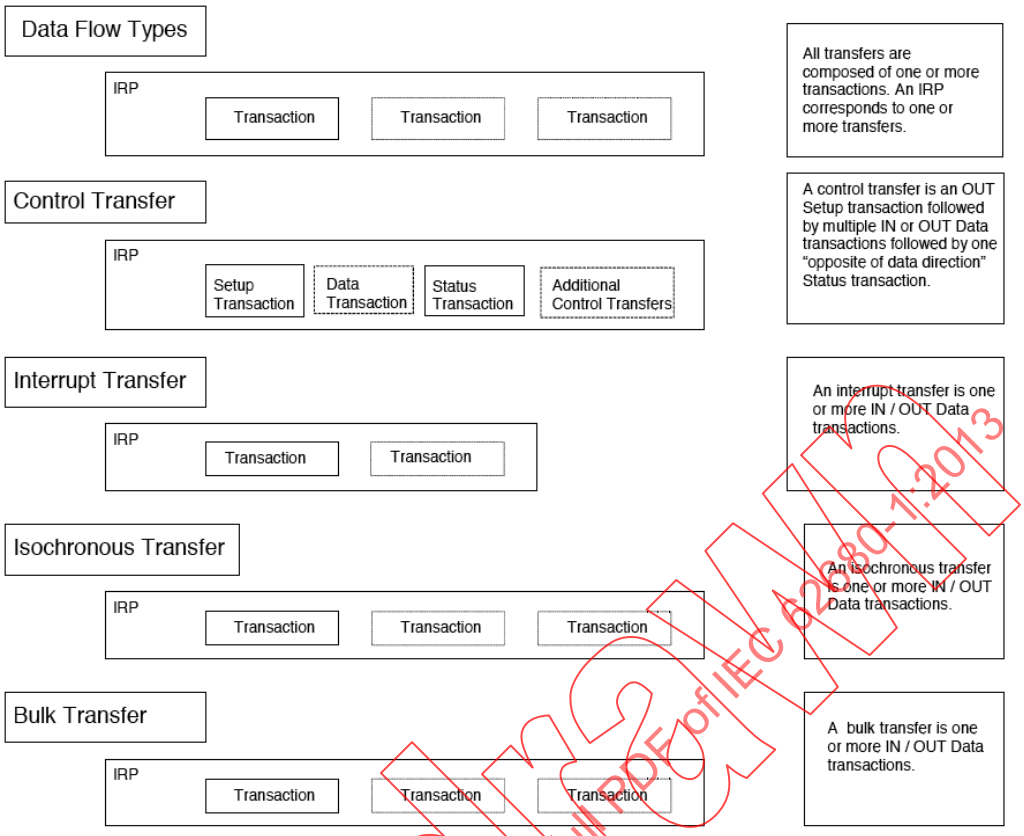
Le contrôleur hôte s'assure que les règles d'accès au bus définies par le protocole sont respectées, telles que les durées inter-paquets, les temporisations, les babillages, etc. L'interface HCD fournit un moyen au contrôleur hôte de participer à la décision de savoir si un nouveau canal est autorisé à accéder au bus. Cela est effectué parce que des restrictions/contraintes peuvent être appliquées aux mises en application du contrôleur hôte sur les durées minimales inter-transactions qu'elles peuvent prendre en charge pour les combinaisons des transactions de bus.

L'interface entre la liste des transactions et le contrôleur hôte est masquée entre un HCD et la mise en application du contrôleur hôte.

5.11.2 Suivi des transactions

Une fonction USB voit les données transmises sur le bus sous forme de paquets, tel que décrit au Chapitre 8. Le contrôleur hôte utilise une représentation dépendante de la mise en application pour suivre les paquets à transférer vers/à partir de tels points d'extrémité à tel moment ou dans tel ordre. La plupart des logiciels clients ne souhaitent pas traiter avec les flux de communication mis en paquets parce que cela implique un degré de complexité et une dépendance d'interconnexion qui limitent la mise en application. Le logiciel système USB (USBD et HCD) fournit la prise en charge pour le respect des exigences relatives au mouvement des données d'un client par rapport aux paquets du bus. Le matériel et le logiciel du contrôleur hôte utilisent les IRP pour suivre les informations concernant une ou plusieurs transactions qui se combinent pour délivrer un transfert d'informations entre le logiciel client et la fonction. La Figure 5-14 résume la manière dont les transactions sont organisées en IRP pour les quatre types de transfert. Des informations détaillées concernant le protocole pour chaque type de transfert peuvent être trouvées au Chapitre 8. De plus amples informations concernant les vues des IRP du logiciel client peuvent être trouvées au Chapitre 10 et dans les informations spécifiques à un système d'exploitation particulier.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



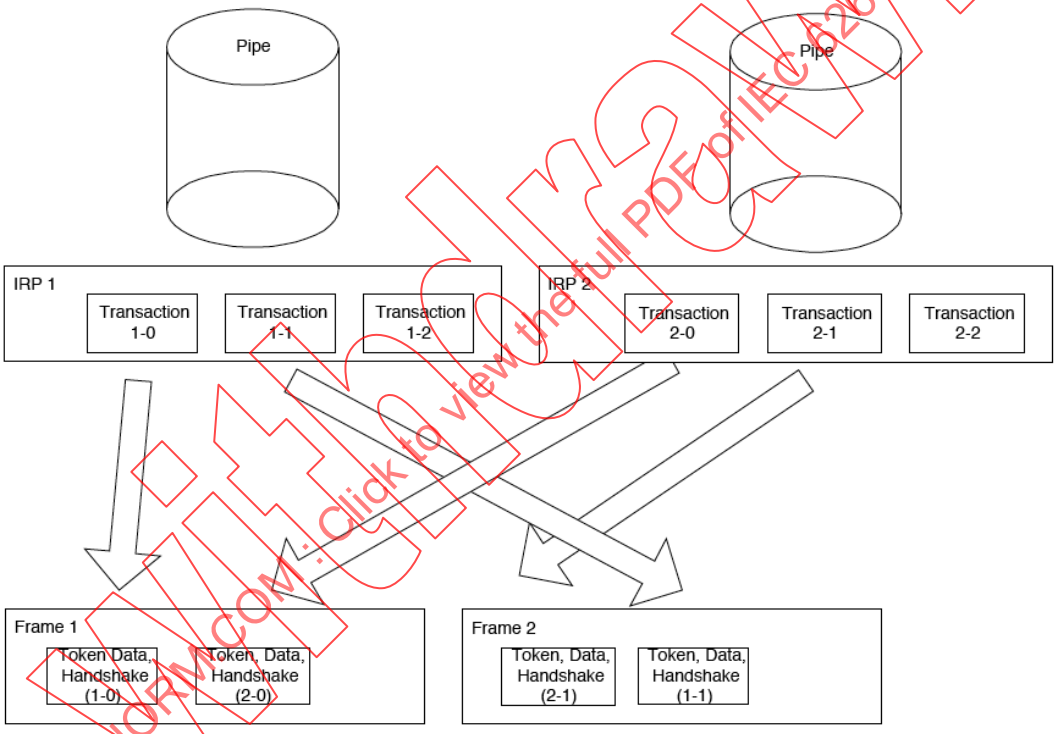
Légende

Anglais	Français
Data Flow Types	Types de flux de données
All transfers are composed of one or more transactions. An IRP corresponds to one or more transfers.	Tous les transferts sont composés d'une ou plusieurs transactions. Un IRP correspond à un ou plusieurs transferts.
Control transfer	Transfert de contrôle
Setup Transaction	Transaction Configuration
Data Transaction	Transaction Données
Status Transaction	Transaction Etat
Additional Control Transfers	Transferts de contrôle supplémentaires
A control transfer is an OUT Setup transaction followed by multiple IN or OUT Data transactions followed by one "opposite of data direction" Status transaction.	Un transfert de contrôle est une transaction Configuration OUT suivie de plusieurs transactions Données IN ou OUR suivies d'une transaction Etat "opposée à la direction des données".
Interrupt Transaction	Transaction d'interruption
Interrupt Transfer	Transfert d'interruption
An interrupt transfer is one or more IN / OUT Data transactions.	Un transfert d'interruption est une ou plusieurs transactions Données IN / OUT
Isochronous Transfer	Transfert isochrones
Bulk Transfer	Transfert en bloc
A bulk transfer is one or more IN / OUT Data transactions.	Un transfert en bloc est une ou plusieurs transactions Données IN / OUT.

Figure 5-14. Transferts des flux de communication

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Bien que les IRP suivent les transactions de bus qui doivent avoir lieu pour déplacer un flux de données spécifiques sur l'USB, les contrôleurs hôtes sont libres de choisir la manière dont les transactions de bus particulières sont déplacées sur le bus faisant l'objet de contraintes définies par l'USB (par exemple, exactement une transaction par (micro)trame pour les transferts isochrones). Dans tous les cas, un point d'extrémité voit les transactions dans l'ordre dans lequel elles apparaissent dans le cadre d'un IRP, à moins que des erreurs n'apparaissent. Par exemple, la Figure 5-15 montre deux IRP, un pour chacun des deux canaux où chaque IRP contient trois transactions. Pour tous les types de transfert, un contrôleur hôte est libre de déplacer la première transaction du premier IRP suivie de la première transaction du deuxième IRP à un endroit quelconque de la (micro)trame 1 tout en déplaçant la deuxième transaction de chaque IRP dans l'ordre inverse à un endroit quelconque de la (micro)trame 2. Dans le cas des types de transferts isochrones, c'est le seul degré de liberté dont dispose un contrôleur hôte. Dans le cas des transferts de contrôle ou en bloc, un contrôleur hôte peut poursuivre le déplacement d'un nombre de transactions plus ou moins important d'un IRP dans le cadre d'une (micro)trame. Les fonctions ne peuvent dépendre de la détection des transactions d'un IRP dos à dos dans le cadre d'une (micro)trame, et il convient qu'elles ne dépendent pas non plus de la non-détection des transactions dos à dos dans le cadre d'une (micro)trame.



Légende

Anglais	Français
Pipe	Canal
IRP 1	IRP 1
Transaction	Transaction
Frame	Trame
Token Data, handshake	Données jeton, poignée de mains

Figure 5-15. Configuration des IRP par rapport aux transactions/(micro)trames

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

5.11.3 Calcul des temps de transaction de bus

Lorsque le logiciel système USB autorise la création d'un nouveau canal pour le bus, il doit calculer le temps de bus nécessaire pour une transaction donnée. Ce temps de bus repose sur les informations relatives à la taille maximum des paquets pour un point d'extrémité, à la surcharge de protocole pour la requête du type de transaction spécifique, à la surcharge due à l'insertion binaire imposée par la signalisation, aux synchronisations inter-paquets requises par le protocole, aux synchronisations inter-transactions, etc. Ces calculs sont requis pour s'assurer que le temps disponible dans une (micro)trame n'est pas dépassé. Les équations utilisées pour déterminer le temps de bus des transactions sont:

KEY:

Data_bc	The byte count of data payload
Host_Delay	The time required for the host or transaction translator to prepare for or recover from the transmission; Host Controller implementation-specific
Floor()	The integer portion of argument
Hub_LS_Setup	The time provided by the Host Controller for hubs to enable low-speed ports; measured as the delay from the end of the PRE PID to the start of the low-speed SYNC; minimum of four full-speed bit times
BitStuffTime	Function that calculates theoretical additional time required due to bit stuffing in signaling; worst case is (1.1667*8*Data_bc)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

High-speed (Input)

```
Non-Isochronous Transfer (Handshake Included)
= (55 * 8 * 2.083) + (2.083 * Floor(3.167  + BitStuffTime(Data_bc))) +
Host_Delay

Isochronous Transfer (No Handshake)
= (38 * 8 * 2.083)  + (2.083 * Floor(3.167  + BitStuffTime(Data_bc))) +
Host_Delay
```

High-speed (Output)

```
Non-Isochronous Transfer (Handshake Included)
= (55 * 8 * 2.083)  + (2.083 * Floor(3.167  + BitStuffTime(Data_bc))) +
Host_Delay

Isochronous Transfer (No Handshake)
= (38 * 8 * 2.083)  + (2.083 * Floor(3.167  + BitStuffTime(Data_bc))) +
Host_Delay
```

Full-speed (Input)

```
Non-Isochronous Transfer (Handshake Included)
= 9107 + (83.54 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay

Isochronous Transfer (No Handshake)
= 7268 + (83.54 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay
```

Full-speed (Output)

```
Non-Isochronous Transfer (Handshake Included)
       = 9107 + (83.54 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay

Isochronous Transfer (No Handshake)
       = 6265 + (83.54 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay
```

Low-speed (Input)

```
       = 64060 + (2 * Hub_LS_Setup) +
             (676.67 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay
```

Low-speed (Output)

```
       = 64107 + (2 * Hub_LS_Setup) +
             (667.0 * Floor(3.167 + BitStuffTime(Data_bc))) + Host_Delay
```

Les temps de bus des équations susmentionnées sont représentés en nanosecondes et prennent en compte les délais de propagation dus à la distance entre l'appareil et l'hôte. Ces équations typiques peuvent être utilisées pour calculer le temps de bus; toutefois, différentes mises en application peuvent choisir d'utiliser des approximations grossières de ces temps.

Le temps de bus actuel pour une transaction donnée est pratiquement toujours inférieur à celui calculé parce que la surcharge de l'insertion binaire dépend des données. Le cas le plus défavorable de l'insertion binaire est égal à 1,1667 (7/6) fois le temps (c'est-à-dire, la fonction BitStuffTime multiplie Data_bc par 8*1,1667 dans les équations). Cela signifie qu'il existe presque toujours du temps inutilisé sur le bus (faisant l'objet de spécificités de modèle des données) après l'achèvement de toutes les transactions organisées périodiquement. Le temps de bus disponible à cause d'une moindre insertion binaire peut être utilisé tel que décrit à la Section 5.11.5.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Le terme Host_Delay utilisé dans les équations dépend du contrôleur hôte, du traducteur de transaction (TT) et du système et permet qu'un contrôleur hôte (ou TT) puisse demander du temps supplémentaire à cause des délais d'obtention d'accès à la mémoire ou d'autres dépendances de mise en application. Ce terme est incorporé à une mise en application de ces équations en utilisant les fonctions de gestion du transfert fournies par l'interface HCD. Ces équations sont généralement mises en application par une combinaison entre l'USBD et le logiciel HCD fonctionnant en coopération. Les résultats de ces calculs sont utilisés pour déterminer si un transfert ou la création d'un canal peut être pris(e) en charge dans une configuration USB donnée.

5.11.4 Calcul des tailles de mémoire tampon dans les fonctions et le logiciel

Le logiciel client et les fonctions doivent allouer un espace de mémoire tampon aux transactions de données en attente de leur tour sur le bus. Pour les canaux non isochrones, cet espace de mémoire tampon doit être suffisamment grand pour contenir le paquet de données suivant. Si plusieurs requêtes de transaction sont en attente pour un point d'extrémité donné, la mise en mémoire tampon de chaque transaction doit être fournie. Les méthodes de calcul de la mise en mémoire tampon précise, absolue et minimum, requises par une fonction à cause des interactions spécifiques définies entre son client logiciel et la fonction, ne relèvent pas du domaine d'application de la présente spécification.

Le contrôleur hôte s'attend à être en mesure de prendre en charge un nombre illimité de transactions en attente du bus soumis à la mémoire tampon du système disponible pour l'espace de la mémoire tampon et du descripteur, etc. Les contrôleurs hôtes sont autorisés à limiter le nombre de (micro)trames qui peut être demandé ultérieurement par une transaction.

Pour les canaux isochrones, la Section 5.12.4 décrit les détails qui ont un impact sur les exigences de mise en mémoire tampon du côté hôte et du côté appareil. En général, les mémoires tampons doivent être fournies de manière à contenir une quantité de données approximativement deux fois supérieure à celle qui peut être transférée en 1 ms pour les points d'extrémité à vitesse intégrale ou en 125 µs pour les points d'extrémité à haute vitesse.

5.11.5 Réclamation de bande passante de bus

La bande passante USB et l'accès au bus sont accordés sur la base d'un calcul du temps de transmission de bus dans le cas le plus défavorable et des latences requises. Toutefois, en raison des contraintes imposées aux différents types de transferts et du fait que la contribution en termes de temps de bus pour l'insertion binaire est calculée sous la forme d'une constante dépendante des données, il y a fréquemment une différence entre le temps de bus restant dans chaque (micro)trame et le temps de transmission d'une (micro)trame calculé. Afin de prendre en charge l'utilisation la plus efficace de la bande passante de bus, les transferts de contrôle et en bloc sont les candidats à déplacer sur le bus dès que le temps de bus est disponible. La prise en charge exacte de cette opération par un contrôleur hôte dépend de la mise en application. Un contrôleur hôte peut prendre en compte les types de transfert des IRP en attente et la connaissance spécifique à la mise en application du temps restant de (micro)trame pour la réutilisation de la bande passante réclamée.

5.12 Considérations spéciales pour les transferts isochrones

La prise en charge du mouvement des données isochrones entre l'hôte et un appareil est l'une des fonctionnalités prises en charge par l'USB. La livraison fiable des données isochrones sur l'USB exige une attention particulière aux détails. La responsabilité relative à une livraison fiable est partagée par plusieurs entités USB:

- Appareil/fonction
- Bus
- Contrôleur hôte
- Un ou plusieurs agents logiciels

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Etant donné que le temps est un élément clé d'un transfert isochrone, il est important que les concepteurs USB comprennent la manière dont le temps est géré par ces différentes entités dans le cadre de l'USB.

Note: les exemples de la présente section décrivent les USB pour un exemple impliquant des points d'extrémité à vitesse intégrale. Les détails généraux des exemples sont également adaptés aux points d'extrémité à haute vitesse lorsque les changements correspondants sont effectués; par exemple, la trame remplacée par une microtrame, 1 ms remplacée par 125 μs, les réglages de débit effectués entre la vitesse intégrale et la haute vitesse, etc.

Tous les appareils isochrones doivent indiquer leurs fonctionnalités sous la forme de descripteurs spécifiques à l'appareil. Il convient que ces fonctionnalités soient également fournies sous une forme pouvant être utilisée par le potentiel client pour décider si l'appareil offre une solution à son(ses) problème(s). Les fonctionnalités spécifiques à un appareil peuvent justifier les différences de prix.

Dans tous les systèmes de communication, le transmetteur et le récepteur doivent être suffisamment synchronisés pour délivrer des données robustes. Dans le cas d'un système de communication asynchrone, les données peuvent être délivrées de façon robuste en permettant au transmetteur de détecter que le récepteur n'a pas correctement reçu un élément de données et de tenter de nouveau la transmission des données.

Dans le cas d'un système de communication isochrone, le transmetteur et le récepteur doivent demeurer synchronisés du point de vue du temps et des données pour délivrer des données robustes. L'USB ne prend pas en charge les nouvelles tentatives de transmission des données isochrones afin que la bande passante minimale puisse être allouée aux transferts isochrones et que la synchronisation temporelle ne soit pas perdue à cause d'un délai de nouvelle tentative. Toutefois, il est primordial qu'une paire transmetteur/récepteur isochrone USB demeure toujours synchronisée dans le cas de transmission normale des données et dans les cas où des erreurs se produisent sur le bus.

Dans de nombreux systèmes qui traitent des données isochrones, une seule horloge globale est utilisée pour la synchronisation de toutes les entités du système. Un exemple d'un tel système est le RTPC (Réseau téléphonique commuté public). Etant donné qu'une grande variété de fréquences naturelles différentes peut être connectée à l'USB, aucune horloge individuelle ne peut fournir toutes les caractéristiques requises pour satisfaire les exigences de synchronisation de tous les appareils et du logiciel tout en prenant en charge les objectifs de coût des produits informatiques de marché de masse. L'USB définit un modèle d'horloge qui permet la coexistence d'une large gamme d'appareils sur le bus avec des mises en application à un coût raisonnable.

La présente section présente les options ou caractéristiques qui peuvent être utilisées par les points d'extrémité isochrones pour minimiser les différences de comportement entre une fonction qui n'est pas conçue pour l'USB et une version USB de la fonction. Un exemple est inclus pour illustrer les similarités et les différences entre les versions non USB et USB d'une fonction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Le reste de la section présente les concepts clés suivants:

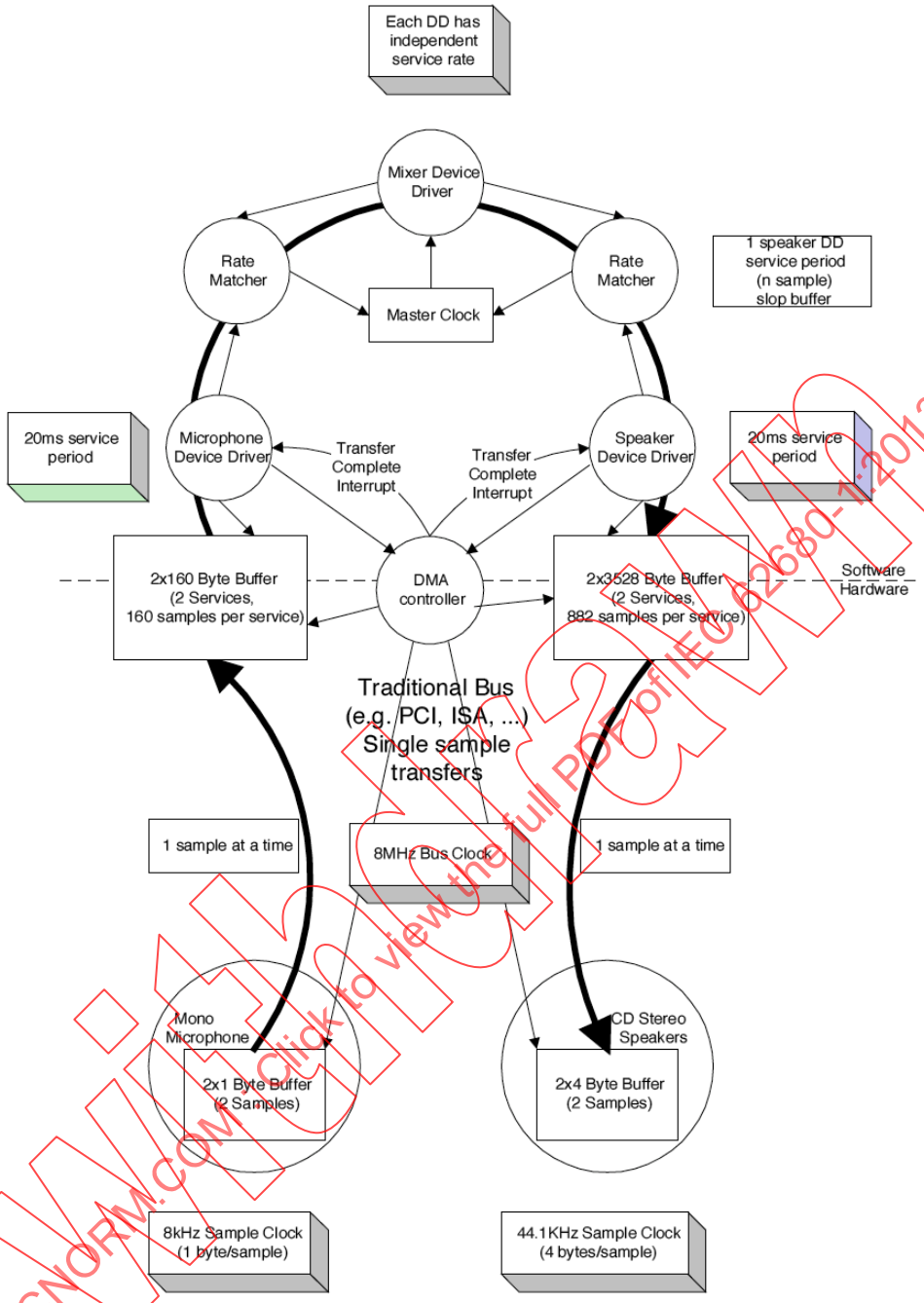
- Modèle d'horloge USB: les horloges présentes dans un système USB avec un impact sur les transferts de données isochrones
- Options de synchronisation de l'horloge horloge-à-fonction des (micro)trames USB: liaison de l'horloge des (micro)trames USB avec une horloge de fonction
- Suivi SOF: responsabilités et opportunités des points d'extrémité isochrones en matière de jeton SOF et de (micro)trames USB
- Mise en mémoire tampon préalable des données: exigences relatives à l'accumulation des données avant la génération, la transmission et la consommation
- Gestion des erreurs: détails spécifiques aux transferts isochrones pour la gestion des erreurs
- Mise en mémoire tampon pour l'adaptation du débit: équations qui peuvent être utilisées pour calculer l'espace de mémoire tampon requis pour les points d'extrémité isochrones

5.12.1 Exemple d'application isochrone non-USB

L'exemple utilisé est raisonnablement généralisé. D'autres cas plus simples ou plus complexes sont possibles et les caractéristiques USB pertinentes identifiées peuvent être ou non utilisées de manière appropriée.

L'exemple consiste en un microphone mono de 8 kHz connecté à un amplificateur-mixeur qui envoie le flux de données d'entrée à des haut-parleurs stéréo de 44 kHz. Le mixeur s'attend à ce que les données soient reçues et transmises avec un débit et un encodage d'échantillonnage. Un amplificateur adaptateur de débit d'entrée et de sortie convertit le débit et l'encodage d'échantillonnage du débit naturel et de l'encodage de l'appareil au débit et à l'encodage escomptés par le mixeur. La Figure 5-16 illustre cet exemple.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Each DD has independent service rate	Chaque DD à un débit de service indépendant
Mixer Device Driver	Pilote d'appareil mixeur
Rate Matcher	Adaptateur de débit
Master Clock	Horloge maître
1 speaker DD service period (n sample) slop buffer	1 mémoire tampon d'épreuve de période de service DD de haut-parleur (n échantillons)
20ms service period	Période de service de 20 ms
Microphone Device Driver	Pilote d'appareil de microphone
Transfer Complete Interrupt	Interruption complète de transfert
Speaker Device Driver	Pilote d'appareil de haut-parleur
2x160 Byte Buffer (2 Services, 160 samples per service)	Mémoire tampon de 2x160 octets (2 Services, 160 échantillons par service)
DMA Controller	Contrôleur DMA
2x3528 Byte Buffer (2 Services, 882 samples per service)	Mémoire tampon de 2x3 528 (2 Services, 882 échantillons par service)
Software/Hardware	Logiciel/Matériel

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Anglais	Français
Traditional Bus (e.g. PCI, ISA, ...) Single sample transfers	Bus traditionnel (par exemple PCI, ISA, ...) Transferts d'échantillons uniques
1 sample at a time	1 échantillon à la fois
8MHz Bus Clock	Horloge de bus de 8 MHz
Mono Microphone	Microphone mono
CD Stereo Speakers	Haut-parleurs stéréo CD
2x1 Byte Buffer (2 Samples)	Mémoire tampon de 2x1 (2 échantillons)
2x4 Byte Buffer (2 Samples)	Mémoire tampon de 2x4 (2 échantillons)
8kHz Sample Clock (1 byte/Sample)	Horloge d'échantillons de 8 kHz (1 octet/échantillon)
44.1KHz Sample Clock (4 bytes/Sample)	Horloge d'échantillons de 44,1 KHz (4 octets/échantillons)

Figure 5-16. Exemple de transfert isochrone non-USB

Une horloge maître (qui peut être fournie par un logiciel à partir de l'horodateur réel) de l'ordinateur est utilisée pour réveiller le mixeur afin de demander des données d'entrée à la source d'entrée et de fournir des données de sortie au destinataire de sortie. Dans cet exemple, supposer qu'il se réveille toutes les 20 ms. Le microphone et les haut-parleurs ont chacun leurs propres horloges d'échantillonnage qui ne sont pas synchronisées les unes par rapport aux autres ou par rapport à l'horloge maître du mixeur. Le microphone produit des données à son débit naturel (échantillons d'un octet, 8 000 fois par seconde) et les haut-parleurs consomment les données à leur débit naturel (échantillons de quatre octets, 44 100 fois par seconde). Les trois horloges du système peuvent connaître des phénomènes de dérive et de gigue les unes par rapport aux autres. Chaque adaptateur de débit peut également fonctionner avec un débit naturel différent de celui de l'amplificateur-mixeur, de la source/amplificateur d'entrée ou du destinataire/amplificateur de sortie.

Les adaptateurs de débit surveillent également le débit des données à long terme de leur appareil comparé à l'horloge maître du mixeur et interpolent un échantillon supplémentaire ou fusionnent deux échantillons pour ajuster le débit de données de leur appareil à celui du mixeur. Ce réglage peut être nécessaire toutes les deux secondes, mais il se produit généralement que très rarement. Les adaptateurs de débit fournissent une mise en mémoire supplémentaire pour réaliser une correspondante de débit.

Note: une autre application peut ne pas être en mesure de tolérer un réglage d'échantillonnage et peut avoir besoin d'autres moyens pour la prise en charge de la dérive horloge maître-à-appareil ou également pour la synchronisation des horloges afin de s'assurer qu'aucune dérive ne peut se produire.

Le mixeur s'attend toujours à recevoir exactement une période de service de données (période de service de 20 ms) de la part de son appareil d'entrée et à produire exactement une période de service de données pour son appareil de sortie. Le mixeur peut être retardé pour une durée inférieure à une période de service en cas d'indisponibilité des données ou de l'espace de son appareil d'entrée/sortie. Le mixeur suppose que de tels délais ne s'accumulent pas.

Les appareils d'entrée et de sortie ainsi que leurs amplificateurs s'attendent à pouvoir placer/obtenir des données en réponse à une interruption matérielle du contrôleur DMA lorsque leur transducteur a traité une période de service de données. Ils s'attendent à placer/obtenir exactement une période de service de données. L'appareil d'entrée produit 160 octets (dix échantillons) pour chaque période de service de 20 ms. L'appareil de sortie consomme 3 528 octets (882 échantillons) pour chaque période de service de 20 ms. Le contrôleur DMA peut déplacer un seul échantillon entre l'appareil et la mémoire tampon de l'hôte à un débit beaucoup plus rapide que le débit d'échantillons d'un des appareils.

Les amplificateurs des appareils d'entrée et de sortie fournissent deux périodes de service pour la mise en mémoire tampon du système. Une mémoire tampon est toujours traitée par le contrôleur DMA. Il est garanti que l'autre mémoire tampon est prête avant que la mémoire tampon actuelle ne soit épuisée. Lorsque la mémoire tampon actuelle est vidée, l'interruption matérielle active l'amplificateur de l'appareil et demande à l'adaptateur de débit de lui

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

attribuer la mémoire tampon. L'amplificateur de l'appareil demande un nouvel IRP avec la mémoire tampon avant que la mémoire tampon actuelle ne soit épuisée.

Les appareils peuvent fournir deux échantillons de mise en mémoire tampon de données pour s'assurer qu'ils ont toujours un échantillon à traiter sur la période d'échantillonnage suivante pendant que le système réagit à l'échantillon précédent/suivant.

Les périodes de service des amplificateurs sont choisies pour survivre aux variations de latence d'interruption qui peuvent apparaître dans l'environnement du système d'exploitation. Différents environnements de système d'exploitation nécessitent différentes périodes de service pour un fonctionnement fiable. Les périodes de service sont également sélectionnées pour placer une charge d'interruption minimum sur le système parce qu'il se peut qu'un autre logiciel du système nécessite du temps de traitement.

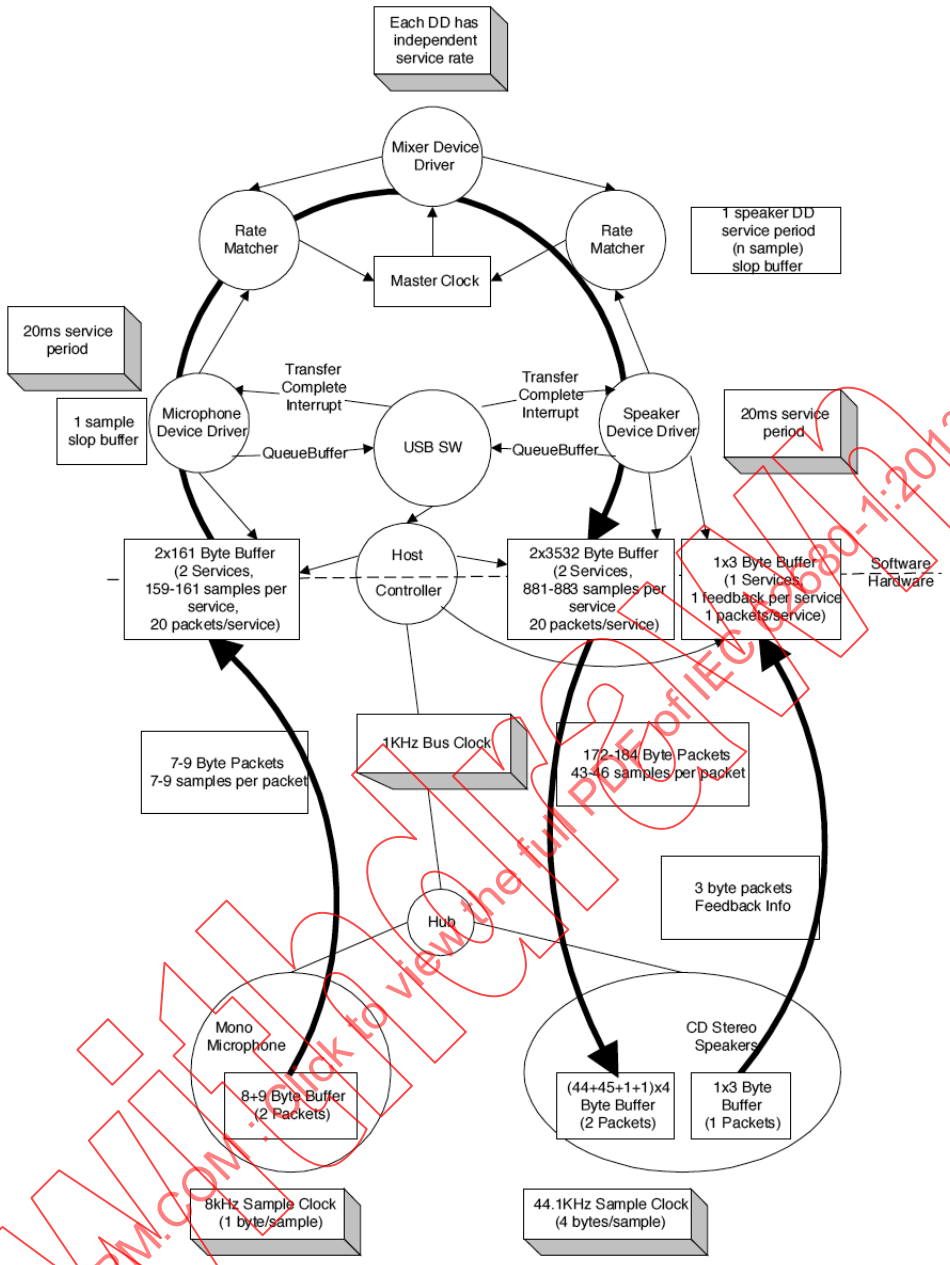
5.12.2 Modèle d'horloge USB

Le temps est présent dans le système USB par l'intermédiaire d'horloges. En fait, il existe dans un système USB plusieurs horloges qui doivent être comprises:

- Horloge d'échantillonnage: cette horloge détermine le débit de données naturel des échantillons qui se déplacent entre le logiciel client de l'hôte et la fonction. Il n'est pas nécessaire que cette horloge soit différente entre les mises en application non USB et USB.
- Horloge de bus: Cette horloge fonctionne avec une période de 1 000 ms (fréquence égale à 1 kHz) sur les segments à vitesse intégrale et de 125 000 µs (fréquence égale à 8 kHz) sur les segments à haute vitesse du bus et est indiquée par le débit des paquets SOF sur le bus. Cette horloge est en quelque sorte équivalente à l'horloge de 8 MHz présentée dans l'exemple non USB. Dans le cas de l'USB, l'horloge de bus est souvent une horloge de fréquence inférieure à celle de l'horloge d'échantillonnage, tandis que l'horloge de bus est presque toujours une horloge de fréquence supérieure à celle de l'horloge d'échantillonnage d'un cas non-USB.
- Horloge de service: Cette horloge est déterminée par le débit auquel le logiciel client fonctionne pour le service des IRP qui peuvent s'être accumulés entre chaque exécution. Cette horloge peut également être identique dans les cas USB et non-USB.

Dans la plupart des systèmes d'exploitation existants, il n'est pas possible de prendre en charge une large gamme de flux de communication isochrones si chaque amplificateur d'appareil doit être interrompu pour chaque échantillon en cas de débits d'échantillonnage rapides. Par conséquent, plusieurs échantillons, voire plusieurs paquets, sont traités par le logiciel client, puis envoyés au contrôleur hôte pour le séquencement sur le bus conformément aux exigences d'accès au bus négociées à l'avance. La Figure 5-17 présente l'exemple d'un environnement d'horloge USB raisonnablement équivalent à l'exemple non-USB de la Figure 5-16.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Légende

Anglais	Français
Each DD has independent service rate	Chaque DD à un débit de service indépendant
Mixer Device Driver	Pilote d'appareil mixeur
Rate Matcher	Adaptateur de débit
Master Clock	Horloge maître
1 speaker DD service period (n sample) slop buffer	1 mémoire tampon d'épreuve de période de service DD de haut-parleur (n échantillons)
20ms service period	Période de service de 20 ms
1 sample slop buffer	1 mémoire tampon d'épreuve d'un échantillon
Microphone Device Driver	Pilote d'appareil de microphone
Transfer Complete Interrupt	Interruption complète de transfert
QueueBuffer	Mémoire tampon de queue
Speaker Device Driver	Pilote d'appareil de haut-parleur
2x161 Byte Buffer (2 Services, 159-161 samples per service, 20 packets/service)	Mémoire tampon de 2x161 (2 services, 159-161 échantillons par service, 20 paquets/service)
Host	Hôte
Controller	Contrôleur
2x3532 Byte Buffer (2 Services, 881-883 samples per service, 20 packets/service)	Mémoire tampon de 2x3 532 (2 services, 881-883 échantillons par service, 20 paquets/service)
1x3 Byte Buffer (1 Services, 1 feedback	Mémoire tampon de 1x3 (1 service, 1 rétroaction par

Anglais	Français
per service/ 1 packets/service)	service/ 1 paquet/service)
Software/Hardware	Logiciel/Matériel
7-9 Byte Packets 7-9 samples per packet	Paquets de 7-9 octets 7-9 échantillons par paquet
1KHz Bus Clock	Horloge de bus de 1KHz
172-184 Byte Packets 43-46 samples per packet	Paquets de 172-184 octets 43-46 échantillons par paquet
3 byte packets Feedback Info	Infos de rétroaction avec paquets de 3 octets
Mono Microphone	Microphone mono
CD Stereo Speakers	Haut-parleurs stéréo CD
8+9 Byte Buffer (2 packets)	Mémoire tampon de 8+9 (2 paquets)
(44+45+1+1)x4 Byte Buffer (2 packets)	Mémoire tampon de (44+45+1+1)x4 (2 paquets)
1x3 Byte Buffer (1 packets)	Mémoire tampon de 1x3 (1 paquet)
8kHz Sample Clock (1 byte/Sample)	Horloge d'échantillons de 8 kHz (1 octet/échantillon)
44.1KHz Sample Clock (4 bytes/Sample)	Horloge d'échantillons de 44,1 KHz (4 octets/échantillons)

Figure 5-17. Application USB isochrone et à vitesse intégrale

La Figure 5-17 illustre un chemin typique d'aller et retour d'informations depuis un microphone (appareil d'entrée) vers un haut-parleur (appareil de sortie). Les horloges, les paquets et la mise en mémoire tampon impliqués sont également illustrés. La Figure 5-17 est explorée dans de plus amples détails aux sections suivantes.

L'objectif de cet exemple est d'identifier les différences introduites par l'USB par rapport à l'exemple précédent non-USB. Les différences résident dans les zones de mise en mémoire tampon, la synchronisation compte tenu de l'existence d'une horloge de bus USB et le délai. Le logiciel client au-dessus des amplificateurs d'appareil peut ne pas être affecté dans la plupart des cas.

5.12.3 Synchronisation d'horloge

Afin d'assurer une manipulation fiable des données isochrones, les trois horloges identifiées ci-dessus doivent être synchronisées de la même manière. Si les horloges ne sont pas synchronisées, plusieurs attributs d'horloge-à-horloge indésirables peuvent apparaître:

- Dérive d'horloge: deux horloges qui fonctionnent à la même vitesse nominale peuvent, en fait, avoir des différences de mise en application, ce qui peut avoir pour conséquence une horloge fonctionnant plus rapidement ou plus lentement que l'autre sur de longues périodes de temps. Si elle est incorrecte, cette différence d'une horloge par rapport à l'autre peut aboutir à une quantité de données trop importante ou trop faible lorsque les données sont censées être toujours présentes au moment requis.
- Gigue d'horloge: une horloge peut varier sa fréquence au fil du temps à cause des changements de température, etc. Elle peut également changer lorsque les données sont réellement délivrées comparé à lorsqu'elles sont censées être délivrées.
- Différences de la phase d'horloge à horloge: Si deux horloges ne sont pas verrouillées en phase, différentes quantités de données peuvent être disponibles à différents moments étant donné que la fréquence de battement des horloges change au fil du temps. Cela peut mener à la quantification/l'échantillonnage des artéfacts associés.

L'horloge de bus fournit une horloge centrale à l'aide de laquelle les appareils matériels USB et le logiciel peuvent se synchroniser à un degré ou à un autre. Toutefois, le logiciel n'est pas en mesure de verrouiller la phase ou la fréquence de façon précise par rapport à l'horloge de bus étant donné la prise en charge actuelle de la programmation "en temps réel" de la plupart des systèmes d'exploitation d'ordinateur. Le logiciel s'exécutant sur l'hôte peut, cependant, savoir que les données déplacées sur l'USB sont mises en paquets. Pour les types de transfert isochrones, une unité de données est déplacée exactement une seule fois par (micro)trame et l'horloge de (micro)trame est raisonnablement précise. Ces informations lui étant fournies, le logiciel peut régler la quantité de données qu'il traite par rapport à la durée de (micro)trame qui s'est écoulée.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Note: pour les points d'extrémité à haute vitesse et à large bande passante, les données échangées entre deux ou trois transactions par microtrame sont encore considérées comme appartenant au même "paquet unique". Une grande quantité de données par paquet est divisée entre deux ou trois transactions uniquement pour des raisons d'efficacité de bus.

5.12.4 Appareils isochrones

L'USB inclut une infrastructure pour les appareils isochrones qui définit les types de synchronisation, la manière dont les points d'extrémité isochrones fournissent une rétroaction sur le débit des données et la façon dont ils peuvent se connecter ensemble. Les appareils isochrones comprennent des appareils analogiques d'échantillonnage (par exemple, les appareils audio et téléphoniques) et des appareils de données synchrones. Le type de synchronisation classe un point d'extrémité selon sa fonctionnalité à synchroniser son débit de données par rapport à celui du point d'extrémité auquel il est connecté. La rétroaction est fournie en indiquant de manière précise le débit de données requis relatif à la fréquence SOF. La capacité à effectuer des connexions dépend de la qualité de la connexion requise, du type de synchronisation du point d'extrémité et des fonctionnalités de l'application hôte qui réalise la connexion. Des informations supplémentaires, spécifiques à la classe de l'appareil, peuvent être requises en fonction de l'application.

Note: le terme "données" est utilisé de façon très générale et peut faire référence aux données qui représentent des informations analogiques échantillonnées (comme l'audio) ou à des informations plus abstraites. Le "débit de données" se réfère au débit auquel les informations analogiques sont échantillonnées ou au débit auquel les données sont cadencées.

Les informations suivantes sont requises afin de déterminer la méthode de connexion des points d'extrémité isochrones:

- Type de synchronisation:
  - Asynchrone: non synchronisée, bien que les destinataires fournissent une rétroaction sur le débit de données
  - Synchrone: synchronisée au SOF de l'USB
  - Adaptative: synchronisée grâce à la rétroaction ou à la prédiction des informations relatives au débit des données
- Débits de données disponibles
- Formats de données disponibles

Les informations concernant le type de synchronisation et le débit de données sont nécessaires pour déterminer s'il y a une correspondance exacte entre le débit de données de la source et celui du destinataire, ou s'il existe un processus de conversion acceptable qui pourrait permettre à la source de se connecter au destinataire. Il est de la responsabilité de l'application de déterminer si la connexion peut être prise en charge dans le cadre des ressources de traitement disponibles et d'autres contraintes (comme le retard). Les classes d'appareil USB spécifiques définissent la façon de décrire les informations concernant le type de synchronisation et le débit de données.

La correspondance du format et la conversion des données sont également requises pour une connexion, mais ce n'est pas l'unique exigence relative aux connexions isochrones. Les détails concernant la conversion du format peuvent être trouvés dans d'autres documents liés aux formats spécifiques.

5.12.4.1 Type de synchronisation

Trois types de synchronisation distincts sont définis. Le Tableau 5-12 présente une vue d'ensemble sur les caractéristiques de synchronisation des points d'extrémité de la source et du destinataire. Les types sont présentés afin d'améliorer la fonctionnalité.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 5-12. Caractéristiques de synchronisation

	Source	Destinataire
Asynchrone	Fréquence indépendante Fs Fournit une prédiction implicite (flux de données)	Fréquence indépendante Fs Fournit une rétroaction explicite (canal isochrone)
Synchrone	Fs verrouillé sur SOF Utilise une rétroaction implicite (SOF)	Fs verrouillé sur SOF Utilise une rétroaction implicite (SOF)
Adaptative	Fs verrouillé sur le destinataire Utilise une rétroaction explicite (canal isochrone)	Fs verrouillé sur le flux de données Utilise une prédiction implicite (flux de données)

5.12.4.1.1 Asynchrone

Les points d'extrémité asynchrones ne peuvent pas se synchroniser au SOF ni à une autre horloge du domaine USB. Elles représentent la source ou le destinataire d'un flux de données isochrone avec un débit de données fixe (points d'extrémité sur une fréquence), un nombre limité de débits de données (32 kHz, 44,1 kHz, 48 kHz, …) ou un débit de données programmable en continu. Si le débit de données est programmable, il est réglé pendant l'initialisation du point d'extrémité isochrone. Les appareils asynchrones doivent indiquer leurs fonctionnalités de programmation dans le descripteur de point d'extrémité spécifique à la classe, tel que décrit dans leur spécification de classe d'appareil. Le débit de données est verrouillé sur une horloge externe à l'USB ou sur une horloge interne en fréquence indépendante. Ces appareils placent le fardeau de la correspondance du débit de données à un autre emplacement de l'environnement USB. Les points d'extrémité de source asynchrone transportent les informations relatives à leur débit de données de manière implicite dans le nombre d'échantillons qu'ils produisent par (micro)trame. Les points d'extrémité de destinataire asynchrone doivent fournir des informations de rétroaction explicites à un amplificateur adaptatif (voir Section 5.12.4.2).

Un exemple de source asynchrone est un lecteur de CD audio qui fournit ses données sur la base d'une horloge ou d'un résonateur externe. Un autre exemple est un récepteur de radiodiffusion sonore numérique (DAB) ou un récepteur satellite numérique (DSR). De la même manière, le débit d'échantillonnage est réglé du côté radiodiffusion et se trouve sous le contrôle de l'USB.

Les points d'extrémité de destinataire asynchrone peuvent être des haut-parleurs de faible coût fonctionnant grâce à leur horloge d'échantillonnage interne.

5.12.4.1.2 Synchrone

Les points d'extrémité peuvent avoir leur propre système d'horloge (notion de temps) contrôlé de l'extérieur par le biais de la synchronisation SOF. Ces points d'extrémité doivent asservir leur horloge d'échantillonnage à la coche SOF de 1 ms (à l'aide d'une PPL programmable). Pour les points d'extrémité à haute vitesse, la présence du SOF de microtrame peut être utilisée pour un meilleur suivi de l'horloge de trame.

Les points d'extrémité synchrones peuvent représenter la source ou le destinataire des flux de données isochrones avec un débit de données fixe (points d'extrémité sur une fréquence), un nombre limité de débits de données (32 kHz, 44,1 kHz, 48 kHz, …) ou un débit de données programmable en continu. S'il est programmable, le débit de données de fonctionnement est réglé pendant l'initialisation du point d'extrémité isochrone. Le nombre d'échantillons ou d'unités de données généré(e)s dans une série de (micro)trames USB est déterministe et périodique. Les appareils synchrones doivent indiquer leurs fonctionnalités de programmation dans le descripteur de point d'extrémité spécifique à la classe, tel que décrit dans leur spécification de classe d'appareil.

Un exemple de source synchrone est un microphone numérique qui synthétise son horloge d'échantillonnage à partir du SOF et produit un nombre fixe d'échantillons audio pour chaque (micro)trame USB. De même, un destinataire synchrone dérive son horloge d'échantillonnage à partir du SOF et consomme un nombre fixe d'échantillons pour chaque (micro)trame USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

5.12.4.1.3 Adaptative

Les points d'extrémité adaptatifs sont les points d'extrémité les plus capables possible. Ils peuvent jouer le rôle de source ou de destinataire de données avec n'importe quel débit dans leur gamme opérationnelle. Les points d'extrémité sources adaptatifs produisent des données avec un débit contrôlé par le destinataire des données. Le destinataire fournit une rétroaction (voir Section 5.12.4.2) à la source, ce qui permet à cette dernière de connaître le débit de données souhaité par le destinataire. Pour les points d'extrémité destinataires, les informations relatives au débit de données sont intégrées dans le flux de données. Le nombre moyen d'échantillons reçus pendant une certaine durée moyenne détermine le débit de données instantané. Si ce nombre change durant le fonctionnement, le débit de données est réglé en conséquence.

La gamme opérationnelle du débit de données peut être axée autour d'un débit (par exemple, 8 kHz), choisir parmi plusieurs débits de données programmables ou à détection automatique (32 kHz, 44,1 kHz, 48 kHz, ...) ou se situer entre un ou plusieurs intervalles (par exemple, 5 kHz à 12 kHz ou 44 kHz à 49 kHz). Les appareils adaptatifs doivent indiquer leurs fonctionnalités de programmation dans le descripteur de point d'extrémité spécifique à la classe, tel que décrit dans leur spécification de classe d'appareil.

Un exemple de source adaptative est un lecteur CD qui contient un convertisseur de débit d'échantillonnage (SRC) entièrement adaptatif afin qu'il ne soit plus nécessaire que la fréquence d'échantillonnage de sortie soit égale à 44,1 kHz, mais qu'elle puisse avoir n'importe quelle valeur dans la gamme opérationnelle du SRC. Les destinataires adaptatifs comprennent des points d'extrémité tels que des haut-parleurs numériques haut de gamme, des combinés, etc.

5.12.4.2 Rétroaction

Un destinataire asynchrone doit fournir une rétroaction explicite à l'hôte en indiquant de manière précise la valeur souhaitée pour son débit de données (F_f) par rapport à la fréquence de (micro)trame USB. Cela permet à l'hôte de régler de manière continue le nombre d'échantillons envoyés au destinataire de sorte qu'aucun engorgement ou débordement ne se produise au niveau de la mémoire tampon des données. De même, une source adaptative doit recevoir une rétroaction explicite de la part de l'hôte afin qu'elle puisse générer le nombre d'échantillons requis par l'hôte. Les points d'extrémité de rétroaction peuvent être spécifiés tels que décrits à la Section 9.6.6 du champ *bmAttributes* du descripteur de point d'extrémité.

Pour générer le débit de données souhaité F_f, l'appareil doit mesurer son débit d'échantillonnage actuel F_s, référencé par rapport à la notion de temps de l'USB, c'est-à-dire la fréquence de (micro)trame USB. Cette spécification exige la résolution du débit de données F_f à vitesse supérieure à un échantillon par seconde (1Hz) afin de permettre la création d'un débit source de grande qualité et tolérer des retards et des erreurs dans la boucle de rétroaction. Pour atteindre cette précision, le temps de mesure T_{meas} doit avoir une valeur au moins égale à 1 seconde. En conséquence:

$$T_{meas} = 2^K$$

où T_{meas} est maintenant exprimé en (micro)trames USB et K=10 pour les appareils à vitesse intégrale (trames de 1 ms) et K=13 pour les appareils à haute vitesse (microtrames de 125 µs). Toutefois, pour la plupart des appareils, le débit d'échantillonnage actuel F_S dérive d'une horloge maître F_m par l'intermédiaire d'un diviseur binaire. En conséquence:

$$F_m = F_s * 2^P$$

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

où *P* est un entier positif (y compris 0 si aucune autre horloge maître à fréquence supérieure n'est disponible). Le temps de mesure T_{meas} peut maintenant être diminué en mesurant F_m à la place de F_s et:

$$T_{meas} = \frac{2^K}{2^P} = 2^{(K-P)}$$

De cette manière, une nouvelle estimation pour F_f devient disponible toutes les $2^{(K-P)}$ (micro)trames. *P* est pratiquement tenu de se trouver dans l'intervalle [0,K] parce qu'il serait inutile d'utiliser une horloge moins rapide que F_s (*P*=0) ni d'essayer de mettre à jour F_f plus d'une fois par (micro)trame (*P*=K). Un destinataire peut déterminer F_f en comptant les cycles de l'horloge maître F_m sur une période de $2^{(K-P)}$(micro)trames. Le compteur est lu en F_f et réinitialisé toutes les $2^{(K-P)}$(micro)trames. Tant qu'aucun cycle d'horloge n'est sauté, le compteur est précis à long terme.

Pour chaque (micro)trame, une source adaptative ajoute F_f à tout comptage d'échantillons fractionnaire restant de la (micro)trame précédente, fournit le nombre d'échantillons dans la partie entière de la somme et conserve le comptage d'échantillons fractionnaire pour la (micro)trame suivante. La source peut rechercher le comportement de F_f sur de nombreuses (micro)trames afin de déterminer un débit encore plus précis, le cas échéant.

F_f est exprimé en nombre d'échantillons par (micro)trame. La valeur F_f est composée d'une partie entière qui représente le nombre (entier) d'échantillons par (micro)trame et d'une partie fractionnaire qui représente la "fraction" d'un échantillon nécessaire pour la correspondance entre la fréquence d'échantillonnage F_s et une résolution supérieure ou égale à 1 Hz. La partie fractionnaire nécessite au moins K bits pour représenter la "fraction" d'un échantillon avec une résolution supérieure ou égale à 1 Hz. La partie entière doit disposer d'un nombre de bits suffisant pour représenter le nombre maximum d'échantillons qui peuvent se produire dans une seule (micro)trame. En supposant que la taille minimum d'échantillon est égale à un octet, ce nombre est limité à 1 023 pour les points d'extrémité à vitesse intégrale. Par conséquent, dix bits sont suffisants pour coder cette valeur. Pour les points d'extrémité à haute vitesse, ce nombre est limité à 3*1 024=3 072 et douze bits sont requis.

En résumé, pour les points d'extrémité à vitesse intégrale, la valeur F_f doit être codée dans un format non signé 10.10 (*K*=10) qui s'inscrit dans trois octets. Etant donné que la valeur entière maximum est fixée à 1 023, le nombre 10.10 est aligné à gauche dans les 24 bits afin d'avoir le format 10.14. Seuls les dix premiers bits après le point binaire sont requis. Les quatre bits de poids faible peuvent être utilisés en option pour élargir la précision de F_f, sinon ils doivent être mis sur zéro. Pour les points d'extrémité à haute vitesse, la valeur F_f doit être codée dans un format non signé 12.13 (*K*=13) qui s'inscrit dans quatre octets. La valeur doit être alignée dans ces quatre octets afin que le point binaire soit situé entre le second et le troisième octet pour d'avoir le format 16.16. Les quatre octets de poids fort doivent être mis sur zéro. Seuls les 13 premiers bits après le point binaire sont requis. Les trois bits de poids faible peuvent être utilisés en option pour élargir la précision de F_f, sinon ils doivent être mis sur zéro.

Un point d'extrémité doit uniquement mettre en application le nombre de bits dont il a réellement besoin pour sa valeur F_f maximum.

Le choix de *P* dépend du point d'extrémité. Utiliser les directives suivantes pour choisir *P*:

- *P* doit se trouver dans l'intervalle [0,K].

ECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Des valeurs supérieures à *P* sont préférées parce qu'elles réduisent la taille du compteur de trames et augmentent le débit auquel *Ff* est mis à jour. Des mises à jour plus fréquentes ont pour résultat un contrôle plus strict du débit de données source, ce qui réduit l'espace de mémoire tampon requis pour traiter les changements de *Ff*.
- Il convient que *P* soit inférieur à *K* afin que *Ff* ait une valeur moyenne sur au moins deux trames pour réduire les effets de gigue du SOF.
- Il convient que *P* ne soit pas égal à zéro afin de conserver la différence du nombre d'échantillons fournis en dessous de 1 dans le cas d'une valeur *Ff* perdue.

Les transferts isochrones sont utilisés pour lire *Ff* à partir du registre de rétroaction. Il convient que le débit souhaité pour l'établissement de rapports de la rétroaction soit égal à $2^{(K-P)}$trames. *Ff* est consigné au moins une fois par période de mise à jour. Il n'y a aucun avantage à consigner la même valeur *Ff* plus d'une fois par période de mise à jour. Le point d'extrémité peut choisir de consigner *Ff* uniquement si la valeur mise à jour est différente de la valeur *Ff* précédente. Si la valeur n'a pas changé, le point d'extrémité peut consigner la valeur *Ff* actuelle ou une charge utile de données de longueur nulle. Il est fortement recommandé pour un point d'extrémité de toujours consigner la valeur *Ff* actuelle à chaque fois qu'il est interrogé.

Il est possible que la source délivre un nombre trop important ou trop faible d'échantillons pendant une longue période à cause des erreurs ou des inexactitudes accumulées par la mesure de *Ff*. Le destinataire doit disposer d'une fonctionnalité de mémoire tampon suffisante pour satisfaire à cette condition. Lorsque le destinataire reconnaît cette condition, il convient qu'il ajuste la valeur *Ff* consignée pour la corriger. Cela peut également se révéler nécessaire pour compenser les dérives d'horloge relatives. La mise en application de ce processus de correction dépend du point d'extrémité et n'est pas spécifiée.

5.12.4.3 Rétroaction implicite

Dans certains cas, la mise en application d'un point d'extrémité explicite séparé peut être évitée. Si un appareil met en application un groupe de points d'extrémité de données isochrone qui sont étroitement liés et si:

- Tous les points d'extrémité du groupe sont synchronisés (c'est-à-dire, ils utilisent des horloges d'échantillonnage qui dérivent d'une horloge maître commune)
- Le groupe contient un ou plusieurs points d'extrémité de données isochrones dans une direction qui nécessite normalement une rétroaction explicite
- Le groupe contient au moins un point d'extrémité de données isochrone dans la direction opposée

Dans ces circonstances, l'appareil peut choisir de ne pas mettre en application un point d'extrémité de rétroaction explicite isochrone. Au contraire, les informations de rétroaction peuvent dériver du point d'extrémité de données dans la direction opposée en observant son débit de données.

Deux cas peuvent apparaître:

- Un ou plusieurs points d'extrémité destinataires asynchrones sont accompagnés d'un point d'extrémité source asynchrone. Le débit de données du point d'extrémité source peut être utilisé comme une information de rétroaction explicite pour ajuster le débit de données du(des) point(s) d'extrémité destinataire(s).
- Un ou plusieurs points d'extrémité sources adaptatifs sont accompagnés d'un point d'extrémité destinataire adaptatif. Le point d'extrémité source peut régler son débit de

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

données sur la base du débit de données reçu en provenance du point d'extrémité destinataire.

Cette spécification fournit l'infrastructure nécessaire à la mise en application de la synchronisation décrite ci-dessus (voir Chapitre 9). Toutefois, la méthode exacte dont le débit de données souhaité *Ff* dérive du débit de données du point d'extrémité de rétroaction implicite dépend de la mise en application.

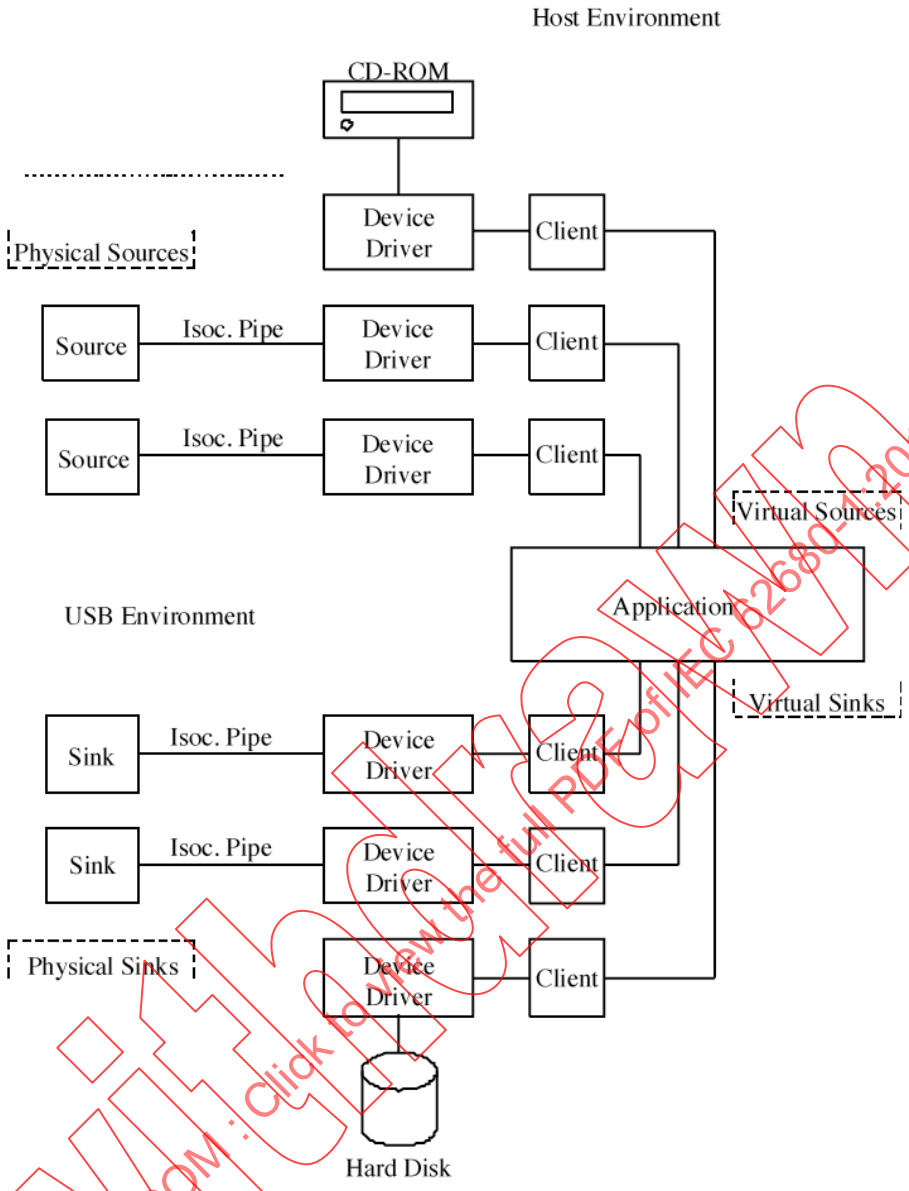
5.12.4.4 Connectivité

Un modèle d'interconnexion est présenté afin de décrire le processus de connectivité entre la source et le destinataire de manière exhaustive. Le modèle indique les différentes composantes impliquées et la manière dont elles interagissent pour établir la connexion.

Le modèle répond à des situations de sources/destinataires multiples. La Figure 5-18 illustre une situation typique (fortement condensée et incomplète). Un appareil physique est connecté au logiciel de l'application hôte par l'intermédiaire de différentes couches matérielles et logicielles, tel que décrit dans la présente spécification.

Au niveau de l'interface client, un appareil virtuel est présenté à l'application. Du point de vue de l'application, seuls les appareils virtuels existent. Le pilote d'appareil et le logiciel client sont les seuls à même de décider de la relation exacte qui existe entre l'appareil physique et virtuel.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Host Environment	Environnement hôte
Physical Sources	Sources physiques
Device Driver	Pilote d'appareil
Client	Client
Source	Source
Isoc. Pipe	Canal Isoc.
Virtual Sources	Sources virtuelles
USB Environment	Environnement USB
Application	Application
Virtual Sinks	Destinataires virtuels
Physical Sinks	Destinataires physiques
Hard Disk	Disque dur

Figure 5-18. Exemple de connectivité source/destinataire

Les fabricants de appareils (ou les vendeurs de systèmes d'exploitation) doivent fournir les logiciels de pilote d'appareil et d'interface client nécessaires pour convertir leur appareil d'une mise en application physique à une mise en application logicielle conforme à l'USB (l'appareil virtuel). Comme déjà indiqué, selon les fonctionnalités du logiciel, l'appareil virtuel peut faire preuve d'un comportement de synchronisation différent de celui de l'appareil physique. Toutefois, la classification de la synchronisation s'applique tout autant aux appareils

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

physiques et virtuels. Tous les appareils physiques appartiennent à l'un des trois types de synchronisation possibles. Par conséquent, les fonctionnalités qui doivent être intégrées au pilote d'appareil et/ou au logiciel client sont les mêmes que celles d'un appareil physique. Le terme "application" doit être remplacé par "pilote d'appareil/logiciel client". Dans le cas de la connexion d'une source physique à une source virtuelle, "appareil source virtuel" doit être remplacé par "appareil source physique" et "appareil destinataire virtuel" doit être remplacé par "appareil source virtuel". Dans le cas de la connexion d'un destinataire virtuel à un destinataire physique, "appareil source virtuel" doit être remplacé par "appareil destinataire virtuel" et "appareil destinataire virtuel" doit être remplacé par "appareil destinataire physique".

Le placement de la fonctionnalité d'adaptation de débit (RA) dans la couche pilote appareil/logiciel client a l'avantage d'isoler toutes les applications, soulageant l'appareil des spécificités et des problèmes associés à l'adaptation du débit. Les applications qui, autrement, auraient eu des débits multiples deviennent de simples systèmes à débit unique.

Note: le modèle n'est pas uniquement limité aux appareils USB. Par exemple, un lecteur CD-ROM contenant une audio de 44,1 kHz peut se manifester comme une source asynchrone, synchrone ou adaptative. Une opération asynchrone signifie que le CD-ROM remplit sa mémoire tampon au débit avec lequel il lit les données d'un disque, et le pilote vide la mémoire tampon conformément à son intervalle de service USB. Une opération synchrone signifie que le pilote utilise l'intervalle de service USB (par exemple, 10 ms) et un débit d'échantillonnage nominal des données (44,1 kHz) pour déterminer la sortie de 441 échantillons sur tous les intervalles de service USB. Une opération adaptative se matérialiserait dans un convertisseur de débit d'échantillonnage pour faire correspondre le débit de sortie du CD-ROM aux différents débits d'échantillonnage du destinataire.

A l'aide de ce modèle de référence, il est possible de définir les opérations nécessaires pour établir des connexions entre différentes sources et différents destinataires. De plus, le modèle indique le niveau auquel ces opérations doivent ou peuvent avoir lieu. Tout d'abord, il y a l'étape durant laquelle les appareils physiques sont mis en correspondance par rapport aux appareils virtuels et vice versa. Cela est accompli par le pilote et/ou le logiciel client. En fonction des fonctionnalités incluses dans ce logiciel, un appareil physique peut être transformé en un appareil virtuel d'un type de synchronisation entièrement différent. La deuxième étape est l'application qui utilise les appareils virtuels. Le placement des fonctionnalités de correspondance du débit au niveau du pilote/client de la pile logicielle soulage les applications qui communiquent avec les appareils virtuels du fardeau de la réalisation de la correspondance du débit pour chaque appareil qui y est connecté. Une fois que les caractéristiques liées à l'appareil virtuel sont décidées, les caractéristiques actuelles de l'appareil ne sont plus aussi intéressantes que les caractéristiques actuelles de l'appareil physique d'un autre pilote.

Par exemple, soit une application de mixeur qui se connecte du côté source à différentes sources, chacune fonctionnant avec ses propres fréquences et horloges. Avant que le mixage ne puisse avoir lieu, tous les flux doivent être convertis à une fréquence commune et verrouillés sur une référence d'horloge commune. Cette action peut être réalisée dans la couche de mise en correspondance physique à virtuelle ou traitée par l'application elle-même pour chaque appareil source de manière indépendante. Des actions similaires doivent être réalisées du côté destinataire. Si l'application envoie le flux de données mixé vers différents appareils destinataires, elle peut soit effectuer elle-même la mise en correspondance du débit de chaque appareil, soit compter sur le pilote/logiciel client pour ce faire, si possible.

Le Tableau 5-13 indique au niveau des intersections les actions que l'application doit entreprendre pour connecter un point d'extrémité source à un point d'extrémité destinataire.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 5-13. Exigences de connexion

	Point d'extrémité source		
Point d'extrémité destinataire	**Asynchrone**	**Synchrone**	**Adaptative**
Asynchrone	Source/destinataire RA async. Voir note 1.	SOF/destinataire RA async. Voir note 2.	Alimentation de données + rétroaction Voir note 3.
Synchrone	Source/SOF RA async. Voir note 4.	RA sync. Voir note 5.	Alimentation de données + rétroaction d'application Voir note 6.
Adaptative	Alimentation de données Voir note 7.	Alimentation de données Voir note 8.	Alimentation de données Voir note 9.

Notes:

1. RA asynchrone dans l'application. Fs_i est déterminé par la source à l'aide des informations de prédiction intégrées au flux de données. Fs_o est déterminé par le destinataire sur la base des informations de rétroaction provenant du destinataire. Si nominalement $Fs_i = Fs_o$, le processus devient une connexion d'alimentation si les glissements/insertions dus au manque de synchronisation sont tolérés. Ces glissements/insertions entraînent une dégradation audible des applications audio.
2. RA asynchrone dans l'application. Fs_i est déterminé par la source, mais est verrouillé pour le SOF. Fs_o est déterminé par le destinataire sur la base des informations de rétroaction provenant du destinataire. Si nominalement $Fs_i = Fs_o$, le processus devient une connexion d'alimentation si les glissements/insertions dus au manque de synchronisation sont tolérés. Ces glissements/insertions entraînent une dégradation audible des applications audio.
3. Si Fs_o est dans l'intervalle de verrouillage de la source adaptative, une connexion d'alimentation peut être établie. $Fs_i = Fs_o$ et les deux valeurs sont déterminées par le destinataire asynchrone sur la base des informations de rétroaction provenant du destinataire. Si Fs_o se trouve hors de l'intervalle de verrouillage de la source adaptative, celle-ci est commutée en mode synchrone et la note 2 s'applique.
4. RA asynchrone dans l'application. Fs_i est déterminé par la source. Fs_o est déterminé par le destinataire et verrouillé pour le SOF. Si nominalement $Fs_i = Fs_o$, le processus devient une connexion d'alimentation si les glissements/insertions dus au manque de synchronisation sont tolérés. Ces glissements/insertions entraînent une dégradation audible des applications audio.
5. RA synchrone dans l'application. Fs_i est déterminé par la source et verrouillé pour le SOF. Fs_o est déterminé par le destinataire et verrouillé pour le SOF. Si $Fs_i = Fs_o$, le processus devient une connexion d'alimentation sans pertes.
6. L'application fournit la rétroaction afin de synchroniser la source au SOF. La source adaptative semble être un point d'extrémité synchrone et la note 5 s'applique.
7. Si Fs_i est dans l'intervalle de verrouillage du destinataire adaptatif, une connexion d'alimentation peut être établie. $Fs_i = Fs_o$ et les deux valeurs sont déterminées par et verrouillées pour la source. Si Fs_i se trouve hors de l'intervalle de verrouillage du destinataire adaptatif, la RA synchrone se réalise sur l'hôte pour fournir une valeur Fs_o dans l'intervalle de verrouillage du destinataire adaptatif.
8. Si Fs_i est dans l'intervalle de verrouillage du destinataire adaptatif, une connexion d'alimentation peut être établie. $Fs_o = Fs_i$ et les deux valeurs sont déterminées par la source et verrouillées pour le SOF. Si Fs_i se trouve hors de l'intervalle de verrouillage du destinataire adaptatif, la RA synchrone se réalise sur l'hôte pour fournir une valeur Fs_o dans l'intervalle de verrouillage du destinataire adaptatif.
9. L'application utilise le contrôle de la rétroaction pour régler la valeur Fs_o de la source adaptative lorsque la connexion est mise en place. La source adaptative fonctionne comme une source asynchrone en l'absence d'informations de rétroaction en continu et la note 7 s'applique.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Dans le cas où la RA est requise, mais non disponible, le processus d'adaptation du débit peut être masqué par un dépôt/une insertion d'échantillons. La connexion peut encore être réalisée, éventuellement avec un avertissement de mauvaise qualité, sinon la connexion ne peut pas être réalisée.

5.12.4.4.1 Connectivité audio

Lorsque les opérations décrites ci-dessus s'appliquent aux flux de données audio, le processus RA est remplacé par la conversion de débit d'échantillons qui est une forme spécialisée d'adaptation du débit. A la place du contrôle des erreurs, une certaine forme d'interpolation d'échantillons est utilisée pour la correspondance entre les débits d'échantillons entrants et sortants. En fonction des techniques d'interpolation utilisées, la qualité audio (distorsion, rapport signal-bruit, etc.) de la conversion peut varier de manière significative. En général, une qualité supérieure nécessite plus de puissance de traitement.

5.12.4.4.2 Connectivité des données synchrones

Dans le cas des données synchrones, la RA est utilisée. Des glissements/insertions occasionnel(le)s peuvent être acceptables pour de nombreuses applications qui mettent en application une certaine forme du contrôle des erreurs. Le contrôle des erreurs incluent la détection et le rejet des erreurs, la détection et la retransmission des erreurs ou la correction des erreurs sans voie de retour. Le rapport glissements/insertions dépend de la non-concordance des horloges entre la source et le destinataire et peut représenter la source d'erreur dominante du canal. Si le contrôle des erreurs est suffisant, la connexion peut encore être réalisée.

5.12.5 Mise en mémoire tampon préalable des données

L'USB exige que les appareils effectuent une mise en mémoire tampon avant de procéder au traitement/à la transmission pour offrir plus de flexibilité à l'hôte dans la gestion du moment auquel la transaction de chaque canal est déplacée sur le bus d'une (micro)trame à l'autre.

Pour les transferts de la fonction vers l'hôte, le point d'extrémité doit accumuler des échantillons pendant la (micro)trame X jusqu'à ce qu'il reçoive le jeton SOF de la (micro)trame X+1. Il "verrouille" les données de la (micro)trame X dans la mémoire tampon de son paquet et est maintenant prêt à envoyer le paquet contenant ces échantillons durant la (micro)trame X+1. Le moment auquel il envoie ces données pendant la (micro)trame est déterminé exclusivement par le contrôleur hôte et peut varier d'une (micro)trame à l'autre.

Pour les transferts de l'hôte vers la fonction, le point d'extrémité accepte un paquet provenant de l'hôte à un certain moment durant la (micro)trame Y. Quand il reçoit le SOF de la (micro)trame Y+1, il peut commencer à traiter les données reçues dans la (micro)trame Y.

Cette approche permet à un point d'extrémité d'utiliser le jeton SOF comme une horloge stable avec très peu de gigue et/ou de dérive lorsque le contrôleur hôte déplace le paquet sur le bus. Cette approche permet également au contrôleur hôte de varier dans le cadre d'une (micro)trame précisément lorsque le paquet est actuellement en déplacement sur le bus. Cette mise en mémoire tampon préalable introduit un certain délai supplémentaire entre le moment où un échantillon est disponible sur un point d'extrémité et celui pendant lequel il se déplace sur le bus par rapport à un environnement dans lequel l'accès au bus se produit exactement au même moment pour le SOF d'une (micro)trame à l'autre.

La Figure 5-19 illustre la séquence temporelle pour un transfert fonction à hôte (processus IN). Les données D_0 s'accumulent pendant la (micro)trame F_i au moment T_i et sont transmises à l'hôte pendant la (micro)trame F_{i+1}. De façon similaire, pour un transfert hôte à fonction (processus OUT), les données D_0 sont reçues par le point d'extrémité pendant la (micro)trame F_{i+1} et traitées pendant la (micro)trame F_{i+2}.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Time:	T_i	T_{i+1}	T_{i+2}	T_{i+3}	------	------	T_m	T_{m+1}	------
(Micro)Frame:	F_i	F_{i+1}	F_{i+2}	M_{i+3}	------	------	F_m	F_{m+1}	------
Data on Bus:	------	D_0	D_1	D_2	------	------	D_0	D_1	------
OUT Process:	------	------	D_0	D_1	------	------	------	D_0	------
IN Process:	D_0	D_1	------	------	------	D_0	------	------	------

Légende

Anglais	Français
Time:	Temps:
(Micro)Frame:	(Micro)trame:
Data on Bus:	Données sur le bus:
OUT Process:	Processus OUT:
IN Process:	Processus IN:

Figure 5-19. Mise en mémoire tampon préalable des données

5.12.6 Suivi SOF

Les fonctions prenant en charge les canaux isochrones doivent recevoir et englober le jeton SOF pour prendre en charge la mise en mémoire tampon préalable telle que décrite précédemment. Etant que les SOF peuvent être corrompus, un appareil doit être prêt à récupérer d'un SOF corrompu. Ces exigences limitent les transferts isochrones aux appareils à vitesse intégrale et haute vitesse uniquement parce que les appareils à faible vitesse ne voient pas les SOF sur le bus. De même, étant donné que les paquets SOF peuvent être endommagés durant la transmission, il est nécessaire que les appareils prenant en charge les transferts isochrones soient en mesure de synthétiser l'existence d'un SOF qu'ils peuvent ne pas voir à cause d'une erreur du bus.

Les transferts isochrones exigent que les données appropriées soient transmises dans la (micro)trame correspondante. Lorsqu'un transfert isochrone est présenté au contrôleur hôte, l'USB requiert l'identification du nombre de (micro)trames de la première (micro)trame. Le contrôleur hôte ne doit pas transmettre la première transaction avant le numéro de (micro)trame indiqué. Chaque transaction ultérieure de l'IRP doit être transmise sous forme de (micro)trames successives (excepté pour les transferts à haute vitesse et à large bande passante lorsqu'un nombre maximum de trois transactions peut se produire dans la même microtrame). Si aucune transaction n'est en attente pour la (micro)trame actuelle, le contrôleur hôte ne doit rien transmettre pour un canal isochrone. Si le numéro de (micro)trame indiqué est passé, le contrôleur hôte doit sauter (c'est-à-dire, ne pas transmettre) toutes les transactions jusqu'à ce que celui qui correspond à la (micro)trame actuelle soit atteint.

5.12.7 Gestion des erreurs

Les transferts isochrones ne fournissent aucune nouvelle tentative de paquets de données (c'est-à-dire, aucune poignée de mains n'est retournée à un transmetteur par un récepteur), si bien que l'opportunité de la livraison des données n'est pas perturbée. Toutefois, il est encore important pour les agents responsables du transport des données de savoir le moment auquel l'erreur se produit et la manière dont elle affecte le débit de communication. En particulier, pour une séquence de paquets de données (A, B, C, D), l'USB permet d'avoir des informations suffisantes de manière qu'un paquet manquant (A, _, C, D) puisse être détecté et ne soit pas inconsciemment transformé en une donnée ou une séquence temporelle incorrecte (A, C, D ou A, _, B, C, D). Le protocole fournit quatre mécanismes pour la prise en charge de ce mécanisme: une périodicité strictement définie pour la transmission des paquets et les mécanismes de séquencement des PID de données pour les points d'extrémité à haute vitesse et à large bande passante, SOF, CRC et retard de transaction de bus.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Les transferts isochrones nécessitent l'occurrence périodique des transactions de données pour un fonctionnement normal. La période doit représenter une puissance exacte des deux (micro)trames. L'USB n'impose pas les données qui doivent être transmises dans chaque trame. Le transmetteur/la source de données détermine de manière spécifique les données qui doivent être fournies. Cette livraison régulière et périodique de données fournit une infrastructure fondamentale pour la détection des erreurs de données manquantes. Pour les points d'extrémité à haute vitesse et à large bande passante, le séquencement des PID de données permet la détection des transactions manquantes ou endommagées pendant une microtrame. Toute phase d'une transaction peut être endommagée pendant la transmission sur le bus. Le Chapitre 8 décrit la manière dont chaque cas d'erreur affecte le protocole.
- Etant donné que chaque (micro)trame est précédée d'un SOF et qu'un récepteur peut voir les SOF sur le bus, un récepteur peut déterminer si la transaction prévue pour cette (micro)trame n'a pas eu lieu entre deux SOF. En outre, puisqu'un SOF peut être endommagé, un appareil doit être en mesure de reconstruire l'existence d'un SOF manquant, tel que décrit à la Section 5.12.6.
- Un paquet de données peut être corrompu sur le bus; par conséquent, la protection CRC permet à un récepteur de déterminer si le paquet de données reçu a été corrompu.
- Le protocole définit les détails qui permettent à un récepteur de déterminer par l'intermédiaire d'un retard de transaction de bus qu'il ne va pas recevoir son paquet de données après avoir détecté avec succès son paquet de jetons.

Une fois qu'un récepteur a déterminé qu'un paquet de données n'a pas été reçu, il se peut qu'il ait besoin de connaître la taille des données manquantes afin de récupérer de l'erreur concernant son comportement fonctionnel. Si le flux de communication a toujours la même taille de données par (micro)trame, la taille est toujours une constante connue. Toutefois, dans certains cas, la taille des données peut varier d'une (micro)trame à l'autre. Dans ce cas, le récepteur et le transmetteur ont un mécanisme dépendant de la mise en application pour déterminer la taille du paquet perdu.

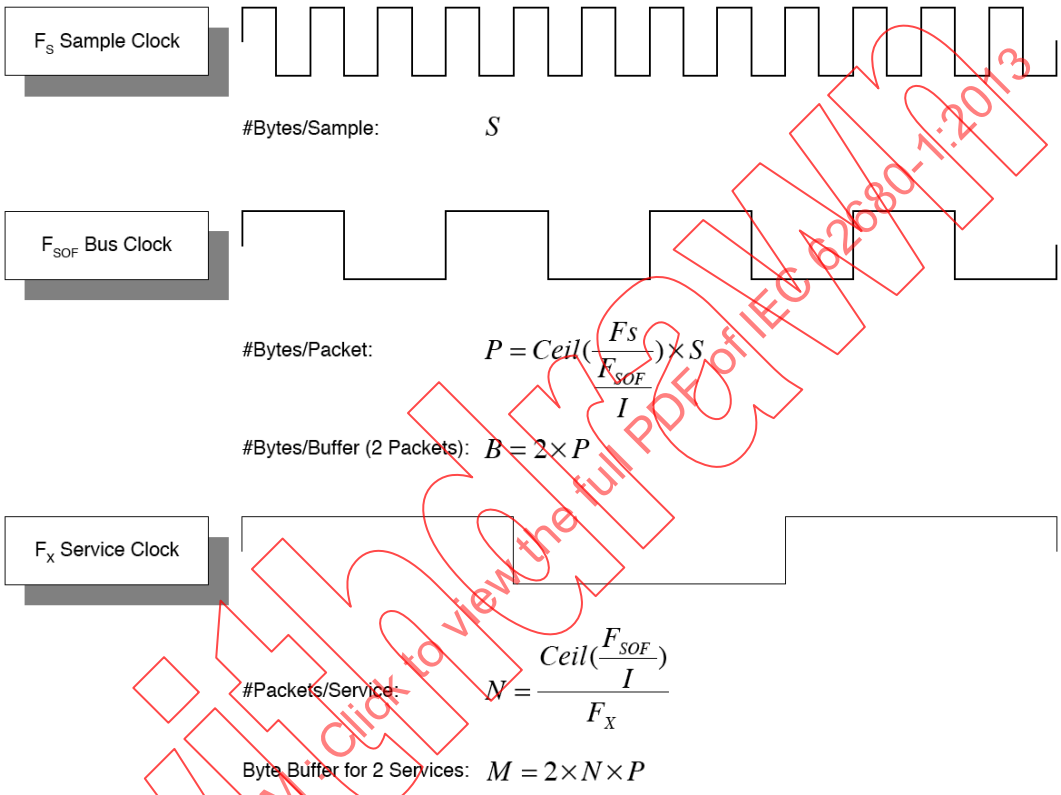
En résumé, qu'une transaction soit actuellement déplacée avec succès ou non sur le bus, le transmetteur et le récepteur avancent toujours les flux de données/mémoire tampon, comme l'indique la période d'accès au bus pour conserver la synchronisation des données en fonction du temps. Les mécanismes détaillés décrits ci-dessus permettent la détection, le suivi et l'établissement de rapports des transactions endommagées afin qu'une fonction ou son logiciel client puisse réagir au dommage de manière appropriée à la fonction. Les détails de cette réaction spécifique à la fonction ou à l'application ne relèvent pas du domaine d'application de la spécification USB.

5.12.8 Mise en mémoire tampon pour la mise en correspondance du débit

Etant qu'il y a plusieurs horloges qui affectent les flux de communication isochrones de l'USB, la mise en mémoire tampon est requise pour la mise en correspondance du débit du flux de communication sur l'USB. Il doit y avoir un espace de mémoire tampon disponible sur l'appareil par point d'extrémité et du côté hôte au nom du logiciel client. Ces mémoires tampons fournissent l'espace nécessaire pour l'accumulation des données jusqu'à ce qu'il soit temps pour un transfert de se déplacer sur l'USB. Etant donné les débits naturels de données de l'appareil, la taille maximum des paquets de données qui se déplacent sur le bus peut également être calculée.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

La Figure 5-20 illustre les équations utilisées pour déterminer la taille de mémoire tampon de l'appareil et de l'hôte et la taille maximum de paquet qui doit être requise pour la prise en charge d'un débit de données souhaité. Ces équations sont une fonction de la vitesse d'horloge de service (F_X), de la vitesse d'horloge de bus (F_{SOF}), de la vitesse d'horloge d'échantillons (F_S), de la période d'accès au bus (I) et de la taille d'échantillons (S). Il convient que ces équations fournissent des informations de conception pour la sélection de la taille de paquet appropriée qu'un point d'extrémité signale dans les informations relatives à ses caractéristiques et des exigences de mémoire tampon appropriées de l'appareil/du point d'extrémité et de son logiciel client. La Figure 5-17 montre les valeurs actuelles de mémoire tampon, de paquet et d'horloge pour un exemple typique de transfert isochrone à vitesse intégrale.



Légende

Anglais	Français
F_s Sample Clock	Horloge d'échantillons F_s
F_{SOF} Bus Clock	Horloge de bus F_{SOF}
F_X Service Clock	Horloge de service F_X
# Bytes/Sample:	Nombre d'octets/échantillon:
# Bytes/Buffer (2 Packets):	Nombre d'octets/Mémoire tampon (2 paquets):
# Bytes/Packet:	Nombre d'octets/paquet:
# Packets/Service:	Nombre de paquets/service:
Byte buffer for 2 Services:	Mémoire tampon d'octets pour 2 services:

Figure 5-20. Formules de taille de paquet et de mémoire tampon pour les transferts isochrones dont le débit est mis en correspondance

Le modèle de données USB suppose que les appareils ont une taille et un débit d'échantillons naturels. L'USB prend en charge la transmission des paquets qui sont multiples de la taille d'échantillon afin de faciliter la manipulation de la récupération des erreurs lorsque les transactions isochrones sont endommagées sur le bus. Si un appareil n'a pas une taille d'échantillon naturelle ou si ses échantillons sont plus grands qu'un paquet, il convient qu'il décrive une taille d'échantillon égale à un octet. Si un échantillon est divisé sur un paquet de données, la récupération des erreurs peut se révéler plus difficile lorsqu'une transaction arbitraire est perdue. Dans certains cas, la synchronisation des données peut être perdue, à moins que le récepteur ne sache le numéro de (micro)trame dans laquelle chaque échantillon partiel est transmis. En outre, si le nombre d'échantillons peut varier à cause de la correction

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

des horloges (par exemple, pour une horloge d'appareil non dérivée), il peut être difficile ou inefficace de savoir lorsqu'un échantillon partiel est transmis. Par conséquent, l'USB ne divise pas les échantillons entre les paquets.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

6 Chapitre 6
Domaine mécanique

Le présent chapitre fournit les spécifications mécaniques et électriques des câbles, des connecteurs et des assemblages de câbles utilisés pour l'interconnexion des appareils USB. La spécification comprend les exigences relatives aux dimensions, aux matériaux, à la partie électrique et à la fiabilité. Le présent chapitre documente les exigences minimales relatives à l'interconnexion USB externe. Du matériel de substitution peut être utilisé tant qu'il satisfait à ces exigences minimales.

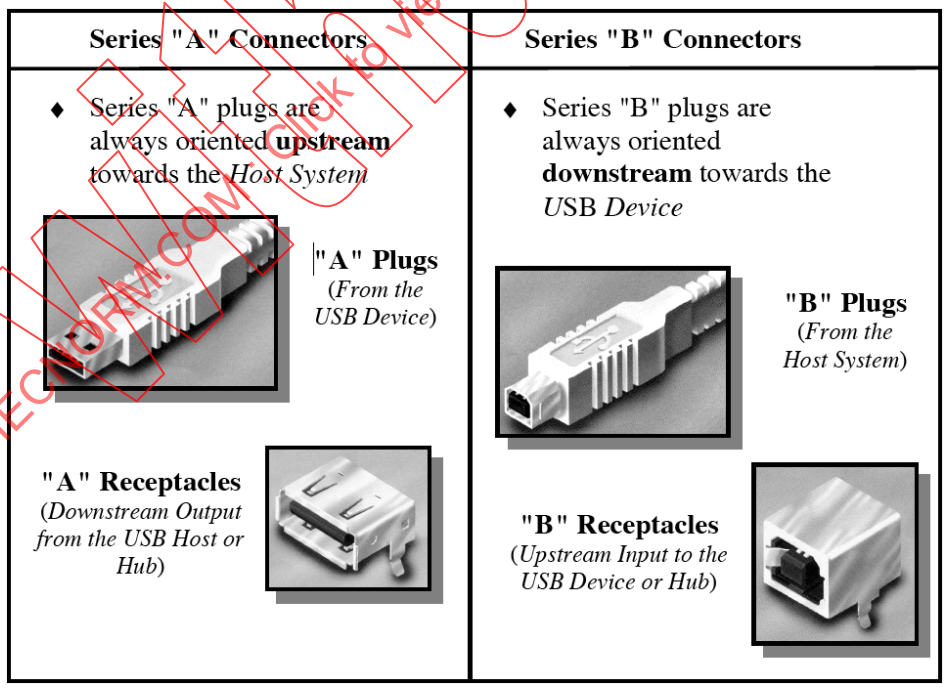
6.1 Vue d'ensemble sur l'architecture

La topologie physique USB consiste à connecter le port du hub en aval au port en amont d'un autre hub ou à un appareil. L'USB peut fonctionner à trois vitesses. La haute vitesse (480 Mb/s) et la vitesse intégrale (12 Mb/s) nécessitent l'utilisation d'un câble blindé avec deux conducteurs d'alimentation et des conducteurs de signal à paire torsadée. La faible vitesse (1,5 Mb/s) recommande, mais n'exige pas, l'utilisation d'un câble avec des conducteurs de signal à paire torsadée.

Les connecteurs sont conçus pour être branchés à chaud. L'icône USB qui se trouve sur les fiches fournit une rétroaction tactile, ce qui facilite l'obtention de l'orientation correcte.

6.2 Protocole de connecteur claveté

Pour minimiser les problèmes de raccordement de l'utilisateur final, l'USB utilise un protocole de "connecteur claveté". La différence physique des gammes de connecteurs "A" et "B" garantit une connectivité correcte pour l'utilisateur final. Le connecteur "A" est le moyen principal utilisé par les appareils USB pour se connecter directement à l'hôte ou au port en aval d'un hub. Tous les appareils USB doivent avoir le connecteur normalisé de série "A" spécifié au présent chapitre. Le connecteur "B" permet aux vendeurs d'appareils de fournir un câble démontable normalisé. Cela facilite le remplacement du câble par l'utilisateur final. La Figure 6-1 illustre le protocole de connecteur claveté.



Légende

Anglais	Français
Series "A" Connectors	Connecteurs de série "A"
Series "A" plugs are always oriented upstream towards the Host System	Les fiches de série "A" sont toujours orientées en amont vers le système hôte
"A" Plugs (From the USB Device)	Fiches "A" (de l'appareil USB)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Anglais	Français
"A" Receptacles (Downstream Output from the USB Host or Hub)	Réceptacles "A" (sortie en aval de l'hôte USB ou du hub)
Series "B" Connectors	Connecteurs de série "B"
Series "B" plugs are always oriented downstream towards the USB Device	Les fiches de série "B" sont toujours orientées en aval vers l'appareil USB
"B" Plugs (From the Host System)	Fiches "B" (du système hôte)
"B" Receptacles (Upstream Input to the USB Device or Hub)	Réceptacles "B" (entrée en amont de l'appareil USB ou du hub)

Figure 6-1. Protocole de connecteur claveté

La liste suivante explique la méthode d'accouplement des fiches et des prises:

- Le réceptacle de série "A" s'accouple à une fiche de série "A". Du point de vue électrique, les réceptacles de série "A" fonctionnent comme des sorties pour les systèmes hôtes et/ou les hubs.
- La prise de série "A" s'accouple à un réceptacle de série "A". La fiche de série "A" est toujours orientée vers le système hôte.
- Le réceptacle de série "B" s'accouple à une fiche (mâle) de série "B". Du point de vue électrique, les réceptacles de série "B" fonctionnent comme des entrées pour les hubs ou les appareils.
- La fiche de série "B" s'accouple à un réceptacle de série "B". La fiche de série "B" est toujours orientée vers le hub ou l'appareil USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

6.3 Câble

Le câble USB consiste en quatre conducteurs, deux conducteurs d'alimentation et deux conducteurs de signal.

Le câble à haute/vitesse intégrale consiste en une paire torsadée de signalisation, des fils VBUS, des fils GND et un blindage intégral. Le câble à haute/vitesse intégrale doit être marqué pour indiquer la conformité à l'utilisation USB (voir Section 6.6.2). Le câble à haute/vitesse intégrale peut être utilisé avec des appareils à faible, vitesse intégrale ou haute vitesse. Lorsque le câble à haute/vitesse intégrale est utilisé avec des appareils à faible vitesse, il doit satisfaire à toutes les exigences de faible vitesse.

La faible vitesse recommande, mais n'exige pas, l'utilisation d'un câble avec des conducteurs de signalisation torsadés.

6.4 Ensemble de câbles

La présente spécification décrit trois ensembles de câbles USB: câble amovible normalisé, câble captif à haute/vitesse intégrale et câble à faible vitesse.

Un câble amovible normalisé est un câble à haute/vitesse intégrale qui se termine d'un côté par une fiche de série "A" et du côté opposé par une fiche de série "B". Un câble captif à haute/vitesse intégrale se termine d'un côté par une fiche de série "A" et dispose d'un moyen de connexion (câblé ou à démontage personnalisé) spécifique au vendeur du côté opposé pour le périphérique à haute/vitesse intégrale. Un câble captif à faible vitesse se termine d'un côté par une fiche de série "A" et dispose d'un moyen de connexion (à brin unique ou à démontage personnalisé) spécifique au vendeur du côté opposé pour le périphérique à faible vitesse. Tout autre ensemble de câbles est interdit.

La couleur utilisée pour l'ensemble de câbles est spécifique au vendeur; les couleurs recommandées sont le blanc, le gris ou le noir.

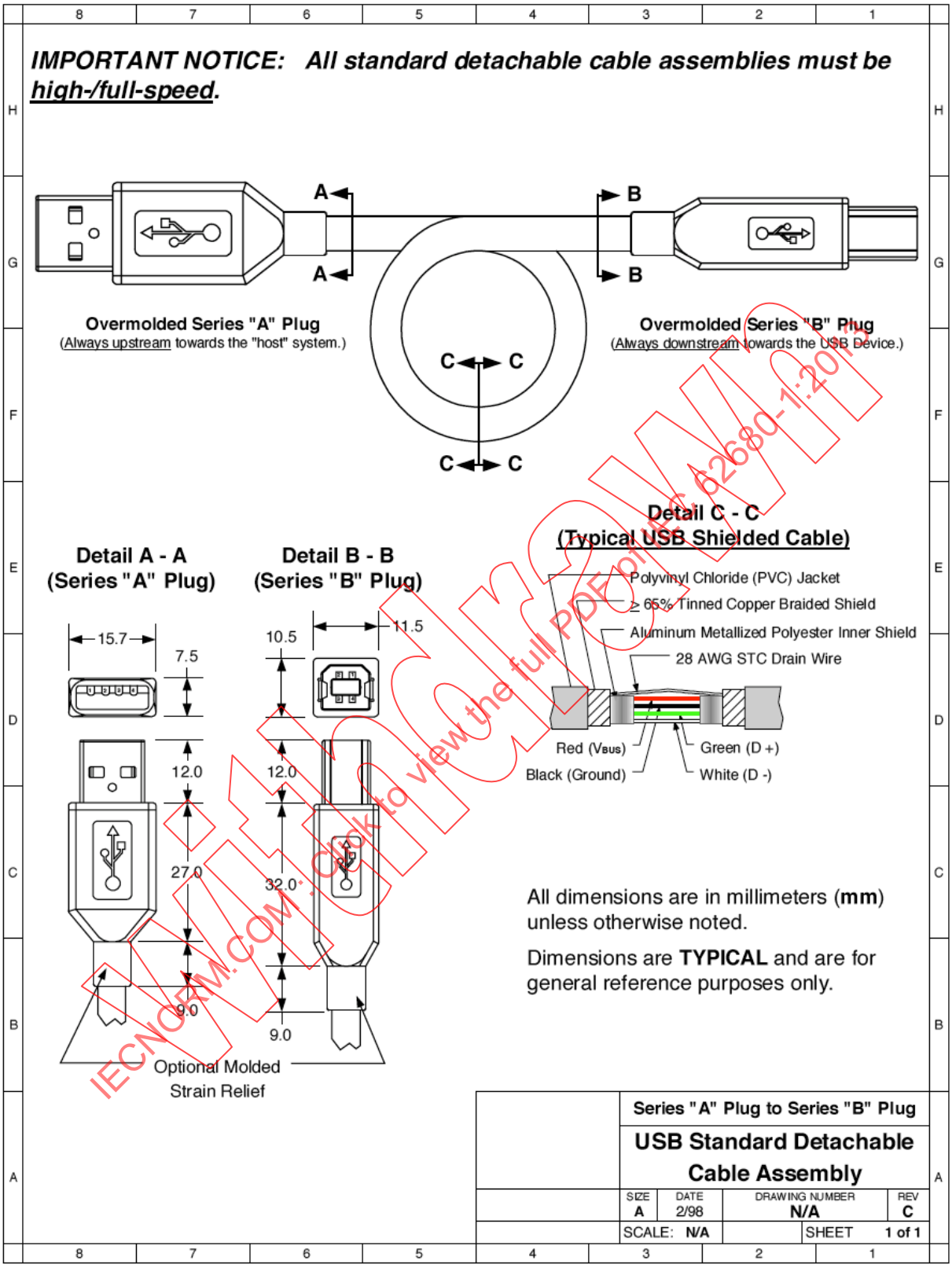
6.4.1 Ensembles de câbles amovibles normalisés

Les appareils à haute et vitesse intégrale peuvent utiliser le connecteur "B". Cela permet à l'appareil d'avoir un câble USB amovible normalisé. Cela élimine le besoin de construire l'appareil avec un câble à brin unique et minimise les problèmes de l'utilisateur final si le remplacement du câble est nécessaire.

Les appareils qui utilisent le connecteur "B" doivent être conçus pour fonctionner avec un câble amovible de longueur maximum dans le cas le plus défavorable. Les ensembles de câbles amovibles normalisés peuvent être utilisés uniquement sur les appareils à haute et vitesse intégrale. L'utilisation d'un câble amovible normalisé à haute/vitesse intégrale sur un appareil à faible vitesse peut dépasser la longueur maximum de câble à faible vitesse.

La Figure 6-2 illustre un ensemble de câbles amovibles normalisés.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
WITHDRAWN

Légende

Anglais	Français
IMPORTANT NOTICE: All standard detachable cables assemblies must be high-/full speed.	AVIS IMPORTANT: tous les ensembles de câbles amovibles normalisés doivent fonctionner à haute vitesse/vitesse intégrale.
Overmolded Series "A" plug (Always upstream towards the "host"	Fiche surmoulée de série "A" (toujours en amont vers le système "hôte")

Anglais	Français
system)	
Overmolded Series "B" plug (Always downstream towards the USB Device)	Fiche surmoulée de série "B" (toujours en aval vers l'appareil USB)
Details A - A (Series "A" Plug)	Détails A - A (fiche de série "A")
Details B - B (Series "B" Plug)	Détails B - B (fiche de série "B")
Details C - C (Typical USB Shielded Cable)	Détails C - C (câble blindé USB typique)
Polyvinyl Chloride (PVC) Jacket	Chemise en polychlorure de vinyle (PVC)
≥65% Tinned Copper Braided Shield	≥65 % de blindage en tresses en cuivre étamé
Aluminium Metallized Polyester Inner Shield	Blindage interne en polyester métallisé d'aluminium
28 AWG STC Drain Wire	Conducteur de drainage toronné en cuivre étamé de 28 AWG
Red (V_{BUS})	Rouge (V_{BUS})
Green (D+)	Vert (D+)
Black (Ground)	Noir (masse)
White (D-)	Blanc (D-)
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
Dimensions are TYPICAL and are for general references purposes only.	Les dimensions sont TYPIQUES et servent en général à des fins de référence uniquement.
Optional Molded Strain Relief	Relaxation des contraintes moulée en option
Series "A" Plug to Series "B" Plug	Fiche de série "A" vers fiche de série "B"
USB Standard Detachable Cable Assembly	Ensemble de câbles amovibles USB normalisés
SIZE	TAILLE
DATE	DATE
DRAWING NUMBER	NUMÉRO DE DESSIN
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-2. Ensembles de câbles USB amovibles normalisés

Les ensembles de câbles amovibles normalisés doivent satisfaire aux exigences électriques:

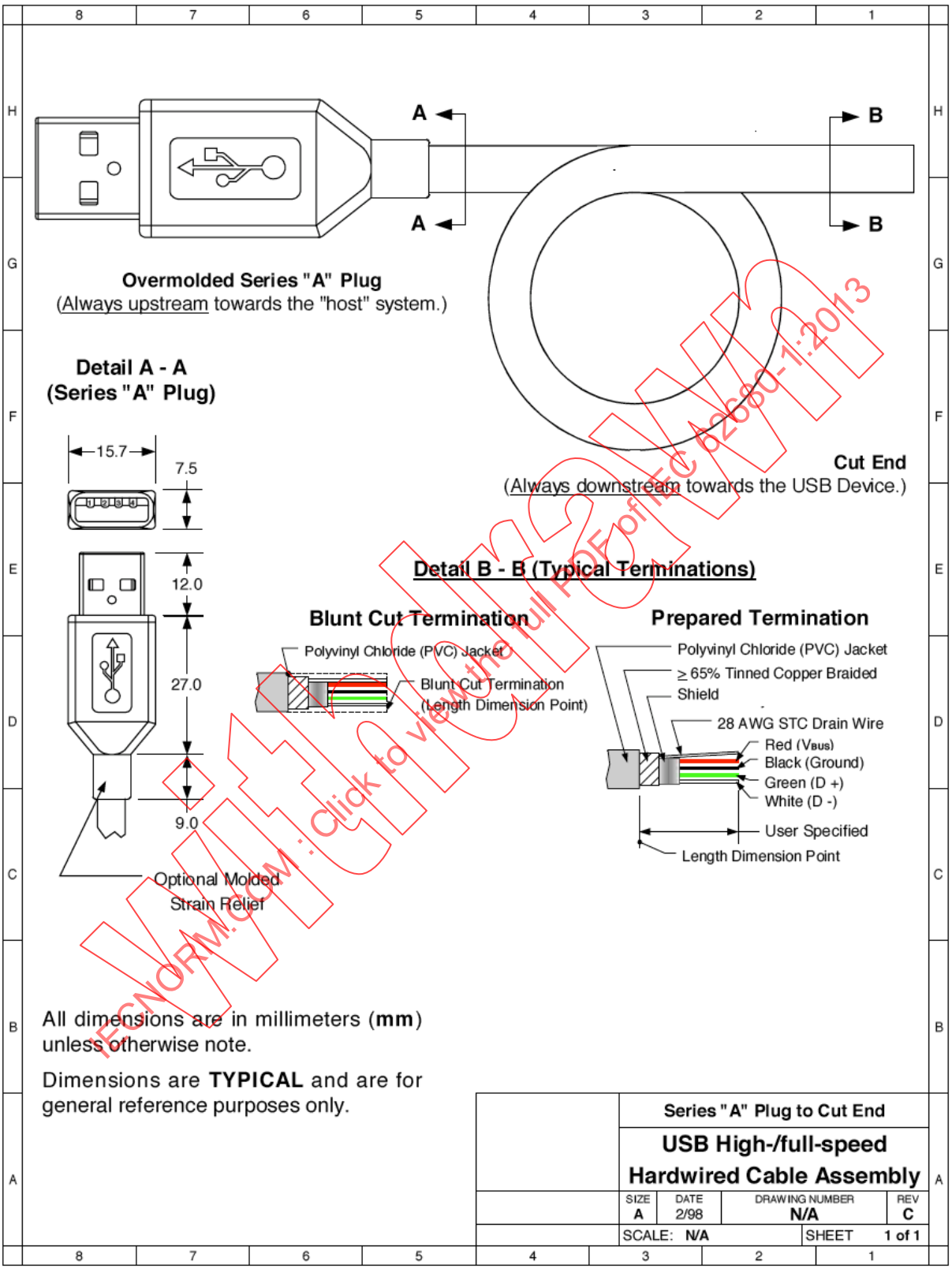
- Le câble doit se terminer d'un côté par une fiche surmoulée de série "A" et du côté opposé par une fiche surmoulée de série "B".
- Le câble doit être assigné pour la haute vitesse et la vitesse intégrale.
- L'impédance du câble doit correspondre à celles des pilotes à haute et vitesse intégrale. Les pilotes sont caractérisés par une impédance de câble spécifique au pilote. Voir la Section 7.1.1 pour les détails.
- La longueur de câble maximum permise est déterminée par l'atténuation de la paire de signaux et le délai de propagation. Voir les Sections 7.1.14 et 7.1.17 pour les détails.
- Les différences de délai de propagation entre les deux conducteurs de signal doivent être minimisées. Voir la Section 7.1.3 pour les détails.
- Le conducteur GND fournit une référence commune à la masse entre les ports en amont et en aval. La longueur de câble maximum est limitée par la chute de potentiel le long du conducteur. Voir la Section 7.2.2 pour les détails. Le calibre de fil électrique maximum permis est calculé en supposant que l'appareil connecté est de grande puissance.
- Le conducteur VBUS alimente l'appareil connecté. Pour les câbles amovibles normalisés, les exigences relatives au VBUS sont les mêmes que celles relatives au conducteur GND.

6.4.2 Ensembles de câbles captifs à haute/vitesse intégrale

Les ensembles sont considérés captifs s'ils sont équipés d'un moyen de connexion (câblée ou à démontage personnalisé) au périphérique spécifique au vendeur. L'ensemble de câbles à haute/vitesse intégrale peut être utilisé avec des appareils à haute, vitesse intégrale ou faible vitesse. Lorsqu'un câble à haute/vitesse intégrale est utilisé avec un appareil à faible vitesse, il doit satisfaire à toutes les exigences de faible vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

La Figure 6-3 illustre un ensemble de câbles à haute/vitesse intégrale.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Overmolded Series "A" plug (Always upstream towards the "host" system)	Fiche surmoulée de série "A" (toujours en amont vers le système "hôte")

Anglais	Français
Details A - A (Series "A" Plug)	Détails A - A (fiche de série "A")
Cut End (Always downstream towards the USB Device)	Extrémité de découpe (toujours en aval vers l'appareil USB)
Details B - B (Typical Terminations)	Détails B - B (terminaisons typiques)
Blunt Cut Termination	Terminaison de coupe arrondie
Polyvinyl Chloride (PVC) Jacket	Chemise en polychlorure de vinyle (PVC)
Blunt Cut Termination (Length Dimension Point)	Terminaison de coupe arrondie (point de dimension de longueur)
Prepared Termination	Terminaison préparée
Polyvinyl Chloride (PVC) Jacket	Chemise en polychlorure de vinyle (PVC)
≥65% Tinned Copper Braided Shield	≥65 % de blindage en tresses en cuivre étamé
28 AWG STC Drain Wire	Conducteur de drainage toronné en cuivre étamé de 28 AWG
Red (V_{BUS})	Rouge (V_{BUS})
Black (Ground)	Noir (masse)
Green (D+)	Vert (D+)
White (D-)	Blanc (D-)
User Specified	Spécifié par l'utilisateur
Length Dimension Point	Point de dimension de longueur
Optional Molded Strain Relief	Relaxation des contraintes moulée en option
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
Dimensions are TYPICAL and are for general references purposes only.	Les dimensions sont TYPIQUES et servent en général à des fins de référence uniquement.
Series "A" Plug to Cut End	Fiche de série "A" vers extrémité de découpe
USB High-/full-speed Hardwired Cable Assembly	Ensemble de câbles à brin unique USB à grande vitesse/vitesse intégrale
SIZE	TAILLE
DATE 2/98	DATE
DRAWING NUMBER N/A	NUMÉRO DE DESSIN N/A
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-3. Ensembles de câbles USB à haute/vitesse intégrale

Les ensembles de câbles captifs à haute/vitesse intégrale doivent satisfaire aux exigences électriques suivantes:

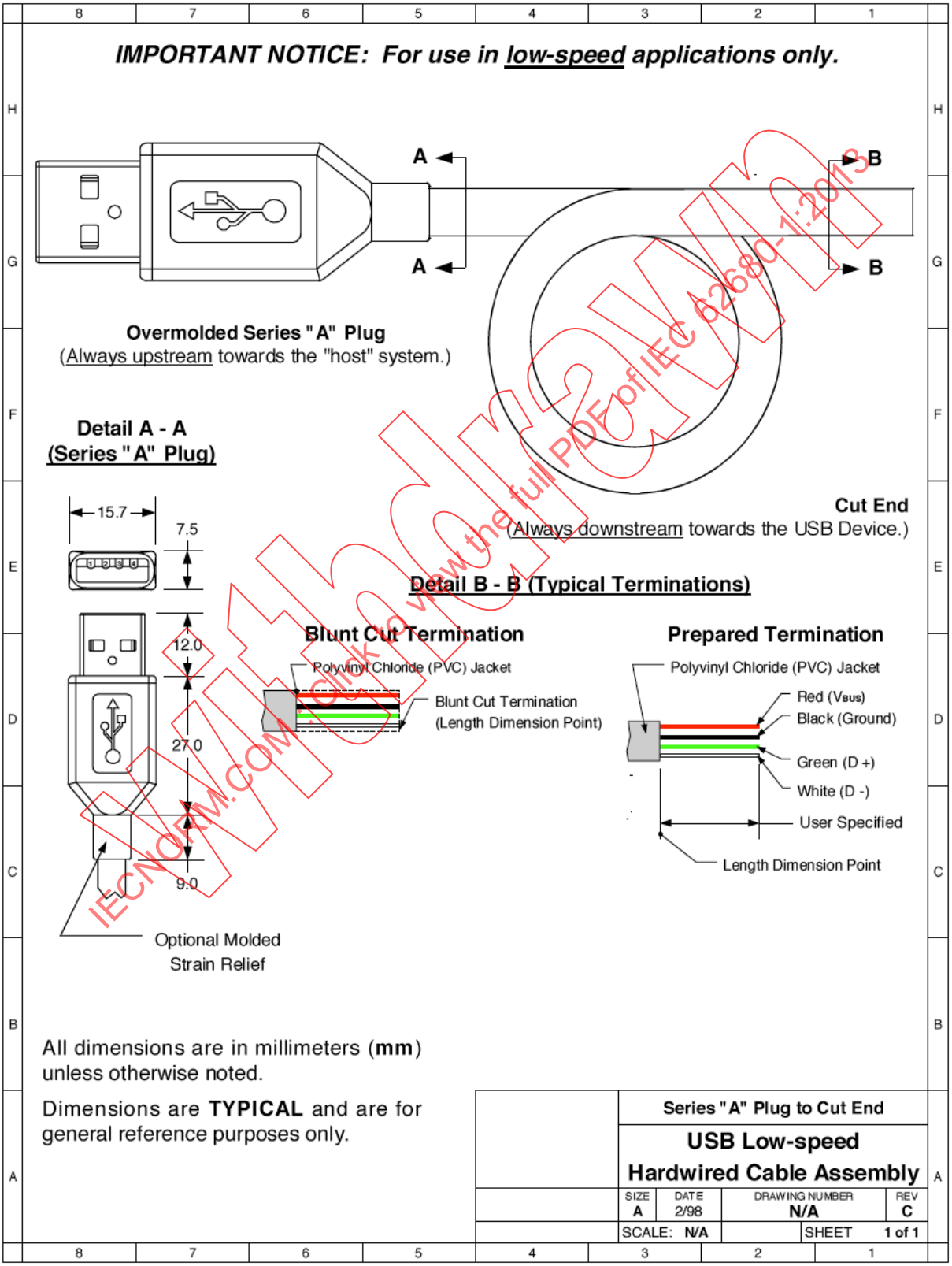
- Le câble doit se terminer d'un côté par une fiche surmoulée de série "A" et le côté opposé est spécifique au vendeur. Si l'interconnexion spécifique au vendeur est branchée à chaud, elle doit satisfaire aux mêmes exigences de performance que le connecteur USB "B".
- Le câble doit être assigné pour la haute et vitesse intégrale.
- L'impédance du câble doit correspondre à celles des pilotes à haute et vitesse intégrale. Les pilotes sont caractérisés par une impédance de câble spécifique au pilote. Voir la Section 7.1.1 pour les détails.
- La longueur de câble maximum permise est déterminée par l'atténuation de la paire de signaux et le délai de propagation. Voir les Sections 7.1.14 et 7.1.17 pour les détails.
- Les différences de délai de propagation entre les deux conducteurs de signal doivent être minimisées. Voir la Section 7.1.3 pour les détails.
- Le conducteur GND fournit une référence commune entre les ports en amont et en aval. La longueur de câble maximum est déterminée par la chute de potentiel le long du conducteur. Voir la Section 7.2.2 pour les détails. Le calibre de fil électrique minimum est calculé à l'aide de la consommation de courant dans le cas le plus défavorable.
- Le conducteur VBUS alimente l'appareil connecté. Le calibre de fil électrique minimum est spécifique au vendeur.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

6.4.3 Ensembles de câbles captifs à faible vitesse

Les ensembles sont considérés captifs s'ils sont équipés d'un moyen de connexion (câblée ou à démontage personnalisé) au périphérique spécifique au vendeur. Les câbles à faible vitesse peuvent être utilisés sur les appareils à faible vitesse uniquement.

La Figure 6-4 illustre un ensemble de câbles à faible vitesse.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
IMPORTANT NOTICE: For use in low-speed applications only.	AVIS IMPORTANT: pour utilisation dans des applications à faible vitesse uniquement.
Overmolded Series "A" plug (Always upstream towards the "host" system)	Fiche surmoulée de série "A" (toujours en amont vers le système "hôte")
Details A - A (Series "A" Plug)	Détails A - A (fiche de série "A")
Cut End (Always downstream towards the USB Device)	Extrémité de découpe (toujours en aval vers l'appareil USB)
Details B - B (Typical Terminations)	Détails B - B (terminaisons typiques)
Blunt Cut Termination	Terminaison de coupe arrondie
Polyvinyl Chloride (PVC) Jacket	Chemise en polychlorure de vinyle (PVC)
Blunt Cut Termination (Length Dimension Point)	Terminaison de coupe arrondie (point de dimension de longueur)
Prepared Termination	Terminaison préparée
Polyvinyl Chloride (PVC) Jacket	Chemise en polychlorure de vinyle (PVC)
Red (V_{BUS})	Rouge (V_{BUS})
Black (Ground)	Noir (masse)
Green (D+)	Vert (D+)
White (D-)	Blanc (D-)
User Specified	Spécifié par l'utilisateur
Length Dimension Point	Point de dimension de longueur
Optional Molded Strain Relief	Relaxation des contraintes moulée en option
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
Dimensions are TYPICAL and are for general references purposes only.	Les dimensions sont TYPIQUES et servent en général à des fins de référence uniquement.
Series "A" Plug to Cut End	Fiche de série "A" vers extrémité de découpe
USB Low-speed Hardwired Cable Assembly	Ensemble de câbles à brin unique USB à faible vitesse
SIZE	TAILLE A
DATE	DATE
DRAWING NUMBER N/A	NUMÉRO DE DESSIN N/A
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-4. Ensembles de câbles USB à faible vitesse

Les ensembles de câbles captifs à faible vitesse doivent satisfaire aux exigences électriques suivantes:

- Le câble doit se terminer d'un côté par une fiche surmoulée de série "A" et le côté opposé est spécifique au vendeur. Si l'interconnexion spécifique au vendeur est branchée à chaud, elle doit satisfaire aux mêmes exigences de performance que le connecteur USB "B".
- Les pilotes à faible vitesse sont caractérisés pour le fonctionnement sur une gamme de charges capacitives. Cette valeur comprend toutes les sources de capacitance des lignes D+ et D-, non seulement le câble. La sélection du câble doit s'assurer que la capacitance de charge totale se situe entre les valeurs minimum et maximum spécifiées. Si la mise en application souhaitée ne satisfait pas aux exigences minimales, la capacitance supplémentaire doit être ajoutée à l'appareil. Voir la Section 7.1.1.2 pour les détails.
- La longueur de câble maximum permise est déterminée par les temps de montée et de descente de la signalisation à faible vitesse. Cela force le câble à faible vitesse à être beaucoup plus court que celui à haute/vitesse intégrale. Voir la Section 7.1.1.2 pour les détails.
- Les différences de délai de propagation entre les deux conducteurs de signal doivent être minimisées. Voir la Section 7.1.3 pour les détails.
- Le conducteur GND fournit une référence commune entre les ports en amont et en aval. La longueur de câble maximum est déterminée par la chute de potentiel le long du conducteur. Voir la Section 7.2.2 pour les détails. Le calibre de fil électrique minimum est calculé à l'aide de la consommation de courant dans le cas le plus défavorable.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- Le conducteur VBUS alimente l'appareil connecté. Le calibre de fil électrique minimum est spécifique au vendeur.

6.4.4 Ensembles de câbles interdits

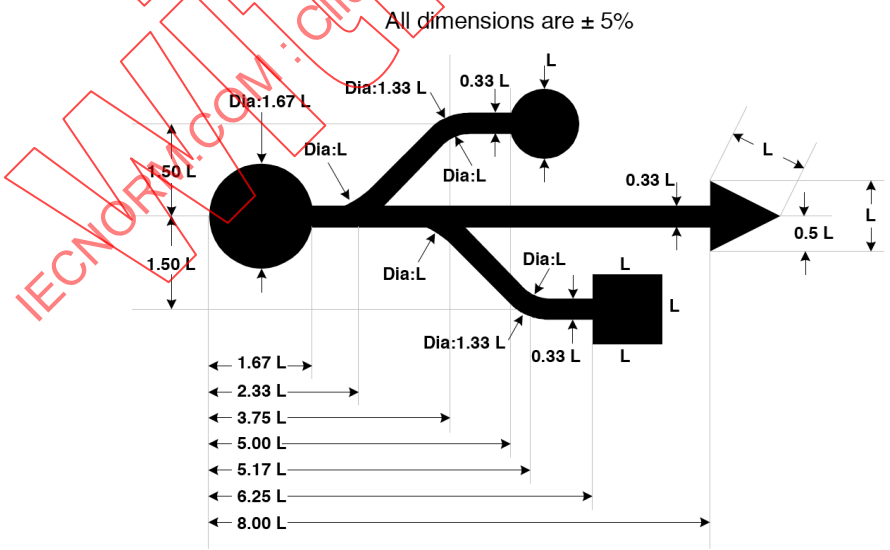
L'USB est optimisé pour la facilité d'utilisation. Si l'appareil peut être branché, on s'attend à ce qu'il fonctionne. Selon les spécifications, les seules conditions qui empêchent l'utilisation réussie d'un appareil USB sont le manque de puissance, le manque de bande passante et la profondeur excessive de la topologie. Ces conditions sont très bien comprises par le logiciel système.

Les ensembles de câbles interdits peuvent fonctionner dans certaines situations, mais il n'est pas garanti qu'ils puissent fonctionner dans toutes les situations.

- **Ensemble de câbles d'extension**
  Ensemble de câbles qui fournit une fiche de série "A" avec un réceptacle de série "A" ou une fiche de série "B" avec un réceptacle de série "B". Cela permet d'effectuer la connexion de plusieurs segments de câbles ensemble, pouvant dépasser la longueur de câble maximum permise.
- **Ensemble de câbles qui viole les règles de la topologie USB**
  Ensemble de câbles dont les deux extrémités se terminent soit par des fiches de série "A", soit par des réceptacles de série "B". Cela permet de connecter directement deux ports en aval.
  Note: cette interdiction n'empêche pas l'utilisation d'un appareil USB pour réaliser un point entre deux bus USB.
- **Câbles amovibles normalisés pour appareils à faible vitesse**
  Il est interdit aux appareils à faible vitesse d'utiliser des câbles amovibles normalisés. Un ensemble de câbles amovibles normalisés doit être à haute/vitesse intégrale. Etant donné qu'un ensemble de câbles amovibles normalisés est nominalement à haute/vitesse intégrale, l'utilisation d'un long câble à haute/vitesse intégrale dépasse la charge capacitive de la faible vitesse.

6.5 Configuration mécanique des connecteurs et exigences en termes de matériel

L'icône USB est utilisée pour identifier les fiches et les réceptacles USB. La Figure 6-5 illustre l'icône USB.



Légende

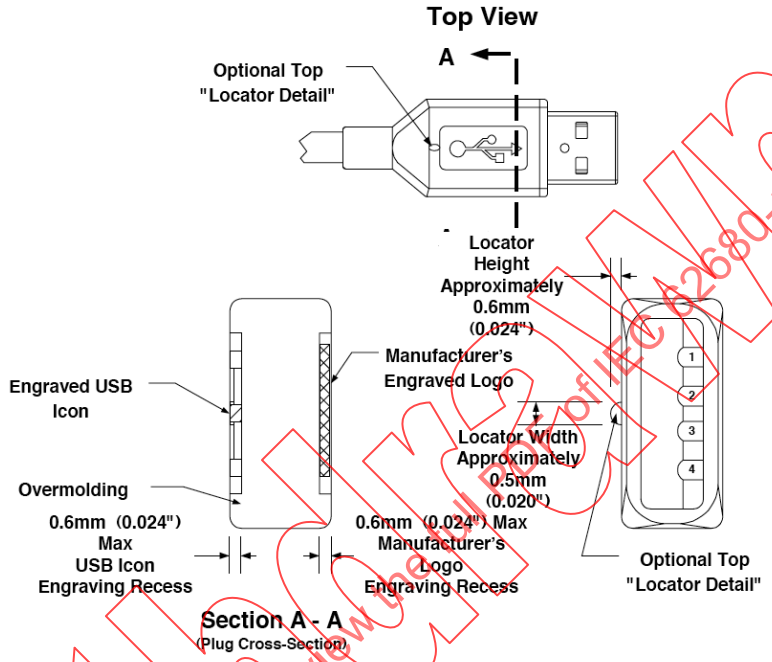
Anglais	Français
All dimensions are ± 5%	Toutes les dimensions sont à ± 5 %

Figure 6-5. Icône USB

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

6.5.1 Emplacement de l'icône USB

L'icône USB est en relief, dans une zone renfoncée, sur la partie supérieure de la fiche USB. Cela permet une identification aisée par l'utilisateur et facilite l'alignement pendant le processus d'accouplement. Il convient que l'icône USB et le logo du fabricant ne se projettent pas au-delà de la surface surmoulée. L'icône USB est requise tandis que le logo du fabricant est recommandé pour les deux ensembles de fiches des séries "A" et "B". L'icône USB se trouve également à côté de chaque réceptacle. Il convient que les réceptacles soient orientés afin de permettre à l'icône de la fiche d'être visible pendant le processus d'accouplement. La Figure 6-6 illustre l'orientation typique de la fiche.



Légende

Anglais	Français
Top View	Vue du dessus
Optional Top "Locator Detail"	"Détails de localisateur" supérieur en option
Locator Height Approximately 0.6mm (0.024")	Hauteur du localisateur environ égale à 0,6 mm (0,024 ")
Manufacturer's Engraved Logo	Logo gravé du fabricant
Engraved USB icon	Icône USB gravée
Overmolding	Surmoulage
0.6mm (0.024") Max USB Icon Engraving Recess	Renfoncement de gravage de l'icône USB de 0,6 mm (0,024 ") max
0.6mm (0.024") Manufacturer's Logo Engraving Recess	Renfoncement de gravage du logo du fabricant de 0,6 mm (0,024 ")
Locator Width Approximately 0.5mm (0.020")	Largeur du localisateur environ égale à 0,5 mm (0,020 ")
Section A - A (Plug Cross-Section)	Section A - A (Section transversale de la fiche)

Figure 6-6. Orientation typique de la fiche USB

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

6.5.2 Données de terminaison des connecteurs USB

Le Tableau 6-1 fournit les attributions de terminaison de contact normalisées selon le numéro et la valeur électrique des connecteurs des séries "A" et "B".

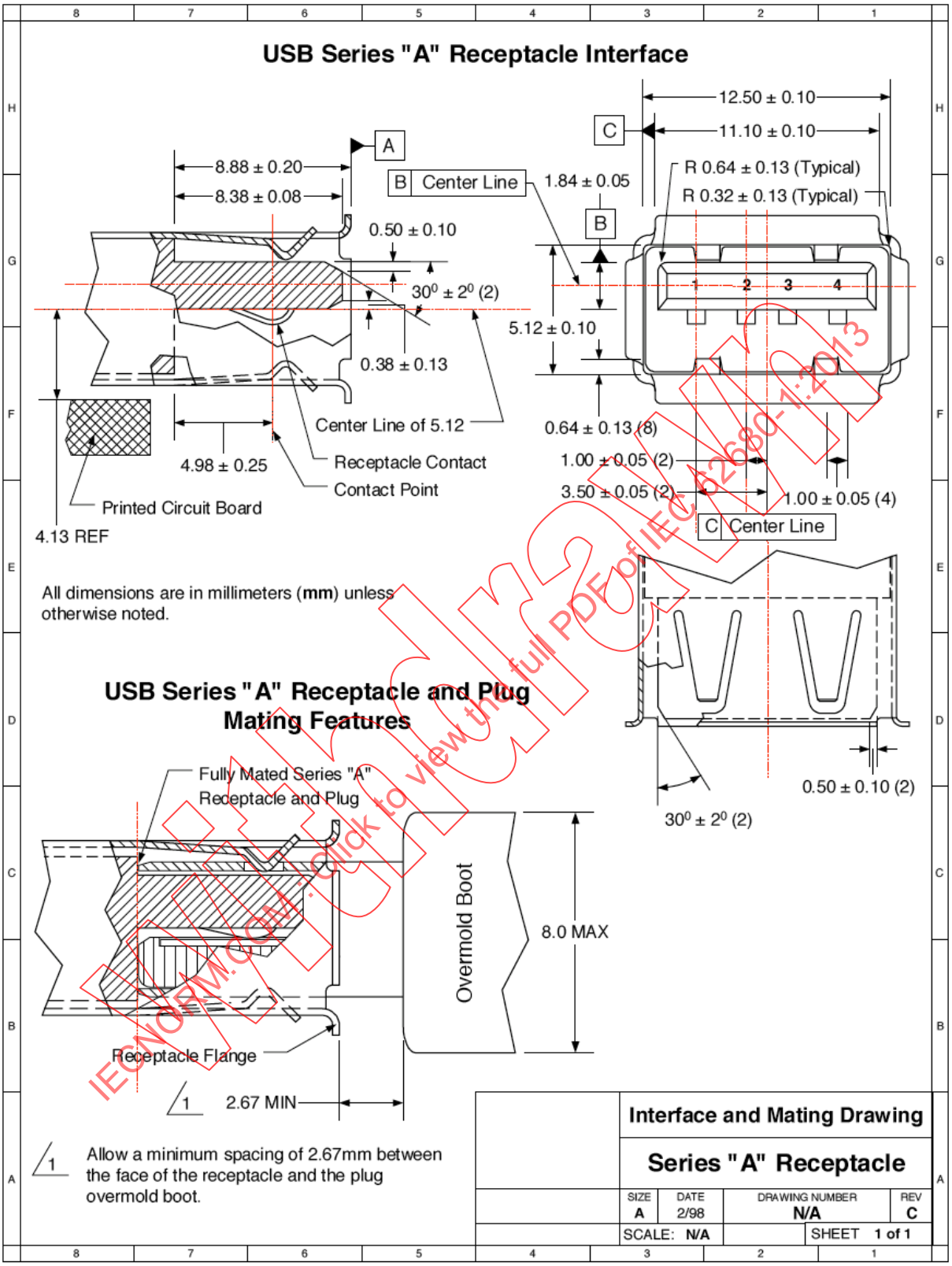
Tableau 6-1. Attribution de terminaison des connecteurs USB

Numéro de contact	Nom du signal	Attribution typique du câblage
1	VBUS	Rouge
2	D-	Blanc
3	D+	Verte
4	GND	Noire
Coque	Blindage	Conducteur de drainage

6.5.3 Réceptacles des séries "A" et "B"

Les données de la configuration des interfaces électriques et mécaniques des réceptacles des séries "A" et "B" sont illustrées à la Figure 6-7 et à la Figure 6-8. Voir aussi la Figure 6-12, à la Figure 6-13, et à la Figure 6-14 de la fin du présent chapitre pour les dispositions typiques des réceptacles PCB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

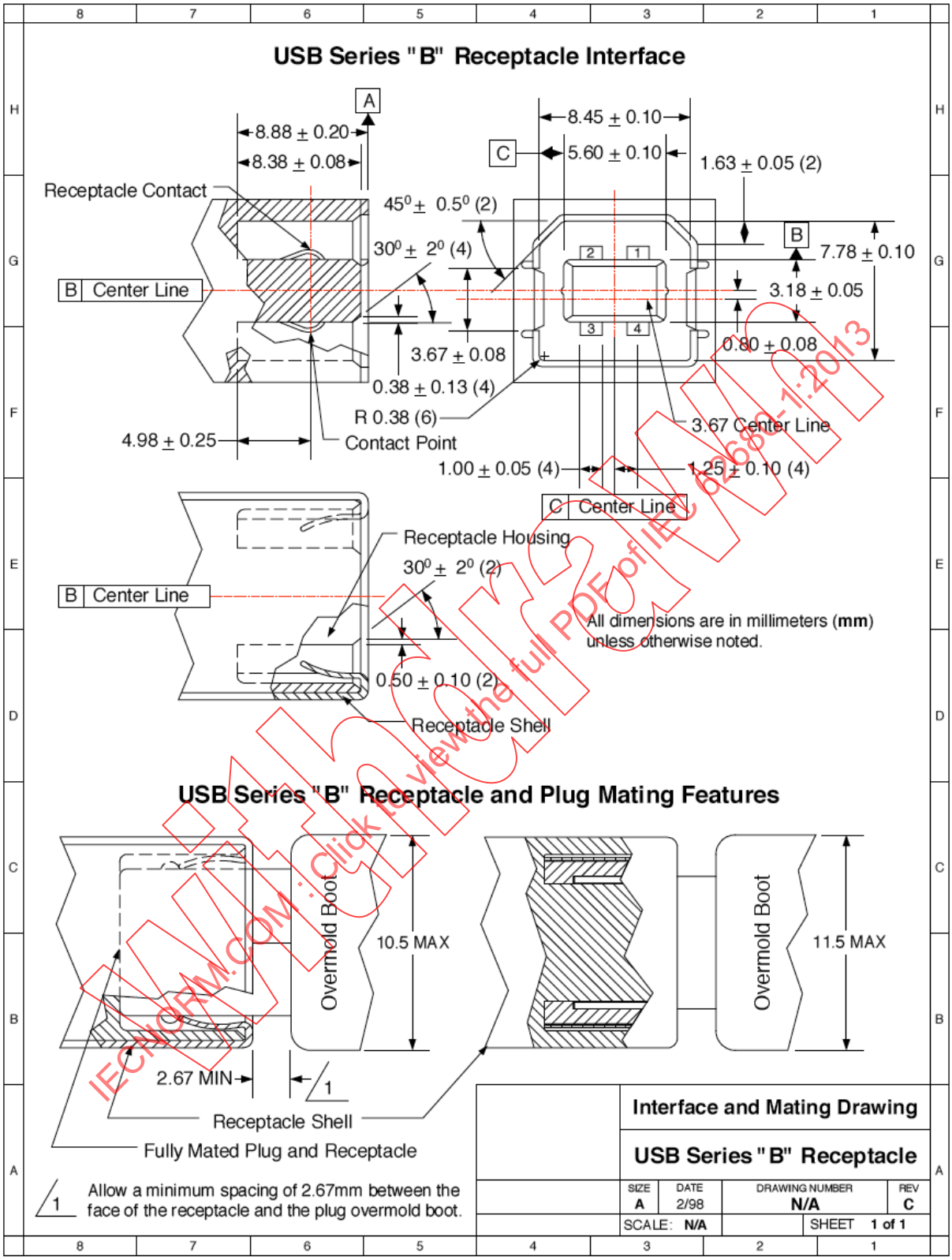
Légende

Anglais	Français
USB Series "A" Receptacle Interface	Interface de réceptacle USB de série "A"
Center Line	Ligne centrale
Typical	Typique
Receptacle Contact	Contact de réceptacle
Contact Point	Point de contact
Printed Circuit Board	Carte à circuit imprimé

Anglais	Français
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
USB Series "A" Receptacle and Plug Mating Features	Réceptacle USB de série "A" et caractéristiques d'accouplement de fiche
Fully Mated Series "A" Receptacle and Plug	Réceptacle de série "A" entièrement couplé à la fiche
Overmold Boot	Coiffe surmoulée
Receptacle Flange	Bride de réceptacle
Allow a minimum spacing of 2.67mm between the face of the receptacle and the plug overmold boot.	Permet un espacement de 2,67 mm entre la face du réceptacle et la coiffe surmoulée de la fiche
Interface and Mating Drawing	Schéma d'interface et d'accouplement
Series "A" Receptacle	Réceptacle de série "A"
SIZE	TAILLE
DATE	DATE
DRAWING NUMBER N/A	NUMÉRO DE DESSIN N/A
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-7. Schéma d'interface et d'accouplement de réceptacle USB de série "A"

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Légende

Anglais	Français
USB Series "B" Receptacle Interface	Interface de réceptacle USB de série "B"
Center Line	Ligne centrale
Receptacle Contact	Contact de réceptacle
Contact Point	Point de contact
Receptacle Housing	Logement de réceptacle
Receptacle Shell	Coque de réceptacle

Anglais	Français
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
USB Series "B" Receptacle and Plug Mating Features	Réceptacle USB de série "B" et caractéristiques d'accouplement de fiche
Overmold Boot	Coiffe surmoulée
Receptacle Shell	Coque de réceptacle
Fully Mated Plug and Receptacle	Fiche et réceptacle entièrement couplés
Allow a minimum spacing of 2.67mm between the face of the receptacle and the plug overmold boot.	Permet un espacement de 2,67 mm entre la face du réceptacle et la coiffe surmoulée de la fiche
Interface and Mating Drawing	Schéma d'interface et d'accouplement
USB Series "B" Receptacle	Réceptacle de série "B"
SIZE	TAILLE
DATE	DATE
DRAWING NUMBER N/A	NUMÉRO DE DESSIN N/A
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-8. Schéma d'interface et d'accouplement de réceptacle USB de série "B"

6.5.3.1 Matériau isolant thermoplastique moulé par injection de réceptacle

Polytéréphtalate de butylène (PBT) ou polytéréphtalate d'éthylène (PET) à renfort de verre de trente pour cent (30 %) avec une norme minimum UL 94-V0 ou de qualité supérieure.

Couleurs typiques: Noir, gris, naturel.

Caractéristiques d'inflammabilité: norme UL 94-V0.

Le paquet ignifuge doit satisfaire ou dépasser les exigences des normes UL, CSA, VDE, etc.

Indice d'oxygène (LOI): supérieur à 21 %. ASTM D 2863.

6.5.3.2 Matériaux des coques de réceptacle

Matériau du substrat: couche de bronze phosphoreux de 0,30 + 0,05 mm, maillechort ou autres matériaux à haute résistance à base de cuivre.

Placage:

1. Sous-placage: Facultatif. Couche de nickel de 1,00 micromètre (40 micropouces) au minimum. En outre, le fabricant peut utiliser un sous-placage en cuivre en dessous du nickel.
2. Extérieur: couche d'étain brillant ou d'étain au plomb brillant de 2,5 micromètres (100 micropouces) au minimum.

6.5.3.3 Matériaux de contact de réceptacle

Matériau du substrat: couche de bronze phosphoreux demi-durci de 0,30 ± 0,05 mm au minimum ou autre matériau à haute résistance à base de cuivre.

Placage: les contacts doivent être plaqués de manière sélective.

A. Option I
    1. Sous-placage: couche de nickel de 1,25 micromètre (50 micropouces) au minimum. Un matériau de base cuivré est facultatif.
    2. Zone d'accouplement: couche d'or de 0,05 micromètre (2 micropouces) au minimum sur couche de palladium de 0,70 micromètre (28 micropouces)
    3. Queues de brasage: couche d'étain au plomb brillant de 3,8 micromètres (150 micropouces) au minimum au-dessus du sous-placage.
B. Option II

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

1. Sous-placage: couche de nickel de 1,25 micromètre (50 micropouces) au minimum. Un matériau de base cuivré est facultatif.
2. Zone d'accouplement: couche d'or de 0,05 micromètre (2 micropouces) au minimum sur couche de nickel de palladium de 0,75 micromètre (30 micropouces)
3. Queues de brasage: couche d'étain au plomb brillant de 3,8 micromètres (150 micropouces) au minimum au-dessus du sous-placage.

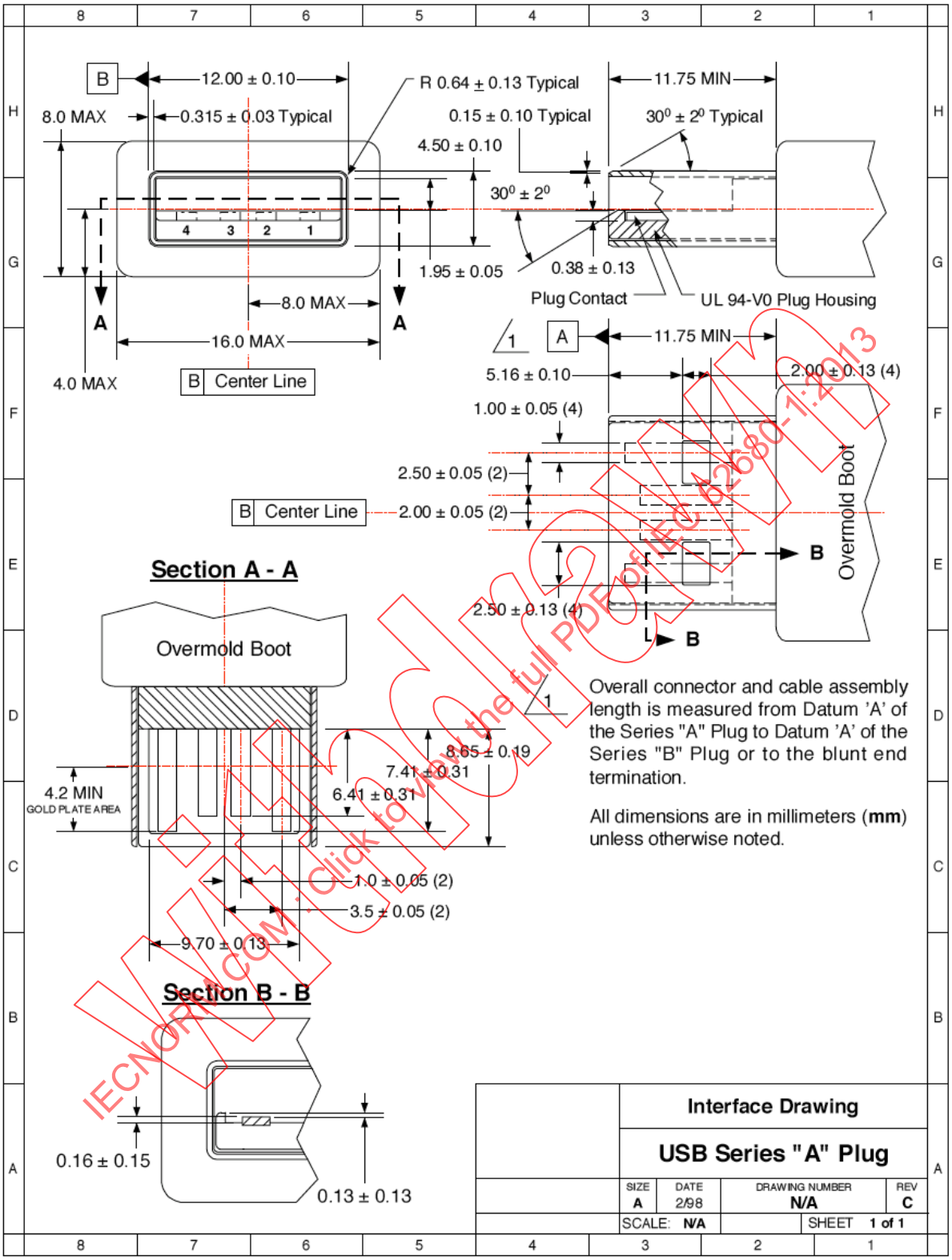
C. Option III

1. Sous-placage: couche de nickel de 1,25 micromètre (50 micropouces) au minimum. Un matériau de base cuivré est facultatif.
2. Zone d'accouplement: couche d'or de 0,75 micromètre (30 micropouces) au minimum.
3. Queues de brasage: couche d'étain au plomb brillant de 3,8 micromètres (150 micropouces) au minimum au-dessus du sous-placage.

6.5.4 Fiches des séries "A" et "B"

Les données de la configuration des interfaces électriques et mécaniques des fiches des séries "A" et "B" sont affichées à la Figure 6-9 et à la Figure 6-10.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

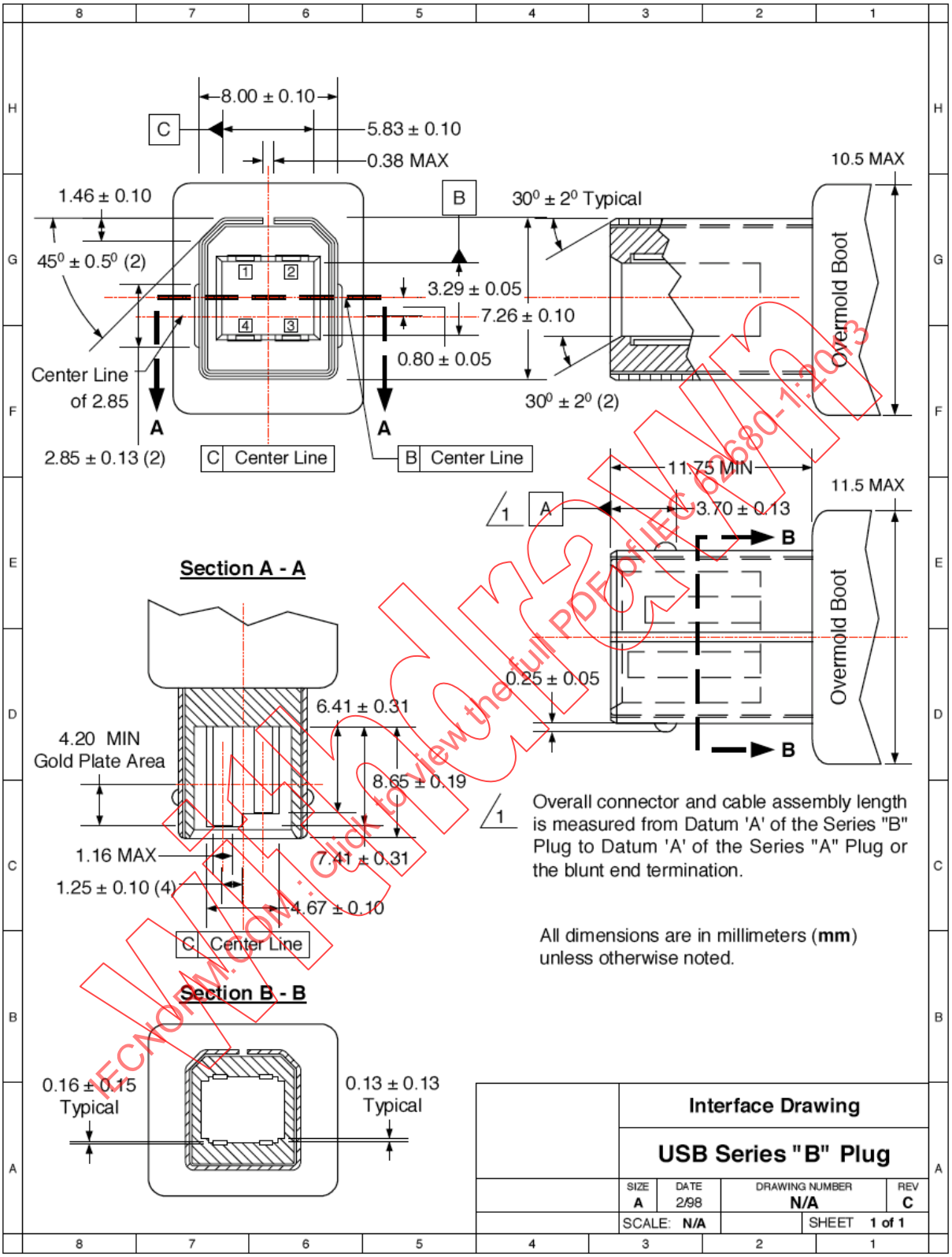
Légende

Anglais	Français
Plug Contact	Contact de fiche
Typical	Typique
Plug Housing	Logement de fiche
Center Line	Ligne centrale
Overmold Boot	Coiffe surmoulée
Section A-A	Section A-A
Overall connector et cable assembly	La longueur totale du connecteur et de l'ensemble de

Anglais	Français
length is measured from Datum 'A' of the Series "A" Plug to Datum 'A' of the Series "B" Plug or to the blunt end termination.	câbles est mesurée à partir du Datum 'A' de la fiche de série "A" jusqu'au Datum 'A' de la fiche de série "B" ou jusqu'à la terminaison d'extrémité arrondie.
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
Section B-B	Section B-B
Interface Drawing	Schéma d'interface
USB Series "A" Plug	Fiche USB de série "A"
SIZE	TAILLE
DATE	DATE
DRAWING NUMBER N/A	NUMÉRO DE DESSIN N/A
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-9. Schéma d'interface de fiche USB de série "A"

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Center Line	Ligne centrale
Typical	Typique
Overmold Boot	Coiffe surmoulée
Section A-A	Section A-A
Overall connector et cable assembly length is measured from Datum 'A' of	La longueur totale du connecteur et de l'ensemble de câbles est mesurée à partir du Datum 'A' de la fiche de

Anglais	Français
the Series "B" Plug to Datum 'A' of the Series "B" Plug or to the blunt end termination.	série "B" jusqu'au Datum 'A' de la fiche de série "B" ou jusqu'à la terminaison d'extrémité arrondie.
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
Section B-B	Section B-B
Interface Drawing	Schéma d'interface
USB Series "B" Plug	Fiche USB de série "B"
SIZE	TAILLE
DATE	DATE
DRAWING NUMBER N/A	NUMÉRO DE DESSIN N/A
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-10. Schéma d'interface de fiche USB de série "B"

6.5.4.1 Matériau isolant thermoplastique moulé par injection de fiche

Polytéréphtalate de butylène (PBT) ou polytéréphtalate d'éthylène (PET) à renfort de verre de trente pour cent (30 %) avec une norme minimum UL 94-V0 ou de qualité supérieure.

Couleurs typiques: Noir, gris, naturel.

Caractéristiques d'inflammabilité: norme UL 94-V0.

Le paquet ignifuge doit satisfaire ou dépasser les exigences des normes UL, CSA, et VDE.

Indice d'oxygène (LOI): 21 %. ASTM D 2863.

6.5.4.2 Matériaux des coques de fiche

Matériau du substrat: couche de bronze phosphoreux de 0,30 ± 0,05 mm, maillechort ou autre matériau approprié.

Placage:

A. Sous-placage: Facultatif. Couche de nickel de 1,00 micromètre (40 micropouces) au minimum. En outre, le fabricant peut utiliser un sous-placage en cuivre en dessous du nickel.

B. Extérieur: couche d'étain brillant ou d'étain au plomb brillant de 2,5 micromètres (100 micropouces) au minimum.

6.5.4.3 Matériaux des contacts de fiche (mâle)

Matériau du substrat: bronze phosphoreux demidurci de 0,30 ± 0,05 mm.

Placage: les contacts doivent être plaqués de manière sélective.

A. Option I

   1. Sous-placage: couche de nickel de 1,25 micromètre (50 micropouces) au minimum. Un matériau de base cuivré est facultatif.
   2. Zone d'accouplement: couche d'or de 0,05 micromètre (2 micropouces) au minimum sur couche de palladium de 0,70 micromètre (28 micropouces)
   3. Queues de brasage: couche d'étain au plomb brillant de 3,8 micromètres (150 micropouces) au minimum au-dessus du sous-placage.

B. Option II

   1. Sous-placage: couche de nickel de 1,25 micromètre (50 micropouces) au minimum. Un matériau de base cuivré est facultatif.
   2. Zone d'accouplement: couche d'or de 0,05 micromètre (2 micropouces) au minimum sur couche de nickel de palladium de 0,75 micromètre (30 micropouces)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

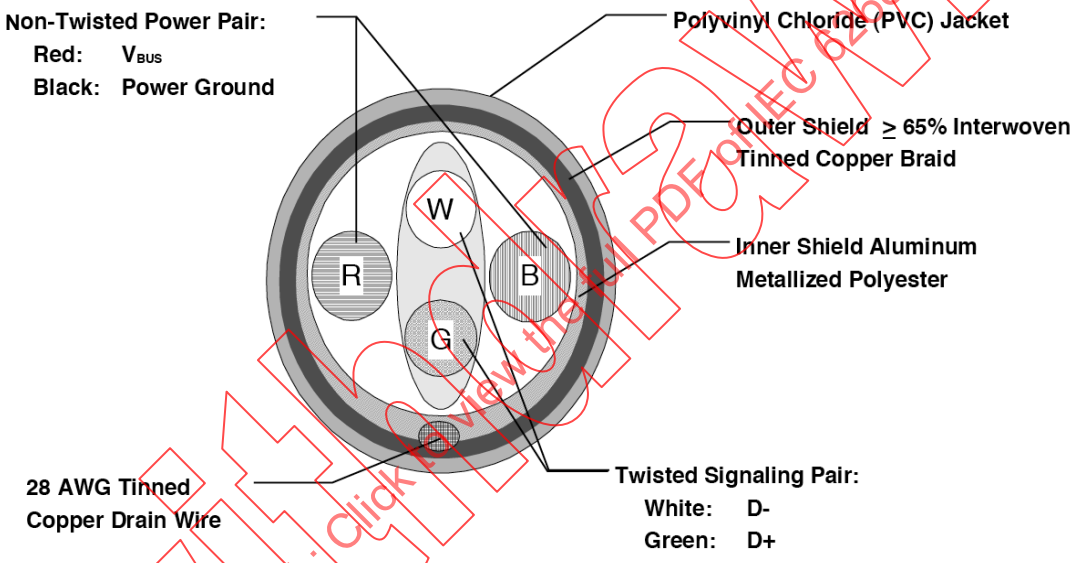
3. Queues de sertissage de fil électrique/brasage couche d'étain au plomb brillant de 3,8 micromètres (150 micropouces) au minimum au-dessus du sous-placage.

C. Option III

1. Sous-placage: couche de nickel de 1,25 micromètre (50 micropouces) au minimum. Un matériau de base cuivré est facultatif.
2. Zone d'accouplement: couche d'or de 0,75 micromètre (30 micropouces) au minimum.
3. Queues de brasage: couche d'étain au plomb brillant de 3,8 micromètres (150 micropouces) au minimum au-dessus du sous-placage.

6.6 Configuration mécanique des câbles et exigences en termes de matériel

Les câbles à haute/vitesse intégrale et à faible vitesse diffèrent en termes de branchement et de blindage des conducteurs de données. La faible vitesse recommande, mais n'exige pas, l'utilisation d'un câble avec des conducteurs de données torsadés. La faible vitesse recommande, mais n'exige pas, l'utilisation d'un câble avec un blindage externe en tresses. La Figure 6-11 illustre une construction typique de câble à haute/vitesse intégrale.



Légende

Anglais	Français
Non-Twisted power pair: Red: V_{BUS} Black: Power Ground	Paire d'alimentation non torsadée: Rouge: V_{BUS} Noir: masse
Polyvinyl Chloride (PVC) Jacket	Chemise en polychlorure de vinyle (PVC)
Outer Shield ≥ 65% Interwoven Tinned Copper Braid	Blindage externe ≥65 % de blindage entrelacé en tresses en cuivre étamé
Inner Shield Aluminium Metallized Polyester	Blindage interne en polyester métallisé d'aluminium
28 AWG Tinned Copper Drain Wire	Conducteur de drainage toronné en cuivre étamé de 28 AWG
Twisted Signaling Pair: White: D- Green: D+	Paire de signalisation torsadée: Blanc: D- Vert: D+

Figure 6-11. Construction typique de câble à haute/vitesse intégrale

6.6.1 Description

Le câble à haute/vitesse intégrale est formé d'une paire d'alimentation non torsadée de 28 à 20 AWG et d'une paire d'alimentation torsadée de 28 AWG avec un blindage interne en polyester métallisé d'aluminium, d'un conducteur de drainage toronné en cuivre étamé de 28 AWG, blindage externe (en tresses) entrelacé en fil de cuivre étamé ≥ 65 % et d'une chemise externe en PVC.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Le câble à faible vitesse est formé d'une paire d'alimentation non torsadée de 28 à 20 AWG et d'une paire de données de 28 AWG (une torsion est recommandée) avec un blindage interne en polyester métallisé d'aluminium, d'un conducteur de drainage toronné en cuivre étamé de 28 AWG et d'une chemise externe en PVC. Un blindage externe (en tresses) entrelacé en fil de cuivre étamé ≥ 65 % est recommandé.

6.6.2 Construction

Les matières premières utilisées dans la fabrication de ce câble doivent être d'une qualité telle que le câble fabriqué soit en mesure de satisfaire ou de dépasser les critères de performance mécanique et électrique de la version la plus récente de la spécification USB et de toutes les exigences locales des agences de sécurité/d'essai locales et internationales; par exemple, UL, CSA, BSA, NEC, etc. pour les câbles de signalisation électronique et de distribution électrique de sa catégorie.

Tableau 6-2. Paire d'alimentation

Système AWG (American Wire Gauge)	Diamètre externe nominal du conducteur	Conducteurs toronnés en cuivre étamé
28	0,381 mm (0,015 ") 0,406 mm (0,016 ")	7 x 36 19 x 40
26	0,483 mm (0,019 ") 0,508 mm (0,020 ")	7 x 34 19 x 38
24	0,610 mm (0,024 ") 0,610 mm (0,024 ")	7 x 32 19 x 36
22	0,762 mm (0,030 ") 0,787 mm (0,031 ")	7 x 30 19 x 34
20	0,890 mm (0,035 ") 0,931 mm (0,037 ")	7 x 28 19 x 32

Note: la construction minimum des conducteurs doit être toronnée en cuivre étamé.

Paire d'alimentation non torsadée:

A. Diamètre de fil: 28 AWG au minimum ou tel que spécifié par les exigences de l'utilisation selon la longueur de câble spécifiée. Voir Tableau 6-2.
B. Isolation de fil: polychlorure de vinyle (PVC) semi-rigide
   1. Epaisseur de paroi d'isolation nominale: 0,25 mm (0,010 ")
   2. Conducteur d'alimentation (V_{BUS}) typique: isolation rouge
   3. Conducteur de terre typique: isolation noire

Paire de signalisation:

A. Diamètre de fil: 28 AWG au minimum. Voir Tableau 6-3.

Tableau 6-3. Paire de signalisation

Système AWG (American Wire Gauge)	Diamètre externe nominal du conducteur	Conducteurs toronnés en cuivre étamé
28	0,381 mm (0,015 ") 0,406 mm (0,016 ")	7 x 36 19 x 40

Note: la construction minimum des conducteurs doit être toronnée en cuivre étamé.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

B. Isolation de fil: polyéthylène haute densité (PEHD), sinon mousse de polyéthylène ou de polypropylène

1. Epaisseur de paroi d'isolation nominale: 0,31 mm (0,012 ")
2. Conducteur Plus (+) de données typique: isolation verte
3. Conducteur Moins (-) de données typique: isolation blanche

C. Nombre nominal de torsions (non requis pour la faible vitesse): une torsion complète tous les 60 mm (2,36 ") à 80 mm (3,15 ")

Blindage interne polyester métallisé d'aluminium (requis pour la faible vitesse):

A. Matériau du substrat: polytéréphtalate d'éthylène (PET) ou matériau équivalent

B. Métallisation: dépôts d'aluminium sous vide

C. Ensemble:

1. Le côté métallisé en aluminium du blindage interne doit être positionné de face afin de garantir un contact direct avec le conducteur de drainage.
2. Le blindage interne métallisé en aluminium doit se chevaucher d'un quart de tour approximativement.

Conducteur de drainage (requis pour la faible vitesse):

A. Diamètre de fil: toronné en cuivre étamé (STC) de 28 AWG au minimum non isolé. Voir Tableau 6-4.

Tableau 6-4. Paire de signalisation de conducteur de drainage

Système AWG (American Wire Gauge)	Diamètre externe nominal du conducteur	Conducteurs toronnés Conducteurs
28	0,381 mm (0,015 ") 0,406 mm (0,016 ")	7 x 36 19 x 40

Blindage externe entrelacé (en tresses) en fil de cuivre étamé (ITCW) (recommandé, mais non requis pour la faible vitesse):

A. Zone de couverture: 65 % au minimum.

B. Ensemble: le blindage externe entrelacé (en tresses) en fil de cuivre étamé doit recouvrir les paires d'alimentation et de signalisation blindées en polytéréphtalate d'éthylène et métallisées en aluminium et être en contact direct avec le conducteur de drainage.

Chemise externe en polychlorure de vinyle (PVC):

A. Ensemble: la chemise externe en PVC doit recouvrir les paires d'alimentation et de signalisation entièrement blindées et être en contact direct avec le blindage externe en cuivre étamé.

B. Epaisseur de paroi nominale: 0,64 mm (0,025 ").

Marquage: le câble doit être marqué lisiblement avec de l'encre permanente à couleurs contrastées.

A. Les informations minimales concernant le marquage du câble à haute/vitesse intégrale doivent inclure:
USB SHIELDED <Gauge/2C + Gauge/2C> UL CM 75 °C - UL Vendor ID.

B. Les informations minimales concernant le marquage du câble à faible vitesse doivent inclure:
le marquage spécifique à l'USB n'est pas requis pour le câble à faible vitesse.

Diamètre externe nominal de câble fabriqué:

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Cette valeur est nominale et peut légèrement varier d'un fabricant à l'autre en fonction des matériaux d'isolation de conducteur et du conducteur spécifiés. Voir Tableau 6-5.

Tableau 6-5. Diamètre nominal de câble

Configuration de câble USB blindé	Diamètre externe nominal de câble
28/28	4,06 mm (0,160 ")
28/26	4,32 mm (0,170 ")
28/24	4,57 mm (0,180 ")
28/22	4,83 mm (0,190")
28/20	5,21 mm (0,205 ")

6.6.3 Caractéristiques électriques

Toutes les caractéristiques électriques doivent être mesurées ou référencées à +20 °C (68 °F).

Tension nominale: 30 V rms maximum.

Résistance des conducteurs: la résistance des conducteurs doit être mesurée conformément à la Section 13 de la norme ASTM-D-4566. Voir Tableau 6-6.

Déséquilibre (paires) de résistance des conducteurs: le déséquilibre de résistance des conducteurs entre deux (2) conducteurs de chaque paire ne doit pas dépasser cinq pour cent (5 %) lorsqu'il est mesuré conformément à la Section 15 de la norme ASTM-D-4566.

La résistance en courant continu d'une coque de fiche à l'autre (ou extrémité de câble intégré) doit être inférieure à 0,6 ohm.

Tableau 6-6. Résistance des conducteurs

Système AWG (American Wire Gauge)	Ohms (Ω) / 100 mètres Maximum
28	23,20
26	14,60
24	9,09
22	5,74
20	3,58

6.6.4 Caractéristiques environnementales des câbles

Plage de températures:

A. Plage de températures de fonctionnement: 0 °C à +50 °C
B. Plage de températures de stockage: -20 °C à +60 °C
C. Valeur nominale de Température: +20 °C

Inflammabilité: Tous les matériaux de plastique utilisés dans la fabrication de ce produit doivent satisfaire ou dépasser les exigences de la norme NEC Article 800 pour les câbles de communication de Type CM (Commercial).

6.6.5 Référence

Le produit doit être référencé UL selon les exigences UL Sujet 444, Classe 2, Type CM des câbles de communication.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

6.7 Normes de conformité électrique, mécanique et environnementale

Le Tableau 6-7 indique les critères d'essai minimum pour tous les câbles, ensembles de câbles et connecteurs USB.

Tableau 6-7. Normes de conformité électrique, mécanique et environnementale USB

Description de l'essai	Procédure d'essai	Exigence de performance
Inspection visuelle et dimensionnelle	EIA 364-18 Inspection visuelle, dimensionnelle et fonctionnelle conformément aux plans d'inspection de qualité USB.	Doit satisfaire ou dépasser les exigences spécifiées par la version la plus récente du Chapitre 6 de la spécification USB.
Résistance d'isolement	EIA 364-21 L'objet de cette procédure d'essai est de détailler une méthode normalisée d'évaluation de la résistance d'isolement des connecteurs USB. Cette procédure d'essai est utilisée pour déterminer la résistance offerte par les matériaux isolants et les différents joints d'un connecteur à un potentiel en courant continu qui a tendance à produire une fuite de courant à travers ou sur la surface de ces membres.	1 000 MΩ au minimum.
Tension de tenue diélectrique	EIA 364-20 L'objet de cette procédure d'essai est de détailler une méthode d'essai visant à prouver qu'un connecteur USB peut fonctionner de manière sûre à sa tension nominale et supporter les surpotentiels à cause de la commutation, des surtensions e/ou d'autres phénomènes similaires.	Le diélectrique doit supporter 500 V c.a. pendant une minute au niveau de la mer.
Résistance de contact de faible niveau	EIA 364-23 L'objet de cet essai est de détailler une méthode normalisée visant à mesurer la résistance électrique sur une paire de contacts accouplés sans que les films isolants, le cas échéant, se rompent ou que la fusion des aspérités se produise.	30 mΩ au maximum lorsque la mesure est effectuée en circuit ouvert de 20 mV maximum à 100 mA. Les contacts d'accouplement d'essai doivent se trouver dans un logement de connecteurs.
Courant de contact nominal	EIA 364-70 - Méthode B L'objet de cette procédure d'essai est de détailler une méthode normalisée d'évaluation de la capacité portante de courant des contacts de connecteurs d'accouplement USB.	1,5 A à 250 V c.a. au minimum lorsque la mesure est effectuée à une température ambiante de 25 °C. Lorsque les contacts sont alimentés, la valeur Δ T ne doit dépasser +30 °C à aucun moment sur le connecteur USB soumis à essai.
Capacitance de contact	EIA 364-30 L'objet de cet essai est de détailler une méthode normalisée visant à déterminer la capacitance entre les éléments conducteurs d'un connecteur USB.	2 pF au maximum par contact non couplé.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 6-7. Normes de conformité électrique, mécanique et environnementale USB (suite)

Description de l'essai	Procédure d'essai	Exigence de performance
Force d'insertion	EIA 364-13 L'objet de cet essai est de détailler une méthode normalisée visant à déterminer les forces mécaniques requises pour insérer un connecteur USB.	35 Newtons au maximum pour une vitesse maximum de 12,5 mm (0,492 ") par minute.
Force d'extraction	EIA 364-13 L'objet de cet essai est de détailler une méthode normalisée visant à déterminer les forces mécaniques requises pour extraire un connecteur USB.	10 Newtons au minimum pour une vitesse maximum de 12,5 mm (0,492 ") par minute.
Durabilité	EIA 364-09 L'objet de cette procédure d'essai est de détailler une méthode d'essai uniforme visant à déterminer les effets causés par la soumission d'un connecteur USB à l'action de mise en condition d'insertion et d'extraction, simulant la durée de vie prévue des connecteurs. Le cycle de durabilité à l'aide d'une jauge est prévu pour produire un effort mécanique uniquement. La durabilité effectuée avec des composantes d'accouplement consiste à produire un effort mécanique et un effet d'usure.	1 500 cycles d'insertion/extraction à une vitesse maximum de 200 cycles par heure.
Retrait de câble	EIA 364-38 Condition d'essai A L'objet de cette procédure d'essai est de détailler une méthode normalisée visant à déterminer l'effet de retenue d'un appareil de serrage de câble de fiche USB sans avoir de conséquence négative sur le câble ou sur les composantes du connecteur lorsque le câble est soumis à des charges axiales de traction par inadvertance.	Après l'application d'une charge axiale en régime permanent de 40 Newtons pendant une minute.
Choc physique	EIA 364-27 Condition d'essai H L'objet de cette procédure d'essai est de détailler une méthode normalisée visant à évaluer la capacité d'un connecteur USB à supporter la sévérité spécifiée du choc mécanique.	Aucune discontinuité de 1 µs ou d'une durée plus longue lorsque les connecteurs USB d'accouplement sont soumis à des chocs demi-sinusoïdaux de 30 G pendant 11 ms. Trois chocs appliqués dans chaque direction sur trois plans perpendiculaires pour un total de 18 chocs.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 6-7. Normes de conformité électrique, mécanique et environnementale USB (suite)

Description de l'essai	Procédure d'essai	Exigence de performance
Vibrations aléatoires	EIA 364-28 Condition d'essai V Lettre d'essai A Cette procédure d'essai s'applique aux connecteurs USB qui peuvent être soumis, en cours de fonctionnement, à des conditions impliquant des vibrations. Il convient que la spécification détaillée du produit indique clairement si un connecteur USB doit fonctionner durant les vibrations ou seulement survivre aux conditions de vibration. Quel que soit le cas, il convient que la spécification pertinente stipule les tolérances de performance acceptables.	Aucune discontinuité de 1 µs ou d'une durée plus longue lorsque les connecteurs USB d'accouplement sont soumis à des valeurs efficaces de 5,35 G. 15 minutes sur chacun des trois plans réciproquement perpendiculaires.
Choc thermique	EIA 364-32 Condition d'essai I L'objet de cet essai est de déterminer la résistance d'un connecteur USB à l'exposition à des températures basses et élevées extrêmes et au choc d'expositions alternées à ces conditions extrêmes simulant les conditions les plus défavorables de stockage, transport et application.	10 cycles -55 °C et +85 °C. Les connecteurs USB soumis à essai doivent être accouplés.
Durée de vie dans des conditions d'humidité	EIA 364-31 Condition d'essai A Méthode III L'objet de cette procédure d'essai est de détailler une méthode d'essai normalisée pour l'évaluation des propriétés des matériaux utilisés dans les connecteurs USB lorsqu'ils sont influencés par les conditions d'humidité et de chaleur élevées.	168 heures au minimum (sept cycles complets). Les connecteurs USB soumis à essai doivent être soumis à essai conformément à l'EIA 364-31.
Brasabilité	EIA 364-52 L'objet de cette procédure d'essai est de détailler une méthode d'essai uniforme pour déterminer l'aptitude au brasage des connecteurs USB. La procédure d'essai décrite ici utilise la technique de brasage par immersion. Elle ne prétend ni soumettre à essai ni évaluer les bornes en coupelle, les œillets à souder, un autre type soudé à la main ou les terminaisons de type SMT.	Les queues de brasage des contacts USB doivent assurer une couverture de 95 % après une heure de vieillissement à la vapeur tel que spécifié dans la Catégorie 2.
Inflammabilité	UL 94 V-0 Cette procédure vise à assurer la conformité de la résine thermoplastique aux normes d'inflammabilité UL.	Le fabricant exige à son vendeur de résine thermoplastique de fournir un certificat de conformité pour chaque livraison de résine. Le certificat de conformité doit clairement indiquer le numéro de liste UL, le numéro de lot, le code de date, etc. de la résine.
Inflammabilité	UL 94 V-0 Cette procédure vise à assurer la conformité de la résine thermoplastique aux normes d'inflammabilité UL.	Le fabricant exige à son vendeur de résine thermoplastique de fournir un certificat de conformité pour chaque livraison de résine. Le certificat de conformité doit clairement indiquer le numéro de liste UL, le numéro de lot, le code de date, etc. de la résine.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 6-7. Normes de conformité électrique, mécanique et environnementale USB (suite)

Description de l'essai	Procédure d'essai	Exigence de performance
Impédance des câbles (requise uniquement pour la haute/vitesse intégrale)	L'objet de cet essai est de s'assurer que les conducteurs de signal ont une impédance correcte. Connecter les sorties du réflectomètre de domaine temporel (TDR) au banc d'essai d'impédance/délai/différence (Note 1). Utiliser 50 Ω câbles différents pour les sorties plus (ou vrai) et moins (ou complément). Régler la tête du TDR sur le mode TDR différentiel. Connecter la fiche de série "A" du câble qui doit être soumis au banc d'essai tout en laissant l'autre extrémité en circuit ouvert. Définir une forme d'onde composée de la différence entre les formes d'onde vraie ou de complément afin de permettre la mesure de l'impédance différentielle. Mesurer les impédances minimum et maximum mesurées entre le connecteur et l'extrémité en circuit ouvert du câble.	L'impédance doit se trouver dans l'intervalle spécifiée au Tableau 7-9 (ZO).
Atténuation des paires de signal (requise uniquement pour la haute/vitesse intégrale)	L'objet de cet essai est de s'assurer que la puissance du signal nécessaire est présentée au récepteur afin de maintenir un faible pourcentage d'erreur. 1. Connecter le port de sortie de l'analyseur de réseau (port 1) au connecteur d'entrée situé sur le banc d'essai d'atténuation (Note 2). 2. Connecter la fiche de série "A" du câble qui doit être soumis au banc d'essai tout en laissant l'autre extrémité en circuit ouvert. 3. Etalonner l'analyseur de réseau et le banc à l'aide des normes d'étalonnage appropriées sur l'intervalle de fréquences souhaité. 4. Suivre la méthode décrite dans la note d'application Hewlett Packard 380-2 pour mesurer la réponse sans terminaison du câble. 5. Court-circuiter l'extrémité de série "B" (ou l'extrémité nue en cas de câble captif) et mesurer la réponse du court-circuit. 6. A l'aide de la note d'app. H-P note 380-2 ou équivalent, calculer la valeur de l'atténuation de câble des effets de résonance dans le câble en fonction des besoins.	Voir la Section 7.1.17 pour l'intervalle de fréquences et l'atténuation permise.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 6-7. Normes de conformité électrique, mécanique et environnementale USB (suite)

Description de l'essai	Procédure d'essai	Exigence de performance
Délai de propagation	L'objectif de cet essai est de vérifier la propagation d'une extrémité à l'autre du câble. 1. Connecter une sortie de la tête d'échantillonnage du TDR aux entrées D+ et D- du banc d'essai d'impédance/délai/différence (Note 1). Utiliser un câble de 50 Ω pour chaque signal et régler la tête du TDR en mode TDR différentiel. 2. Connecter le câble qui doit être soumis au banc d'essai. Si le câble est amovible, brancher les deux connecteurs dans les connecteurs correspondants du banc. Si le câble est captif, brancher la fiche de série "A" dans le connecteur correspondant du banc et souder les fils de sortie dénudés à l'autre extrémité du banc d'essai. 3. Mesurer le délai de propagation du banc d'essai en connectant une petite pièce de fil électrique le long du banc d'une entrée à une sortie et enregistrer le délai. 4. Déposer la petite pièce de fil électrique et mesurer de nouveau le délai de propagation. Le soustraire du délai du banc d'essai mesuré à l'étape précédente.	Haute/vitesse intégrale. Voir Section 7.1.1.1, Section 7.1.4, Section 7.1.16 et Tableau 7-9 (TFSCBL). Faible vitesse. Voir Section 7.1.1.2, Section 7.1.16 et Tableau 7-9 (TLSCBL).
Différence de délai de propagation	Cet essai s'assure que le signal des deux lignes D+ et D- arrivent au niveau du récepteur au même moment. 1. Connecter le TDR au banc avec le câble de l'échantillon d'essai, comme dans la section précédente. 2. Mesurer la différence de délai pour les deux conducteurs du câble d'essai. Utiliser les curseurs TDR pour trouver l'extrémité en court-circuit de chaque conducteur (où l'impédance tend vers l'infini) et soustraire la différence de temps entre les valeurs.	La différence de propagation doit satisfaire les exigences énumérées à la Section 7.1.3.
Charge capacitive Requise pour la faible vitesse uniquement.	L'objectif de cet essai est de s'assurer que la capacitance distribuée entre les fils électriques est inférieure à la capacitance globale spécifiée par le pilote de transmission à faible vitesse. 1. Connecter le fil de sortie à l'analyseur d'impédance à la broche D+ du banc d'impédance/délai/différence (Note 1) et l'autre fil de sortie à la broche D-. 2. Connecter la fiche de série "A" au banc, avec les fils de sortie de l'extrémité de série "B" étant court-circuités. 3. Régler l'analyseur d'impédance à une fréquence de 100 kHz pour mesurer la capacitance.	Voir Section 7.1.1.2 et Tableau 7-7 (CLINUA).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Note 1: banc d'essai d'impédance, de délai de propagation et de différence
Ce banc est utilisé avec le TDR pour la mesure de la performance du domaine temporel du câble soumis à essai. Il convient que l'impédance du banc corresponde à l'équipement, généralement 50 Ω. Il convient que le banc soit équipé de connecteurs coaxiaux pour la connexion du TDR.

Note 2: banc d'essai d'atténuation
Ce banc fournit un moyen de connexion de l'analyseur de réseau à la fiche de série "A". Etant donné que les signaux USB sont de nature différentielle et qu'ils fonctionnent sur un câble déséquilibré, l'utilisation d'un transformateur ou un transformateur symétrique-dissymétrique (North Hills NH13734 ou équivalent) est idéale. Le transformateur convertit le signal déséquilibré (également appelé asymétrique) provenant du générateur de signaux qui représente généralement une sortie de 50 Ω pour l'impédance équilibrée (également appelée différentielle) et probablement différemment chargée et présentée par le câble. Il convient qu'un deuxième transformateur ou transformateur symétrique-dissymétrique soit utilisé sur l'autre extrémité du câble soumis à essai pour reconvertir le signal sous la forme équilibrée de l'impédance pour correspondre à l'analyseur de réseau.

6.7.1 Documents applicables

American National Standard/Electronic Industries Association

ANSI/EIA-364-C (12/94)	Procédures d'essai des connecteurs/prises électriques, y compris les classifications environnementales

American Standard Test Materials

ASTM-D-4565	Propriétés de performance physique et environnementale d'isolation et chemise pour fil et câble de télécommunication, méthode d'essai normalisée
ASTM-D-4566	Propriétés de performance électrique d'isolation et chemise pour fil et câble de télécommunication, méthode d'essai normalisée

Underwriters' Laboratory, Inc.

UL STD-94	Essai d'inflammabilité des matériaux plastiques utilisés pour les pièces des appareils
UL Subject-444	Câbles de communication

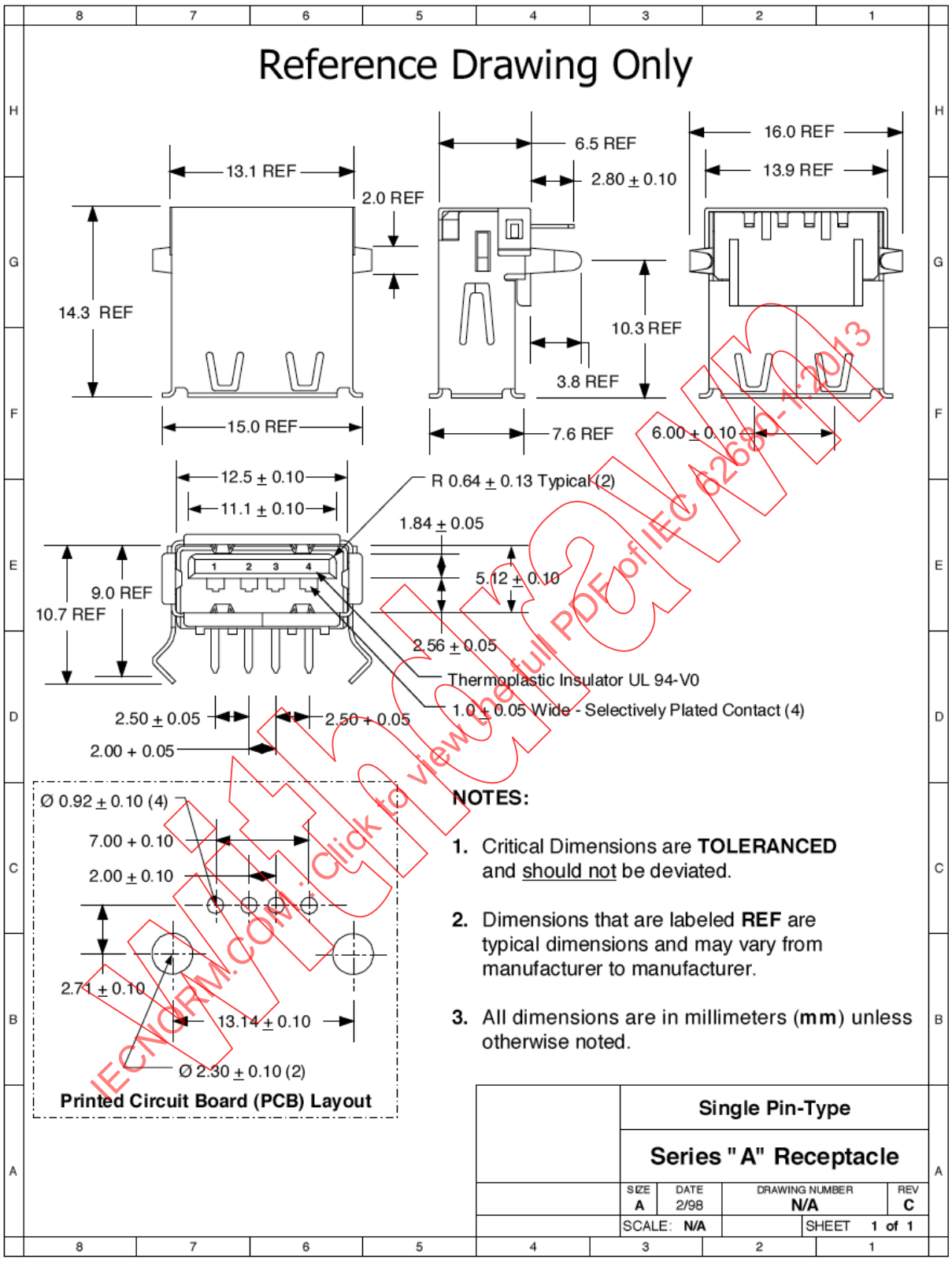
6.8 Mise à la terre USB

Le blindage doit se terminer au niveau de la fiche de connecteur pour les ensembles complets. Le blindage et le châssis reliés ensemble. Le schéma de mise à la terre sélectionné par l'utilisateur pour les appareils et les câbles USB doit être cohérent avec les pratiques reconnues dans l'industrie et les normes des organismes de réglementation pour la sécurité et les perturbations électromagnétiques/électrostatiques/radioélectriques (EMI/ESD/RFI).

6.9 Schémas de référence PCB

Les plans présentés à la Figure 6-12, à la Figure 6-13 et à la Figure 6-14 décrivent les interfaces PCB de réceptacle typiques. Ces plans sont fournis à titre d'information uniquement.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

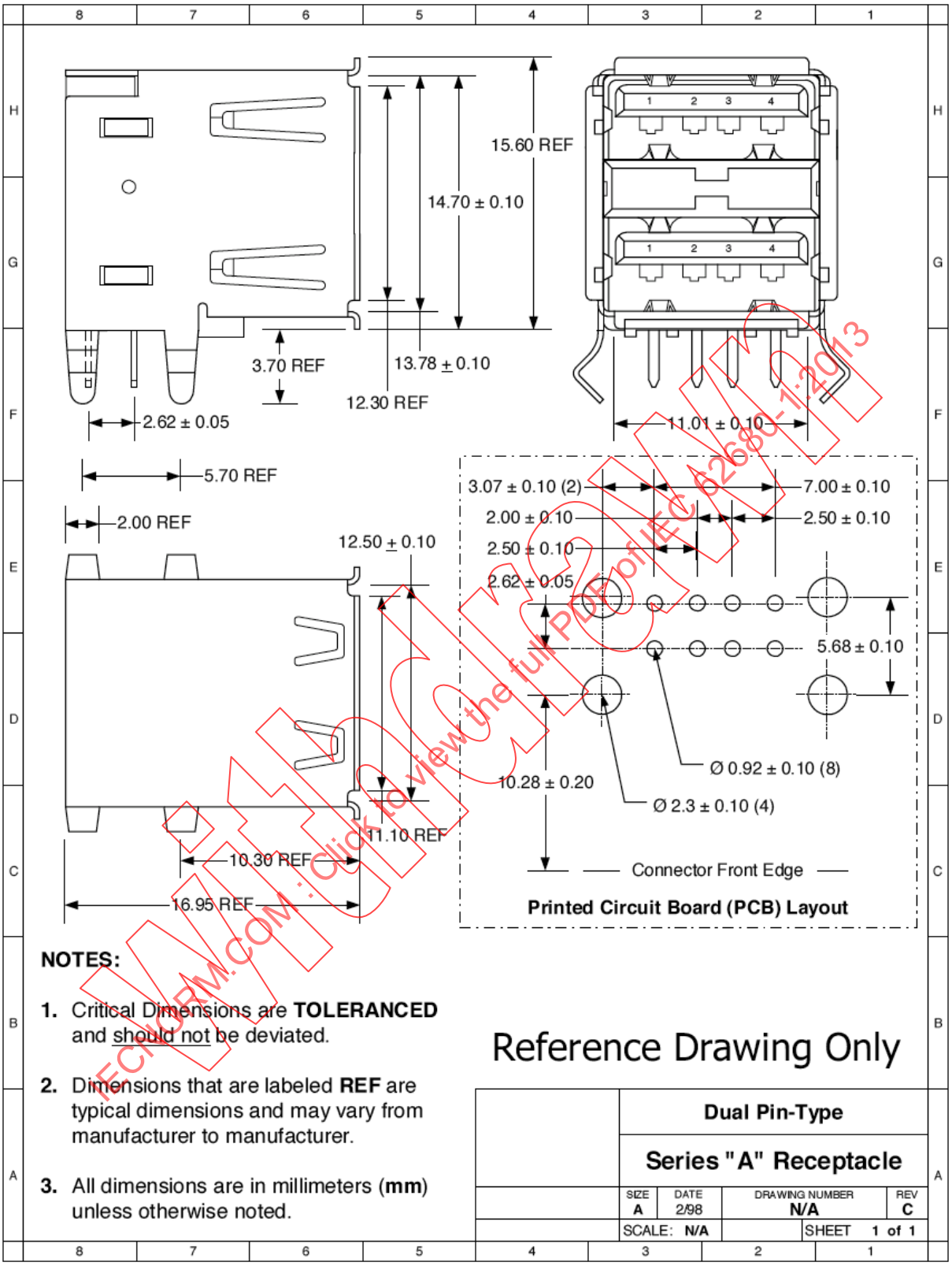
Légende

Anglais	Français
Reference Drawing Only	Schéma de référence uniquement
Thermoplastic Insulator UL 94-V0	Isolant thermoplastique UL 94-V0
1.0 ±0.05 Wide - Selectively plated Contact (4)	1,0 ± 0,05 de largeur - Contact plaque de manière sélective (4)
Printed Circuit Board (PCB) Layout	Disposition de carte de circuit imprimé (PCB)
NOTES:	NOTES:

Anglais	Français
Critical Dimensions are TOLERANCED and should not be deviated.	Les dimensions critiques sont TOLÉRÉES et il convient de ne pas s'en écarter.
Dimensions that are labeled REF are typical dimensions and may vary from manufacturer to manufacturer.	Les dimensions étiquetées RÉF sont les dimensions typiques et elles peuvent varier d'un fabricant à l'autre.
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
Single Pin-Type	Type à broche unique
Series "A" Receptacle	Réceptacle de série "A"
SIZE	TAILLE
DATE	DATE
DRAWING NUMBER N/A	NUMÉRO DE DESSIN N/A
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-12. Réceptacle de série "A" de type à broche unique

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



NOTES:

1. Critical Dimensions are **TOLERANCED** and should not be deviated.
2. Dimensions that are labeled **REF** are typical dimensions and may vary from manufacturer to manufacturer.
3. All dimensions are in millimeters (**mm**) unless otherwise noted.

Reference Drawing Only

	Dual Pin-Type			
	Series "A" Receptacle			
	SIZE A	DATE 2/98	DRAWING NUMBER N/A	REV C
	SCALE: N/A		SHEET 1 of 1	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

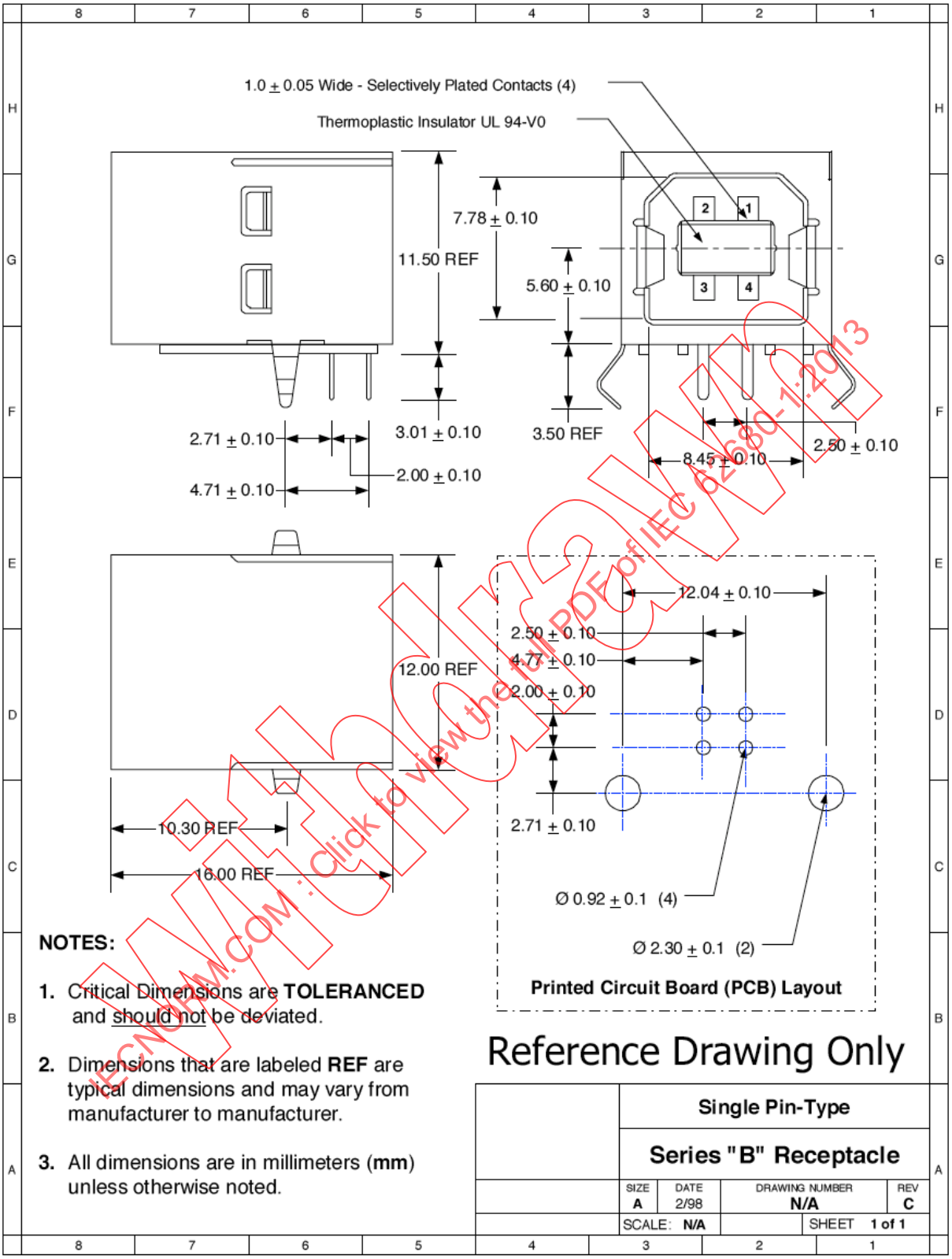
Légende

Anglais	Français
Connector Front Edge	Bord avant de connecteur
Printed Circuit Board (PCB) Layout	Disposition de carte de circuit imprimé (PCB)
NOTES:	NOTES:
Critical Dimensions are TOLERANCED and should not be deviated.	Les dimensions critiques sont TOLÉRÉES et il convient de ne pas s'en écarter.
Dimensions that are labeled REF are	Les dimensions étiquetées RÉF sont les dimensions

Anglais	Français
typical dimensions and may vary from manufacturer to manufacturer.	typiques et elles peuvent varier d'un fabricant à l'autre.
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
Reference Drawing Only	Schéma de référence uniquement
Dual Pin-Type	Type à double broche
Series "A" Receptacle	Réceptacle de série "A"
SIZE	TAILLE
DATE	DATE
DRAWING NUMBER N/A	NUMÉRO DE DESSIN N/A
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-13. Réceptacle de série "A" de type à double broche

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
1.0 ± 0.05 Wide - Selectively plated Contact (4)	1.0 ± 0.05 de largeur - Contact plaque de manière sélective (4)
Thermoplastic insulator UL 94-V0	Isolant thermoplastique UL 94-V0
Printed Circuit Board (PCB) Layout	Disposition de carte de circuit imprimé (PCB)
NOTES:	NOTES:
Critical Dimensions are TOLERANCED	Les dimensions critiques sont TOLÉRÉES et il convient

Anglais	Français
and should not be deviated.	de ne pas s'en écarter.
Dimensions that are labeled REF are typical dimensions and may vary from manufacturer to manufacturer.	Les dimensions étiquetées REF sont les dimensions typiques et elles peuvent varier d'un fabricant à l'autre.
All dimensions are in millimeters (mm) unless otherwise noted.	Toutes les dimensions sont en millimètres (mm), sauf indication contraire.
Reference Drawing Only	Schéma de référence uniquement
Single Pin-Type	Type à broche unique
Series "B" Receptacle	Réceptacle de série "B"
SIZE	TAILLE
DATE	DATE
DRAWING NUMBER N/A	NUMÉRO DE DESSIN N/A
REV	RÉV
SCALE: N/A	ÉCHELLE: N/A
SHEET 1 of 1	PAGE 1 sur 1

Figure 6-14. Réceptacle de série "B" de type à broche unique

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7 Chapitre 7
Domaine électrique

Le présent chapitre décrit la spécification électrique de l'USB. Il contient les spécifications des couches de signalisation, distribution électrique et physique. Cette spécification ne traite pas de la conformité réglementaire. Il est de la responsabilité des concepteurs de produit de s'assurer que leurs dessins sont conformes à toutes les exigences réglementaires applicables.

La spécification USB 2.0 exige que les hubs prennent en charge le mode à faible vitesse. Les appareils USB 2.0 ne doivent pas prendre en charge le mode à haute vitesse. Un émetteur-récepteur à haute vitesse dirigé en amont ne doit pas prendre en charge le mode de signalisation à faible vitesse. Un émetteur-récepteur USB 2.0 dirigé en aval doit prendre en charge les modes à haute, vitesse intégrale et faible vitesse.

Pour garantir un fonctionnement fiable à des débits de données à haute vitesse, la présente spécification nécessite l'utilisation de câbles conformes à toutes les spécifications actuelles relatives aux câbles.

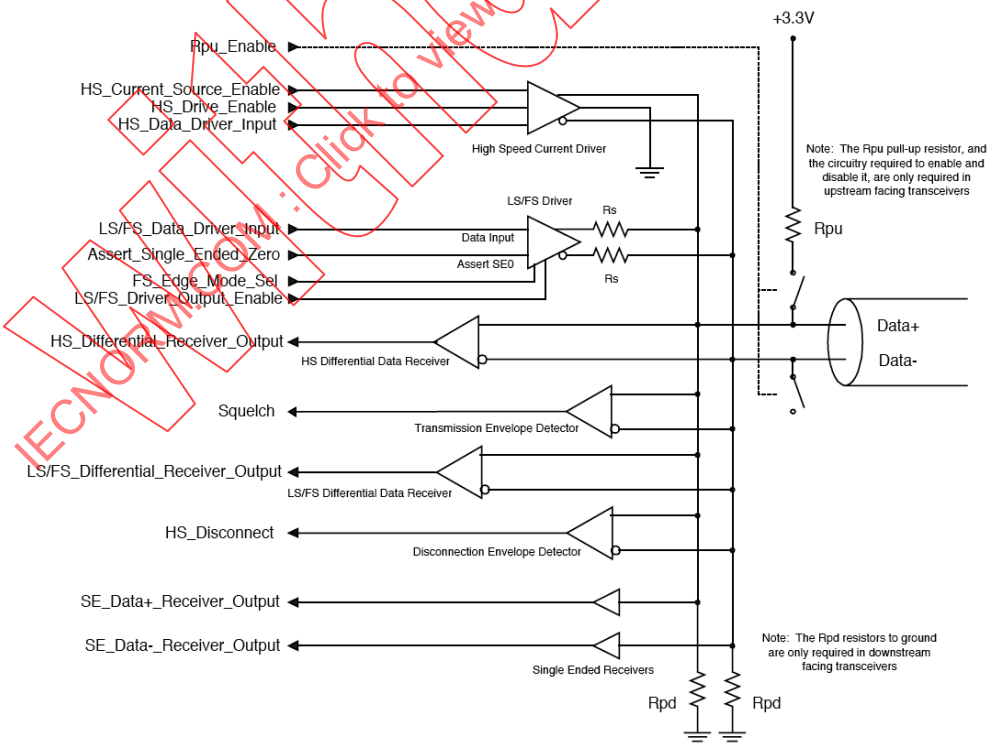
Dans le présent chapitre, il existe de nombreuses références aux caractères J et K ou aux chiffres 1 et 0. Dans chacun de ces cas, le symbole situé le plus à gauche est transmis/reçu en premier tandis que celui situé le plus à droite est transmis/reçu en dernier.

7.1 Signalisation

La spécification de signalisation pour l'USB est décrite dans les sous-sections suivantes.

Vue d'ensemble sur la signalisation à haute vitesse

Une connexion USB à haute vitesse est réalisée par l'intermédiaire d'un câble blindé à paire torsadée qui est conforme à toutes les spécifications actuelles des câbles USB.



Légende

Anglais	Français
High Speed Current Driver	Pilote de courant à haute vitesse
Note: The Rpu pull-up resistor, and the circuitry required to enable and disable it,	Note: la résistance de polarisation et les circuits nécessaires à son activation et à sa désactivation sont

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
are only required in upstream facing transceivers.	uniquement requis dans les émetteurs-récepteurs dirigés en amont.
LS/FS Driver	Pilote LS/FS
Data Input	Entrée de données
Assert SE0	Affirmation SE0
HS Differential Data Receiver	Récepteur différentiel de données HS
Transmission Envelope Detector	Détecteur d'enveloppe de transmission
LS/FS Differential Data Receiver	Récepteur différentiel de données LS/FS
Disconnection Envelope Detector	Détecteur d'enveloppe de déconnexion
Single Ended Receivers	Récepteurs à sortie unique
Note: The Rpd resistors to ground are only required in downstream facing transceivers	Note: les résistances Rpd à la masse sont requises uniquement dans les émetteurs-récepteurs dirigés en aval.

Figure 7-1. Exemple de circuit d'émetteur-récepteur capable de fonctionner à haute vitesse

La Figure 7-1 illustre un exemple de mise en application qui utilise largement des éléments d'émetteur-récepteur USB 1.1 et ajoute les nouveaux éléments requis pour le fonctionnement à haute vitesse.

Le fonctionnement à haute vitesse prend en charge la signalisation à 480 Mb/s. Pour atteindre une signalisation fiable à cette vitesse de débit, le câble est terminé à chaque extrémité par une résistance de chaque fil électrique relié à la masse. La valeur nominale de cette résistance (sur chaque fil électrique) est réglée sur la moitié de l'impédance différentielle du câble ou sur 45 Ω. Cela représente une terminaison différentielle de 90 Ω.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Pour une liaison fonctionnant en mode haute vitesse, l'état de repos à haute vitesse se produit lorsque les émetteurs-récepteurs des deux extrémités de câble présentent des terminaisons à haute vitesse à la masse et lorsqu'aucun émetteur-récepteur ne transmet le courant de signalisation dans les lignes D+ ou D-. Cet état est atteint en utilisant le pilote à faible/vitesse intégrale pour affirmer un zéro à sortie unique et contrôler étroitement le total combiné de l'impédance intrinsèque de sortie de pilote et de la résistance Rs (valeur nominale de 45 Ω). La pratique recommandée est de régler l'impédance intrinsèque de pilote à un niveau aussi bas que possible et de laisser Rs contribuer avec la plus grande partie possible des 45 Ω. Cela mène généralement à la meilleure précision de terminaison avec la charge la moins parasite.

Afin d'effectuer des transmissions en mode haute vitesse, un émetteur-récepteur active une source de courant interne qui dérive de sa tension d'alimentation positive et dirige ce courant dans l'une des deux lignes de données par l'intermédiaire d'un commutateur de direction de courant à haute vitesse. De cette façon, l'émetteur-récepteur génère l'état J ou K à haute vitesse sur le câble.

La commutation dynamique de ce courant dans la ligne D+ ou D- respecte le même schéma de codage des données NRZI utilisé pour le fonctionnement à faible vitesse ou à vitesse intégrale et dans le comportement d'insertion binaire. Pour signaler un J, le courant est dirigé dans la ligne D+ et, pour signaler un K, le courant est dirigé dans la ligne D-. Le champ SYNC et les délimiteurs EOP ont été modifiés pour le mode haute vitesse.

L'amplitude de la source de courant et la valeur des résistances de terminaison sont contrôlées par rapport aux tolérances spécifiées et elles déterminent ensemble les niveaux de tension actuels. La résistance en courant continu provenant de la ligne D+ ou D- menant à la masse de l'appareil doit être égale à 45 Ω ± 10 % lorsqu'elle est mesurée sans charge, et la tension de sortie différentielle mesurée sur les lignes (dans l'état J ou K) doit être égale à ± 400 mV ± 10 % lorsque les lignes D+ et D- sont terminées par des résistances avec une précision de 45 Ω reliées à la masse.

La tension différentielle développée sur les lignes sert à trois objectifs:

- Un récepteur différentiel situé à l'extrémité de réception du câble reçoit le signal différentiel des données.
- Un détecteur différentiel d'enveloppe situé à l'extrémité de réception du câble détermine lorsque la liaison est dans l'état Silence. Un récepteur utilise la détection silencieuse comme indication pour savoir si le signal n'est pas valide au niveau de son connecteur.
- Dans le cas émetteur-récepteur de hub dirigé en aval, un détecteur différentiel d'enveloppe surveille si le signal se trouve dans l'état de haute vitesse au niveau de son connecteur. Un émetteur-récepteur dirigé en aval qui fonctionne en mode haute vitesse est nécessaire pour soumettre cet état à essai en un point donné dans le temps lorsqu'il transmet un paquet SOF, tel que décrit à la Section 7.1.7.3. Il est utilisé pour détecter une déconnexion d'appareil. En l'absence des terminaisons d'extrémité, la tension différentielle nominale est double (comparé à lorsqu'un appareil à haute vitesse est présent) lorsqu'un état J ou K à haute vitesse est maintenu de façon continue pendant une période supérieure au délai de propagation en boucle du câble et des pistes conductrices de carte entre les deux émetteurs-récepteurs.

L'USB 2.0 exige qu'un émetteur-récepteur dirigé en aval doive être capable de fonctionner dans les modes de signalisation à faible, vitesse intégrale et haute vitesse. Un émetteur-récepteur à haute vitesse dirigé en amont ne doit pas fonctionner en mode de signalisation à faible vitesse, mais il doit être capable de fonctionner en mode de signalisation à vitesse intégrale. Par conséquent, une polarisation de 1,5 kΩ sur la ligne D- n'est pas admise pour un appareil à haute vitesse étant donné qu'un émetteur-récepteur à haute vitesse ne doit jamais signaler un fonctionnement à faible vitesse au port du hub auquel il est connecté.

Le Tableau 7-1 décrit les éléments fonctionnels requis d'un émetteur-récepteur à haute vitesse, avec le schéma illustré à la Figure 7-1 en exemple.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 7-1. Description des éléments fonctionnels illustrés dans l'exemple de la Figure 7-1

Elément	Description
Pilote à faible/vitesse intégrale	Le pilote à faible/vitesse intégrale est utilisé pour la transmission à faible vitesse et à vitesse intégrale. Il est nécessaire de satisfaire à toutes les spécifications énoncées pour l'USB 1.1 pour le fonctionnement à faible et à vitesse intégrale, à une seule exception près. L'exception est celle que, dans le cas des émetteurs-récepteurs à haute vitesse, l'impédance de chaque sortie, y compris la contribution de RS, doit être égale à 45 Ω ± 10 %. Les terminaisons de ligne pour le fonctionnement à haute vitesse sont créées lorsque ce pilote relie D+ et D- à la masse. (Cela revient à mettre SE0 en mode vitesse intégrale ou faible vitesse). Les exigences d'impédance de sortie décrites ci-dessus permettent de fournir une terminaison à haute vitesse bien maîtrisée sur chaque ligne de données à la masse. Cela équivaut à une terminaison différentielle de 90 Ω.
Récepteur différentiel à faible/vitesse intégrale	Le récepteur différentiel à faible/vitesse intégrale est utilisé pour la réception des données à faible vitesse et à vitesse intégrale.
Récepteurs à sortie unique	Les récepteurs à sortie unique sont utilisés pour la signalisation à faible et vitesse intégrale.
Pilote de courant à haute vitesse	Le pilote de courant à haute vitesse est utilisé pour la transmission des données à haute vitesse. Une source de courant dérivée d'une alimentation positive est transférée sur la ligne D+ ou D- pour signaler un état J ou K, respectivement. La valeur nominale de la source de courant est égale à 17,78 mA. Lorsque ce courant s'applique à une ligne de données avec une terminaison à la masse de 45 Ω à chaque extrémité, la tension nominale de niveau élevé (VHSOH) est de +400 mV. La tension différentielle nominale à haute vitesse (D+ - D-) s'élève donc à 400 mV pour un état J et à -400 mV pour un état K. La source de courant doit satisfaire aux gabarits d'œil de transmission spécifiés à la Section 7.1.2.2, à commencer par le premier symbole d'un paquet. Un moyen d'y parvenir est de laisser la source de courant fonctionner en continu lorsqu'un émetteur-récepteur fonctionne en mode haute vitesse. Si cette approche est utilisée, le courant peut être dirigé vers la masse du port lorsque l'émetteur-récepteur n'émet pas (la conception donnée en exemple à la Figure 7-1 illustre une ligne de contrôle appelée S_Current_Source_Enable pour rétablir le courant et une autre appelée HS_Drive_Enable pour diriger le courant dans les lignes de données). Le désavantage de cette approche est le courant de repos de 17,78 mA pour chaque émetteur-récepteur activé du système. La conception préférée est la source de courant lorsque l'émetteur-récepteur n'émet pas.
Récepteur différentiel de données à haute vitesse	Le récepteur différentiel de données à haute vitesse est utilisé pour recevoir des données à haute vitesse. Les concepteurs d'émetteurs-récepteurs doivent choisir entre incorporer des récepteurs différents à haute vitesse et à faible/vitesse intégrale, tels qu'illustrés à la Figure 7-1, ou combiner les deux fonctions dans un récepteur unique.
Détecteur d'enveloppe de transmission	Ce détecteur d'enveloppe est utilisé pour indiquer les données invalides lorsque l'amplitude du signal différentiel au niveau des entrées d'un récepteur tombe en dessous du seuil silencieux (VHSSQ). Il doit indiquer Silence lorsque le signal est inférieur à une amplitude différentielle de 100 mV et il doit indiquer que la ligne ne se trouve pas dans l'état Silence lorsque le signal dépasse une amplitude différentielle de 150 mV. Le temps de réponse du détecteur doit être suffisamment rapide pour permettre à un récepteur de détecter la transmission de données, mettre un verrou DLL et détecter la fin du champ SYNC sur des périodes de 12 bits, le nombre minimum de bits SYNC qu'un récepteur est sûr de voir. Ce détecteur d'enveloppe doit incorporer un mécanisme de filtrage qui empêche l'indication du silence pendant les transitions différentielles de données les plus longues permises par les spécifications du modèle en œil du récepteur.
Détecteur d'enveloppe de déconnexion	Ce détecteur d'enveloppe est requis dans les ports dirigés en aval pour détecter l'état Déconnecté à haute vitesse sur la ligne (VHSDSC). La déconnexion doit être indiquée lorsque l'amplitude du signal différentiel au niveau du connecteur du pilote dirigé en aval est égale à ≥625 mV, mais elle ne doit pas être indiquée lorsque l'amplitude du signal est égale à ≤525 mV. La sortie de ce détecteur est échantillonnée à un moment spécifique de la transmission de l'EOP SOP à haute vitesse, tel que décrit à la Section 7.1.7.3.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 7-1. Description des éléments fonctionnels illustrés dans l'exemple de la Figure 7-1 (suite)

Elément	Description
Résistance de polarisation (RPU)	Cette résistance est requise uniquement dans les émetteurs-récepteurs dirigés en amont et utilisée pour indiquer la fonctionnalité de vitesse de signalisation. Un appareil à haute vitesse doit d'abord se connecter comme un appareil à vitesse intégrale et doit passer en haute vitesse tel que décrit dans la présente spécification. Lorsqu'elle fonctionne à haute vitesse, la résistance de 1,5 kΩ doit être déposée du circuit du point de vue électrique. A la Figure 7-1, une ligne de contrôle appelée RPU_Enable est indiquée à cette fin. Le mode de réalisation préféré est de connecter les appareils de connexion correspondants aux lignes D+ et D- de manière à maintenir l'équilibre de la charge parasite des lignes, même si une résistance de polarisation ne doit jamais être utilisée sur la ligne D- d'un émetteur-récepteur à haute vitesse dirigé en amont. Lorsqu'elle est connectée, cette polarisation doit satisfaire à toutes les spécifications énoncées pour le fonctionnement à vitesse intégrale.
Résistances de polarisation (RPD)	Ces résistances sont requises uniquement pour les émetteurs-récepteurs dirigés en aval et doivent se conformer aux mêmes spécifications énoncées pour le fonctionnement à faible et vitesse intégrale.

7.1.1 Caractéristiques du Pilote USB

L'USB utilise un pilote de sortie différentielle pour diriger le signal de données USB sur le câble USB.

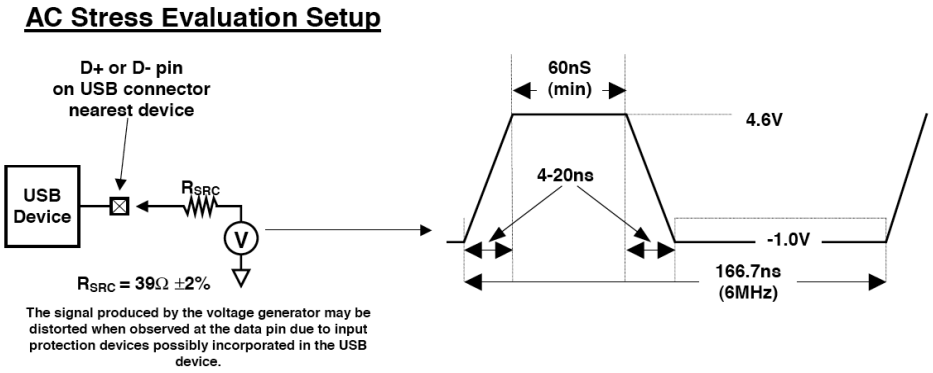
Pour le fonctionnement à faible et vitesse intégrale, les impulsions statiques de sortie du pilote à leur niveau le plus bas doivent être inférieures au V_{OL} (max) de 0,3 V avec une charge de 1,5 kΩ à 3,6 V, et, à son niveau le plus élevé, elles doivent être supérieures au V_{OH} (min) de 2,8 V avec une charge à la masse de 15 kΩ, tel qu'indiqué au Tableau 7-7. Les pilotes à vitesse intégrale ont des exigences encore plus rigoureuses, telles que décrites à la Section 7.1.1.1. Les impulsions de sortie entre les niveaux différentiels élevés et bas doivent être bien équilibrées afin de minimiser la différence de signal. Le contrôle de la vitesse de balayage du pilote est nécessaire pour minimiser l'émission du bruit et la diaphonie. Les sorties du pilote doivent prendre en charge le fonctionnement à trois états pour atteindre le fonctionnement bidirectionnel en semi-duplex.

Les pilotes USB à faible et vitesse intégrale ne doivent jamais générer "de manière intentionnelle" un SE1 sur le bus. SE1 est un état dans lequel les lignes D+ et D- ont une tension supérieure à V_{OSE1} (min), qui a une valeur égale à 0,8 V.

Les pilotes à haute vitesse utilisent des niveaux de signalisation nettement différents, tels que décrits à la Section 7.1.1.3.

Les ports USB doivent être en mesure de supporter une exposition continue aux formes d'onde décrites à la Figure 7-2 dans tous les états de transmission. Ces formes d'onde s'appliquent directement à chaque broche de données USB à partir de la source de tension avec une impédance de sortie de 39 Ω. La tension de circuit ouvert de la source indiquée à la Figure 7-2 repose sur le surdépassement et le sous-dépassement attendu dans le cas le plus défavorable.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
AC Stress Evaluation Setup	Configuration d'évaluation de l'effet c.a.
D+ or D- pin on USB connector nearest device	Broche D+ ou D- sur le connecteur USB de l'appareil le plus proche
USB Device	Appareil USB
The signal produced by the voltage generator may be distorted when observed at the data pin due to input protection devices possibly incorporated in the USB device.	Le signal produit par le générateur de tension peut être distordu lorsqu'il est observé au niveau de la broche de données à cause des appareils de protection d'entrée éventuellement incorporés dans l'appareil USB.

Figure 7-2. Formes d'onde d'entrée maximum pour la signalisation USB

Résistance aux courts-circuits

Un émetteur-récepteur USB doit supporter un court-circuit continu de D+ et/ou D- au niveau du VBUS, du GND, d'une autre ligne de données ou du blindage de câble du connecteur pendant une durée minimum de 24 heures sans subir de dégradation. Il est recommandé que les émetteurs-récepteurs soient conçus pour supporter ces courts-circuits indéfiniment. L'appareil ne doit pas être endommagé dans une situation de court-circuit lorsqu'il transmet 50 % du temps et reçoit 50 % du temps (à toutes les vitesses prises en charge). La phase de transmission consiste en un signal symétrique qui bascule entre la transmission élevée et la transmission faible. Cette exigence doit être satisfaite pour la valeur max de VBUS (5,25 V).

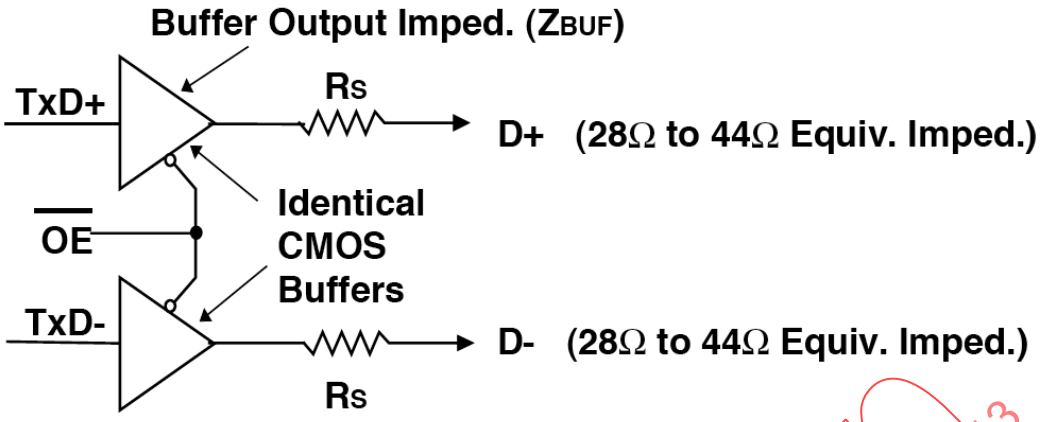
Il est recommandé que ces effets c.a. et de court-circuit soient utilisés comme critères de qualification pour l'évaluation de chaque appareil par rapport à la fiabilité à long terme.

7.1.1.1 Caractéristiques de pilote à vitesse intégrale (12 Mb/s)

Une connexion USB à vitesse intégrale se réalise par l'intermédiaire d'un câble blindé à paire torsadée avec une impédance différentielle des caractéristiques (Z_0) de 90 Ω ±15 %, une impédance en mode commun (Z_{CM}) de 30 Ω ±30 % et un délai unilatéral maximum (T_{FSCBL}) de 26 ns. Lorsque le pilote à vitesse intégrale ne fait pas partie d'un émetteur-récepteur à haute vitesse, l'impédance de chaque pilote (Z_{DRV}) doit être située entre 28 Ω et 44 Ω, c'est-à-dire, dans la zone grisée de la Figure 7-4. Lorsque le pilote à vitesse intégrale fait partie d'un émetteur-récepteur à haute vitesse, l'impédance de chaque pilote (Z_{HSDRV}) doit être située entre 40,5 Ω et 49,5 Ω, c'est-à-dire, dans la zone grisée de la Figure 7-5.

Pour une mise en application CMOS, l'impédance du pilote est généralement réalisée avec un pilote CMOS dont l'impédance est nettement inférieure à celle d'une résistance de gamme discrète réalisant l'équilibre, tel qu'illustré à la Figure 7-3. La résistance de série RS est comprise dans l'exigence de l'impédance de mémoire tampon illustrée à la Figure 7-4 et à la Figure 7-5. Le reste du chapitre fait référence à une mémoire tampon qui est supposée avoir l'impédance de série, sauf indication contraire.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Buffer Output Imped. (Z_{BUF})	Impéd. de sortie de mémoire tampon (Z_{BUF})
(28Ω to 44Ω Equiv. imped.)	(impéd. équiv. allant de 28 Ω à 44 Ω)
Identical CMOS buffers	Mémoires tampons CMOS identiques

Figure 7-3. Exemple de circuit de pilote CMOS à vitesse intégrale (incapable de fonctionner à haute vitesse)

Mémoires tampons à vitesse intégrale dans les émetteurs-récepteurs qui ne peuvent pas fonctionner à haute vitesse

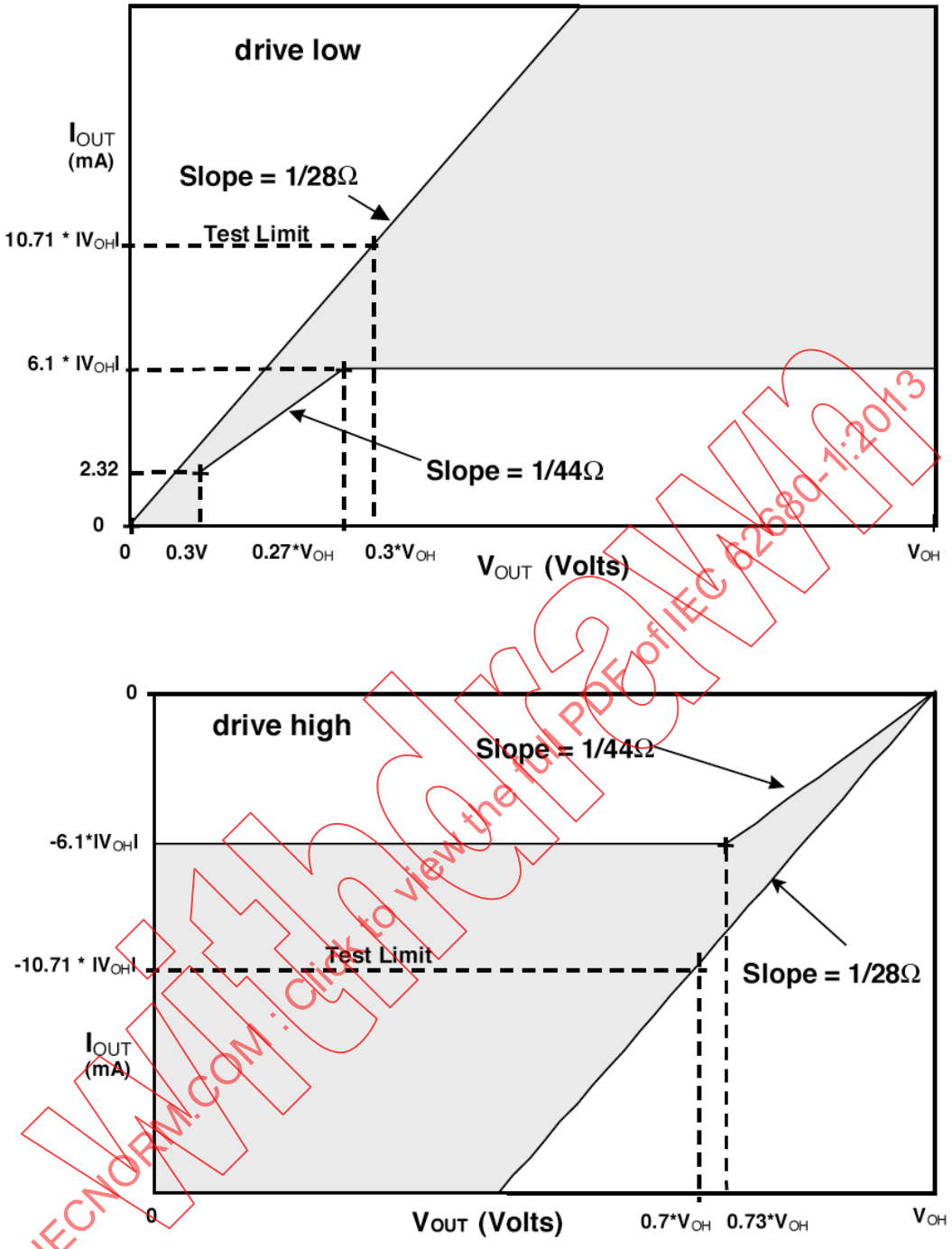
L'impédance de mémoire tampon doit être mesurée pour le haut rendement ainsi que pour le faible rendement. La Figure 7-4 illustre les caractéristiques V/I composites des pilotes à vitesse intégrale équipés d'une résistance d'amortissement de série (RS). Les caractéristiques sont normalisées aux impulsions de sortie à l'état stable et à vide du pilote. Les caractéristiques normalisées du pilote se calculent en divisant les tensions et les courants mesurés par les impulsions actuelles du pilote soumis à essai. La courbe V/I normalisée du pilote doit se trouver entièrement à l'intérieur de la région grisée. La région V/I est limitée au-dessus par l'impédance minimum du pilote et en dessous par l'impédance maximum du pilote. La région de rendement minimum est traversée par une région de courant constant de |6,1VOH| à faible rendement et de -|6,1VOH| mA à haut rendement. Dans le cas spécial d'un pilote à vitesse intégrale qui fonctionne à faible rendement et qui fait partie d'un émetteur-récepteur à haute vitesse, la région à faible rendement est traversée par une région de courant constant de 22,0 mA. C'est le niveau de courant de commande minimum nécessaire pour s'assurer que la forme d'onde au niveau du récepteur traverse le niveau de commutation opposé à extrémité unique de la première réflexion.

Durant l'essai, il n'est pas nécessaire que le courant qui entre ou sort de l'appareil dépasse ± 10,71*VOH mA, que la tension appliquée à D+/D- dépasse 0,3*VOH en cas de basse génération et soit inférieure à 0,7*VOH en cas de génération élevée.

Mémoires tampons dans les émetteurs-récepteurs à vitesse intégrale

La Figure 7-5 illustre les caractéristiques V/I d'une mémoire tampon à vitesse intégrale qui fait partie d'un émetteur-récepteur à haute vitesse. L'impédance de sortie, ZHSDRV (y compris la contribution de RS), doit être comprise entre 40,5 Ω et 49,5 Ω. De plus, la tension de sortie doit se trouver dans un intervalle de 10 mV de la masse lorsqu'aucun courant n'entre ni ne sort de la broche (VHSTERM).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

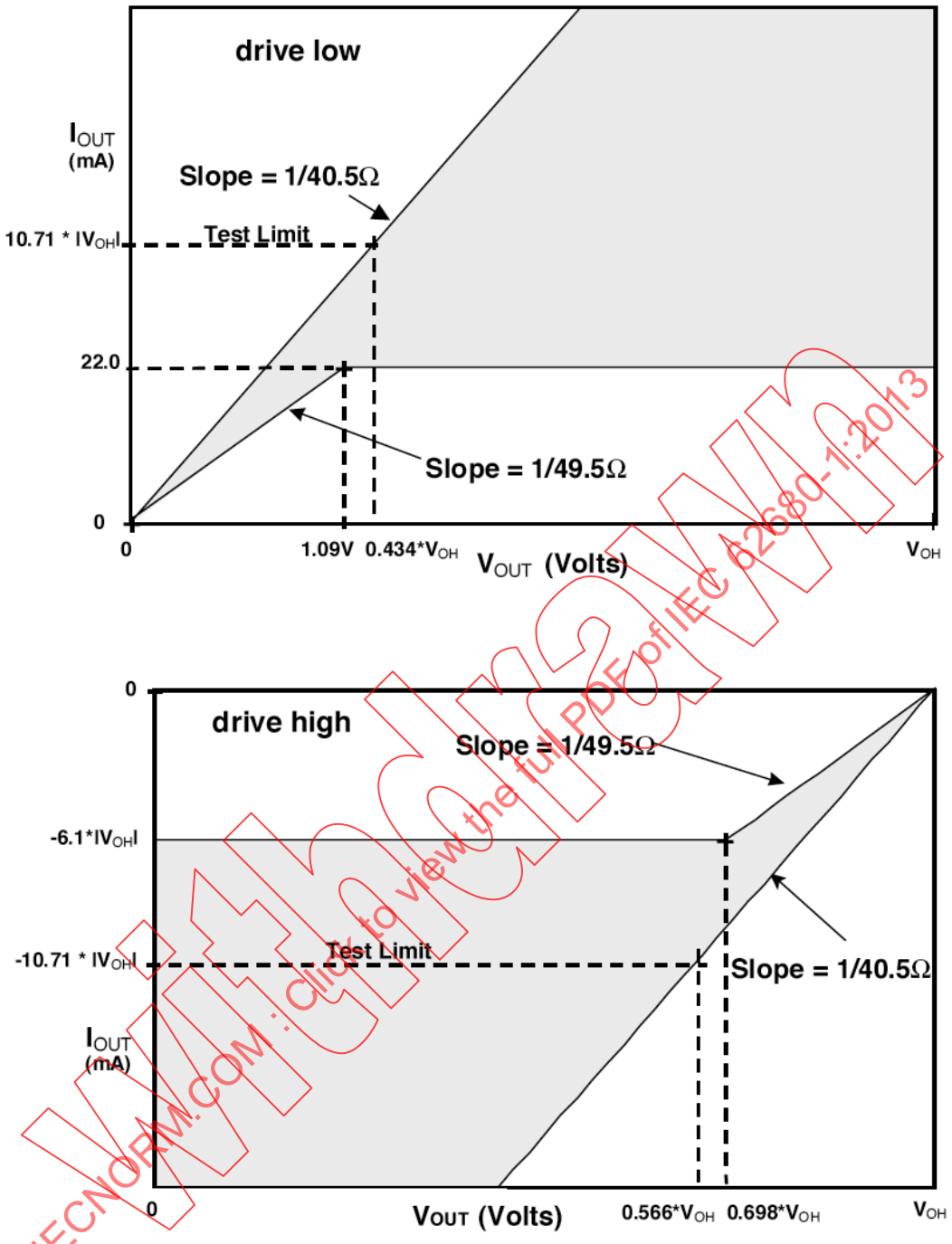


IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Drive low	Faible excitation
Slope	Pente
Test Limit	Limite d'essai
Drive high	Forte excitation

Figure 7-4. Caractéristiques V/I de mémoire tampon à vitesse intégrale



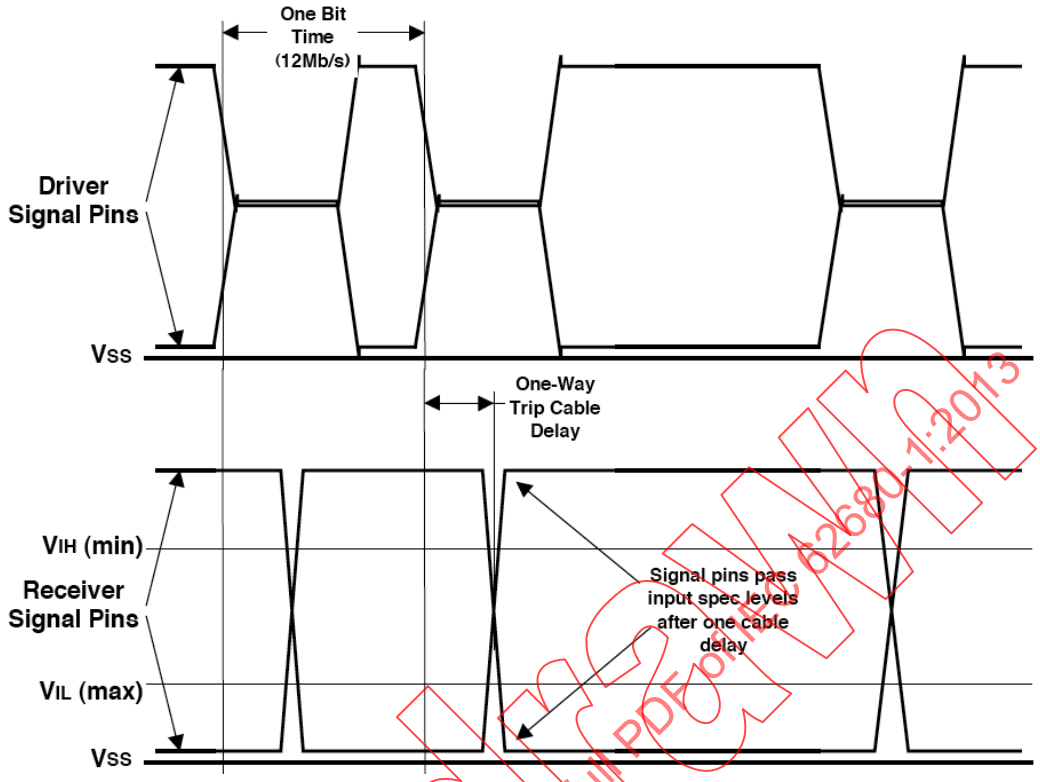
Légende

Anglais	Français
Drive low	Faible excitation
Slope	Pente
Test Limit	Limite d'essai
Drive high	Forte excitation

Figure 7-5. Caractéristiques V/I de mémoire tampon à vitesse intégrale d'émetteur-récepteur à haute vitesse

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

La Figure 7-6 illustre les formes d'onde des signaux du pilote à vitesse intégrale.



Légende

Anglais	Français
One Bit Time	Temps à un bit
Driver Signal Pins	Broches de signalisation de pilote
One-Way Trip Cable Delay	Délai de câble à une seule direction
Receiver Signal Pins	Broches de signalisation de récepteur
Signal pins pass input spec levels after one cable delay	Niveaux de spéc. d'entrée de passage des broches de signalisation après un délai de câble

Figure 7-6. Formes d'onde des signaux à vitesse intégrale

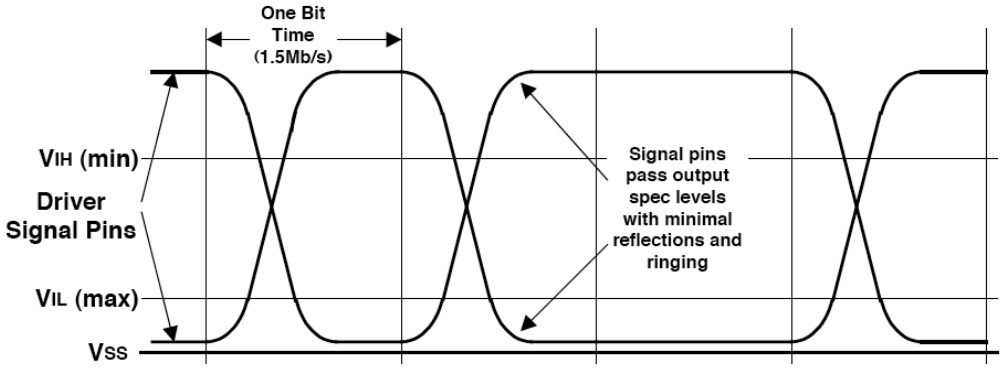
7.1.1.2 Caractéristiques de pilote à faible vitesse (1,5 Mb/s)

Un appareil à faible vitesse doit être équipé d'un câble captif avec le connecteur de série A sur l'extrémité de la fiche. La combinaison du câble et de l'appareil doit avoir une capacitance à extrémité unique qui ne doit être ni inférieure à 200 pF, ni supérieure à 450 pF sur les lignes D+ ou D-.

Le délai de propagation (TLSCBL) d'un câble à faible vitesse doit être inférieur à 18 ns. Cela garantit que la réflexion se produit dans la première moitié de la montée/descente du signal, ce qui permet d'obtenir une approximation du câble par une capacitance globale.

La Figure 7-7 illustre les formes d'onde des signaux du pilote à faible vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
One Bit Time	Temps à un bit
Driver Signal Pins	Broches de signalisation de pilote
Signal pins pass output spec levels with minimal reflections and ringing	Niveaux de spéc. de sortie de passage des broches de signalisation avec réflexions et oscillations minimales

Figure 7-7. Formes d'onde des signaux de pilote à faible vitesse

7.1.1.3 Caractéristiques de pilote à haute vitesse (480 Mb/s)

Une connexion USB à haute vitesse se réalise par l'intermédiaire d'un câble blindé à paire torsadée avec une impédance différentielle des caractéristiques (Z_0) de 90 Ω ± 15 %, une impédance en mode commun (Z_{CM}) de 30 Ω ± 30 % et un délai unilatéral maximum (T_{FSCBL}) de 26 ns. Il convient que les pistes de carte de circuit D+ et D- qui passent entre un émetteur-récepteur et son connecteur associé aient également une impédance différentielle nominale de 90 Ω, et ensemble ils peuvent avoir 4 ns de délai supplémentaire entre les émetteurs-récepteurs. (voir la Section 7.1.6 pour les détails concernant les spécifications d'impédance des cartes et des émetteurs-récepteurs). L'impédance de sortie différentielle d'un appareil à haute vitesse doit être égale à 90 Ω ± 10 %. Lorsque les lignes D+ ou D- sont fortement excitées, V_{HSOH} (la tension de sortie de niveau élevé en mode haute vitesse transmise sur une ligne de données avec une charge de précision de 45 Ω vers le GND) doit être égale à 400 mV ± 10 %. Si une ligne n'est pas excitée, soit parce que l'émetteur-récepteur ne transmet pas, soit parce que la ligne opposée est fortement excitée, V_{HSOL} (la tension de sortie de bas niveau en mode haute vitesse transmise sur une ligne de données avec une charge de 45 Ω au GND) doit être égale à 0 V ± 10 mV.

Note: sauf indication contraire, toutes les mesures de tension doivent être effectuées conformément au circuit de mise à la masse local.

Note: la présente spécification exige qu'un émetteur-récepteur à haute vitesse fonctionnant en mode vitesse intégrale ou faible vitesse doive avoir une impédance (Z_{HSDRV}) de 45 Ω ± 10 %. Il est recommandé que les impédances d'excitation se trouvent dans un intervalle de 5 Ω dans un émetteur-récepteur. Pour les émetteurs-récepteurs dirigés en amont qui ne prennent pas en charge le mode haute vitesse, l'impédance de sortie d'excitation (Z_{DRV}) doit être comprise sur une plage allant de 28 Ω à 44 Ω.

Sur les ports dirigés en aval, les résistances R_{PD} (15 kΩ ± 5 %) doivent connecter D+ et D- à la masse.

Lorsqu'un émetteur-récepteur à haute vitesse passe en mode haute vitesse, l'état de repos à haute vitesse est atteint en entraînant SE0 avec les pilotes à faible/vitesse intégrale à chaque extrémité de la liaison (afin de fournir les terminaisons requises) et en déconnectant la résistance de polarisation D+ de l'émetteur-récepteur dirigé en amont.

Dans le mode de réalisation préféré, un émetteur-récepteur active son pilote de courant uniquement lors de la transmission de signaux à haute vitesse. Cela représente un défi de conception potentiel. Toutefois, étant donné que l'amplitude de signal et les spécifications de synchronisation doivent être satisfaites même sur le premier symbole dans le cadre d'un

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

paquet. Comme alternative moins efficace, un émetteur-récepteur peut provoquer l'activation continue de sa source de courant à haute vitesse en mode haute vitesse. Lorsque l'émetteur-récepteur ne transmet pas, le courant peut être dirigé dans la masse de l'appareil au lieu de traverser le commutateur de direction de courant utilisé pour la signalisation des données. Dans le circuit exemplifié, la direction du courant vers la masse est accomplie en attribuant une valeur réduite à HS_Drive_Enable.

Dans les implantations CMOS, l'impédance du pilote est typiquement réalisée par la combinaison de l'impédance de sortie intrinsèque du pilote et RS. Pour contrôler de manière optimale ZHSDRV et minimiser les parasites, il est préférable que l'impédance du pilote soit minimisée (en dessous de 5 Ω) et il convient que le composant RS apporte le solde de 45 Ω.

Lorsqu'un émetteur-récepteur, fonctionnant en mode à grande vitesse transmet, le courant de transmission est dirigé dans l'une ou l'autre des lignes de données D+ ou D-. Un J est affirmé en dirigeant le courant vers la ligne D+, un K est affirmé en le dirigeant vers la ligne D-.

Lorsque chacune des lignes de données se termine par une résistance de 45 Ω à la masse de l'appareil, la résistance de charge effective est de 22,5 Ω. Par conséquent, la ligne dans laquelle le courant de fonctionnement est dirigé passe à 17,78 ma * 22,5 Ω ou 400 mV (nominal). L'autre ligne demeure à la tension de masse de l'appareil. Lorsque le courant est dirigé vers la ligne opposée, ces tensions sont inversées.

7.1.2 Montée et chute du signal de données, diagrammes en œil

Les sections suivantes spécifient les temps de montée et de chute du signal de données pour une signalisation à vitesse intégrale et à basse vitesse, ainsi que le temps de montée et les diagrammes en œil pour la signalisation à grande vitesse.

7.1.2.1 Montée et chute du signal de données à basse vitesse et à vitesse intégrale

En ce qui concerne la basse vitesse et la vitesse intégrale, les temps de montée et de chute de sortie sont mesurés entre 10 % et 90 % du signal (Figure 7-8). Les exigences en matière de temps de montée et de chute s'appliquent aux transitions différentielles, ainsi qu'aux transitions entre la signalisation différentielle et à extrémité unique.

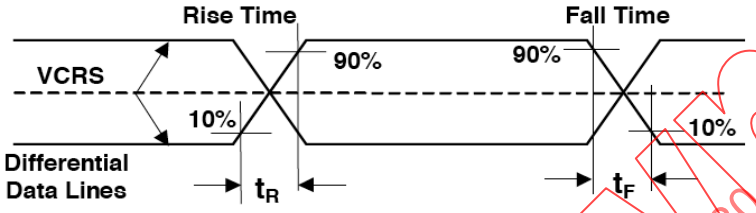
Les temps de montée et de chute pour les mémoires tampons à vitesse intégrale sont mesurés avec la charge indiquée à la Figure 7-9. Les temps de montée et de chute doivent se situer entre 4 ns et 20 ns et maintenus à ± 10 %, afin de minimiser les émissions RFI et la dérive du signal. Les transitions doivent être monotones.

Les temps de montée et de chute pour les mémoires tampons à basse vitesse sont mesurés avec la charge indiquée à la Figure 7-10. La charge capacitive indiquée à la Figure 7-10 est représentative de la charge la plus défavorable autorisée par la spécification. Un émetteur-récepteur orienté vers l'aval d'une capacité d'entrée/sortie de 150 pF est autorisé (CIND). Un appareil à basse vitesse (câble inclus) peut avoir une capacité aussi infime que 200 pF et aussi importante que 450 pF. Cela offre une gamme comprise entre 200 pF et 600 pF de charge capacitive qu'une mémoire tampon basse vitesse orientée en aval peut rencontrer. Les mémoires tampons orientées en amont sur les appareils basse vitesse doivent être conçues afin de pouvoir transmettre la capacité du câble relié plus 150 pF supplémentaires. Si une mémoire tampon basse vitesse est conçue pour une application où l'on sait que la capacité de charge rentre dans une gamme différente, la charge d'essai peut être réglée pour correspondre à l'application réelle. Les mémoires tampons basse vitesse présentes sur des hôtes et hubs reliés aux réceptacles USB doivent être conçues pour la gamme comprise entre 200 pF et 600 pF. Le temps de montée et de chute doit se situer entre 75 ns et 300 ns pour chaque charge d'essai capacitive équilibrée. Dans tous les cas, les bords doivent être maintenus à ± 20 %, afin de minimiser les émissions RFI et la dérive du signal. Les transitions doivent être monotones.

Pour la signalisation à vitesse intégrale et à basse vitesse, la tension croisée (V_{CRS}) doit se situer entre 1,3 V et 2,0 V.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

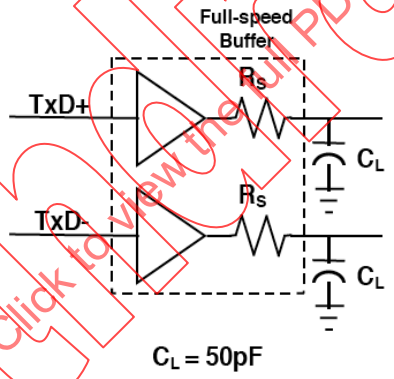
Pour la basse vitesse et la vitesse intégrale, cette spécification n'exige aucune variation concordante du signal qui corresponde à un degré plus important que celui décrit ci-dessus. Toutefois, lors de la signalisation, il convient de préférence que la tension moyenne sur les lignes D+ et D- soit constante. Cela signifie qu'il convient que l'amplitude de la variation du signal sur les deux lignes D+ et D- soit la même; il convient que les transitions en cours élevées et basses débutent au même moment et changent au même taux; et il convient que la tension croisée soit similaire lors d'une commutation vers J ou K. Les déviations en matière de concordance du signal résultent en un bruit de mode commun qui rayonne et affecte la capacité des appareils et systèmes à passer les essais mandatés par les agences gouvernementales.



Légende

Anglais	Français
Rise Time	Temps de montée
Fall Time	Temps de chute
Differential Data Lines	Lignes de données différentielles

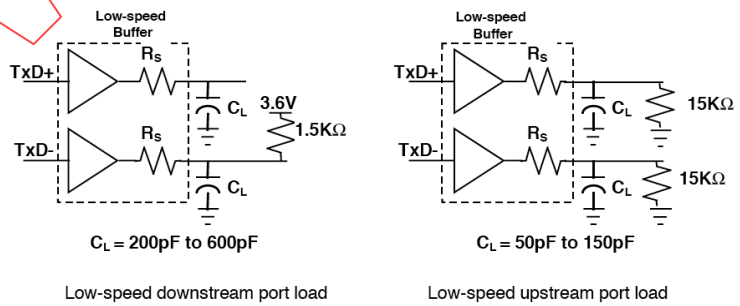
Figure 7-8. Temps de montée et de chute pour une signalisation à grande vitesse



Légende

Anglais	Français
Full-speed Buffer	Mémoire tampon à vitesse intégrale

Figure 7-9. Charge à vitesse intégrale



Légende

Anglais	Français
Low-speed Buffer	Mémoire tampon à basse vitesse
Low-speed downstream port load	Charge du port en aval à basse vitesse
Low-speed upstream port load	Charge du port en amont à basse vitesse

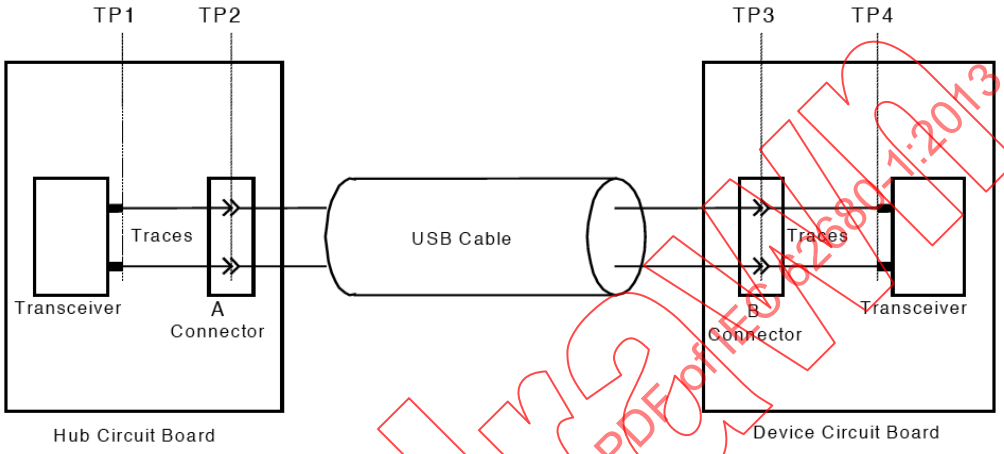
Figure 7-10. Charges du port à basse vitesse

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Note: Le CL pour la charge du port à basse vitesse représente uniquement la plage de chargement qui peut être ajoutée lorsque l'appareil à basse vitesse est relié à un hub. La mémoire tampon à basse vitesse doit être conçue pour transférer la charge de son câble relié plus le CL. Une conception de mémoire tampon à basse vitesse qui peut transporter la charge d'essai en aval doit être en mesure de véhiculer chacune des charges en amont légitimes.

7.1.2.2 Diagrammes en œil et temps de montée et de chute du signal à grande vitesse

Les spécifications suivantes s'appliquent à la signalisation en mode à grande vitesse. Tous les bits, notamment le premier bit et le dernier bit d'un paquet doivent satisfaire aux exigences du diagramme en œil suivant, en matière de synchronisation et d'amplitude.



Légende

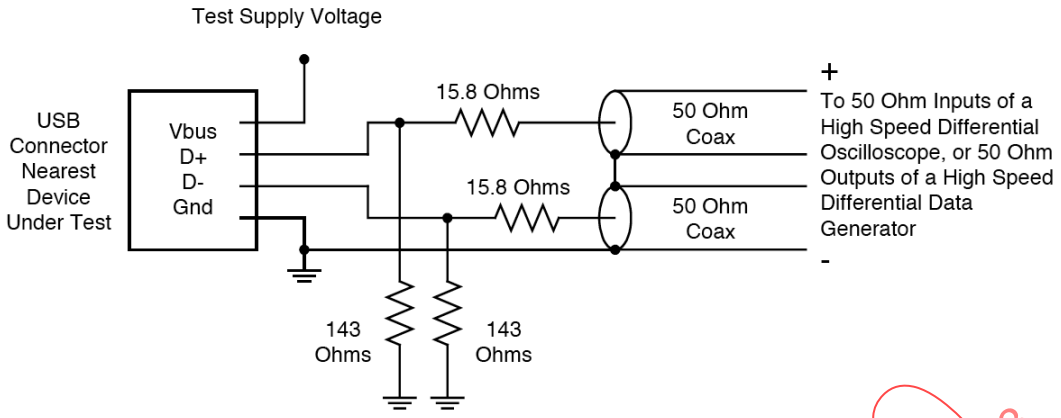
Anglais	Français
Traces	Traces
Transceiver	Emetteur-récepteur
Connector	Connecteur
USB Cable	Câble USB
Hub Circuit Board	Circuit imprimé de hub
Device Circuit Board	Circuit imprimé d'appareil

Figure 7-11. Plans de mesures

La Figure 7-11 définit quatre plans d'essai qui sont référencés dans cette section. TP1 et TP4 correspondent aux points où les contacts IC de l'émetteur-récepteur sont brasés aux circuits imprimés du hub et de l'appareil, respectivement. TP2 se trouve au niveau des contacts branchés du connecteur A et TP3 figure au niveau des contacts branchés du connecteur B (ou, en cas d'un câble captif, à l'endroit où le câble est relié au circuit imprimé). Les modèles de diagrammes en œil différentiels suivants spécifient les exigences de la forme d'onde de transmission et de la sensibilité de réception au niveau d'une variété de points et sous des conditions variées.

Lors de l'essai des émetteurs et récepteurs à grande vitesse, les mesures sont prises avec le montage d'essai de l'émetteur/récepteur représenté à la Figure 7-12. Dans les deux cas, la fixation est reliée au connecteur USB le plus proche de l'émetteur-récepteur soumis à l'essai.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Test Supply Voltage	Tension d'alimentation d'essai
USB Connector Nearest Device Under Test	Connecteur USB de l'appareil le plus proche soumis à essai
To 50 Ohm Inputs of a High Speed Differential Oscilloscope, or 50 Ohm Outputs of a High Speed Differential Data Generator	Entrées de 50 Ohms d'un oscilloscope différentiel à haute vitesse ou sorties de 50 Ohms d'un générateur de données différentiel à haute vitesse
50 Ohm Coax	Câble coaxial de 50 Ohms

Figure 7-12. Montage d'essai de l'émetteur/récepteur

Note: Lors de l'essai du port d'un appareil orienté vers l'aval, VBUS doit être fourni à partir du moment où l'appareil est placé dans le mode d'essai approprié et ce jusqu'à ce que l'essai soit terminé. Cette exigence nécessite certainement une fonctionnalité de commutation supplémentaire dans le montage de l'essai (par exemple, la commutation des lignes D+ et D- entre le contrôleur d'hôte et l'instrument d'essai). De tels ajouts doivent avoir un impact minimal sur les résultats de mesure à haute fréquence.

Les diagrammes en œil de transmission spécifient les limites minimales et maximales, ainsi que les limites sur la gigue temporelle, au sein de laquelle un pilote doit transporter les signaux au niveau de chacun des plans d'essai spécifiés. Les diagrammes en œil de transmission spécifient les limites minimales et maximales, ainsi que les limites sur la gigue temporelle, au sein de laquelle un récepteur doit récupérer les données.

Une conformité avec les modèles 1, 2, 3 et 4 est requise pour les hubs et appareils USB 2.0:

Modèle 1: les exigences en matière de forme d'onde de transmission pour un hub mesuré au niveau de TP2 et pour un appareil (sans câble captif) mesuré au niveau de TP3

Modèle 2: les exigences en matière de forme d'onde de transmission pour un appareil (avec un câble captif) mesuré au niveau de TP2

Modèle 3: les exigences en matière de sensibilité du récepteur pour un appareil (avec un câble captif), lorsque le signal est appliqué au niveau de TP2

Modèle 4: les exigences en matière de sensibilité du récepteur pour un appareil (sans câble captif), lorsque le signal est appliqué au niveau de TP3, et pour un hub lorsqu'un signal est appliqué au niveau de TP2.

Les modèles 5 et 6 correspondent aux directives recommandées pour les concepteurs:

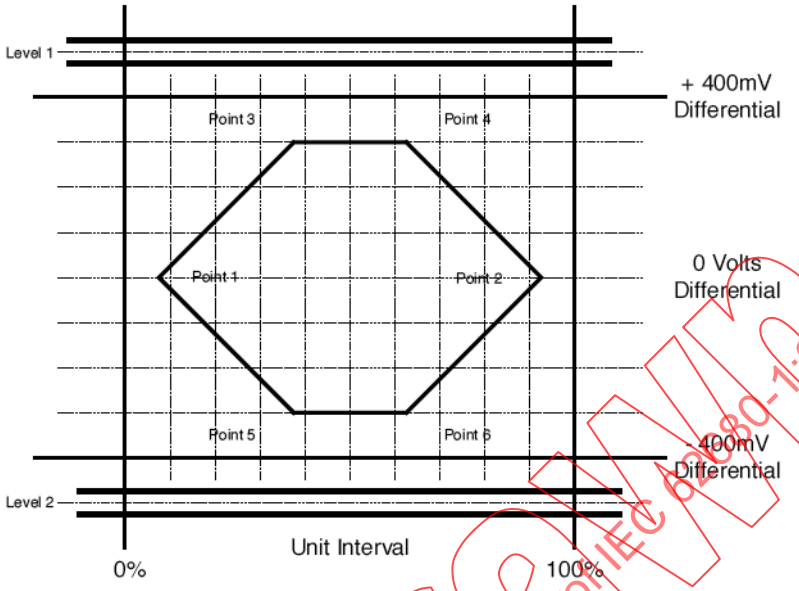
Modèle 5: les exigences en matière de forme d'onde de transmission pour un émetteur-récepteur, qui fait office de hub, mesuré au niveau de TP1 et pour un émetteur-récepteur qui fait office d'appareil, mesuré au niveau de TP4.

Modèle 6: les exigences en matière de sensibilité du récepteur pour un émetteur-récepteur qui fait office d'appareil, lorsque le signal est appliqué au niveau de TP4 et pour un émetteur-récepteur, qui fait office de hub, lorsque le signal est appliqué au niveau de TP1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Modèle 1

La Figure 7-13 illustre les exigences en matière de forme d'onde de transmission pour un hub mesuré au niveau de TP2 et pour un appareil (sans câble captif) mesuré au niveau de TP3.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition, -475 in all others	N/A
Point 1	0 V	7.5% UI
Point 2	0 V	92.5% UI
Point 3	300 mV	37.5% UI
Point 4	300 mV	62.5% UI
Point 5	-300 mV	37.5% UI
Point 6	-300 mV	62.5% UI

Légende

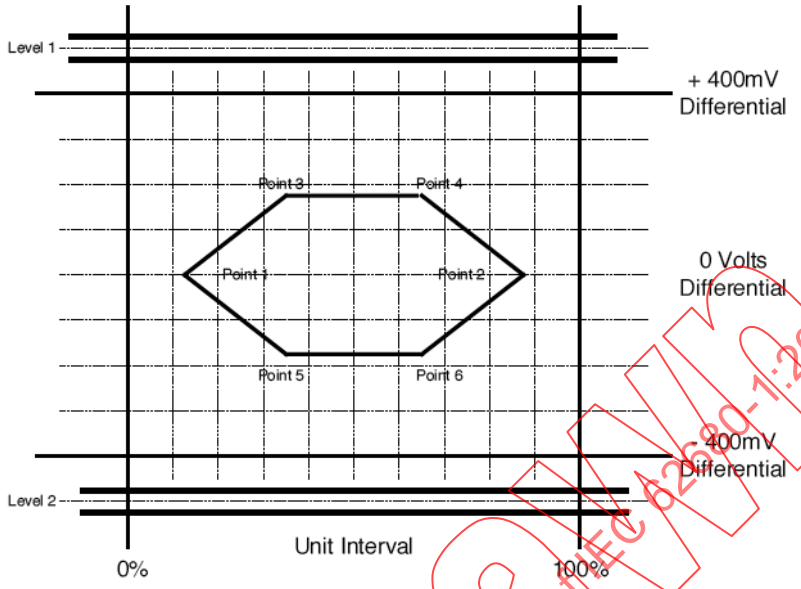
Anglais	Français
+400mV Differential	+400 mV différentiel
0 Volts Differential	0 Volt différentiel
-400mV Differential	-400 mV différentiel
Unit Interval	Intervalle d'unité
Voltage Level (D+ D-)	Niveau de tension (D+ D-)
Time (% of Unit Interval)	Temps (% d'intervalle d'unité)
Level 1	Niveau 1
525 mV in UI following a transition, 475 mV in all others	525 mV dans UI après une transition, 475 mV dans les autres
N/A	N/A
Level 2	Niveau 2
-525 mV in UI following a transition, -475 mV in all others	-525 mV dans UI après une transition, -475 mV dans les autres

Figure 7-13. Modèle 1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Modèle 2

La Figure 7-14 illustre les exigences en matière de forme d'onde de transmission pour un appareil (avec un câble captif) mesuré au niveau de TP2.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition, -475 in all others	N/A
Point 1	0 V	12.5% UI
Point 2	0 V	87.5% UI
Point 3	175 mV	35% UI
Point 4	175 mV	65% UI
Point 5	-175 mV	35% UI
Point 6	-175 mV	65% UI

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

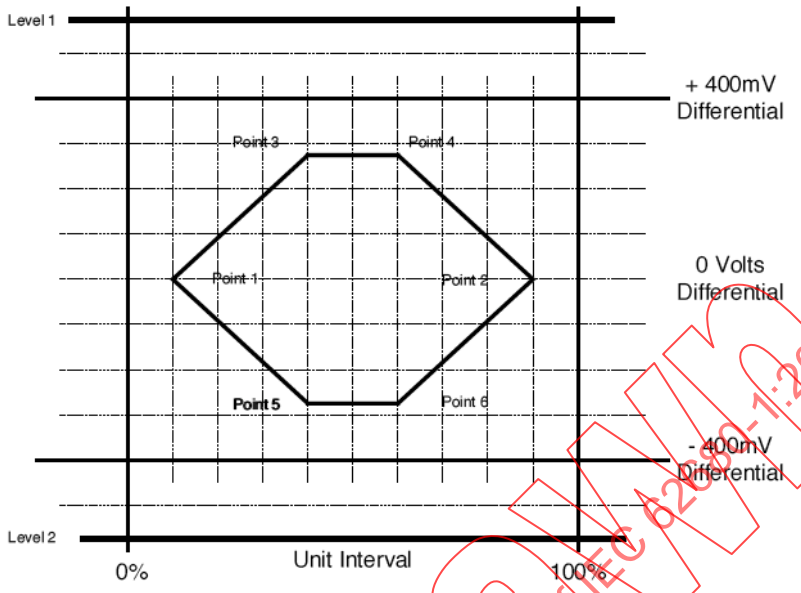
Légende

Anglais	Français
+400mV Differential	+400 mV différentiel
0 Volts Differential	0 Volt différentiel
-400mV Differential	-400 mV différentiel
Unit Interval	Intervalle d'unité
Voltage Level (D+ D-)	Niveau de tension (D+ D-)
Time (% of Unit Interval)	Temps (% d'intervalle d'unité)
Level 1	Niveau 1
525 mV in UI following a transition, 475 mV in all others	525 mV dans UI après une transition, 475 mV dans les autres
N/A	N/A
Level 2	Niveau 2
-525 mV in UI following a transition, -475 mV in all others	-525 mV dans UI après une transition, -475 mV dans les autres

Figure 7-14. Modèle 2

Modèle 3

La Figure 7-15 illustre les exigences en matière de sensibilité du récepteur pour un appareil (avec un câble captif), lorsqu'un signal est appliqué au niveau de TP2.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	575 mV	N/A
Level 2	-575 mV	N/A
Point 1	0 V	10% UI
Point 2	0 V	90% UI
Point 3	275 mV	40% UI
Point 4	275 mV	60% UI
Point 5	-275 mV	40% UI
Point 6	-275 mV	60% UI

Légende

Anglais	Français
+400mV Differential	+400 mV différentiel
0 Volts Differential	0 Volt différentiel
-400mV Differential	-400 mV différentiel
Unit Interval	Intervalle d'unité
Voltage Level (D+ D-)	Niveau de tension (D+ D-)
Time (% of Unit Interval)	Temps (% d'intervalle d'unité)
Level 1	Niveau 1
N/A	N/A
Level 2	Niveau 2

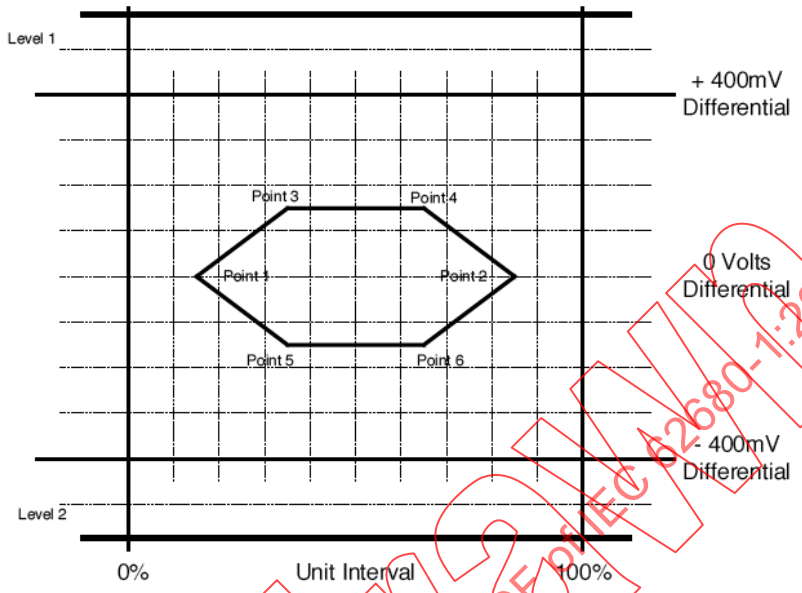
Figure 7-15. Modèle 3

Note: Cet œil a pour objet la spécification des exigences en matière de sensibilité du récepteur des données différentielles. Les niveaux 1 et 2 figurent à l'extérieur des valeurs de seuil de déconnexion, mais une déconnexion est détectée à la source (après un minimum de 32 temps bit sans aucune transition) et pas au niveau du récepteur cible.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Modèle 4

La Figure 7-16 illustre les exigences en matière de sensibilité du récepteur pour un appareil (sans câble captif), lorsqu'un signal est appliqué au niveau de TP3 et pour un hub lorsqu'un signal est appliqué au niveau de TP2.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	575 mV	N/A
Level 2	-575 mV	N/A
Point 1	0 V	15% UI
Point 2	0 V	85% UI
Point 3	150 mV	35% UI
Point 4	150 mV	65% UI
Point 5	-150 mV	35% UI
Point 6	-150 mV	65% UI

Légende

Anglais	Français
+400mV Differential	+400 mV différentiel
0 Volts Differential	0 Volt différentiel
-400mV Differential	-400 mV différentiel
Unit Interval	Intervalle d'unité
Voltage Level (D+ D-)	Niveau de tension (D+ D-)
Time (% of Unit Interval)	Temps (% d'intervalle d'unité)
Level 1	Niveau 1
N/A	N/A
Level 2	Niveau 2

Figure 7-16. Modèle 4

Note: Cet œil a pour objet la spécification des exigences en matière de sensibilité du récepteur des données différentielles. Les niveaux 1 et 2 figurent à l'extérieur des valeurs de

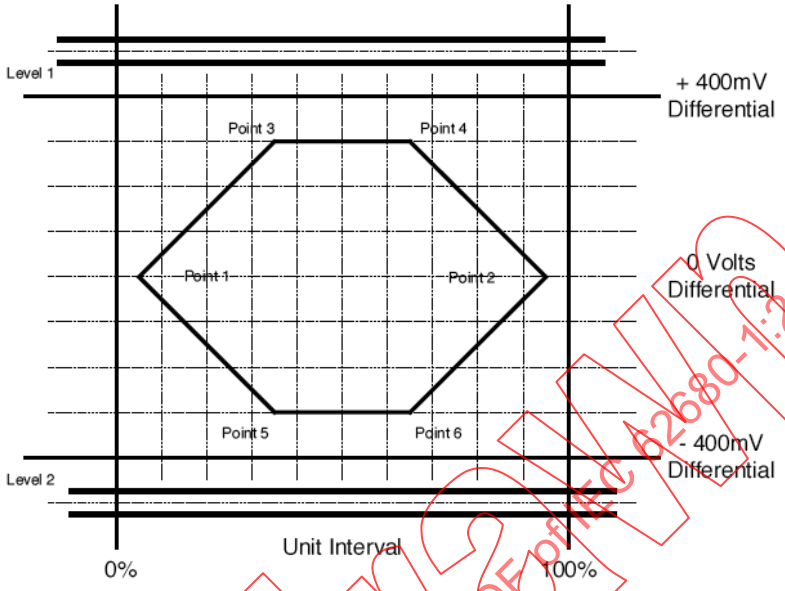
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

seuil de déconnexion, mais une déconnexion est détectée à la source (après un minimum de 32 temps bit sans aucune transition) et pas au niveau du récepteur cible.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Modèle 5

La Figure 7-17 illustre les exigences en matière de forme d'onde pour un émetteur-récepteur, qui fait office de hub, mesuré au niveau de TP1 et pour un émetteur-récepteur qui fait office d'appareil, mesuré au niveau de TP4.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	525 mV in UI following a transition, 475 mV in all others	N/A
Level 2	-525 mV in UI following a transition, -475 in all others	N/A
Point 1	0 V	5% UI
Point 2	0 V	95% UI
Point 3	300 mV	35% UI
Point 4	300 mV	65% UI
Point 5	-300 mV	35% UI
Point 6	-300 mV	65% UI

Légende

Anglais	Français
+400mV Differential	+400 mV différentiel
0 Volts Differential	0 Volt différentiel
-400mV Differential	-400 mV différentiel
Unit Interval	Intervalle d'unité
Voltage Level (D+ D-)	Niveau de tension (D+ D-)
Time (% of Unit Interval)	Temps (% d'intervalle d'unité)
Level 1	Niveau 1
525 mV in UI following a transition, 475 mV	525 mV dans UI après une transition, 475 mV dans

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

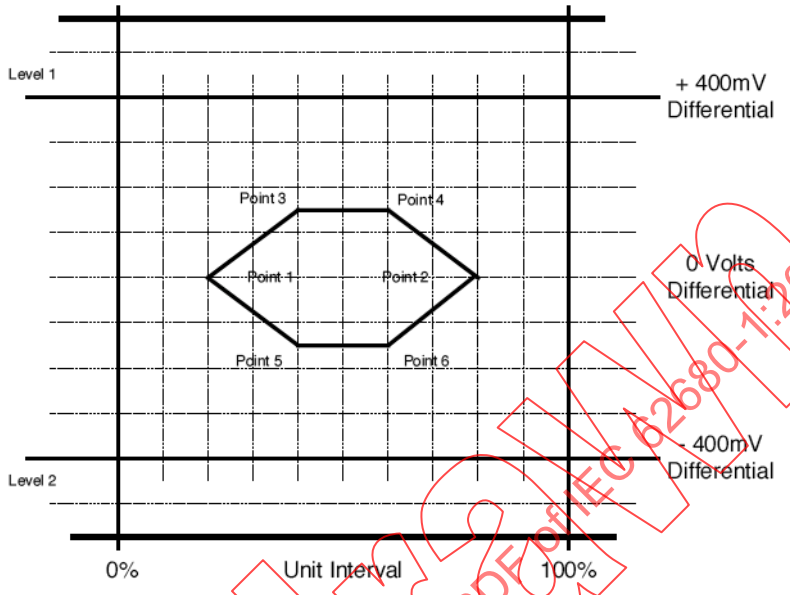
Anglais	Français
in all others	les autres
N/A	N/A
Level 2	Niveau 2
-525 mV in UI following a transition, -475 mV in all others	-525 mV dans UI après une transition, -475 mV dans les autres

Figure 7-17. Modèle 5

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Modèle 6

La Figure 7-18 illustre les exigences en matière de sensibilité du récepteur pour un émetteur-récepteur qui fait office d'appareil, lorsqu'un signal est appliqué au niveau de TP4 et pour un émetteur-récepteur, qui fait office de hub, lorsqu'un signal est appliqué au niveau de TP1.



	Voltage Level (D+ - D-)	Time (% of Unit Interval)
Level 1	575 mV	N/A
Level 2	-575 mV	N/A
Point 1	0 V	20% UI
Point 2	0 V	80% UI
Point 3	150 mV	40% UI
Point 4	150 mV	60% UI
Point 5	-150 mV	40% UI
Point 6	-150 mV	60% UI

Légende

Anglais	Français
+400mV Differential	+400 mV différentiel
0 Volts Differential	0 Volt différentiel
-400mV Differential	-400 mV différentiel
Unit Interval	Intervalle d'unité
Voltage Level (D+ D-)	Niveau de tension (D+ D-)
Time (% of Unit Interval)	Temps (% d'intervalle d'unité)
Level 1	Niveau 1
N/A	N/A
Level 2	Niveau 2

Figure 7-18. Modèle 6

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Note: Cet œil a pour objet la spécification des exigences en matière de sensibilité du récepteur des données différentielles. Les niveaux 1 et 2 figurent à l'extérieur des valeurs de seuil de déconnexion, mais une déconnexion est détectée à la source (après un minimum de 32 temps bit sans aucune transition) et pas au niveau du récepteur cible.

Temps de montée et de chute pour une signalisation à grande vitesse

Le temps de transition d'un pilote à grande vitesse ne doit pas être inférieur au temps minimum de montée et de chute différentielles minimum spécifié admissible (THSR et THSF). Les temps de transition sont mesurés lors du transport d'une charge de référence de 45 Ω à la masse sur D+ et D-. La Figure 7-12 indique un "montage d'essai de l'émetteur" recommandé pour effectuer ces mesures.

Pour un hub ou pour un appareil doté d'un câble détachable, les temps de montée et de chute différentielles à grande vitesse de 10 % à 90 % doivent être d'au minimum 500 ps, lorsqu'ils ont été mesurés au niveau des réceptacles A ou B (respectivement).

Pour un appareil doté d'un ensemble de câbles captifs, une directive de conception recommandée indique que les temps de montée et de chute différentielles à grande vitesse de 10 % à 90 % doivent être d'au minimum 500 ps, lorsqu'ils ont été mesurés au niveau du point où le câble est relié au circuit imprimé de l'appareil.

Il est nécessaire que les transitions de données à grande vitesse soient monotones sur les ouvertures verticales minimum spécifiées dans les modèles de diagramme en œil précédents.

7.1.2.3 Utilisation du pilote

Les ports de fonctions orientés en amont doivent utiliser une et seulement une des trois configurations de pilote suivantes:

1. Basse vitesse - Uniquement les pilotes à basse vitesse
2. Vitesse intégrale - Uniquement les pilotes à vitesse intégrale
3. Vitesse intégrale/grande vitesse - Une combinaison de pilotes à vitesse intégrale et à grande vitesse

Les ports de hubs USB 2.0 orientés en amont doivent utiliser des pilotes à vitesse intégrale/grande vitesse. Ces ports doivent être en mesure de transmettre des données à basse vitesse et à vitesse intégrale avec une signalisation à vitesse intégrale, et à grande vitesse en utilisant une signalisation à grande vitesse. Les ports orientés en aval (hôte inclus) doivent prendre en charge la signalisation à basse vitesse, à vitesse intégrale et à grande vitesse et doivent être en mesure de transmettre les données à chacune des trois vitesses de transfert associées.

Dans cette section, il existe une référence à une situation dans laquelle les opérations à grande vitesse sont "interdites". Ce sujet est abordé en profondeur dans le Chapitre 11 de cette spécification. En bref, les ports orientés en aval d'un hub capable de fonctionner à grande vitesse ont "l'interdiction d'opérer à grande vitesse" si le hub n'est pas en mesure d'établir une connexion à grande vitesse sur son port orienté en amont. Par exemple, cela s'avère exact pour les ports orientés en aval d'un hub capable de fonctionner à grande vitesse, lorsque le hub est connecté à un contrôleur d'hôte USB 1.1.

Lorsqu'un appareil opérant à vitesse intégrale/grande vitesse est relié à un hub pré-USB 2.0, ou à un port de hub pour lequel la grande vitesse est interdite, il doit se comporter comme un appareil à même de fonctionner à vitesse intégrale uniquement. Lorsqu'un appareil opérant à vitesse intégrale/grande vitesse est relié à un hub USB 2.0 pour lequel la grande vitesse est autorisée, il doit fonctionner avec une signalisation et un débit de données à grande vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.1.3 Dérivation de câble

La dérivation maximale introduite par le câble entre la paire de signalisations différentielles (c.-à-d, D+ et D- (TSKEW)) doit être inférieure à 100 ps et est mesurée de la manière décrite en 6.7.

7.1.4 Caractéristiques des récepteurs

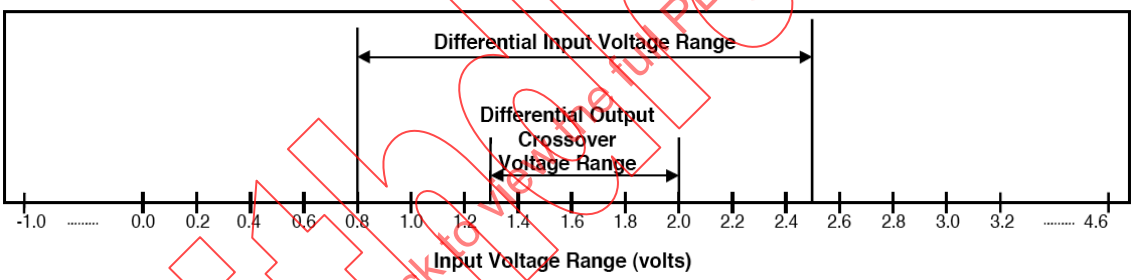
Cette section aborde les caractéristiques des récepteurs pour les émetteurs-récepteurs à basse vitesse, vitesse intégrale et vitesse intégrale/grande vitesse.

7.1.4.1 Caractéristiques des récepteurs à basse vitesse et à vitesse intégrale

Un récepteur à entrée différentielle doit être utilisé pour accepter le signal de données USB. Le récepteur doit présenter une sensibilité d'entrée (VDI) d'au moins 200 mV lorsque les deux entrées de données différentielles figurent dans la gamme de mode commun différentiel (VCM) allant de 0,8 V à 2,5 V, comme représenté à la Figure 7-19

Outre le récepteur différentiel, un récepteur à extrémité -unique doit être présent pour chacune des deux lignes de données. Les récepteurs doivent disposer d'un seuil de commutation compris entre 0,8 V (VIL) et 2,0 V (VIH). Il est recommandé que les récepteurs à extrémité unique incorporent une hystérésis afin de réduire leur sensibilité au bruit.

Les deux lignes D+ et D- peuvent temporairement être inférieures à VIH (min) au cours des transitions de signaux différentiels. Cette période peut aller jusqu'à 14 ns (TFST) pour les transitions à vitesse intégrale et jusqu'à 210 ns (TLST) pour les transitions à basse vitesse. La logique dans le récepteur doit assurer que cela n'est pas interprété en tant que SE0.



Légende

Anglais	Français
Differential Input Voltage Range	Plage de tension d'entrée différentielle
Differential Output Crossover Voltage Range	Plage de tension de croisement de sortie différentielle
Input Voltage Range (volts)	Plage de tension d'entrée (volts)

Figure 7-19. Gamme de sensibilité pour les entrées différentielles à basse vitesse/vitesse intégrale

7.1.4.2 Caractéristiques des récepteurs à grande vitesse

Un émetteur-récepteur capable de fonctionner à grande vitesse doit être conforme aux spécifications des caractéristiques du récepteur signalées dans la Section 7.1.4.1, lors de la réception en modes à basse vitesse et à vitesse intégrale.

Comme illustré dans la Figure 7-1, un émetteur-récepteur capable de fonctionner à grande vitesse et qui opère en mode grande vitesse "détecte" un flux de données sérielles entrantes avec le récepteur des données différentielles à grande vitesse et le détecteur de l'enveloppe de transmission. En outre, un émetteur-récepteur capable de fonctionner à grande vitesse et orienté en aval surveille l'amplitude de la tension différentielle sur les lignes avec le détecteur de l'enveloppe de déconnexion.

Lors d'une réception en mode à grande vitesse, le récepteur différentiel doit être en mesure de recevoir de manière fiable les signaux conformes aux modèles de diagramme en œil du récepteur illustré dans la Section 7.1.2. En outre, une directive fortement recommandée

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

stipule qu'il convient qu'un récepteur à grande vitesse puisse recevoir de manière fiable de tels signaux, lors de la présence d'un composant de tension en mode commun (V_{HSCM}), sur une plage allant de -50 mV à 500 mV (le composant en mode commun nominal d'une signalisation à grande vitesse est de 200 mV). Il convient que la signalisation J et K d'un chirp basse fréquence, qui se produit au cours d'un rappel pour le protocole de transfert, soit reçue de manière fiable avec une plage de tension en mode commun comprise entre -50 mV et 600 mV.

La réception des données est qualifiée par la sortie du détecteur de l'enveloppe de transmission. Le récepteur doit désactiver la récupération des données, lorsque le signal chute en dessous du niveau du silencieux (V_{HSSQ}) donné dans le Tableau 7-3. (Le détecteur doit indiquer le silencieux, lorsque la magnitude de l'enveloppe de tension différentielle se monte à ≤ 100 mV et ne doit pas indiquer le silencieux si l'amplitude de l'enveloppe de tension différentielle est de ≥ 150 mV.) La détection du silencieux doit être effectuée avec un détecteur d'enveloppe différentielle, tel que celui illustré dans la Figure 7-1. Le détecteur d'enveloppe utilisé pour détecter l'état du silencieux doit incorporer un mécanisme de filtrage qui empêche l'indication du silencieux au cours des croisements des données différentielles.

La définition d'un modèle SYNC de paquet à grande vitesse, de concert avec les exigences en matière de hubs, garantit qu'un récepteur peut voir au moins 12 bits de SYNC (KJKJKJKJKJKK) suivis par la portion de données du paquet. Cela signifie que la combinaison temps de réponse du silencieux, temps de verrouillage DLL et fin de la détection SYNC doit se produire dans 12 temps bit. Cela est requis pour garantir la réception correcte du premier bit de la charge utile du paquet.

Dans le cas d'un port orienté en aval, un émetteur-récepteur capable de fonctionner à grande vitesse doit inclure un détecteur d'enveloppe différentielle qui indique à quel moment le signal sur les données excède le niveau de déconnexion à grande vitesse (V_{HSDSC}) comme donné dans le Tableau 7-3. (Le détecteur ne doit pas indiquer que le seuil de déconnexion a été dépassé si l'amplitude du signal différentiel est de ≤525 mV, et doit mentionner que le seuil a été dépassé si l'amplitude du signal différentiel est de ≥625 mV.)

Lorsqu'il est contrôlé au moment opportun, ce détecteur indique que l'appareil a été déconnecté. Les informations détaillées sur la manière dont le détecteur de l'enveloppe de déconnexion est utilisé sont décrites dans la Section 7.1.7.3.

7.1.5 Identification de la vitesse de l'appareil

Les sections suivantes spécifient les mécanismes d'identification de la vitesse, pour la basse vitesse, la vitesse intégrale et la grande vitesse.

7.1.5.1 Identification de la vitesse de l'appareil fonctionnant à basse/vitesse intégrale

L'USB se termine au niveau du hub et la fonction s'achève de la manière représentée à la Figure 7-20 et à la Figure 7-21. Les appareils à vitesse intégrale et à basse vitesse sont différenciés par la position de la résistance de rappel à la source de l'extrémité orientée en aval du câble:

- Les appareils à vitesse intégrale sont terminés de la manière illustrée dans la Figure 7-20, avec la résistance de rappel à la source placée sur la ligne D+.
- Les appareils à basse vitesse sont terminés de la manière illustrée dans la Figure 7-21, avec la résistance de rappel à la source placée sur la ligne D-.
- Les terminateurs de rappel à la masse sur les ports orientés en aval sont des résistances de 15 kΩ ± 5 % connectées à la masse.

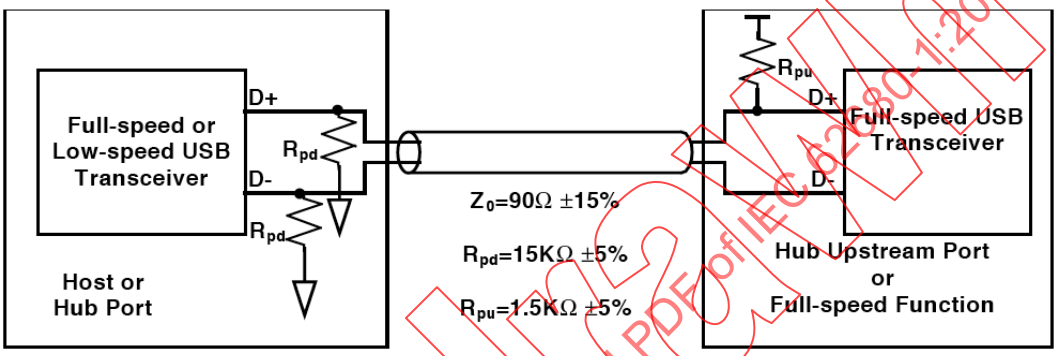
La conception de la résistance de rappel à la source doit assurer que le niveau des signaux satisfait aux exigences spécifiées dans le Tableau 7-2. La conception de la résistance de rappel à la source doit garantir que les niveaux de signal satisfont aux exigences spécifiées dans le Tableau 7-2. Afin de faciliter l'évaluation de l'état du bus qui peut être effectuée à la fin d'une réinitialisation, la conception doit être en mesure de tracter D+ ou D- à partir de 0 V

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

jusqu'à VIH (min.) pendant la durée de relaxation minimale de la réinitialisation de 2,5 µ. Un appareil doté d'un câble détachable doit utiliser une résistance de 1,5 kΩ ± 5 % reliée à une source de tension située entre 3,0 V et 3,6 V (VTERM) pour satisfaire à ces exigences. Les appareils équipés de câbles captifs peuvent utiliser des moyens de terminaison alternatifs. Toutefois, la résistance de Thevenin de chaque terminaison ne doit pas être inférieure à 900 Ω.

Note: La résistance de Thevenin d'une terminaison n'inclut pas la résistance de 15 kΩ ± 5 % sur l'hôte/le hub.

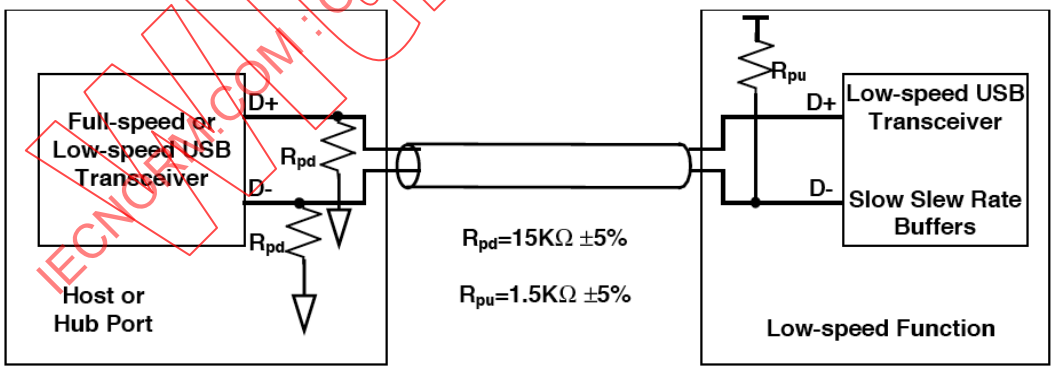
La source de tension sur la résistance de rappel à la source doit être dérivée de, ou contrôlée par la puissance fournie sur le câble USB, de manière à ce que lorsque VBUS est retiré, la résistance de rappel à la source ne transmette pas de courant sur la ligne de données à laquelle elle est rattachée.



Légende

Anglais	Français
Full-speed or Low-speed USB Transceiver	Emetteur-récepteur USB à vitesse intégrale ou faible vitesse
Full-speed USB Transceiver	Emetteur-récepteur USB à vitesse intégrale
Host or Hub Port	Hôte ou port de hub
Hub Upstream Port or Full-speed Function	Port en amont de hub ou fonction à vitesse intégrale

Figure 7-20. Connexions du câble de l'appareil à vitesse intégrale et de la résistance



Légende

Anglais	Français
Full-speed or Low-speed USB Transceiver	Emetteur-récepteur USB à vitesse intégrale ou faible vitesse
Low-speed USB Transceiver	Emetteur-récepteur USB à vitesse faible
Slow Slew Rate Buffers	Mémoires tampons à faible vitesse de balayage
Host or Hub Port	Hôte ou port de hub
Low-speed Function	Fonction à faible vitesse

Figure 7-21. Connexions du câble de l'appareil à basse vitesse et de la résistance

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.1.5.2 Identification de la vitesse de l'appareil fonctionnant à grande vitesse

Les mécanismes de détection et de réinitialisation suivent le modèle de comportement pour la basse vitesse/vitesse intégrale. Lorsque la réinitialisation est terminée, la liaison doit fonctionner dans son mode de signalisation approprié (basse vitesse, vitesse intégrale ou grande vitesse, comme régi par les règles d'utilisation précédentes) et les bits d'indication de la vitesse dans le registre d'état du port rapportent ce mode de manière correcte. Le logiciel peut uniquement initier l'assertion de réinitialisation et lire le registre d'état du port sur notification de fin de la réinitialisation.

Les appareils en mesure de fonctionner à grande vitesse sont initialement liés aux appareils opérant à vitesse intégrale. Cela signifie que pour les ports orientés en amont capables d'opérer à grande vitesse, RPU (1,5 kΩ ± 5 %) doit être connecté, à partir de D+ vers une alimentation de 3,3 V (comme illustré dans la Figure 7-1), par le biais d'un commutateur qui peut être ouvert sous commande logicielle.

Après la fixation initiale, les émetteurs-récepteurs capables de fonctionner à grande vitesse s'engagent dans un protocole de niveau bas au cours de la réinitialisation, afin d'établir une liaison à grande vitesse et d'indiquer l'opération à grande vitesse dans le registre d'état du port approprié. Ce protocole est décrit dans la Section 7.1.7.5.

7.1.6 Caractéristiques d'entrée

Les sections suivantes décrivent les caractéristiques d'entrée pour les émetteurs-récepteurs opérant en modes basse vitesse, vitesse intégrale et grande vitesse.

7.1.6.1 Caractéristiques d'entrées à basse vitesse et à vitesse intégrale

Il convient que l'impédance d'entrée de D+ ou D-, sans terminaison soit de > 300 kΩ (Z_{INP}). La capacité d'entrée d'un port est mesurée au niveau des broches du connecteur. Différentes valeurs de capacité sont autorisées pour les ports orientés en amont et en aval. La capacité maximum (différentiel ou à extrémité unique) (C_{IND}) autorisée sur un port ou un hub ou un hôte orienté en aval est de 150 pF sur D+ ou D- lors d'un fonctionnement à basse vitesse ou à vitesse intégrale. Ce nombre comprend un maximum de 75 pF de capacité globale à la masse sur chaque ligne au niveau de l'émetteur-récepteur et dans le connecteur et une capacité supplémentaire de 75 pF sur chaque conducteur dans la ligne de transmission, entre le réceptacle et l'émetteur-récepteur. La ligne de transmission entre le réceptacle et RS doit être de 90 Ω ± 15 %.

La capacité maximale sur un port orienté en amont d'un appareil à vitesse intégrale doté d'un câble détachable (C_{INUB}) est de 100 pF sur D+ ou D-. Ce nombre comprend un maximum de 75 pF de capacité globale à la masse sur chaque ligne au niveau de l'émetteur-récepteur et dans le connecteur et une capacité supplémentaire de 25 pF sur chaque conducteur dans la ligne de transmission, entre le réceptacle et l'émetteur-récepteur. La différence en matière de capacité entre D+ et D- doit être inférieure à 10 %.

Pour les appareils à vitesse intégrale équipés de câbles captifs, l'appareil en lui-même peut avoir une capacité globale allant jusqu'à 75 pF à la masse, sur D+ et D-. Le câble justifie le reste de la capacité d'entrée.

Un appareil à basse vitesse doit disposer d'un câble captif. La capacité d'entrée d'un appareil à basse vitesse inclut le câble. La capacité d'entrée différentielle ou à extrémité unique d'un appareil à basse vitesse est de 450 pF (C_{LINUA}).

Pour les appareils équipés de câbles captifs, la capacité d'entrée à extrémité unique doit être consistante avec le schéma de terminaison utilisé. La terminaison doit être en mesure de charger la ligne D+ ou D- de 0 à V_{IH} (min) dans 2,5 µs. La capacité sur D+/D- inclut la capacité d'entrée à extrémité unique de l'appareil (mesurée à partir des broches sur le connecteur du câble) et les 150 pF de capacité d'entrée sur l'hôte/le hub.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Une implantation peut utiliser de petits hubs au niveau de l'émetteur-récepteur à des fins de contrôle de débit du bord. La somme de la capacité du hub ajouté (CEDGE), de l'émetteur-récepteur et du hub reliant la trace ainsi que de l'émetteur-récepteur vers RS ne doit pas dépasser 75 pF (extrémité unique ou différentielle) et la capacité doit être équilibrée dans les 10 %. Le cas échéant, le condensateur ajouté doit être placé entre les broches de l'émetteur-récepteur et le RS (voir la Figure 7-22).

L'utilisation de perles de ferrite sur les lignes D+ ou D- des appareils à vitesse intégrale n'est pas encouragée.

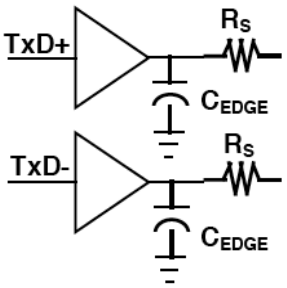
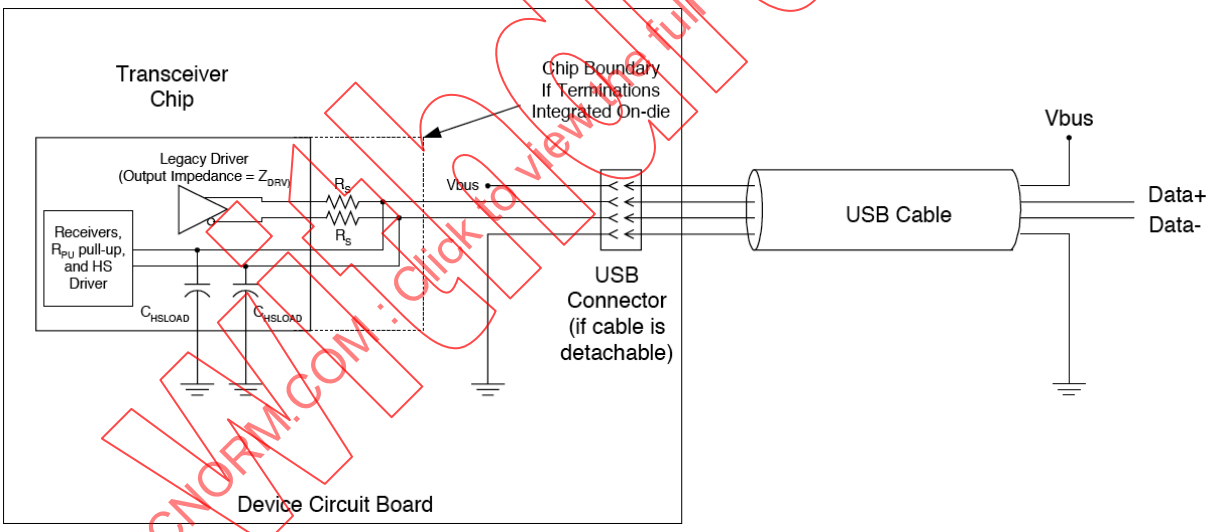


Figure 7-22. Placement de condensateurs de contrôle de débit du bord optionnels pour la basse vitesse/vitesse intégrale

7.1.6.2 Caractéristiques d'entrée à grande vitesse

La Figure 7-23 illustre le circuit de chargement simple équivalent d'un appareil USB fonctionnant en mode de réception à grande vitesse.



Légende

Anglais	Français
Transceiver Chip	Puce d'émetteur-récepteur
Chip Boundary If Terminations Integrated On-die	Limite de puce si les terminaisons sont intégrées
Legacy Driver (Output Impedance = Z_{DRV})	Pilote hérité (impédance de sortie = Z_{DRV})
Receivers R_{UP} pull-up, and HS Driver	Récepteurs de rappel R_{UP} et pilote HS
USB Connector (if cable is detachable)	Connecteur USB (si le câble est amovible)
USB Cable	Câble USB
Data+	Données+
Data-	Données-
Device Circuit Board	Circuit imprimé d'appareil

Figure 7-23. Diagramme pour le circuit de chargement équivalent à grande vitesse

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Lorsqu'il fonctionne en mode de signalisation à grande vitesse, un émetteur-récepteur doit satisfaire aux spécifications de chargement suivantes:

1. Spécifications de la tension de sortie continue et de la résistance
2. Spécification de chargement TDR

Il convient en outre, et il est fortement recommandé, qu'un composant de l'émetteur-récepteur fonctionnant en mode de signalisation à grande vitesse satisfasse à la directive suivante traitant de la capacité globale.

L'utilisation de ferrites sur les lignes de données à grande vitesse est fortement déconseillée.

Spécifications de la tension de sortie continue et de la résistance - Un émetteur-récepteur qui se trouve en mode à grande vitesse doit présenter une charge CC sur chacune des lignes de données, nominalement équivalente à 45 Ω à la masse. La résistance réelle, Z_{HSDRV}, doit être de 40,5 Ω ≤ Z_{HSDRV} ≤ 49,5 Ω. La tension de sortie dans l'état de repos (V_{HSTERM}) à grande vitesse est spécifiée dans le Tableau 7-3.

Spécification de chargement TDR - Les spécifications de chargement CA d'un émetteur récepteur dans l'état de repos à grande vitesse sont spécifiées en termes de mesures TDR (Time Domain Reflectometer) différentielles.

Ces mesures régissent les discontinuités maximales admissibles sur la ligne de transmission pour le connecteur du port, l'interconnexion conduisant le connecteur à l'émetteur-récepteur, l'ensemble émetteur-récepteur et l'IC de l'émetteur-récepteur. Dans le cas particulier où un appareil à même de fonctionner à grande vitesse est équipé d'un câble captif, les discontinuités de la ligne de transmission pour l'ensemble de câbles sont également régies.

Les spécifications suivantes doivent être satisfaites avec le temps de montée incident du TDR différentiel réglé sur 400 ps. Il est important de noter que tous les temps restent "tels qu'ils sont affichés" sur le TDR et par conséquent correspondent "au temps aller-retour".

L'impédance de terminaison (Z_{HSTERM}) est mesurée sur la trace TDR à un temps de mesure spécifique suivant le temps de référence du connecteur. Le temps de référence du connecteur est déterminé en déconnectant la connexion TDR du connecteur du port et en notant le temps de l'étape à circuit ouvert. Pour un connecteur A, le temps de mesure est de 8 ns après la position de référence du connecteur. Pour un connecteur B, le temps de mesure est de 4 ns après la position de référence du connecteur. L'impédance de terminaison différentielle doit être de:

$$80\ \Omega \leq Z_{HSTERM} \leq 100\ \Omega$$

Dans le passage de l'impédance (ZHSTHRU) figure l'impédance mesurée à partir de 500 ps avant la position de référence du connecteur jusqu'au temps régi par la spécification de l'impédance de terminaison.

$$70\ \Omega \leq Z_{HSTHRU} \leq 110\ \Omega$$

Dans la fenêtre d'exception (une fenêtre glissante de 1,4 ns à l'intérieur de la fenêtre de temps de passage de l'impédance), l'impédance différentielle peut dépasser les limites de passage. Aucune simple excursion, toutefois, ne peut dépasser les limites de passage pour plus de deux fois le temps de montée TDR (400 ps).

Dans le cas particulier où un appareil à même de fonctionner à grande vitesse est équipé d'un câble captif, les mêmes spécifications doivent être satisfaites, mais les mesures TDR doivent être prises au travers de l'ensemble de câbles. La détermination du temps de référence du connecteur peut être plus difficile dans ce cas, vu que le câble peut ne pas être retiré facilement du port en cours d'essai. Il incombe au responsable de l'essai d'un appareil spécifique de déterminer la position de référence du connecteur par un quelconque moyen mis à sa disposition.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Directive en matière de capacité globale pour le composant de l'émetteur-récepteur

Lors de la caractérisation d'une puce d'un émetteur-récepteur en tant que composant isolé, la mesure peut être effectuée de manière efficace au niveau de la limite de la puce illustrée dans la Figure 7-23, sans connecteurs, ni câbles USB. La capacité parasite du montage d'essai peut être corrigée en mesurant la capacité du montage en lui-même et en soustrayant ce résultat à la valeur prise avec l'émetteur-récepteur inséré. Si les terminaisons sont en dehors de la puce, il convient que des résistances RS discrètes soient mises en place au cours des mesures et il convient que les mesures soivent prises sur le "côté connecteur" des résistances. Il convient que l'émetteur-récepteur se trouve en mode Test_SE0_NAK au cours de l'essai.

Les mesures de capacité sont prises à partir de chacune des lignes de données à la masse alors que l'autre ligne est laissée ouverte. L'instrument utilisé pour effectuer cette mesure doit être capable de déterminer la capacité effective à la masse en présence de la résistance parallèle effective à la masse.

Capacitance à la masse sur chaque ligne: $C_{HSLOAD} \leq 10$ pF
Correspondance de la capacitance à la masse: ≤ 1,0 pF

La directive est de ne pas autoriser plus de 5,0 pF pour le dé de l'émetteur-récepteur en lui-même et pas plus de 5 pF supplémentaires pour l'ensemble. Il convient que la capacitance différentielle au travers des entrées de l'émetteur-récepteur ne dépasse pas 5,0 pF

7.1.7 Niveaux de signalisation

Les sections suivantes spécifient les niveaux de signalisation, pour les opérations à basse vitesse, à vitesse intégrale et à grande vitesse.

7.1.7.1 Niveaux de signalisation à basse vitesse/vitesse intégrale

Le Tableau 7-2 résume les niveaux de signalisation USB. La source doit conduire les niveaux spécifiés dans la seconde colonne et la cible doit quant à elle identifier l'état correct du bus lorsqu'il consulte les niveaux dans la troisième colonne. (Les récepteurs cibles peuvent être plus sensibles tant qu'ils demeurent dans les limites spécifiées dans la quatrième colonne).

Tableau 7-2. Niveaux de signalisation à basse vitesse/vitesse intégrale

Etat du bus	Niveaux de signalisation		
	Au niveau du connecteur de la source d'origine (à la fin du temps bit)	Au niveau du connecteur de la cible finale	
		Obligatoire	Acceptable
Différentiel "1"	D+ > VOH (min) et D- < VOL (max)	(D+) - (D-) > 200 mV et D+ > VIH (min)	(D+) - (D-) > 200 mV
Différentiel "0"	D- > VOH (min) et D+ < VOL (max)	(D-) - (D+) > 200 mV et D- > VIH (min)	(D-) - (D+) > 200 mV
Une seule extrémité 0 (SE0)	D+ et D- < VOL (max)	D+ et D- < VIL (max)	D+ et D- < VIH (min)
Une seule extrémité 1 (SE0)	D+ et D- > VOSE1(min)	D+ et D- < VIL (max)	
Etat J des données: Basse vitesse Vitesse intégrale	 Différentiel "0" Différentiel "1"	 Différentiel "0" Différentiel "1"	
Etat K des données: Basse vitesse Vitesse intégrale	 Différentiel "1" Différentiel "0"	 Différentiel "1" Différentiel "0"	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Etat du bus	Niveaux de signalisation		
	Au niveau du connecteur de la source d'origine (à la fin du temps bit)	Au niveau du connecteur de la cible finale	
		Obligatoire	Acceptable
Inactif: Basse vitesse Vitesse intégrale	NA	 D- > VIHZ (min) et D+ < VIL (max) D+ > VIHZ (min) et D- < VIL (max)	 D- > VIHZ (min) et D+ < VIH (min) D+ > VIHZ (min) et D- < VIH (min)
Etat de reprise	Etat K des données	Etat K des données	
Début du paquet (SOP)	Commutation des lignes de données de l'état de repos à l'état K		
Fin du paquet (EOP)[4]	SEO pour environ 2 temps bit[1] suivi par un J pour 1 temps bit[3]	SE0 pour ≥ 1 temps bit[2] suivi par un état J pour 1 temps bit	SE0 pour ≥ 1 temps bit[2] suivi par un état J
Déconnexion (au niveau du port en aval)	NA	SE0 pour ≥2,5 µs	
Connexion (au niveau du port en aval)	NA	Repos pour ≥ 2 ms	Repos pour ≥2,5 µs
Réinitialisation	D+ et D- < VOL (max) pour ≥10 ms	D+ et D- < VIL (max) pour ≥10 ms	D+ et D- < VIL (max) pour ≥2,5 µs

Note 1: La largeur de l'EOP est définie en temps bit relatifs à la vitesse de transmission. (Les largeurs EOP de spécification sont indiquées dans le Tableau 7-7 et le Tableau 7-8.)

Note 2: La largeur de l'EOP est définie en temps bit relatifs au type d'appareil recevant l'EOP. Le temps bit est approximatif.

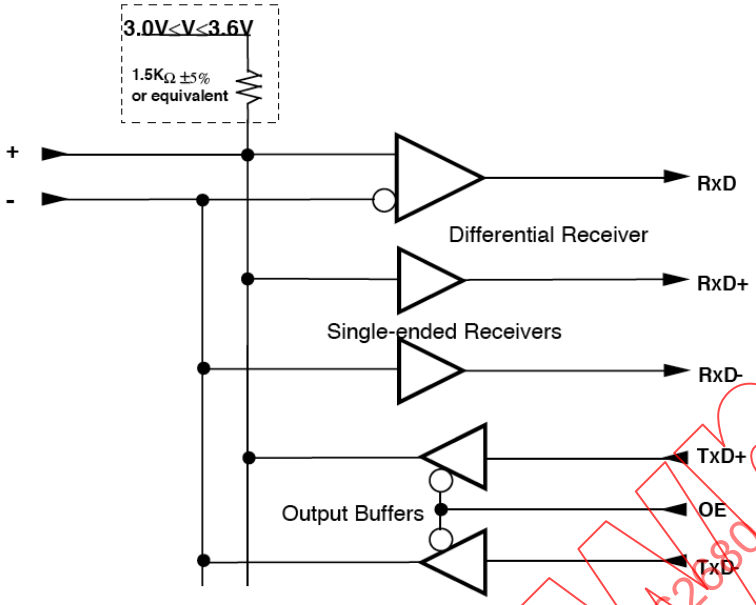
Note 3: La largeur de l'état J suivant l'EOP est définie en temps bit relatifs au débit du bord de la mémoire tampon. L'état J d'une mémoire tampon à basse vitesse doit être un temps bit à basse vitesse et, à partir d'une mémoire tampon à vitesse intégrale, une largeur de temps bit à vitesse intégrale.

Note 4: Le maintien en réserve active constitue l'EOP à basse vitesse.

Les états de données J et K correspondent à deux niveaux logiques utilisés pour communiquer les données différentielles dans le système. La signalisation différentielle est mesurée à partir du point où les signaux des lignes de données se croisent. La signalisation des données différentielles n'est pas concernée par le niveau traversé par les signaux, tant que la tension croisée satisfait aux exigences de la Section 7.1.2. A noter que, au niveau du récepteur, les états de repos et de reprise sont logiquement équivalents aux états J et K respectivement.

Comme illustré dans le Tableau 7-2, les états J et K pour la signalisation à vitesse intégrale sont inversés par rapport à ceux pour la signalisation à basse vitesse. Le sens de la signalisation de données, du repos et de la reprise est défini par le type d'appareil relié à un port. Si un appareil à vitesse intégrale est connecté à un port, ce segment de l'USB utilise les conventions de signalisation à vitesse intégrale (et les temps de montée et de chute rapides), même si les données envoyées à travers les lignes de données sont à un débit de données à basse vitesse. Les conventions de signalisation à basse vitesse indiquées au Tableau 7-2 (plus les temps de montée et de chute) sont utilisées uniquement entre un appareil à basse vitesse et le port auquel il est relié.

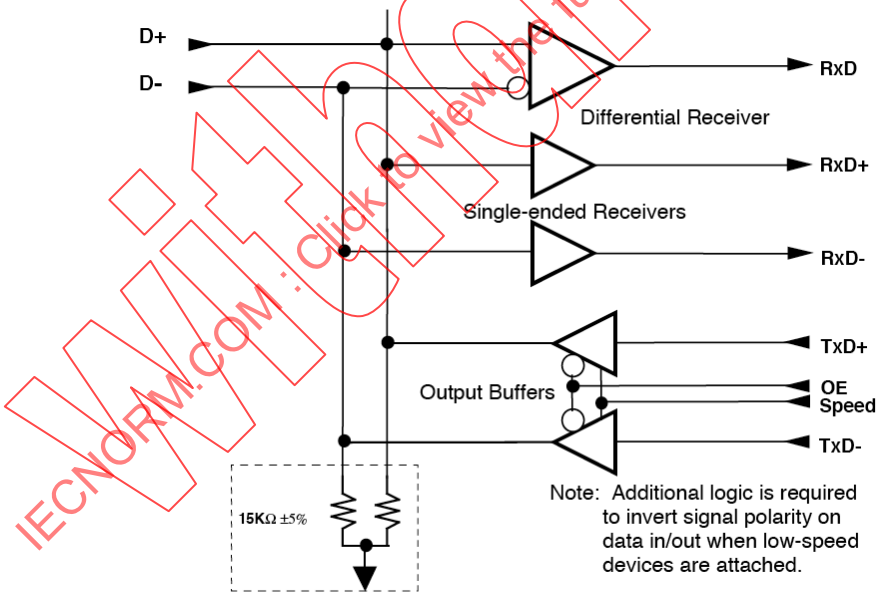
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
1.5K$_{\Omega \pm 5\%}$ or equivalent	1,5 K$_{\Omega \pm 5\%}$ ou équivalent
Differential Receiver	Récepteur différentiel
Single-ended Receivers	Récepteurs à extrémité unique
Output Buffers	Mémoires tampon de sortie

Figure 7-24. Emetteur-récepteur de port orienté en amont à vitesse intégrale



Légende

Anglais	Français
Differential Receiver	Récepteur différentiel
Single-ended Receivers	Récepteurs à extrémité unique
Output Buffers	Mémoires tampons de sortie
Note: Additional logic is required to invert signal polarity on data in/out when low-speed devices are attached.	Note: une logique supplémentaire est requise pour inverser la polarité du signal d'entrée/sortie des données lorsque des appareils à basse vitesse sont connectés.
OE speed	Vitesse OE

Figure 7-25. Emetteur-récepteur de port orienté en aval à basse vitesse/vitesse intégrale

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.1.7.2 Niveaux de signalisation à vitesse intégrale/grande vitesse

Les spécifications de la tension de signalisation à grande vitesse indiquées dans le Tableau 7-3 doivent être satisfaites lors de la prise de mesure au niveau du connecteur le plus proche de l'émetteur-récepteur, en utilisant des résistances de charge de 45 Ω vers la masse de l'appareil, en temps que charges de référence. Toutes les mesures de tension sont prises par rapport à la masse de l'appareil local.

Tableau 7-3. Niveaux de signalisation à vitesse intégrale/grande vitesse

Etat du bus	Niveau de signalisation requis au niveau du connecteur source	Niveau de signalisation requis au niveau du connecteur cible
Différentiel à grande vitesse "1"	**Niveaux CC:** VHSOH (min) ≤ D+ ≤ VHSOH (max) VHSOL (min) ≤ D- ≤ VHSOL (max) Voir note 1. **Niveaux différentiels CA:** Un émetteur doit être conforme aux modèles de diagramme en œil signalés en 7.1.2. Voir note 2.	**Niveaux différentiels CA** Le signal au niveau du connecteur cible doit être récupérable, comme défini par les modèles de diagramme en œil indiqués dans la Section 7.1.2. Voir note 2.
Différentiel à grande vitesse "0"	**Niveaux CC:** VHSOH (min) ≤ D- ≤ VHSOH (max) VHSOL (min) ≤ D- ≤ VHSOL (max) Voir note 1. **Niveaux différentiels CA:** Un émetteur doit être conforme aux modèles de diagramme en œil signalés en 7.1.2. Voir note 2.	**Niveaux différentiels CA:** Le signal au niveau du connecteur cible doit être récupérable, comme défini par les modèles de diagramme en œil indiqués dans la Section 7.1.2. Voir note 2.
Etat J à grande vitesse	Différentiel à grande vitesse "1"	Différentiel à grande vitesse "1"
Etat K à grande vitesse	Différentiel à grande vitesse "0"	Différentiel à grande vitesse "0"
Etat J du Chirp (tension différentielle; s'applique uniquement au cours d'une réinitialisation lorsqu'aussi bien le hub que l'appareil sont en mesure de fonctionner à grande vitesse)	**Niveaux CC:** VCHIRPJ (min) ≤ (D+ - D-) ≤ VCHIRPJ (max)	**Niveaux différentiels CA** Le signal différentiel au niveau du connecteur cible doit être de ≥ 300 mV
Etat K du Chirp (tension différentielle; s'applique uniquement au cours d'une réinitialisation lorsqu'aussi bien le hub que l'appareil sont en mesure de fonctionner à grande vitesse)	**Niveaux CC:** VCHIRPK (min) ≤ (D+ - D-) ≤ VCHIRPK (max)	**Niveaux différentiels CA** Le signal différentiel au niveau du connecteur cible doit être de ≤ -300 mV
Etat du silencieux à grande vitesse	NA	VHSSQ - Le récepteur doit indiquer le silencieux lorsque la magnitude de tension différentielle est de ≤100 mV; le récepteur ne doit pas indiquer le silencieux si la magnitude de tension différentielle est de ≥150 mV. Voir note 3.
Inactif à grande vitesse	NA	**Niveaux CC:** VHSOI min ≤ (D+, D-) ≤ VHSOI max Voir note 1. **Niveaux différentiels CA:** La magnitude de la tension différentielle est de ≤ 100 mV Voir note 3.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 7-3. Niveaux de signalisation à vitesse intégrale/grande vitesse (suite)

Etat du bus	Niveau de signalisation requis au niveau du connecteur source	Niveau de signalisation requis au niveau du connecteur cible
Début du paquet à grande vitesse (HSSOP)	Les lignes de données commutent du repos à grande vitesse à l'état J à grande vitesse ou à l'état K à grande vitesse.	
Fin du paquet à grande vitesse (HSEOP)	Les lignes de données commutent de l'état J ou K à grande vitesse à l'état de repos à grande vitesse.	
Etat de déconnexion à grande vitesse (au niveau du port orienté en aval)	NA	VHSDSC - Le port orienté en aval ne doit pas indiquer la déconnexion de l'appareil si la puissance différentielle est de ≤ 525 mV, et doit indiquer la déconnexion de l'appareil lorsque la magnitude de la tension différentielle est de ≥ 625 mV au temps d'échantillon abordé dans la Section 7.1.7.3.

Note 1: note 1: Mesuré avec une résistance de 45 Ω à la masse à chaque ligne de données, en utilisant les modes d'essai Test_J et Test_K

Note 2: note 2: Mesuré à l'aide du mode d'essai Test_Packet avec le montage illustré dans la Figure 7-12

Note 3: note 3: Mesuré avec le montage illustré dans la Figure 7-12, utilisant le mode d'essai SE0_NACK

Note 4: note 4: Un pilote à grande vitesse ne doit jamais générer "de manière intentionnelle" un signal dans lequel D+ et D- sont amenés à un niveau supérieur à 200 mV. Il convient naturellement que la conception de pilotage de courant à grande vitesse exclue cette possibilité.

7.1.7.3 Signalisation de connexion et de déconnexion

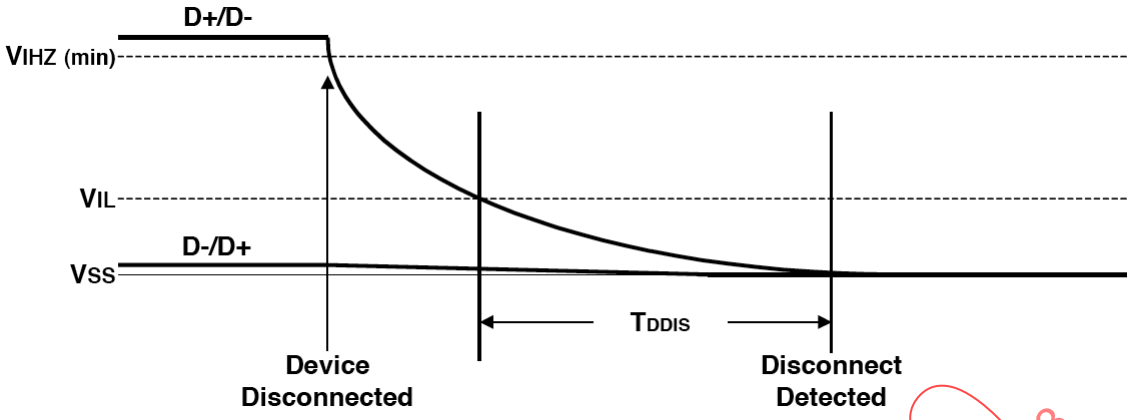
Lorsqu'aucune fonction n'est reliée au port orienté en aval d'un hôte ou d'un hub en basse vitesse/vitesse intégrale, les résistances de rappel à la masse présentes ici entraînent la traction de D+ et D- en dessous du seuil inférieur à extrémité unique de l'émetteur-récepteur faisant office d'hôte ou de hub, lorsque ce port n'est pas conduit par le hub. Cela crée un état SE0 sur le port orienté en aval. Une condition de déconnexion est indiquée si l'hôte ou le hub ne conduit pas les lignes de données et qu'un SEO persiste sur un port orienté en aval pour plus de TDDIS (voir Figure 7-26). Les spécifications pour TDDIS et TDCNN sont définies dans le Tableau 7-13.

Une condition de connexion est détectée lorsque le hub détecte que l'une des lignes de données est tractée au-dessus de son seuil VIH pour plus de TDCNN (voir la Figure 7-27 et la Figure 7-28).

Les hubs doivent déterminer la vitesse de l'appareil relié en échantillonnant l'état du bus, immédiatement avant de conduire SE0, afin d'indiquer une condition de réinitialisation vers l'appareil.

Tous les niveaux de signalisation présents dans le Tableau 7-2 sont définis pour ce segment de bus (et uniquement pour ce segment), une fois que la vitesse de l'appareil relié est déterminée. Les rouages de détection de la vitesse sont décrits dans la Section 11.8.2.

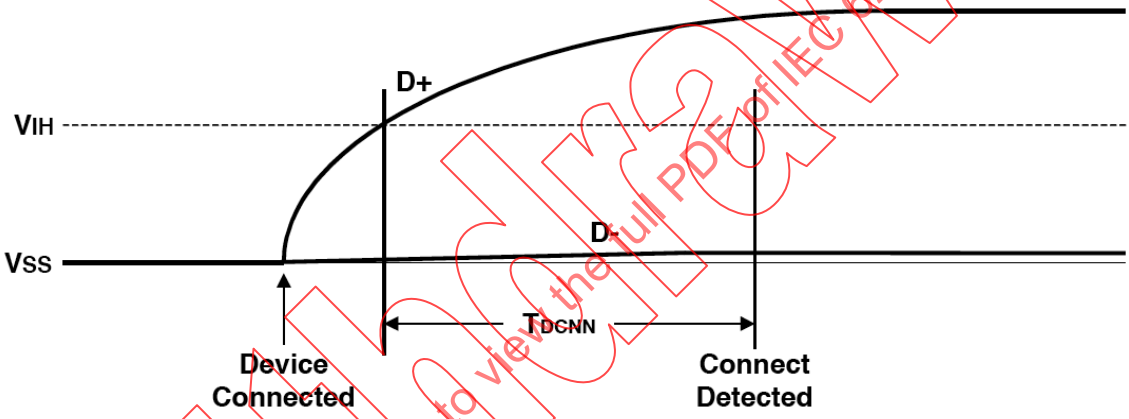
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Device Disconnected	Appareil déconnecté
Disconnect Detected	Déconnexion détectée

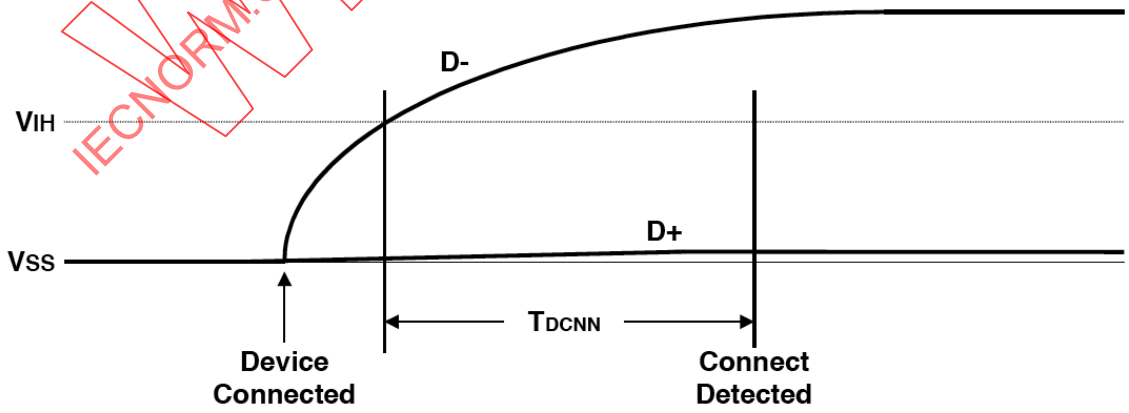
Figure 7-26. Détection de déconnexion à basse vitesse/vitesse intégrale



Légende

Anglais	Français
Device Connected	Appareil connecté
Connect Detected	Connexion détectée

Figure 7-27. Détection de connexion d'appareil à vitesse haute/vitesse intégrale



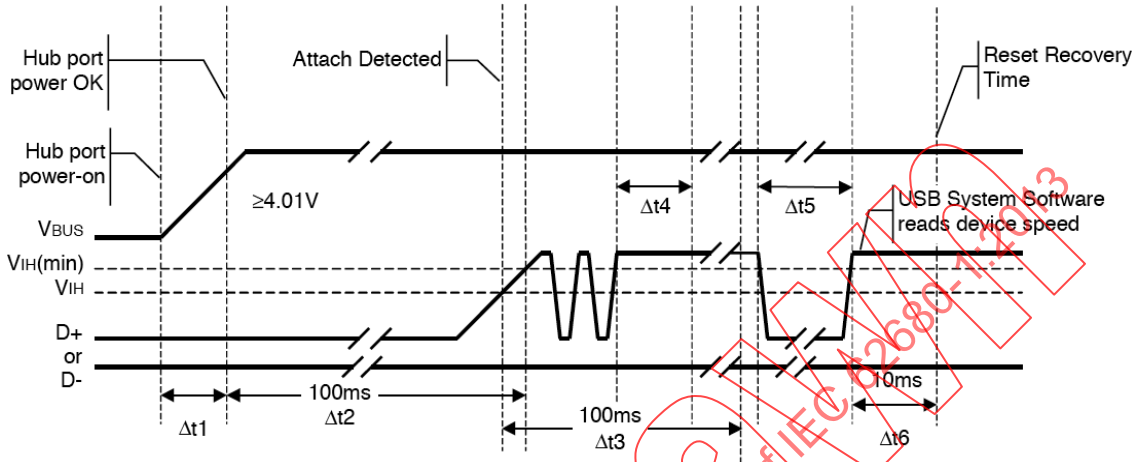
Légende

Anglais	Français
Device Connected	Appareil connecté
Connect Detected	Connexion détectée

Figure 7-28. Détection de connexion d'appareil à basse vitesse

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Parce que des composants USB peuvent être branchés à chaud et que les hubs peuvent implanter l'ouverture et la fermeture électronique d'un circuit de puissance, il est nécessaire de comprendre les délais entre l'ouverture et la fermeture électronique d'un circuit de puissance et/ou la connexion à un appareil et quand la puissance interne de l'appareil s'est stabilisée. La Figure 7-29 illustre tous les évènements associés à l'activation de l'alimentation d'un port avec un appareil connecté et au branchement à chaud d'un appareil. Six délais et une séquence d'évènements sont définis par cette spécification.



Légende

Anglais	Français
Hub port power OK	Alimentation du port de hub OK
Attach Detected	Connexion détectée
Reset Recovery Time	Temps de récupération sur réinitialisation
Hub port power-on	Mise sous tension du port de hub
USB System Software reads device speed	Le logiciel système USB lit la vitesse de l'appareil

Figure 7-29. Temporisation d'évènements de mise sous tension et de connexion

Δt1 Il s'agit de la durée requise par le commutateur d'alimentation du port du hub pour fonctionner. Ce délai est une fonction du type de commutateur du port du hub. Les hubs transmettent cette durée au descripteur du hub (voir Section 11.15.2.1), qui peut être lue via une requête au contrôleur de hub (voir Section 11.16.2.4). Si un appareil a été branché dans un port non commuté ou déjà commuté, Δt1 est égal à zéro.

Δt2 (TSIGATT) Il s'agit de la durée maximale pendant laquelle VBUS atteint un niveau valide (4,01 V) jusqu'à ce qu'un appareil doive signaler une connexion. Δt2 représente le temps nécessaire pour que le rail conducteur interne de l'appareil se stabilise et que D+ ou D- atteigne VIH (min) au niveau du hub. Δt2 doit être inférieur à 100 ms pour toutes les mises en œuvre de hub et d'appareil. (Cette exigence ne s'applique que si le courant électrique est appelé du bus par l'appareil.)

Δt3 (TATTDB) Il s'agit d'un intervalle antirebond avec une durée minimum de 100 ms fournie par le logiciel du système USB. Il garantit la stabilité de la connexion électrique et mécanique avant que le logiciel tente de réinitialiser l'appareil connecté. L'intervalle débute lorsque le logiciel du système USB a été notifié d'une détection de connexion. L'intervalle redémarre en cas de déconnexion. L'intervalle antirebond garantit la stabilité de l'alimentation au niveau de l'appareil pendant au moins 100 ms, avant l'envoi d'une requête à l'appareil.

Δt4 (T2SUSP) Chaque fois qu'un appareil n'observe plus aucune activité sur le bus, il doit se conformer aux règles qui l'obligent à passer en mode d'interruption (voir la Section 7.1.7.6).

Δt5 (TDRST) Il s'agit de la période où le lecteur des hubs effectue une réinitialisation vers un appareil. Voir 7.1.7.5 et 11.5.1.5 pour plus de détails.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Δt6 (TRSTRCY) Le logiciel du système USB garantit un minimum de 10 ms pour la récupération sur réinitialisation. La réponse d'un appareil à chaque transaction de bus vers l'adresse de l'appareil par défaut au cours de la période de récupération sur réinitialisation n'est pas définie.

Les appareils en mesure de fonctionner à grande vitesse doivent tout d'abord se connecter en tant que appareils à vitesse intégrale et doivent satisfaire à toutes les exigences de connexion à vitesse intégrale. Un port orienté en aval, en mesure d'opérer à grande vitesse, doit détecter correctement la connexion des appareils à basse vitesse et à vitesse intégrale et doit aussi satisfaire à tous les comportements de connexion à basse vitesse et à vitesse intégrale.

La transition vers une signalisation à grande vitesse s'accomplit au moyen d'un protocole électrique de niveau bas qui apparaît au cours de la réinitialisation. Ce protocole est spécifié dans la Section 7.1.7.5.

Un émetteur-récepteur orienté en aval et opérant en mode à grande vitesse détecte la déconnexion d'un appareil à grande vitesse en percevant le doublage dans l'amplitude du signal différentiel, parmi les lignes D+ et D-, qui peut se produire lorsque les terminaisons de l'appareil sont retirées. La sortie du détecteur de l'enveloppe de déconnexion passe au niveau haut lorsque l'émetteur-récepteur orienté en aval transmet et que les réflexions positives de la ligne ouverte sont retournées avec une phase, qui constitue un additif, et le signal du pilote de l'émetteur-récepteur. Les signaux avec des amplitudes différentielles de ≥ 625 mV doivent activer de manière fiable le détecteur de l'enveloppe de déconnexion. Les signaux avec des amplitudes différentielles de ≤ 525 mV ne doivent jamais activer le détecteur de l'enveloppe de déconnexion.

Pour garantir que cet effet additif se produise pendant une durée suffisante afin qu'il soit détecté, l'EOP à l'extrémité du SOF à grande vitesse est prolongé d'une chaîne continue de 40 bits sans aucune transition, de la manière abordée à la Section 7.1.13.2. Cette longueur est suffisante pour garantir que la tension au niveau du connecteur du port orienté en aval double, puisque le délai maximum admissible du signal aller-retour est de 30 temps bit.

Lorsqu'un port orienté en aval transmet en mode à grande vitesse et détecte qu'il a envoyé 32 bits sans une transition, la sortie du détecteur de l'enveloppe de déconnexion doit être contrôlée au moins une fois au cours de la transmission des 8 bits suivants au niveau de la sortie de l'émetteur-récepteur. (En l'absence d'erreur de bus, les 8 bits suivants n'incluent pas une transition.) Si le contrôle indique que le seuil de détection de la déconnexion a été dépassé, le port orienté en aval doit indiquer que l'appareil à grande vitesse a été déconnecté. Voir la Section 11.12.4.

7.1.7.4 Signalisation des données

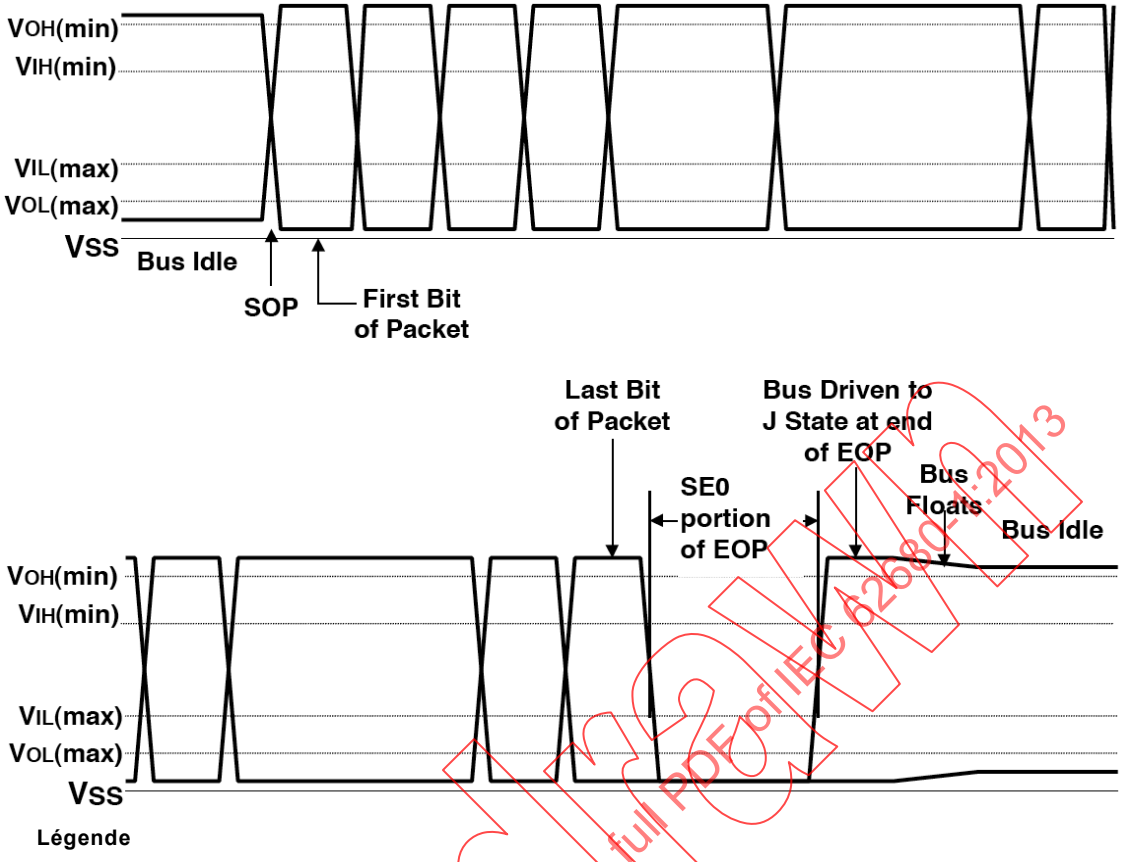
La transmission des données au sein d'un paquet est effectuée avec des signaux différentiels.

7.1.7.4.1 Signalisation à basse vitesse/vitesse intégrale

Le début d'un paquet (SOP) est signalé par le port d'origine qui fait passer les lignes D+ et D- de l'état de repos au niveau logique opposé (état K). Cette commutation de niveaux représente le premier bit du champ SYNC. Les hubs doivent limiter les changements dans la largeur du premier bit de SOP lorsqu'il est retransmis à moins de ± 5 ns. La distorsion peut être minimisée en faisant correspondre le délai nominal des données à travers le hub, avec le délai de validation de sortie du hub.

L'état SE0 est utilisé pour signaler la fin d'un paquet (EOP). L'EOP est signalé en faisant passer D+ et D- en état SE0 pour deux temps bit, suivi par le passage des lignes en état J pour un temps bit. La transition de SEO en état J définit la fin du paquet au niveau du récepteur. L'état J est affirmé pour un temps bit puis, les deux pilotes de sortie D+ et D- sont placés dans leur état de grande impédance. Les résistances de terminaison du bus maintiennent le bus dans l'état de repos. La Figure 7-30 illustre la signalisation pour le début et la fin d'un paquet.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Bus Idle	Bus à l'état de repos
First Bit of Packet	Premier bit du paquet
Last Bit of packet	Dernier bit du paquet
SE0 portion of EOP	Portion SE0 de l'EOP
Bus Driven to J State at end of EOP	Bus amené à l'état J à la fin de l'EOP
Bus Floats	Nombres en virgule flottante de bus

Figure 7-30. Niveaux de tension des paquets à basse vitesse/vitesse intégrale

7.1.7.4.2 Signalisation à grande vitesse

L'état de repos à grande vitesse correspond à l'instant où les deux lignes se trouvent nominalement au niveau de GND.

La source du paquet signale le début du paquet (SOP) en mode à grande vitesse en faisant passer les lignes D+ et D- de l'état de repos à grande vitesse à l'état K. Ce K représente le premier symbole du modèle SYNC (séquence NRZI KJKJKJKJ KJKJKJKJ KJKJKJKJ KJKJKJKK), comme décrit dans la Section 7.1.10.

La fin du paquet à grande vitesse (EOP) débute par une transition qui va du dernier symbole avant l'EOP au symbole opposé. Ce symbole opposé représente le premier symbole dans le modèle EOP (NRZ 01111111 avec le bourrage désactivé), comme décrit dans la Section 7.1.13.2. Après l'achèvement du modèle EOP, le pilote cesse d'injecter du courant dans la ligne D+ ou D- et les lignes retournent à l'état de repos à grande vitesse. Le SOF EOP à grande vitesse constitue un cas spécial. Ce SOF EOP comprend 40 symboles sans une transition (plutôt que 8 pour un paquet non SOF).

Le fait que le premier symbole dans le modèle EOP force une transition simplifie le processus de détermination, précisément parce qu'il s'agit du dernier bit dans le paquet avant le délimiteur EOP.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.1.7.5 Signalisation de réinitialisation

Un hub signale une réinitialisation au port orienté en aval en conduisant un SE0 jusqu'au niveau du port. Une fois la réinitialisation supprimée, l'appareil se trouve en état par défaut (voir la Section 9.1).

La signalisation de réinitialisation peut être générée sur un port de hub ou de contrôleur de l'hôte par le biais d'une requête émise par le logiciel du système USB. La signalisation de réinitialisation doit être menée pendant un minimum de 10 ms (TDRST). Après la réinitialisation, le port du hub effectue une transition vers l'état Activé (voir la Section 11.5).

En tant qu'exigence supplémentaire, les contrôleurs d'hôte et le logiciel du système USB doivent garantir que les réinitialisations émises vers les ports racines sont assez longues pour écraser chacune des tentatives de reprise concurrente effectuées par les appareils en aval. Il est nécessaire que les réinitialisations émises à partir des ports racines aient une durée d'au moins 50 ms (TDRSTR). Il n'est pas exigé que ce soit 50 ms de signalisation continue de réinitialisation. Toutefois, si une réinitialisation n'est pas continue, l'intervalle(s) entre la signalisation de réinitialisation doit être inférieur à 3 ms (TRHRSI) et la durée de chaque assertion SE0 doit être d'au moins 10 ms (TDRST).

Un appareil opérant en mode basse vitesse/vitesse intégrale qui observe pendant plus de 2,5 µs un SE0 sur son port orienté en amont (TDETRST) peut traiter ce signal en tant qu'une réinitialisation. La réinitialisation doit avoir pris effet avant la fin de la signalisation.

Les hubs propagent le trafic vers un port nouvellement réinitialisé une fois que le port est en état Activé. L'appareil relié à ce port doit reconnaître cette activité de bus et ne pas entrer en mode d'interruption.

Les hubs doivent être en mesure d'accepter toutes les requêtes d'un hub et les appareils doivent être capables d'accepter une requête SetAddress() (voir la Section 11.24.2 et à la Section 9.4, respectivement) après le temps de récupération sur réinitialisation, c'est-à-dire 10 ms (TRSTRCY) suivant la suppression de la réinitialisation. Tout manquement dans l'acceptation de cette requête peut entraîner la non-reconnaissance par l'appareil du logiciel du système USB. Les hubs et les appareils doivent achever les commandes dans les temps spécifiés dans le Chapitre 9 et le Chapitre 11.

La réinitialisation doit faire sortir l'appareil de l'état d'interruption.

Il est nécessaire qu'un appareil capable de fonctionner à grande vitesse puisse être réinitialisé dans les états Alimenté, Par défaut, Adresse, Configuré ou Interrompu illustrés dans la Figure 9-1. La signalisation de réinitialisation est compatible avec la réinitialisation à basse vitesse/vitesse intégrale. La signalisation de réinitialisation est compatible avec une réinitialisation à basse vitesse/vitesse intégrale. Cela signifie qu'un hub doit réussir la réinitialisation de tout appareil (même les appareils USB 1.X) et un appareil doit pouvoir être réinitialisé avec succès par n'importe quel hub (même les hubs USB1.X).

Si, et uniquement si, un appareil à même d'opérer à grande vitesse est réinitialisé par un hub à même de fonctionner à grande vitesse, dont la grande vitesse n'a pas été interdite, le hub et l'appareil doivent fonctionner dans l'état par défaut en mode de signalisation à grande vitesse, à la fin de la réinitialisation. Le registre de l'état du port du hub doit indiquer que le port est en mode de signalisation à grande vitesse. Cette exigence est satisfaite en ayant cet appareil et ce hub engagés dans un protocole de niveau bas au cours de la période de signalisation de la réinitialisation. Le protocole est défini de manière à ce que les appareils USB 1.X ne soient pas dérangés dans leurs comportements de réinitialisation habituels.

Note: Parce que le port orienté en aval n'est pas en état de transmission au cours du protocole de réinitialisation, les niveaux de signalisation de Chirp à grande vitesse ne provoquent pas la détection de déconnexion. (Voir 7.1.7.3 et 11.5.1.7.)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Protocole de réinitialisation pour les hubs et appareils capables de fonctionner à grande vitesse

1. Le hub effectue une vérification pour s'assurer que l'appareil relié ne fonctionne pas à basse vitesse. (Un appareil à basse vitesse ne peut pas prendre en charge les opérations à grande vitesse. Si le hub détermine qu'il est connecté à un appareil à basse vitesse, il ne conduit pas le protocole de détection à grande vitesse suivant au cours de la réinitialisation.)
2. Le hub conduit le SE0. Dans cette description du protocole de réinitialisation et de l'établissement de liaison de détection à grande vitesse, le début de SE0 est référé sous forme de temps T0.
3. L'appareil détecte l'assertion de SE0.
   a) Si l'appareil est réinitialisé à partir du mode d'interruption, il débute alors un établissement de liaison de détection à grande vitesse après la détection de SE0, pour non moins de 2,5 µs ($T_{FILTSE0}$). Vu que l'oscillateur d'horloge est en général désactivé dans un appareil interrompu, la détection de SE0 entraîne le redémarrage de l'oscillateur. L'horloge doit être utilisable (mais pas nécessairement réglée précisément à 500 ppm) à temps pour détecter le Chirp du hub à grande vitesse, de la manière décrite dans l'Etape 8.
   b) Si l'appareil est réinitialisé à partir de l'état à vitesse intégrale non interrompu, il débute alors un établissement de liaison de détection à grande vitesse, après la détection de SE0, pour non moins que 2,5 µs et pour pas plus que 3,0 ms ($T_{WTRSTFS}$).
   c) Si l'appareil est réinitialisé à partir d'un état à grande vitesse non suspendu, il doit alors patienter non moins que 3,0 ms et pas plus de 3,125 ms (T_{WTREV}) avant de revenir au mode à vitesse intégrale. Le retour au mode à vitesse intégrale est accompli en retirant la terminaison à grande vitesse et en reconnectant la résistance de rappel à la source D+. L'appareil contrôle l'état du bus et recherche le SE0 (réinitialisation par opposition à interruption), pendant pas moins de 100 µs et pendant pas plus que 875 µs ($T_{WTRSTHS}$) après le début du retour au mode à vitesse intégrale. Si un SE0 (réinitialisation) est détecté, l'appareil commence alors à établir une liaison de détection.

Etablissement d'une liaison de détection à grande vitesse (non effectuée si un appareil à basse vitesse a été détecté par le hub):

Note: Dans l'établissement de liaison suivante, il faut que le hub et l'appareil détectent le J et le K d'un Chirp d'une durée minimum spécifiée. Il convient, et est fortement recommandé, que les "écarts" dans ces signaux Chirp, de temps aussi courts que 16 bits à grande vitesse, fassent redémarrer le chronomètre de durée.

4. L'appareil à grande vitesse maintient la connexion de la résistance de rappel à la source D+, laisse les terminaisons à grande vitesse désactivées et conduit le courant de signalisation à grande vitesse dans la ligne D-. Cela crée un K de Chirp sur le bus. Le Chirp de l'appareil doit durer au moins 1,0 ms (T_{UCH}) et doit se terminer 7,0 ms (T_{UCHEND}) au maximum après le T0 du temps de réinitialisation à grande vitesse.
5. Le hub doit détecter le Chirp de l'appareil après avoir constaté l'assertion d'un K de Chirp pendant au moins 2,5 µs (T_{FILT}). Si le hub ne détecte pas de Chirp de l'appareil, il doit continuer l'assertion de SE0 jusqu'à la fin de la réinitialisation.
6. Pas plus de 100 µs (TWTDCH) après que le bus ait quitté l'état K de Chirp, le hub doit commencer à envoyer une séquence alternative de K et de J de Chirp. Il ne doit pas exister d'états de repos sur le bus entre le J et le K. Cette séquence doit se poursuivre jusqu'à une durée (TDCHSE0) inférieure ou égale à 500 s avant et supérieure ou égale à 100 µs après la fin de la réinitialisation. (Cela garantit que le bus reste actif en évitant que l'appareil entre dans l'état d'interruption à grande vitesse.) Chaque K de Chirp et J de Chirp individuel doit durer au moins 40 µs et pas plus que 60 µs (TDCHBIT).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7. Après l'achèvement de la séquence de Chirp du hub, le hub affirme un SE0 jusqu'à la fin de la réinitialisation. A la fin de la réinitialisation, le hub doit effectuer une transition vers l'état Activé à grande vitesse sans entraîner aucune transition sur les lignes de données.
8. Une fois que l'appareil a terminé son Chirp, il recherche le Chirp du hub à grande vitesse. Au minimum, il est nécessaire que l'appareil voie le Chirp séquentiel K-J-K-J-K-J, afin de pouvoir détecter un Chirp de hub valide. Chaque K de Chirp et J de Chirp individuel doit être détecté pendant au moins 2,5 µs (TFILT).
   a) Si l'appareil détecte le Chirp séquentiel K-J-K-J-K-J, donc avant 500 µs (TWTHS) après la détection, il faut que l'appareil déconnecte la résistance de rappel à la source D+, qu'il active les terminaisons à grande vitesse et qu'il entre dans l'état à grande vitesse par défaut.
   b) Si l'appareil n'a pas détecté le Chirp séquentiel K-J-K-J-K-J dans une période comprise entre 1,0 ms et 2,5 ms (TWTFS) après avoir complété son propre Chirp, l'appareil est alors tenu de revenir à l'état à vitesse intégrale par défaut et de patienter jusqu'à la fin de la réinitialisation.

7.1.7.6 Interruption

Tous les appareils doivent prendre en charge l'état d'interruption. Les appareils peuvent passer à l'état d'interruption depuis n'importe quel état actif. Ils commencent la transition vers l'état d'interruption une fois avoir constaté un état constant de repos sur leurs lignes de bus orientées en amont pendant plus de 3,0 ms. L'appareil doit effectivement être suspendu, en appelant uniquement le courant suspendu du bus après une durée d'inactivité maximale de 10 ms sur l'ensemble de ses ports. Toute activité du bus sur le port orienté en amont fait sortir un appareil de l'état d'interruption. En l'absence de trafic autre du bus, le jeton SOF (voir la Section 8.4.3) apparaît une fois par (micro)trame, afin d'éviter l'interruption des appareils à vitesse intégrale/grande vitesse. En l'absence de trafic à basse vitesse, les appareils à basse vitesse observent au moins un maintien en réserve active (défini dans le Tableau 7-2) dans chaque trame dans laquelle un SOF apparaît, ce qui permet d'éviter leur interruption. Les hubs génèrent ce maintien en réserve active, comme décrit dans la Section 11.8.4.1.

En état d'interruption, un appareil doit continuer à alimenter sa résistance de rappel à la source D+ (vitesse intégrale/grande vitesse) ou D- (basse vitesse) afin de maintenir l'interruption, de manière à ce que le hub en amont puisse continuer à conserver l'état de connectivité correct pour l'appareil.

Exigences supplémentaires pour les appareils capables de fonctionner à grande vitesse

Dans l'optique d'un appareil fonctionnant en mode à grande vitesse, une réinitialisation et une interruption sont initialement indiscernables. Par conséquent, la première partie de la réponse de l'appareil est similaire à celle d'une réinitialisation. Lorsqu'un appareil fonctionnant en mode à grande vitesse détecte que les lignes de données se sont trouvées en état de repos à grande vitesse pendant au moins 3,0 ms, il doit revenir à la configuration à vitesse intégrale dans les 3,125 ms (TWTREV) après le début de l'état de repos. Le retour au mode à vitesse intégrale est accompli en déconnectant ses résistances de terminaison et en reconnectant sa résistance de rappel à la source D+. Pas plus tôt que 100 µs et pas plus tard que 875 µs (TWTRSTHS) après le retour à l'état à vitesse intégrale, l'appareil doit contrôler l'état de la ligne. Si l'état est un J à vitesse intégrale, l'appareil continue le processus d'interruption. (Le SE0 doit avoir indiqué que le port orienté en aval est en train de mener une réinitialisation et l'appareil doit être entré dans l'état "Etablissement d'une liaison de détection à grande vitesse", comme décrit dans la Section 7.1.7.5.)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Un appareil ou un port orienté en aval, dont les opérations à grande vitesse ont été interrompues, effectue réellement une transition vers une signalisation à vitesse intégrale au cours du processus d'interruption, mais doit se souvenir qu'il fonctionnait en mode à grande vitesse lorsqu'il a été interrompu. Lors de la reprise, l'appareil ou l'émetteur-récepteur orienté en aval doit retourner au mode à grande vitesse, comme indiqué dans la Section 7.1.7.7, sans qu'une réinitialisation ne soit nécessaire.

7.1.7.6.1 Interruption globale

L'interruption globale est utilisée lorsqu'aucune communication n'est souhaitée nulle part sur le bus et que le bus au complet est placé en état d'interruption. L'hôte signale le début d'une interruption globale en cessant toutes ses transmissions (y compris le jeton SOF). Comme chacun des appareils sur le bus reconnaît que le bus est en état de repos pour l'intervalle de temps approprié, ils se placent en état d'interruption.

Après 3,0 ms d'état de repos continu, un émetteur-récepteur orienté en aval, opérant à grande vitesse doit retourner à la configuration de repos à vitesse intégrale (terminaisons à grande vitesse désactivées), mais il n'active pas la détection de déconnexion à vitesse intégrale avant passé 1,0 ms. Cela pour s'assurer que l'appareil retourne en état de repos à vitesse intégrale avant l'activation de la détection de déconnexion à vitesse intégrale, et éviter ainsi une détection de déconnexion non souhaitée. Après la réactivation du mécanisme de détection de déconnexion à vitesse intégrale, le hub poursuit le processus d'interruption.

7.1.7.6.2 Interruption sélective

Les segments du bus peuvent être interrompus de manière sélective en envoyant la commande SetPortFeature(PORT_SUSPEND) au port du hub auquel ce segment est relié. Le port suspendu bloque les activités vers le segment du bus suspendu et les appareils sur ce segment entrent en état d'interruption après le délai approprié, comme décrit ci-dessus.

Lorsqu'un port orienté en aval, opérant en mode à grande vitesse reçoit la commande SetPortFeature(PORT_SUSPEND), le port retourne immédiatement en état de repos à vitesse intégrale et bloque toutes les activités vers le segment suspendu. La détection de déconnexion à vitesse intégrale est désactivée jusqu'à ce que le port soit resté dans l'état de repos à vitesse intégrale pendant 4,0 ms. Cela évite une détection de déconnexion non souhaitée. Cela permet d'empêcher une détection de déconnexion involontaire. Après la réactivation du mécanisme de détection de déconnexion à vitesse intégrale, le hub poursuit le processus d'interruption.

La Section 11.5 décrit l'état d'interruption du port et son interaction avec le diagramme d'état du port. L'interruption est décrite plus en détail dans la Section 11.9.

7.1.7.7 Reprise

Si un appareil se trouve en état d'interruption, son fonctionnement est repris lorsqu'une signalisation de non-repos est reçue sur le port orienté en amont. En outre, l'appareil peut signaler au système de reprendre ses opérations, si sa fonctionnalité de réveil à distance a été activée par le logiciel du système USB. La signalisation de reprise est utilisée par l'hôte ou un appareil pour ramener un segment de bus interrompu à une condition active. Les hubs jouent un rôle important dans la propagation et la génération des signaux de reprise. La description suivante souligne les grandes lignes d'une séquence de reprise globale générale. Une description complète de la séquence de reprise, les cas spéciaux causés par une interruption sélective et le rôle du hub sont fournis en 11.9.

L'hôte peut signaler une reprise (T_{DRSMDN}) à tout moment. Il doit envoyer le signal de reprise pendant au moins 20 ms, puis terminer la signalisation de reprise de l'une ou de deux manières, suivant la vitesse à laquelle son port fonctionnait lorsqu'il a été interrompu. Si le port opérait à basse vitesse/vitesse intégrale lors de son interruption, la signalisation de reprise doit être terminée à l'aide de l'EOP à basse vitesse normalisé (deux temps bit à basse vitesse de SE0 suivi par un J). Si le port fonctionnait à grande vitesse lors de son interruption, la signalisation de reprise doit être achevée avec une transition vers l'état de repos à grande vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Les 20 ms de signalisation de reprise garantissent que tous les appareils du réseau activés pour observer la reprise sont réveillés. La connectivité établie par la signalisation de reprise est détruite par la fin de la reprise, qui prépare les hubs afin qu'ils fonctionnent à nouveau . Après la reprise du bus, l'hôte doit commencer à envoyer le trafic du bus (au moins le jeton SOF), dans les 3 ms à partir du début de l'état de repos, afin d'empêcher le système de revenir en état d'interruption.

Un appareil avec une fonctionnalité de réveil à distance peut ne pas générer un signal de reprise, à moins que le bus soit resté, sans interruption, en état de repos pendant 5 ms (TWTRSM). Cela permet aux hubs d'entrer dans leur état d'interruption et se préparer à propager le signal de reprise. L'appareil de réveil à distance doit maintenir le signal de reprise pendant au moins 1 ms, mais au maximum pendant 15 ms (TDRSMUP). A la fin de cette période, l'appareil arrête d'actionner le bus (met ses pilotes en état de haute impédance et n'entraîne pas le bus vers l'état J).

Si l'amont du hub d'un appareil de réveil à distance est interrompu, il propage le signal de reprise vers son port orienté en amont et vers tous ses ports activés, orientés en aval, y compris le port qui, à l'origine, a signalé la reprise. Lorsqu'un hub propage le signal de reprise à partir d'un appareil en aval, il peut effectuer une transition de l'état de repos à l'état K avec un temps de montée plus rapide que celui normalement autorisé. Le hub doit débuter cette rediffusion (TURSM) du signal de reprise dans 1 ms après la réception de la reprise d'origine. Le signal de reprise se propage en amont de cette manière, jusqu'à ce qu'il atteigne l'hôte ou un hub non interrompu (voir la Section 11.9), ce qui reflète l'aval de reprise et la prise de contrôle de la synchronisation de reprise. Ce hub est appelé le hub de contrôle. Les hubs intermédiaires (hubs placés entre l'initiateur de reprise et le hub de contrôle) entraînent pendant au moins 1 ms le signal de reprise (TDRSMUP) vers leur port orienté en amont, laps de temps pendant lequel ils l'emmènent aussi sur les ports activés, orientés en aval. Un hub intermédiaire arrête la diffusion du signal de reprise sur le port orienté en amont et inverse le sens de connectivité, (d'aval en amont) dans les 15 ms suivant la première reprise affirmée sur son port orienté en amont. Lorsque tous les hubs intermédiaires ont inversé le sens de connectivité, le signal de reprise est entraîné vers tous les hubs de contrôle en passant par tous les hubs intermédiaires et tous les ports activés. Le hub de contrôle doit rediffuser le signal de reprise dans les 1 ms (TURSM) et s'assurer que la reprise est signalée pendant au moins 20 ms (TDRSMDN). Le hub peut alors fonctionner normalement en terminant le processus de reprise, comme décrit ci-dessus.

Le logiciel du système USB doit offrir un temps de récupération sur reprise de 10 ms (TRSMRCY), au cours duquel il n'essaie pas d'accéder à un appareil connecté au segment de bus affecté (juste activé).

Les connexions et déconnexions du port peuvent aussi entraîner l'envoi par un hub, du signal de reprise et réveiller le système. Ces évènements n'entraînent la transmission par un hub d'un signal de reprise que si le hub a été activé en tant que source de réveil à distance. Voir la Section 11.4.4 pour plus de détails.

Voir la Section 7.2.3 pour une description de la régulation de puissance au cours d'une interruption et d'une reprise.

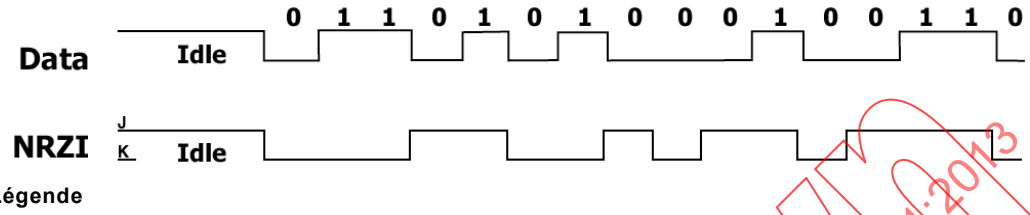
Si le port du hub et de l'appareil fonctionnaient à grande vitesse avant l'interruption, il est requis qu'ils se "rappellent" avoir précédemment opéré en mode à grande vitesse et ils doivent effectuer une transition vers les opérations à grande vitesse, sans arbitrage et cela dans deux temps bit à basse vitesse pendant la transition de K à SE0. Les chronomètres d'inactivité doivent être démarrés par deux temps bit à basse vitesse après la transition de K à SE0. A noter que la transition de SE0 à J, qui se produirait normalement à la fin de la signalisation de la reprise à vitesse intégrale, est omise si la liaison opérait à grande vitesse au moment où elle a été interrompue.

Il faut que l'hôte envoie SOF à temps afin d'éviter l'interruption de l'arborescence à grande vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.1.8 Codage/décodage des données

L'USB emploie le système de codage des données NRZI lorsqu'il transmet des paquets. Dans le code NRZI, un "1" signifie aucun changement dans le niveau et un "0" indique un changement dans un niveau. La Figure 7-31 illustre un flux de données et son équivalence NRZI. Le niveau haut représente l'état J sur les lignes de données dans ces figures et les figures ultérieures, qui illustrent le système de codage NRZI. Une chaîne de zéros entraîne le basculement de chaque temps bit par les données NRZI. Une chaîne de uns engendre de longues périodes sans aucune transition dans les données.



Légende

Anglais	Français
Data	Données
Idle	Inactif

Figure 7-31. Encodage de données NRZI

7.1.9 Insertion de bits

Afin de garantir des transitions de signaux adéquates, l'insertion de bits est employée par l'appareil de transmission lors de l'envoi d'un paquet sur USB (voir la Figure 7-32 et la Figure 7-34). Un zéro est inséré après six uns consécutifs dans le flux de données avant que les données ne soient codées en NRZI. Cela afin de forcer une transition dans le flux de données NRZI. Cela fournit à la logique du récepteur une transition des données, au moins une fois tous les sept temps bit, afin de garantir le verrouillage des données et de l'horloge. L'insertion binaire est activée en commençant par le modèle Sync. La donnée "un" qui termine le modèle Sync est spécifiée en tant que premier un dans une séquence. L'insertion binaire par l'émetteur est toujours imposée, sauf au cours de l'EOP à grande vitesse. Si requis par les règles d'insertion binaire, un bit zéro est inséré, même s'il s'agit du dernier bit avant le signal de fin de paquet (EOP).

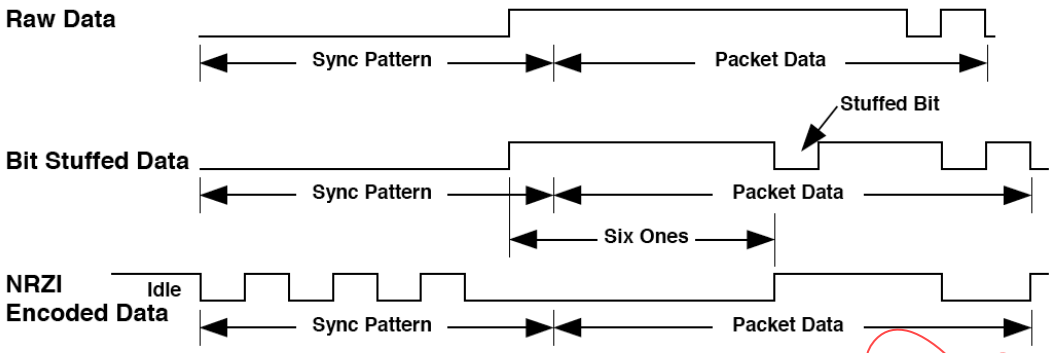
Le récepteur doit décoder les données NRZI, reconnaître les bits insérés et les rejeter.

7.1.9.1 Vitesse intégrale/basse vitesse

La signalisation à vitesse intégrale/basse vitesse utilise sans exception l'insertion de bits au travers du paquet. Si le récepteur en détecte sept consécutifs n'importe où dans le paquet, une erreur d'insertion de bits s'est alors produite et il convient d'ignorer le paquet. L'intervalle de temps juste avant l'EOP constitue un cas spécial. Le dernier bit de données avant l'EOP peut être étendu par les dérivations de commutation du hub. Cela est connu sous le nom de dribble et peut mener au cas illustré dans la Figure 7-33 qui montre où le dribble introduit un sixième bit qui ne requiert pas une insertion binaire. Toutefois, le récepteur doit accepter un paquet pour lequel il existe jusqu'à six temps bit pleins au niveau du port, sans transition, précédant l'EOP.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

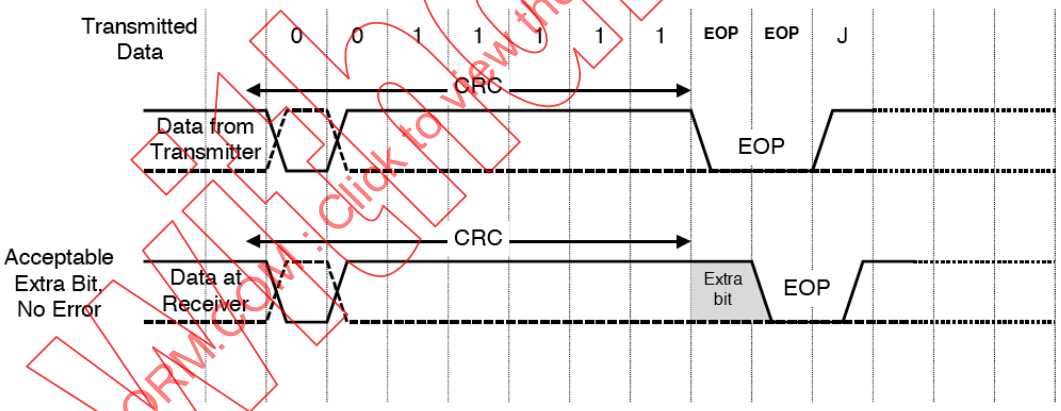
Data Encoding Sequence:



Légende

Anglais	Français
Data Encoding Sequence:	Séquence de codage des données:
Raw Data	Données brutes
Sync Pattern	Modèle SYNC
Packet Data	Données de paquet
Bit Stuffed Data	Données à insertion binaire
Sync Pattern	Modèle SYNC
Stuffed Bit	Bit inséré
Packet Data	Données de paquet
Six Ones	Six uns
NZRI Encoded Data	Données codées NZRI
Idle	Inactif
Sync Pattern	Modèle SYNC
Packet Data	Données de paquet

Figure 7-32. Insertion de bits

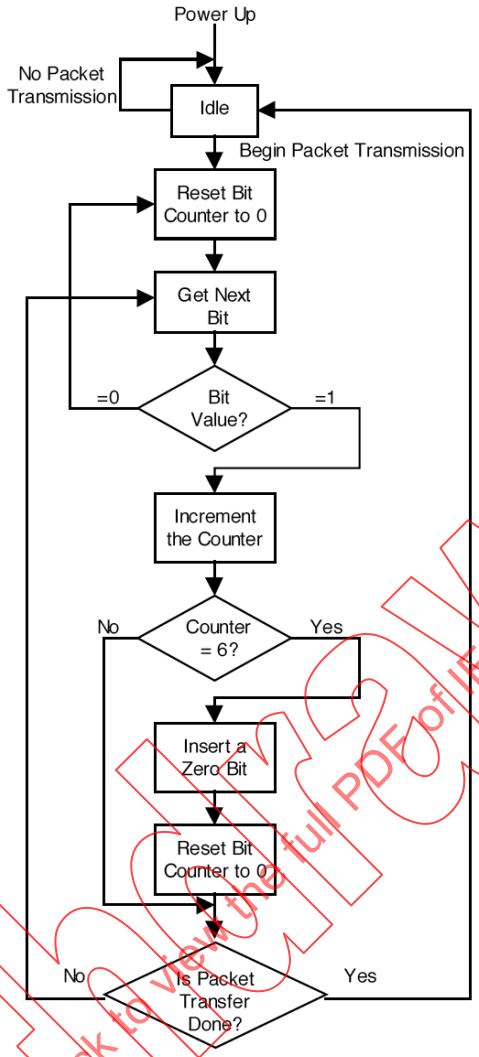


Légende

Anglais	Français
Transmitted Data	Données transmises
Data from Transmitter	Données provenant du transmetteur
Acceptable Extra Bit, No Error	Bit supplémentaire acceptable, aucune erreur
Data at Receiver	Données au niveau du récepteur
Extra bit	Bit supplémentaire

Figure 7-33. Illustration du bit supplémentaire précédant l'EOP (vitesse intégrale/basse vitesse)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Power Up	Mise sous tension
No Packet Transmission	Aucune transmission de données
Idle	Inactif
Begin Packet Transmission	Démarrer la transmission de données
Reset Bit Counter to 0	Réinitialiser le compteur de bits à zéro
Get Next Bit	Obtenir le bit suivant
Bit Value?	Valeur de bit?
Increment the Counter	Incrémenter le compteur
Counter = 6?	Compteur = 6?
Insert a Zero Bit	Insérer un bit zéro
Reset Bit Counter to 0	Réinitialiser le compteur de bits à zéro
Is Packet Transfer Done?	Le transfert du paquet est-il terminé?
Yes	Oui
No	Non

Figure 7-34. Schéma relatif à l'insertion binaire

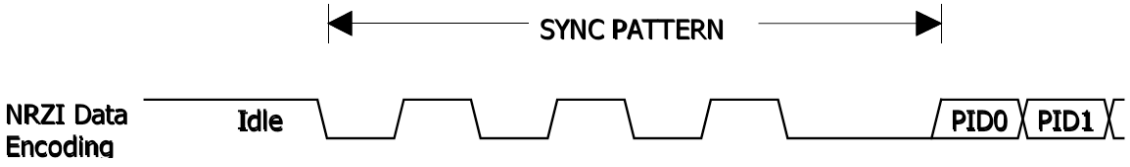
7.1.9.2 Grande vitesse

La signalisation à grande vitesse utilise l'insertion binaire au travers du paquet, à l'exception des erreurs d'insertion binaire intentionnelles utilisées dans l'EOP à grande vitesse, comme décrit dans cette Section 7.1.13.2.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

7.1.10 Modèle Sync

Le modèle SYNC utilisé pour la transmission à basse/vitesse intégrale doit consister en 3 paires KJ suivies par 2 K pour un total de huit symboles. La Figure 7-35 illustre le modèle binaire NRZI qui précède chaque paquet à basse vitesse/vitesse intégrale.



Légende

Anglais	Français
SYNC PATTERN	MODÈLE SYNC
NZRI Data Encoding	Codage des données NZRI
Idle	Inactif

Figure 7-35. Modèle Sync (basse vitesse/vitesse intégrale)

Il faut que le modèle SYNC utilisé pour la transmission à grande vitesse se compose de 15 paires KJ suivies par 2 K pour un total de 32 symboles. Il est admis que les hubs abandonnent jusqu'à 4 bits à partir du début du modèle SYNC lors de la répétition de paquets. Les hubs ne doivent toutefois pas corrompre les bits répétés du champ SYNC. Ainsi, après avoir été répété par 5 hubs, un champ SYNC de paquet peut être aussi court que 12 bits.

7.1.11 Débit de signalisation des données

Le débit de données à grande vitesse (THSDRAT) est nominalement de 480,00 Mb/s, avec une précision de débit binaire de ± 500 ppm. Pour les hôtes, hubs et fonctions capables d'opérer à grande vitesse, la précision du débit de données requis lors de la transmission à n'importe quelle vitesse, est de ± 0,05 % (500 ppm). Le débit à vitesse intégrale pour de tels hubs et fonctions est TFDRATHS. Le débit à basse vitesse pour de tels hubs est TLDRATHS (une fonction à basse vitesse ne doit pas prendre en charge la grande vitesse).

Le débit de données à vitesse intégrale est nominalement de 12 000 Mb/s. Pour les fonctions qui n'opèrent qu'à vitesse intégrale, le débit de données requis lors de la transmission (TFDRATE) est de 12 000 Mb/s ± 0,25 % (2 500 ppm).

Le débit de données à basse vitesse est nominalement de 1,50 Mb/s. Pour les fonctions qui n'opèrent qu'à basse vitesse, le débit de données requis lors de la transmission (TLDRATE) est de 1,50 Mb/s ±1,5 % (15 000 ppm). Cela permet l'utilisation de résonateurs dans les appareils bon marché à basse vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Les hôtes et hubs doivent être en mesure de recevoir les données en provenance de n'importe quelle source à basse vitesse, vitesse intégrale ou grande vitesse conforme. Les fonctions capables d'opérer à grande vitesse doivent être en mesure de recevoir les données en provenance de n'importe quelle source à vitesse intégrale ou à grande vitesse conforme. Les fonctions qui n'opèrent qu'à vitesse intégrale doivent être en mesure de recevoir les données en provenance de n'importe quelle source à vitesse intégrale conforme. Les fonctions qui n'opèrent qu'à vitesse intégrale doivent être en mesure de recevoir les données en provenance de n'importe quelle source à vitesse intégrale conforme.

Les valeurs de précision ci-dessus incluent une contribution de toutes les sources:

- Précision en fréquence initiale
- Charge capacitive piézo-électrique
- Tension d'alimentation sur l'oscillateur
- Température
- Vieillissement

7.1.12 Intervalle de trame

L'USB définit un intervalle de trame (T_{FRAME}) d'une longueur de 1,000 ms ±500 ns. L'USB définit un intervalle de microtrame (T_{HSFRAM}) d'une longueur de 125,0 µs ±62,5 ns. L'intervalle de (micro)trame est mesuré à partir d'un point quelconque du jeton SOF dans une (micro)trame jusqu'au même point dans le jeton SOF de la (micro)trame suivante.

Vu que le contrôleur d'hôte et les hubs doivent satisfaire aux spécifications en matière de précision de l'horloge à ±0,05 %, ils sont automatiquement conformes aux exigences de l'intervalle de trame sans qu'il soit nécessaire de les régler.

La répétabilité de l'intervalle de trame, T_{RFI} (différence dans l'intervalle de trame entre deux trames successives), doit être inférieure à 0,5 temps bit à vitesse intégrale. La répétabilité de l'intervalle de microtrame, T_{HSRFI} (différence dans l'intervalle de microtrame entre deux microtrames successives, mesuré au niveau de l'hôte) doit être inférieure à 4 temps bit à grande vitesse. Chaque hub peut introduire au maximum 4 bits supplémentaires à grande vitesse d'une gigue de microtrame.

Les hubs et certaines fonctions à vitesse intégrale/grande vitesse doivent suivre l'intervalle de (micro)trame. Ils sont également tenus de disposer de suffisamment de réglages de synchronisation de trame pour compenser leur propre imprécision en matière de fréquence.

7.1.13 Signalisation de la source de données

Cette section couvre les caractéristiques de synchronisation des données produites et envoyées par un port (la source de données). La Section 7.1.14 couvre les caractéristiques de synchronisation des données transmises au travers de la section répéteur d'un hub. Dans cette section, T_{PERIOD} est défini en tant que la période réelle du débit de données qui peut disposer d'une gamme, comme définie dans la Section 7.1.11.

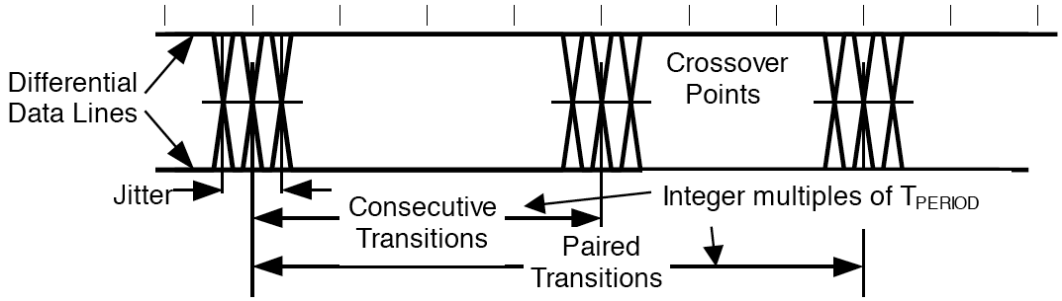
7.1.13.1 Gigue de source de données

Cette section décrit la gigue de source de données admissible maximale pour la signalisation à basse vitesse, à vitesse intégrale et à grande vitesse.

7.1.13.1.1 Gigue de source de données à basse vitesse/vitesse intégrale

La source de données peut disposer d'une certaine variation (gigue) dans la synchronisation des bords des données transmises. Le temps entre un ensemble de transitions de données est le temps de gigue (N * T_{PERIOD})±, où "N" correspond au nombre de bits entre les transitions. La gigue de données est mesurée avec la même charge que celle utilisée pour les temps de montée et de chute maximum et est mesurée aux points de croisement des lignes de données, comme illustré dans la Figure 7-36.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Differential Data Lines	Lignes de données différentielles
Crossover Points	Points de croisement
Jitter	Gigue
Consecutive Transitions	Transitions consécutives
Paired Transitions	Transitions jumelées
Integer multiples of T_{PERIOD}	Multiples entiers de T_{PERIOD}

Figure 7-36. Taxinomie de gigue de données

- Pour les transmissions à vitesse intégrale, le temps de gigue de n'importe quelle transition de données différentielles consécutive doit figurer dans les 2,0 ns et dans ± 1,0 ns pour chaque ensemble de transitions de données différentielles jumelées (JK- jusqu'à la transition JK suivante ou KJ jusqu'à la prochaine transition KJ).
- Pour les transmissions à basse vitesse, le temps de gigue pour n'importe quelle transition de données différentielles consécutive doit se situer dans les ± 25 ns et dans les ± 10 ns pour chaque ensemble de transitions de données différentielles jumelées.

Ces valeurs de gigue incluent les variations de synchronisation dues au délai de la mémoire tampon différentielle et aux discordances dans les temps de chute, à la gigue source de l'horloge interne, au bruit ainsi qu'à d'autres effets aléatoires.

7.1.13.1.2 Gigue de la source de données à grande vitesse

Les données à grande vitesse au sein d'un seul paquet ne doivent pas être transmises avec plus de gigues que ce qui est admis par les diagrammes en œil, définis dans la Section 7.1.2, lorsqu'elles ont été mesurées sur une fenêtre glissante de 480 temps bit à grande vitesse.

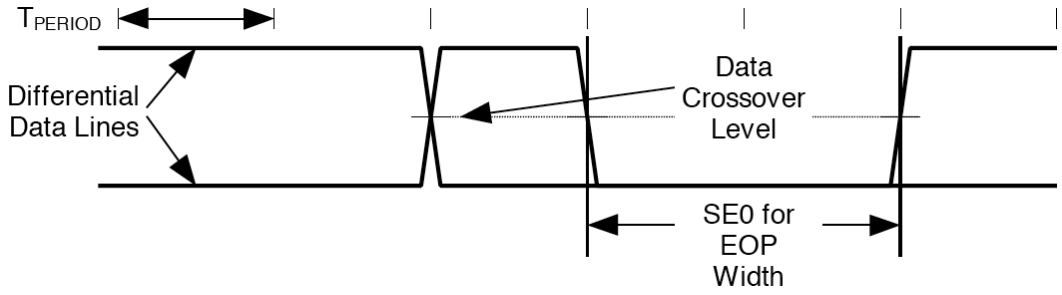
7.1.13.2 Largeur de l'EOP

Cette section décrit la largeur d'EOP à basse vitesse, vitesse intégrale et à grande vitesse.

7.1.13.2.1 EOP à basse vitesse/vitesse intégrale

La largeur du SE0 dans l'EOP est d'environ 2 * TPERIOD. La largeur du SE0 est mesurée avec la même charge que celle utilisée pour les temps de montée et de chute maximum et est mesurée au même niveau que les points de croisement du signal différentiel des lignes de données (voir la Figure 7-37).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Differential Data Lines	Lignes de données différentielles
Data Crossover Level	Niveau de croisement des données
SE0 for EOP Width	SE0 pour la largeur de l'EOP

Figure 7-37. SE0 pour la temporisation de largeur d'EOP

- Pour les transmissions à vitesse intégrale, le SE0 pour la largeur de l'EOP à partir de l'émetteur doit se situer entre 160 ns et 175 ns.
- Pour les transmissions à basse vitesse, le SE0 pour la largeur de l'EOP de l'émetteur doit se situer entre 1,25 μs et 1,50 μs.

Ces fourchettes incluent les variations de synchronisation dues au délai de la mémoire tampon différentielle et aux discordances dans les temps de chute, au bruit ainsi qu'à d'autres effets aléatoires.

Un récepteur doit accepter chaque EOP valide. Il convient que le concepteur du récepteur note que la tension du seuil d'entrée à extrémité unique peut être différente de la tension de croisement différentiel et les transitions SE0 sont en général asynchrones vers l'horloge codée dans le flux NRZI.

- L'EOP à vitesse intégrale peut disposer d'un intervalle réduit à 82 ns (T_{FEOPR}) et un intervalle SEO à basse vitesse peut être aussi bref que 670 ns (T_{LEOPR}).

Un hub peut démanteler une connectivité s'il observe un SE0 T_{FST} ou T_{FST} au moins, suivi par une transition vers l'état J. Un hub doit démanteler la connectivité sur chaque EOP valide.

7.1.13.2.2 EOP à grande vitesse

Dans la signalisation à grande vitesse, une erreur d'insertion binaire est intentionnellement générée afin d'indiquer l'EOP. Un récepteur est tenu d'interpréter chaque erreur d'insertion binaire en tant qu'EOP.

Pour les paquets à grande vitesse, autres que les SOF, il faut que le délimiteur de l'EOP transmis soit un octet NRZ de 01111111 sans insertion de bits. Par exemple, si le dernier symbole précédent le champ d'EOP est un J, cela entraîne l'EOP de KKKKKKKK.

Pour les SOF à haute vitesse, le délimiteur de l'EOP transmis doit être composé de 5 octets NRZ sans insertion de bits, avec 01111111 11111111 11111111 11111111 11111111. Par conséquent, si le dernier bit avant le champ EOP est un J, cela donne lieu à 40 K sur le fil, à l'extrémité duquel les lignes doivent revenir à l'état de repos haute vitesse. Cette longueur d'EOP supplémentaire n'a aucune signification pour le récepteur; elle est utilisée pour la détection de déconnexion, telle qu'elle est abordée dans la Section 7.1.7.3.

Un hub peut ajouter 4 bits aléatoires au maximum à la fin du champ EOP lors de la répétition d'un paquet. Ainsi, après 5 répéteurs, un paquet peut disposer d'un maximum de 20 bits aléatoires supplémentaires, suivant le champ EOP. Toutefois, un hub ne doit pas corrompre aucun des 8 (ou 40 dans le cas d'un SOF) bits requis du champ EOP.

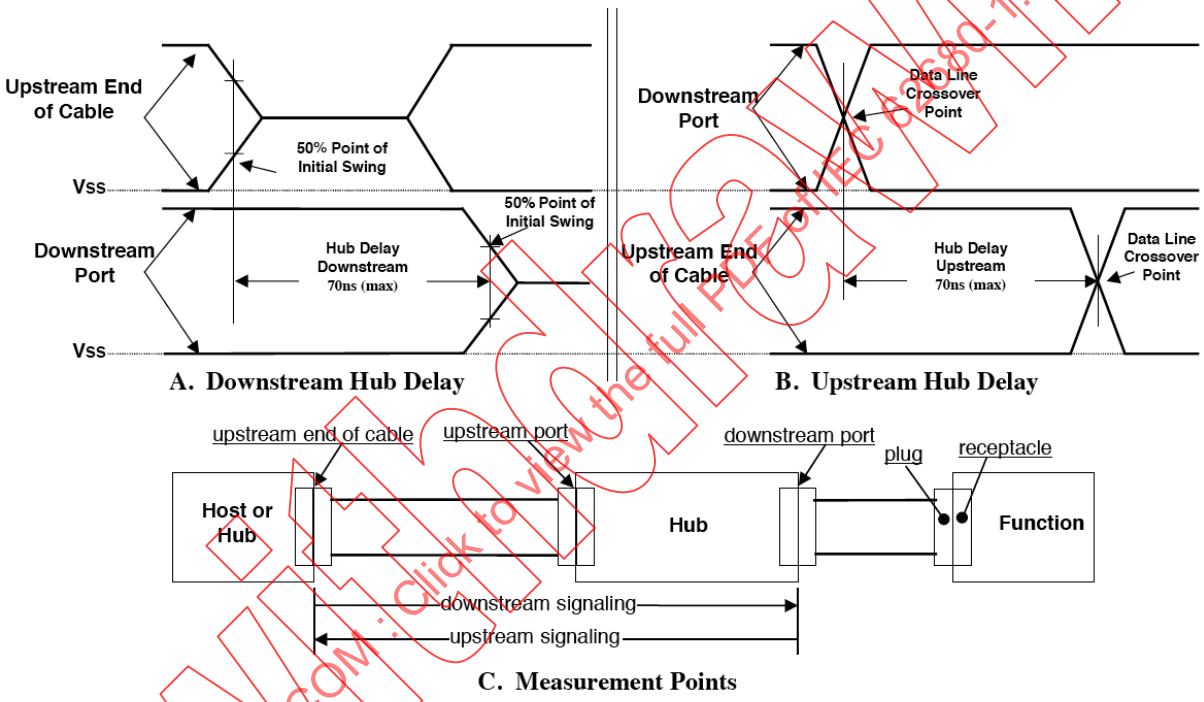
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.1.14 Synchronisations de signalisation d'un hub

Cette section décrit la largeur des synchronisations de signalisation d'un hub à basse vitesse, vitesse intégrale et à grande vitesse.

7.1.14.1 Synchronisations de signalisation d'un hub à basse vitesse/vitesse intégrale

La propagation d'un signal de données différentiel à vitesse intégrale à travers un hub est illustrée à la Figure 7-38. La signalisation en aval est mesurée sans câble connecté au port et avec la charge utilisée pour la mesure des temps de montée et de chute. Le délai total au travers du câble et de l'électronique du hub en amont doit être de 70 ns (T_{HDD1}) au maximum. Si le hub dispose d'un câble USB détachable, le délai (T_{HDD2}), au travers de l'électronique du hub et de la ligne de transmission connexe doit alors être de 44 ns pour permettre un délai maximum du câble de 26 ns (T_{FSCBL}). Le délai au travers de ce hub est mesuré dans les deux sens, en aval et en amont, comme illustré dans la Figure 7-38B, à partir de la ligne de données au niveau du port d'entrée, jusqu'au croisement de la ligne de données au niveau du port de sortie.



Légende

Anglais	Français
Upstream End of Cable	Extrémité en amont du câble
50% Point of Initial Swing	50 % du point de la variation initiale
Downstream Port	Port en aval
Data Line Crossover Point	Point de croisement des lignes de données
Hub Delay Downstream 70ns (max)	Délai de hub en aval 70 ns (max)
Upstream End of Cable	Extrémité en amont du câble
Hub Delay Upstream 70ns (max)	Délai de hub en amont 70 ns (max)
Downstream Hub Delay	Délai de hub en aval
Upstream Hub Delay	Délai de hub en amont
upstream end of cable	extrémité en amont du câble
upstream port	port en amont
plug	fiche
receptacle	réceptacle
Host or Hub	Hôte ou hub
Hub	Hub
Function	Fonction
downstream signaling	Signalisation en aval
upstream signaling	Signalisation en amont
Measurement Points	Points de mesure

Figure 7-38. Délai de propagation du hub de signaux différentiels à vitesse intégrale

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Le délai de propagation à basse vitesse pour les signaux différentiels est mesuré de façon similaire à la signalisation à vitesse intégrale. Le délai maximum du hub à basse vitesse est de 300 ns (TLHDD). Cela permet un délai de propagation de la mémoire tampon à basse vitesse et des temps de montée et de chute plus lents. Cela offre également du temps au hub pour resynchroniser les données à basse vitesse dans le sens amont.

Lorsque le hub fait office de répéteur, il doit reproduire précisément le signal à vitesse intégrale reçu sur ses sorties. Cela signifie que pour les signaux différentiels, les délais de propagation d'une transition de l'état J à K doivent correspondre étroitement aux délais de transition de l'état K à J. Pour la propagation à vitesse intégrale, la différence maximale autorisée entre ces deux délais (THDJ1) (voir la Figure 7-38 et la Figure 7-52) pour un hub plus le câble est de ± 3,0 ns. De manière similaire, la différence dans le délai entre chacune des deux transitions J à K et K à J à travers un hub (THDJ2) doit être inférieure à ± 1,0 ns. Pour la propagation à basse vitesse dans le sens aval, la gigue correspondante admissible (TLDHJ1) est de ± 45 ns et (TLDHJ2) ± 15 ns, respectivement. Pour la propagation à basse vitesse en amont, la gigue admissible est de ± 45 ns dans les deux cas (TLUHJ1 et TLUHJ2).

La dérivation qui peut être introduite dans la transition de l'état K à l'état de repos au niveau du SOP (TFSOP et TLSOP) constitue une exception à ce cas (voir la Section 7.1.7.4). Dans ce cas, le délai pour le port opposé inclut le temps nécessaire à l'activation de la mémoire tampon de sortie. Toutefois, il convient que les délais correspondent étroitement au délai normal du hub et la différence maximum dans le délai supplémentaire sur une transition normale de J à K est de ± 5,0 ns. Cela limite la distorsion maximale du premier bit dans le paquet.

Note: Compte tenu de cette distorsion de la transition SOP, relative à la prochaine transition de l'état K à J, il convient que le premier bit du champ SYNC ne soit pas utilisé pour synchroniser le récepteur avec le flux de données.

L'EOP doit être diffusé à travers un hub, de manière similaire à la signalisation différentielle. Le délai de propagation pour détecter un SE0 ne doit pas être inférieur au délai le plus important de transition des données différentielles J à K ou K à J (afin d'éviter la troncature du dernier bit de données dans un paquet), mais ne doit pas être supérieur de plus de 15 ns que le délai différentiel le plus élevé à vitesse intégrale et de 200 ns à basse vitesse (pour empêcher la création d'une erreur d'insertion binaire à la fin du paquet). Les délais EOP sont illustrés dans la Figure 7-53.

Parce que les niveaux de détection pour l'état SE0 ne figurent pas sur le point médian de la variation du signal, la largeur de l'état SE0 est changée lorsqu'il traverse chaque hub. Un hub peut ne pas modifier la largeur d'un état SE0 dans l'EOP à vitesse intégrale de plus de ± 15 ns (TFHESK), comme mesurée par la différence entre les délais du flanc avant et du flanc arrière de l'état SE0 (voir Figure 7-53). Un SE0 d'un appareil à basse vitesse dispose de longs temps de montée et de chute et est sujet à une dérivation plus importante, mais ces conditions existent uniquement sur le câble qui relie l'appareil à basse vitesse et le port. Par la suite, le signal utilise les mémoires tampons à vitesse intégrale et leurs temps de montée et de chute plus rapides. Le SEO de l'appareil à basse vitesse ne peut pas être modifié de plus de ± 300 ns (TLHESK) lors de son passage à travers le hub auquel l'appareil est connecté. Ce temps permet un certain conditionnement du signal dans l'émetteur-récepteur à basse vitesse, afin de réduire sa sensibilité au bruit.

7.1.14.2 Synchronisation de la signalisation du hub à grande vitesse

Lorsqu'un hub agit en tant que répéteur pour les données à grande vitesse, le délai du hub (THSHDD) ne doit pas excéder 36 temps bit à grande vitesse, plus 4 ns (les délais de trace autorisés pour le circuit imprimé du hub). Ce délai est mesuré entre le dernier bit du champ SYNC, au niveau du connecteur d'entrée, et le dernier bit du champ SYNC au niveau du connecteur de sortie.

Un hub-répéteur doit resynchroniser numériquement les données mises en mémoire tampon, afin que la gigue cumulative ne puisse pas être autorisée (dans un seul paquet), lors du

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

passage d'un paquet à grande vitesse à travers les multiples étapes du répéteur. Dans un seul paquet, la gigue ne doit pas dépasser les modèles de diagramme en œil, définis dans la Section 7.1.2, sur une fenêtre glissante de 480 temps bit à grande vitesse.

Du fait du processus de synchronisation des données, le délai de propagation d'un hub-répéteur peut varier au plus de 5 temps bit à grande vitesse (THSHDV). Le délai qui inclut cette variation autorisée ne doit pas excéder 36 temps bit à grande vitesse, plus 4 ns. (Cela permet une certaine incertitude, quant au moment où un paquet entrant arrive au niveau du hub par rapport à la phase de l'horloge de synchronisation.)

7.1.15 Gigue des données du récepteur

Cette section décrit la gigue des données du récepteur à basse vitesse, vitesse intégrale et à grande vitesse.

7.1.15.1 Gigue des données du récepteur à basse vitesse/vitesse intégrale

Les récepteurs de données pour tous les types d'appareils doivent être en mesure de décoder correctement les données différentielles en présence d'une gigue. Plus le front de données, devant encore être décodées, peut occuper de cellules binaires, plus le transfert de données est fiable. Les récepteurs de données doivent décoder les transitions de données différentielles qui se produisent dans une fenêtre, plus et moins une cellule binaire d'un quart nominal, à partir de la position du front de données (centré) nominal. (Un simple DPLL de diagramme d'état avec suréchantillonnage 4X qui satisfait à ces exigences peut être construit.) Cette exigence découle des Tableaux 7-4 et 7-5. Les tableaux adoptent la topologie la plus défavorable, c'est-à-dire cinq hubs placés entre l'hôte et l'appareil et le pire en matière de nombres de bits, à savoir sept, entre les transitions. Les nombres dérivés sont arrondis pour faciliter la spécification.

La gigue est engendrée par les discordances en matière de délai abordées ci-dessus et par les disparités dans les taux des données source et de destination (fréquences). Les budgets de gigue de données de réception à vitesse intégrale et à basse vitesse sont donnés aux Tableau 7-4 et Tableau 7-5. Ces tableaux indiquent la valeur et les chiffres totaux de chacune des sources de gigue à la fois pour les deux transitions consécutives (suivante) et pour les transitions jumelées. Noter que le composant de gigue lié à la tolérance en matière de fréquence source ou de destination a été alloué à l'appareil approprié (c.-à-d que la gigue source inclut des décalages de bits dus à l'imprécision de la fréquence source sur l'intervalle de transition des données le plus défavorable). La gigue du pilote de sortie peut être sacrifiée au profit de la précision de l'horloge de l'appareil dans une implantation particulière, pour autant que la spécification de la gigue soit satisfaite.

Le tableau des budgets de gigue à basse vitesse dispose d'une ligne supplémentaire parce que la gigue introduite par le hub auquel l'appareil à basse vitesse est relié, est différente de tous les autres appareils présents dans le chemin des données. Les appareils restants opèrent avec des conventions de signalisation à vitesse intégrale (bien qu'à un débit de données à basse vitesse).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 7-4. Budget de la gigue à vitesse intégrale

Source de la gigue	Vitesse intégrale			
	Transition suivante		Transition jumelée	
	Chaque (ns)	Total (ns)	Chaque (ns)	Total (ns)
Gigue du pilote source	2,0	2,0	1,0	1,0
Tolérance de la fréquence source (pire des scénarios)	0,21/bit	1,5	0,21/bit	3,0
Gigue source Total		**3,5**		**4,0**
Gigue du hub	3,0	15,0	1,0	5,0
Spécification de la gigue		**18,5**		**9,0**
Tolérance de la fréquence de destination	0,21/bit	1,5	0,21/bit	3,0
Budget de la gigue de réception		**20,0**		**12,0**

Tableau 7-5. Budget de la gigue à basse vitesse

Source de la gigue	Amont à basse vitesse			
	Transition suivante		Transition jumelée	
	Chaque (ns)	Total (ns)	Chaque (ns)	Total (ns)
Gigue du pilote de fonction	25,0	25,0	10,0	10,0
Tolérance de la fréquence de fonction (pire des scénarios)	10,0/bit	70,0	10,0/bit	140,0
Gigue (de fonction) source Total		**95,0**		**150,0**
Hub avec une gigue d'appareil à basse vitesse	45,0	45,0	45,0	45,0
Gigue des hubs (à vitesse intégrale) restants	3,0	12,0	1,0	4,0
Spécification de la gigue		**152,0**		**199,0**
Tolérance de fréquence de l'hôte	1,7/bit	12,0	1,7/bit	24,0
Budget de la gigue du récepteur de l'hôte		**164,0**		**223,0**
	Aval à basse vitesse			
	Transition suivante		**Transition jumelée**	
	Chaque (ns)	**Total (ns)**	**Chaque (ns)**	**Total (ns)**
Gigue du pilote d'hôte	2,0	2,0	1,0	1,0
Tolérance de fréquence de l'hôte (pire des scénarios)	1,7/bit	12,0	1,7/bit	24,0
Gigue (d'hôte) source Total		**14,0**		**25,0**
Hub avec une gigue d'appareil à basse vitesse	45,0	45,0	15,0	15,0
Gigue des hubs (à vitesse intégrale) restants	3,0	12,0	1,0	4,0
Spéc. de la gigue		**71,0**		**44,0**
Tolérance de la fréquence de fonction	10,0/bit	70,0	10,0/bit	140,0
Budget de la gigue du récepteur de fonction		**141,0**		**184,0**

Note: Ce tableau décrit l'hôte qui transmet des données à un appareil à basse vitesse, à un débit basse vitesse, à l'aide d'une signalisation à vitesse intégrale, en passant par le nombre maximum de hubs. Lorsque l'hôte est directement connecté à l'appareil à basse vitesse, il utilise un débit de données à basse vitesse et une

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

signalisation à basse vitesse et l'hôte doit satisfaire à la gigue source répertoriée dans la rangée "Spécification de la gigue".

7.1.15.2 Gigue des données du récepteur à grande vitesse

Un récepteur à même de fonctionner à grande vitesse doit récupérer de manière fiable les données à grande vitesse, lorsque les formes d'ondes au niveau de ses entrées sont conformes aux modèles de diagramme en œil traitant de la sensibilité du récepteur. Les modèles indiqués dans la Section 7.1.2.2, spécifient l'ouverture du diagramme en œil horizontal et vertical sur une fenêtre glissante de 480 temps bit et sur la durée d'un paquet. Ainsi, par exemple, un récepteur à grande vitesse dans une fonction doit récupérer de manière fiable les données avec une gigue de pic de 30 %, mesurée au niveau de son réceptacle B (comme décrit par le modèle 4).

Cette conformité est soumise à un essai à l'aide du mode d'essai Test_Packet, de la manière définie dans la Section 7.1.20.

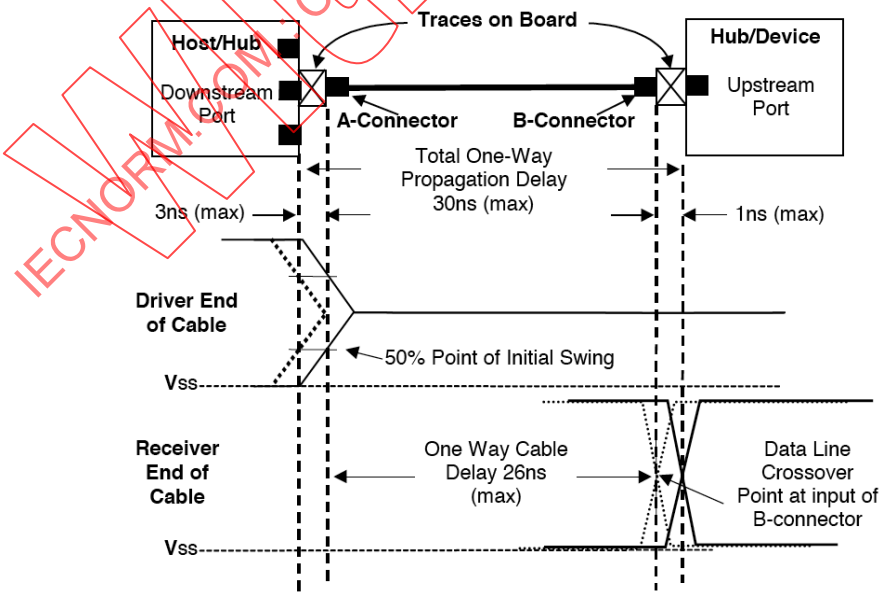
Il convient qu'une directive de conception recommande qu'un BER de récepteur soit <= 10-12, lorsque l'exigence en matière de sensibilité du récepteur est satisfaite.

7.1.16 Délai du câble

Le délai de propagation total maximal autorisé du signal dans un sens ou l'autre est de 30 ns. L'allocation pour le délai du câble est de 26 ns. Un délai maximum de 3 ns est admis entre l'émetteur-récepteur orienté en aval d'un hôte ou d'un contrôleur de hub et le connecteur externe orienté en aval, alors qu'un délai maximum de 1 ns est permis entre le connecteur orienté en amont et l'émetteur-récepteur orienté en amont de n'importe quel appareil. Pour un câble USB détachable normalisé, le délai de câble est mesuré entre les broches du connecteur de série A et les broches du connecteur de série B et ne doit pas dépasser 26 ns. Pour les autres câbles, le délai est mesuré depuis les broches du connecteur de série A jusqu'au point où le câble est connecté à l'appareil. Le délai de câble doit également être inférieur à 5,2 ns par mètre.

Le délai maximum des données transmises dans un sens ou l'autre sur un câble à vitesse intégrale est mesuré de la manière montrée dans la Figure 7-39.

Le délai de câble dans un sens ou l'autre pour les câbles à basse vitesse doit être inférieur à 18 ns. Il est mesuré conformément à la Figure 7-40.



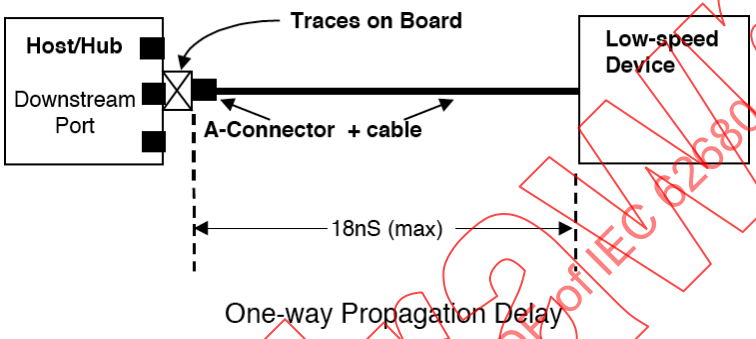
Légende

Anglais	Français
Host/Hub	Hôte/Hub
Downstream Port	Port en aval

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Anglais	Français
Traces on Board	Pistes du circuit
Host/Device	Hôte/Appareil
Upstream Port	Port en amont
A-Connector	Connecteur A
B-Connector	Connecteur B
Total One-Way Propagation Delay 30ns (max)	Délai de propagation total dans un sens ou l'autre de 30 ns (max)
Driver End of Cable	Extrémité pilote du câble
50% Point of Initial Swing	50 % du point de la variation initiale
Receiver End of Cable	Extrémité récepteur du câble
One Way Cable Delay 26ns (max)	Délai de câble dans un sens ou l'autre de 26 ns (max)
Data Line Crossover Point at input of B-Connector	Point de croisement des lignes de données au niveau de l'entrée du connecteur B

Figure 7-39. Délai du câble à vitesse intégrale



Légende

Anglais	Français
Host/Hub	Hôte/Hub
Downstream Port	Port en aval
Traces on Board	Pistes du circuit
Low-speed Device	Appareil à basse vitesse
A-Connector	Connecteur A
+ cable	câble +
One-way propagation Delay	Délai de propagation dans un sens ou l'autre

Figure 7-40. Délai du câble à basse vitesse

7.1.17 Atténuation de câble

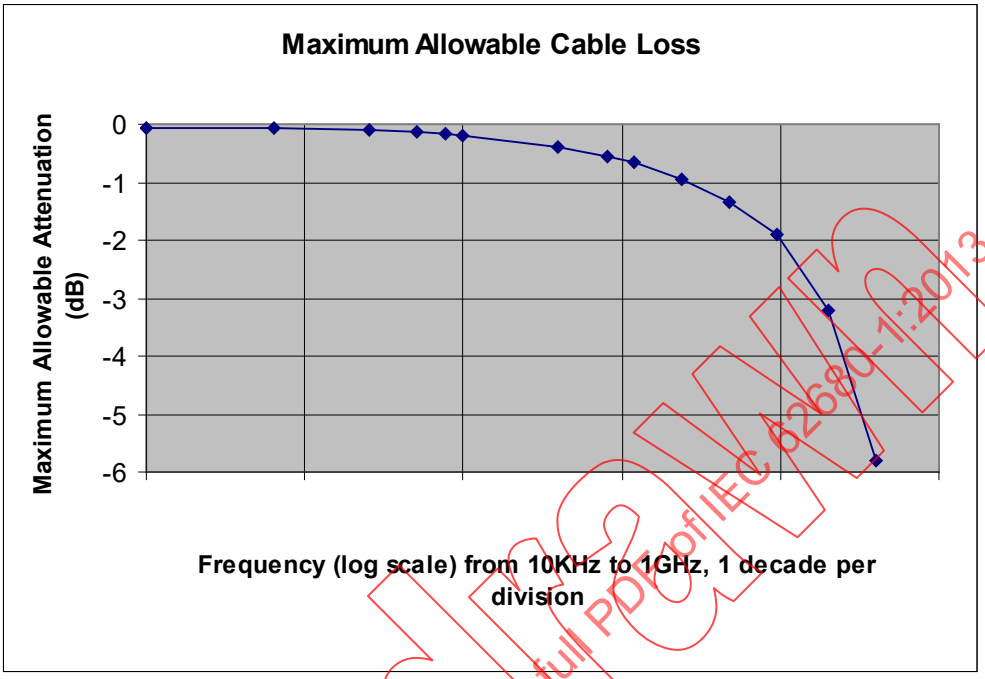
Les câbles USB ne doivent pas dépasser les pertes tolérables indiquées au Tableau 7-6. Entre les fréquences énoncées dans le tableau, il convient que la perte due au câble ne soit pas supérieure à celle présentée dans le graphique d'accompagnement.

Tableau 7-6. Perte maximale admissible due au câble

Fréquence (MHz)	Atténuation (maximum) dB/câble
0,064	0,08
0,256	0,11
0,512	0,13
0,772	0,15
1,000	0,20
4,000	0,39
8,000	0,57
12,000	0,67
24,000	0,95
48,000	1,35
96,000	1,9

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Fréquence (MHz)	Atténuation (maximum) dB/câble
200,00	3,2
400,00	5,8



Légende

Anglais	Français
Maximum Allowable Cable Loss	Perte de câble maximale admissible
Maximum Allowable Attenuation	Atténuation maximale admissible
Frequency (log scale) from 10 KHz to 1GHz, 1 decade per division	Fréquence (échelle logarithmique) entre 10 kHz et 1 GHz, 1 décennie par division

7.1.18 Temps d'inversion du bus et délai interpaquet

Cette section décrit le temps d'inversion et le délai interpaquet d'un bus à basse vitesse, vitesse intégrale et grande vitesse.

7.1.18.1 Temps d'inversion du bus et délai interpaquet d'un bus à basse vitesse/vitesse intégrale

Les délais interpaquet sont mesurés entre la transition SE0 à J au niveau de l'EOP et la transition J à K qui débute le paquet suivant.

Un appareil doit fournir au moins deux temps bit de délai interpaquet. Le délai est mesuré au niveau de l'appareil répondant avec un temps bit défini en fonction de la réponse. Cela permet d'offrir un temps adéquat pour l'appareil qui envoie l'EOP au lecteur J pour un temps bit, puis qui désactive ses mémoires tampons de sortie.

L'hôte doit fournir au moins deux temps bit de J, après le SE0 de l'EOP et le début d'un nouveau paquet (T_{IPD}). Si l'on attend d'une fonction qu'elle fournisse une réponse à une transmission d'hôte, le délai interpaquet maximum pour une fonction ou un hub doté d'un câble détachable ($T_{RSPIPD1}$) est de 6,5 temps bit mesurés au niveau du réceptacle de série B. Si l'appareil dispose d'un câble captif, le délai interpaquet ($T_{RSPIPD2}$) doit être inférieur à 7,5 temps bit, comme mesuré au niveau de la fiche de série A. Ces durées s'appliquent aux appareils à vitesse intégrale et à basse vitesse et les temps bit sont référencés pour le débit de données du paquet.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Le délai interpaquet maximum pour une réponse d'hôte est de 7,5 temps bit, mesurés au niveau des connecteurs du port d'hôte. Il n'existe aucun délai interpaquet maximum entre les paquets dans les transactions non associées.

7.1.18.2 Temps d'inversion et délai interpaquet du bus à grande vitesse

Les délais interpaquet à grande vitesse sont mesurés entre le moment où la ligne retourne en état de repos à grande vitesse, à la fin d'un paquet et le moment où la ligne quitte l'état de repos à grande vitesse au début du paquet suivant.

Lors d'une transmission après la réception d'un paquet, les hôtes et appareils doivent fournir un délai interpaquet d'au moins 8 temps bit ($T_{HSIPDOD}$) mesurés au niveau de leurs connecteurs A ou B (réceptacles ou fiches).

En outre, si un hôte transmet deux paquets dans une ligne, le délai interpaquet doit être de 88 temps bit au minimum ($T_{HSIPDSD}$), mesurés au niveau du réceptacle A de l'hôte. Cela garantit un délai interpaquet d'au moins 32 temps bit au niveau de tous les appareils (lors de la réception de paquets coup sur coup). Le délai interpaquet maximum fourni par un hôte est de 192 temps bit au sein d'une transaction ($T_{HSRSPIPD1}$), mesurés au niveau du réceptacle A. Lorsqu'un hôte répond à un paquet depuis un appareil, il fournit un délai interpaquet de 192 temps bit au maximum, mesurés au niveau du réceptacle A. Il n'existe aucun délai interpaquet maximum entre les paquets dans les transactions non associées.

Lorsqu'un appareil doté d'un câble détachable répond à un paquet depuis un hôte, il fournit un délai interpaquet de 192 temps bit au maximum, mesurés au niveau du réceptacle B. Si l'appareil dispose d'un câble captif, il fournit un délai interpaquet de 192 temps bit plus 52 ns (2 fois la longueur maximale du câble), mesurés au niveau de la fiche A du câble ($T_{HSRSPIPD2}$).

7.1.19 Délai maximum du signal de bout en bout

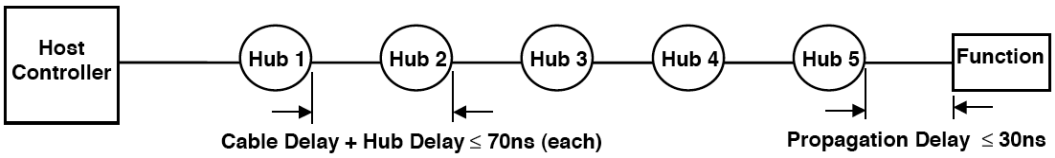
Cette section décrit le délai de bout en bout à basse vitesse, vitesse intégrale et à grande vitesse.

7.1.19.1 Délai du signal de bout en bout à basse vitesse/vitesse intégrale

Un appareil qui attend une réponse à une transmission va invalider la transaction s'il n'observe pas la transition de début de paquet (SOP) dans la période de temps mort, après la fin de la transmission (après la transition de l'état SE0 à l'état J dans l'EOP). Cela peut se produire entre un jeton IN et le paquet de données suivant ou entre un paquet de données et le paquet d'établissement de liaison (voir Chapitre 8). L'appareil qui attend la réponse ne génère pas de temps mort avant 16 temps bit, mais en crée un avant 18 temps bit (mesurés au niveau des broches de données de l'appareil et à partir de la transition SE0 à J, à la fin de l'EOP). L'hôte attend une réponse pendant au moins 18 temps bit avant de démarrer une nouvelle transaction.

La Figure 7-41 dépeint la configuration de six bonds de signal (câbles) qui entraîne le délai de signal le plus défavorable admissible. Le délai de propagation maximum entre l'extrémité en amont d'un câble de hub et un connecteur orienté en aval sur ce hub est de 70 ns.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Host Controller	Contrôleur hôte
Function	Fonction
Cable Delay + Hub Delay ≤ 70ns (each)	Délai de câble + délai de hub ≤ 70 ns (chacun)
Propagation Delay ≤ 30ns	Délai de propagation ≤ 30 ns

Figure 7-41. Modèle de délai de signal de bout en bout le plus défavorable pour basse vitesse ou vitesse intégrale

7.1.19.2 Délai de bout en bout à grande vitesse

Un hôte ou un appareil à grande vitesse qui espère une réponse à une transmission ne doit pas générer de temps mort dans la transaction si le délai interpaquet est inférieur à 736 temps bit, et doit créer un temps mort dans la transaction si aucun signal n'est observé dans les 816 temps bit.

Ces limites de temps mort permettent la visualisation de la réponse, même pour le délai du signal aller-retour le plus défavorable. En mode à grande vitesse, le modèle de délai du signal aller-retour le plus défavorable correspond à la somme des composants suivants:

112 délais câble longueur max. (6 câbles)	= 312 ns	
10 délais max. de hub (5 hubs)	= 40 ns +	360 temps bit
1 temps de réponse max de l'appareil		=192 temps bit
Délai aller-retour le plus défavorable	= 352 ns +	552 temps bit = 721 temps bit

7.1.20 Prise en charge du mode d'essai

Pour faciliter les essais de conformité, les contrôleurs d'hôtes, les hubs et les fonctions capables d'opérer à grande vitesse doivent prendre en charge les modes d'essais suivants:

- Le mode d'essai Test_SE0_NAK: Sur commande, un émetteur-récepteur de port doit entrer dans le mode de réception à grande vitesse et demeurer dans ce mode jusqu'à l'action de sortie. Cela permet de soumettre à un essai l'impédance de sortie, la tension de sortie de niveau bas et les caractéristiques de chargement. En outre, lorsqu'ils se trouvent dans ce mode, les ports orientés en amont (et uniquement les ports orientés en amont) doivent répondre à tous les paquets de jetons IN avec un établissement de liaison NAK (seulement si le CRC de paquet a été déterminé comme étant correct) pendant le temps de réponse normal autorisé de l'appareil. Cela permet de soumettre à un essai le circuit du niveau de silencieux de l'appareil et de fournir, en outre, un essai de stimulation/réponse d'usage général pour l'essai fonctionnel de base.
- Le mode d'essai Test_J: Sur commande, un émetteur-récepteur de port doit passer à l'état J à grande vitesse et rester dans cet état jusqu'à l'action de sortie. Cela permet de soumettre à un essai le niveau du lecteur de sortie haut sur la ligne D+.
- Le mode d'essai Test_K: Sur commande, un émetteur-récepteur de port doit passer à l'état K à grande vitesse et rester dans cet état jusqu'à l'action de sortie. Cela permet de soumettre à un essai le niveau du lecteur de sortie haut sur la ligne D-.
- Le mode d'essai Test_Packet: Sur commande, un port doit transmettre de matière répétitive le paquet d'essai suivant, jusqu'à l'action de sortie. Cela permet de soumettre à un essai les temps de montée et de chute, les diagrammes en œil, la gigue et toute autre spécification de formes d'ondes dynamiques.
  Le paquet d'essai est fabriqué en concaténant les chaînes suivantes. (Note: Pour les données J/K NRZI et les données NRZ, le bit de gauche est transmis en premier. "S"

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

indique qu'une insertion binaire s'est produite et qu'elle a introduit un bit de donnée NRZI "supplémentaire". "* N" indique des occurrences N d'une chaîne de bits ou de symboles.)

Symboles NRZI (champs)	Chaînes de bits NRZ	Nombre de bits NRZ
{KJ * 15}, KK (SYNC)	{00000000 * 3}, 00000001	32
KKJKJKKK (DATA0 PID)	11000011	8
JKJKJKJK * 9	00000000 * 9	72
JJKKJJKK * 8	01010101 * 8	64
JJJJKKKK * 8	01110111 * 8	64
JJJJJJJKKKKKKK * 8	0, {111111S *15}, 111111	97
JJJJJJJK * 8	S, 111111S, {0111111S * 7}	55
{JKKKKKKK * 10}, JK	00111111, {S0111111 * 9}, S0	72
JJJKKKJJKKKKJKKK (CRC16)	0110110101110011	16
JJJJJJJJ (EOP)	01111111	8

Un port en mode Test_Packet doit envoyer ce paquet de manière répétitive. La synchronisation interpaquet doit être inférieure à la discontinuité interpaquet admissible, comme défini dans la Section 7.1.18 et pas plus importante que 125 µs.

- Le mode d'essai Test_Force_Enable: Sur commande, les ports de hub orientés en aval (et uniquement les ports de hub orientés en aval) doivent être activés en mode à grande vitesse, même si aucun appareil n'est connecté. Les paquets qui arrivent au niveau du port orienté en amont doivent être répétés sur le port qui figure dans ce mode d'essai. Cela permet de soumettre à un essai la détection de déconnexion du hub; le bit de détection de déconnexion peut être interrogé tout en variant la charge sur le port, ce qui permet de mesurer la tension du seuil de détection de déconnexion.

Mode d'essai - Entrée et sortie

On entre en mode d'essai d'un port en utilisant une requête normalisée spécifique à l'appareil (pour un port orienté en amont) ou une requête de classe de hub particulière (pour un port orienté en aval). La requête d'appareil normalisée SetFeature(TEST_MODE) est définie à la Section 9.4.9. La requête de classe de hub SetPortFeature(PORT_TEST) est définie à la Section 11.24.2.13. Tous les appareils /hubs capables de fonctionner à grande vitesse doivent prendre en charge ces requêtes. Ces requêtes ne sont pas supportées pour les appareils qui n'opèrent pas à grande vitesse.

La transition vers le mode d'essai doit être terminée avant 3 ms suivant l'achèvement de l'étape d'état de la requête.

Pour un port orienté en amont, l'action de sortie doit être le redémarrage de l'appareil. Pour un port orienté en aval, l'action de sortie doit être la réinitialisation du hub, comme défini dans la Section 11.24.2.13.

7.2 Distribution de l'alimentation

Cette section décrit les spécifications de distribution de l'alimentation USB.

7.2.1 Classes de appareils

Les exigences en matière de source d'alimentation et d'appareil d'écoulement de courant des différentes classes d'appareils peuvent être simplifiées grâce à l'introduction du concept de charge unitaire. Une charge unitaire est définie sous 100 mA. Le nombre de charges unitaires qu'un appareil peut appeler représente le maximum absolu et pas une moyenne mesurée dans le temps. Un appareil peut être soit basse puissance à une charge unitaire soit à grande puissance et consommer jusqu'à cinq charges unitaires. Tous les appareils fonctionnent à basse puissance par défaut. La transition vers la grande puissance reste sous contrôle du logiciel. Il incombe au logiciel de garantir qu'une puissance adéquate est disponible avant d'autoriser les appareils à consommer de l'électricité en état de grande puissance.

L'USB prend en charge une gamme d'agents source de courant et consommateurs de courant, dont:

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- **Hubs de port racine:** Qui sont directement reliés au contrôleur USB hôte. La puissance du hub est dérivée de la même source que celle du contrôleur hôte. Les systèmes qui obtiennent une puissance d'exploitation CA ou CC de l'extérieur, doivent fournir au moins cinq charges unitaires à chaque port. Ces ports sont appelés ports de grande puissance. Les systèmes alimentés par batterie peuvent fournir une ou cinq charges unitaires. Les ports qui peuvent offrir une charge unitaire sont intitulés ports basse puissance.
- **Les hubs alimentés par bus:** Ils tirent toute leur puissance du VBUS présent sur le port orienté en amont du hub, afin d'alimenter n'importe quelle fonction interne et n'importe quel port orienté en aval. Les hubs alimentés par bus ne peuvent appeler qu'une charge unitaire lors de leur mise sous tension et cinq charges unitaires après la configuration. L'allocation de la puissance de configuration est répartie entre le hub, toutes les fonctions non amovibles et les ports externes. Les ports externes dans un hub alimenté par bus ne peuvent fournir qu'une charge unitaire par port, quel que soit l'appel de courant sur les autres ports de ce hub. Ce hub doit être en mesure de fournir du courant à ce port, lorsqu'il se trouve en état actif ou interrompu.
- **Les hubs autoalimentés:** L'alimentation en électricité pour les fonctions internes et les ports orientés en aval ne proviennent pas du VBUS. Toutefois, l'interface USB du hub peut appeler une charge unitaire du VBUS présent sur son port orienté en amont, afin de permettre à l'interface de fonctionner lorsque le reste du hub a été mis hors tension. Les hubs qui obtiennent une puissance d'exploitation de l'extérieur (de l'USB) doivent fournir cinq charges unitaires à chaque port. Les hubs alimentés par batterie peuvent fournir une ou cinq charges unitaires par port.
- **Les fonctions basse puissance alimentées par bus:** Toute l'alimentation électrique vers ces appareils provient du VBUS. Elles ne peuvent pas appeler plus d'une charge unitaire à la fois.
- **Les fonctions grande puissance alimentées par bus:** Toute l'alimentation électrique vers ces appareils provient du VBUS. Elles ne doivent pas appeler plus d'une charge unitaire à la mise sous tension et peuvent appeler jusqu'à cinq charges unitaires après avoir été configurées.
- **Les fonctions autoalimentées:** Peuvent appeler jusqu'à une charge unitaire du VBUS, afin de permettre à l'interface USB de fonctionner lorsque le reste de la fonction a été mise hors tension. Toute autre source électrique provient d'une source (USB) externe.

Aucun appareil ne doit fournir du courant (source) sur le VBUS, au niveau de son port orienté vers l'amont et cela à aucun moment. Un appareil ne peut appeler du courant (écoulement) que du VBUS sur son port orienté en amont. Il peut ne pas fournir de l'électricité à la résistance de rappel à la source sur D+/D-, à moins que le VBUS soit présent (voir la Section 7.1.5). Lorsque le VBUS est retiré, l'appareil doit supprimer dans les 10 secondes, toute alimentation électrique dans la résistance de rappel à la source. A la mise sous tension, un appareil doit garantir que son port orienté en amont n'entraîne pas le bus, afin que l'appareil soit en mesure de recevoir le signal de réinitialisation. Avant leur configuration, les appareils doivent aussi garantir que le courant d'exploitation maximum appelé correspond à une charge unitaire. Tout appareil qui appelle du courant du bus doit être en mesure de détecter un manque d'activité sur le bus, passer en état interrompu et réduire sa consommation électrique à partir du VBUS (voir Sections 7.2.3 et 9.2.5.1).

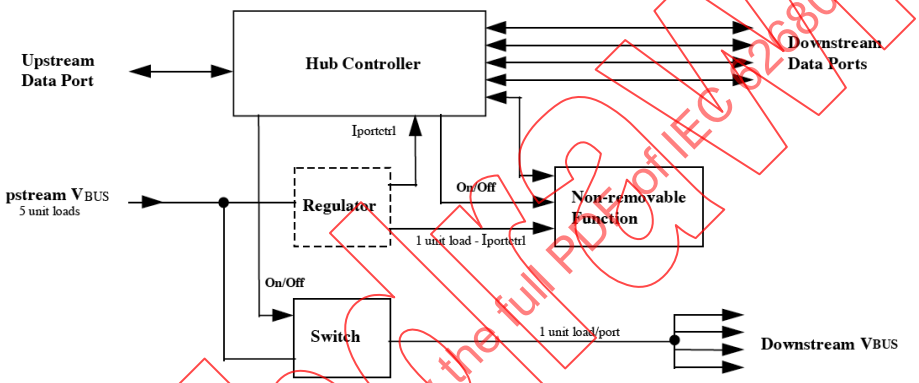
7.2.1.1 Hubs alimentés par bus

Les exigences électriques d'un hub alimenté par bus peuvent être satisfaites avec un circuit d'alimentation et de régulation, comme celui illustré dans la Figure 7-42. Les hubs alimentés par bus contiennent en général au moins une fonction non amovible. Les hubs alimentés par bus contiennent souvent au moins une fonction non amovible. Le contrôleur du hub est toujours alimenté en électricité, ce qui permet à l'hôte d'accéder à la gestion d'énergie et à d'autres registres de configuration au cours du processus d'énumération. Une fonction(s) non amovible peut exiger la commutation de sa puissance électrique, afin que lors de la mise sous tension, l'appareil en entier (hub et fonction(s) non amovible) ne consomme pas plus qu'une charge unitaire. La commutation électrique sur toute fonction non amovible peut être implantée en coupant sa source électrique ou en éteignant l'horloge. La commutation sur une

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

fonction non amovible n'est pas requise si le courant tiré par la fonction et le contrôleur de hub est inférieur à une charge unitaire. Toutefois, tant que le port du hub associé à la fonction est en état hors tension, la fonction doit être logiquement réinitialisée et l'appareil doit apparaître comme n'étant pas connecté. Le courant total tiré par l'appareil alimenté par bus correspond à la somme du courant diffusé vers le contrôleur de hub, de toutes les fonctions non amovibles et des ports orientés en aval.

La Figure 7-42 illustre le partitionnement du courant électrique, basé sur l'appel de courant maximum (depuis l'amont) de cinq charges unitaires: une charge unitaire pour le contrôleur du hub et la fonction non amovible et une charge unitaire pour chacun des ports orientés en aval. Si plus de quatre ports externes sont requis, le hub doit alors s'autoalimenter. Si la fonction(s) non amovible et le contrôleur du hub appellent plus d'une charge unitaire, le nombre de ports externes doit alors être réduit de manière appropriée. La régulation de puissance vers un hub alimenté par bus peut exiger un régulateur. Dans ce cas, le régulateur est toujours activé pour alimenter en électricité le contrôleur du hub. Le régulateur peut aussi alimenter la fonction(s) non amovible. Un limiteur de courant d'appel doit aussi être incorporé dans le sous-système du régulateur.



Légende

Anglais	Français
Upstream Data Port	Port de données en amont
Hub Controller	Contrôleur de hub
Downstream Data Ports	Port de données en aval
Upstream V_{BUS} 5 unit loads	V_{BUS} en amont 5 charges unitaires
Regulator	Régulateur
On/Off	Marche/Arrêt
Non-removable Function	Fonction non amovible
1 unit load - $I_{portctrl}$	1 charge unitaire - $I_{portctrl}$
Switch	Commutateur
1 unit load/port	1 charge unitaire/port
Downstream V_{BUS}	V_{BUS} en aval

Figure 7-42. Hub composé alimenté par bus

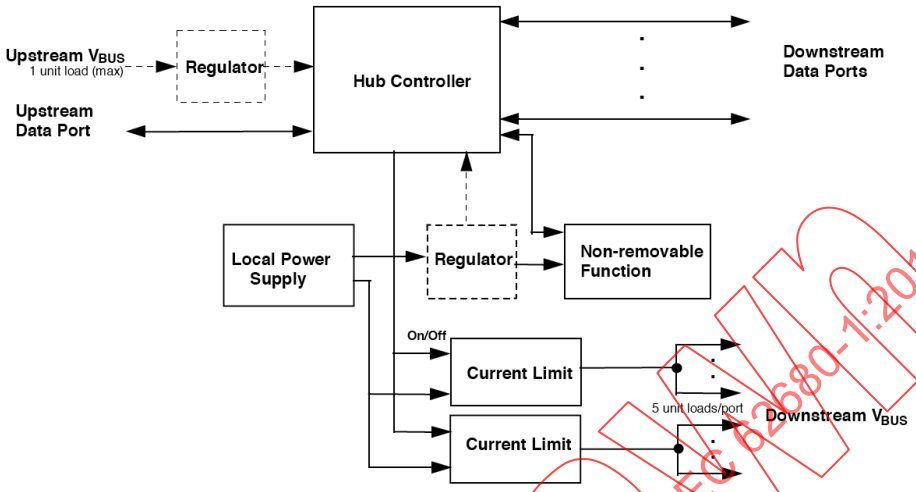
La puissance électrique vers les ports externes orientés en aval d'un hub alimenté par bus, doit être commutée. Le contrôleur du hub fournit, à partir de l'hôte, un signal on/off commandé par logiciel, qui figure en état "off" lorsque l'appareil est mis sous tension ou après une signalisation de réinitialisation. Lorsqu'il est passé en état "on", le commutateur implémente une fonction souple de mise sous tension qui évite un appel excessif de courant transitoire depuis l'amont. La tension chute dans le câble, les connecteurs en amont et la commutation dans un hub alimenté par bus ne doit pas dépasser 350 mV au courant assigné maximum.

7.2.1.2 Hubs autoalimentés

Les hubs autoalimentés disposent d'une source électrique locale qui fournit de l'électricité à toute fonction non amovible et à tous les ports orientés en aval, comme le montre la Figure 7-43. L'alimentation du contrôleur de hub peut cependant être assurée à partir du VBUS amont (un hub alimenté de manière "hybride") ou à partir de la source électrique locale.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

L'avantage de pouvoir alimenter le contrôleur du hub depuis la source en aval est que la communication à partir de l'hôte est rendue possible, même si l'appareil reste hors tension. Cela permet de différencier un appareil déconnecté d'un appareil hors tension. Si un appel de courant à partir du VBUS est effectué par le hub pour son port orienté en amont, il ne peut pas appeler plus d'une charge unitaire.



Légende

Anglais	Français
Upstream V_{BUS}	V_{BUS} en amont
1 unit load (max)	1 charge unitaire (max)
Regulator	Régulateur
Hub Controller	Contrôleur de hub
Downstream Data Ports	Ports de données en aval
Local Power Supply	Alimentation électrique locale
Regulator	Régulateur
Non-removable Function	Fonction non amovible
On/Off	Marche/Arrêt
Current Limit	Limite actuelle
5 unit loads/port	5 charges unitaires/port
Downstream V_{BUS}	V_{BUS} en aval

Figure 7-43. Hub composé autoalimenté

Le nombre de ports qui peuvent être pris en charge n'est limité que par la fonctionnalité d'adressage du hub et de la source électrique locale.

Les hubs autoalimentés peuvent expérimenter une perte de puissance. Cela peut faire suite à la déconnexion du cordon d'alimentation ou à l'épuisement de la batterie. Sous ces conditions, le hub peut forcer sa ré-énumération en tant que hub alimenté par bus. Cela requiert du hub l'implantation d'une commutation d'alimentation sur tous les ports externes. Lorsqu'il y a perte de puissance, le hub doit garantir que le courant en amont n'excède pas la basse puissance. Toutes les règles propres à un hub alimenté par bus s'appliquent alors.

7.2.1.2.1 Protection contre les surintensités

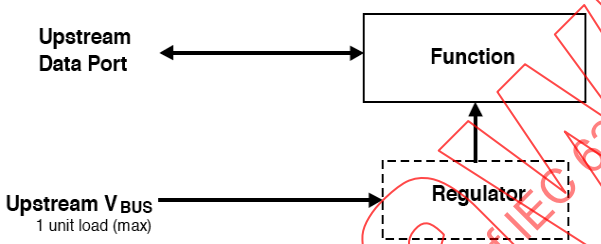
L'hôte, ainsi que tous les hubs autoalimentés doivent implanter un système de protection contre les surintensités et cela pour des raisons de sécurité; et le hub doit disposer d'un moyen de détection d'une condition de surintensité et la signaler au logiciel USB. Si le courant global appelé par un ensemble de ports orientés vers l'aval dépasse une valeur prédéfinie, il convient que le circuit de protection contre les surintensités élimine ou réduise l'alimentation électrique de tous les ports orientés vers l'aval qui sont affectés. La condition de surintensité est signalée par le hub au contrôleur hôte, tel que décrit à la Section 11.12.5. La valeur prédéfinie ne peut dépasser 5,0 A et doit être suffisamment supérieure au courant maximal admissible du port de sorte que les courants transitoires (par exemple lors de la mise sous tension, de la connexion dynamique ou de la reconfiguration) ne déclenchent pas le protecteur contre les surintensités. Si une condition de surintensité se produit sur l'un des

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

ports, les opérations ultérieures de l'USB ne sont pas garanties et une fois que la condition est éliminée, il peut s'avérer nécessaire de réinitialiser le bus, de la manière effectuée lors de la mise sous tension. Le mécanisme de limitation des surintensités doit être réinitialisable sans intervention mécanique de l'utilisateur. Les commutateurs polymères PTC et à semi-conducteurs sont des exemples de méthodes qui peuvent être utilisées pour la limitation des surintensités.

7.2.1.3 Fonctions basse puissance alimentées par bus

Une fonction basse puissance est une fonction qui appelle une charge unitaire du câble USB, lorsqu'elle est opérationnelle. La Figure 7-44 illustre une fonction basse puissance alimentée par bus typique, comme une souris. La régulation basse puissance peut être intégrée dans le silicone de la fonction. Les fonctions basse puissance doivent être en mesure de fonctionner avec des tensions d'entrée VBUS aussi basses que 4,40 V, mesurées au niveau de la fiche à l'extrémité du câble.



Légende

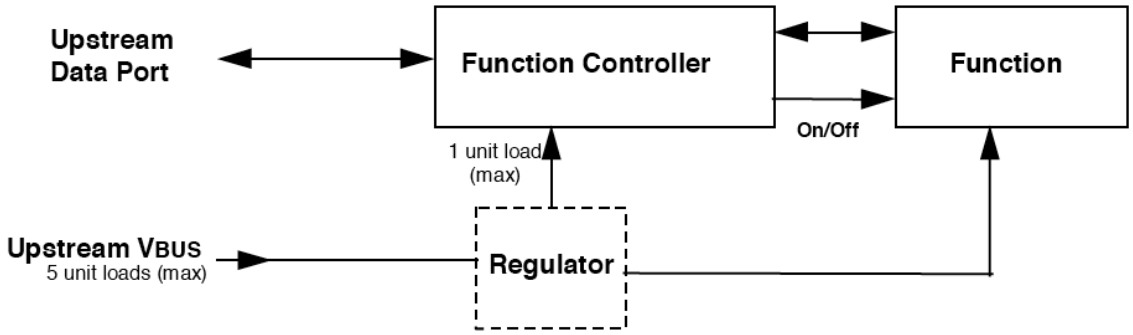
Anglais	Français
Upstream Data Port	Port de données en amont
Function	Fonction
Upstream V_{BUS}	V_{BUS} en amont
5 unit load (max)	5 charges unitaires (max)
Regulator	Régulateur

Figure 7-44. Les fonctions basse puissance alimentées par bus

7.2.1.4 Fonctions grande puissance alimentées par bus

Une fonction est définie comme étant à grande puissance si, lorsqu'elle est pleinement alimentée, elle appelle du câble USB plus d'une charge unitaire, mais pas plus de cinq. Une fonction grande puissance requiert une commutation de puissance étalonnée. Elle doit tout d'abord être activée dans un état de puissance réduit, inférieur à une charge unitaire. Au moment de l'énumération de bus, ses exigences totales en matière de puissance sont obtenues et comparées par rapport au budget de puissance disponible. Si une puissance suffisante existe, le reste de la fonction peut être mis sous tension. Une fonction grande puissance type est fournie à la Figure 7-45. Les systèmes électroniques d'une fonction ont été divisés en deux sections. Le contrôleur de fonction contient la quantité minimale de circuits, nécessaires pour permettre l'énumération et la budgétisation de la puissance. Le reste de la fonction réside dans le bloc de fonction. Les fonctions grande puissance doivent être en mesure de fonctionner dans leur mode basse puissance (une charge unitaire) avec une tension d'entrée aussi faible que 4,40 V, afin qu'elles puissent être détectées et énumérées même lorsqu'elles sont branchées à un hub alimenté par bus. Elles doivent aussi être capables d'opérer à pleine puissance (jusqu'à cinq charges unitaires) avec une tension VBUS de 4,75 V, mesurée au niveau de la fiche d'extrémité en amont du câble.

ECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



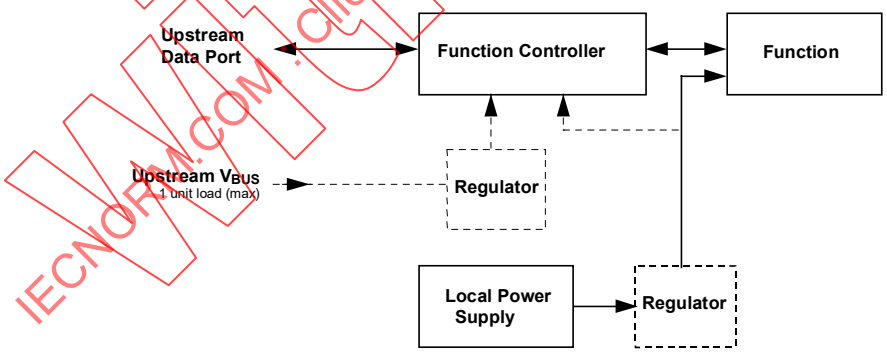
Légende

Anglais	Français
Upstream Data Port	Port de données en amont
Function Controller	Contrôleur de fonction
Function	Fonction
1 unit load (max)	1 charge unitaire (max)
On/Off	Marche/Arrêt
Upstream V_{BUS}	V_{BUS} en amont
5 unit loads (max)	5 charges unitaires (max)
Regulator	Régulateur

Figure 7-45. Fonction grande puissance alimentée par bus

7.2.1.5 Les fonctions autoalimentées

La Figure 7-46 illustre une fonction autoalimentée typique. Le contrôleur de fonction est alimenté à partir du bus en amont, via un régulateur basse puissance ou depuis la source d'alimentation électrique locale. L'avantage du premier schéma est qu'il permet la détection et l'énumération d'une fonction autoalimentée, dont la source électrique locale a été débranchée. La puissance en amont maximale que le contrôleur de fonction peut appeler est une charge unitaire et le bloc de régulation doit implanter une limitation de courant d'appel. La quantité de courant que le bloc de fonction peut appeler n'est limitée que par la source d'alimentation locale. Parce que la source d'alimentation locale n'est pas requise pour alimenter les ports du bus en aval, il n'est pas nécessaire d'implanter une limitation de courant, un démarrage souple ou une commutation d'alimentation.



Légende

Anglais	Français
Upstream Data Port	Port de données en amont
Function Controller	Contrôleur de fonction
Function	Fonction
Upstream V_{BUS}	V_{BUS} en amont
1 unit load (max)	1 charge unitaire (max)
Regulator	Régulateur
Local Power Supply	Alimentation électrique locale
Regulator	Régulateur

Figure 7-46. Fonction autoalimentée

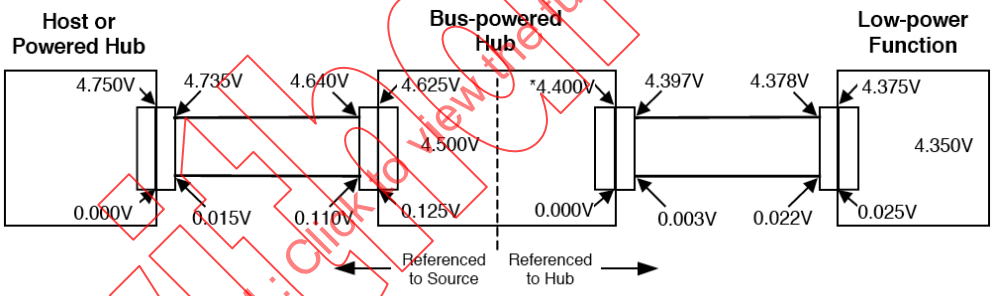
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

7.2.2 Budget de chute de tension

Le budget de chute de tension est déterminé à partir des éléments suivants:

- La tension fournie par les ports de hub de grande puissance est comprise entre 4,75 V et 5,25 V.
- La tension fournie par les ports de hub de basse puissance est comprise entre 4,4 V et 5,25 V.
- Les hubs alimentés par bus peuvent disposer d'une chute maximale de 350 mV depuis leur fiche de câble (à l'endroit où ils sont connectés à une source d'alimentation) jusqu'à leurs connecteurs de port de sortie (où ils fournissent de l'électricité).
- La chute maximale de tension (pour les câbles détachables) entre la fiche de série A et la fiche de série B sur le VBUS est de 125 mV (VBUSD).
- La chute maximale de tension pour tous les câbles entre l'amont et l'aval sur un GND est de 125 mV (VGNDD).
- Tous les hubs et fonctions doivent être en mesure de fournir des informations de configuration avec aussi peu de courant que 4,40 V au niveau de l'extrémité du connecteur de leurs câbles en amont. Seules les fonctions basse puissance doivent être opérationnelles avec ce minimum de tension.
- Les fonctions qui appellent plus d'une charge unitaire doivent fonctionner avec une tension d'entrée minimum de 4,75 V, au niveau de l'extrémité du connecteur de leurs câbles en amont.

La Figure 7-47 illustre les tensions admissibles minimales dans la topologie la plus défavorable qui comprend un hub alimenté par bus entraînant une fonction alimentée par bus.



Légende

Anglais	Français
Host or Powered Hub	Hôte ou hub alimenté
Bus-powered Hub	Hub alimenté par bus
Low-power Function	Fonction basse puissance
Referenced to Source	Référencée à la source
Referenced to Hub	Référencée au hub
*Under transient conditions, supply at hub can drop from 4.400V to 4.070V	*Dans des conditions transitoires, l'alimentation du hub peut chuter de 4 400 V à 4 070 V

Figure 7-47. Topologie de chute de tension la plus défavorable (Etat stable)

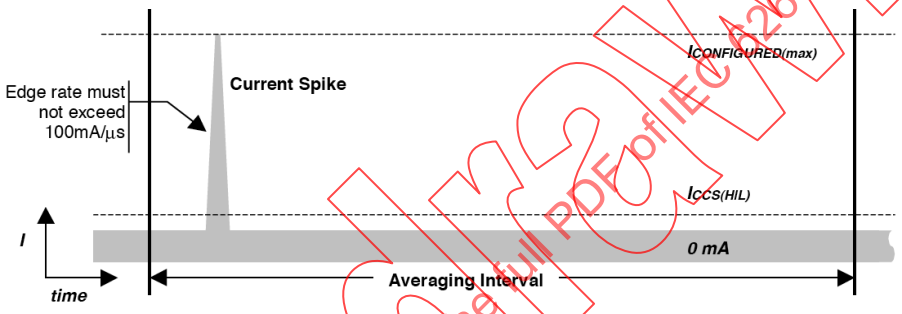
7.2.3 Régulation de puissance au cours d'une interruption/reprise

Le courant de veille est une fonction d'allocation de charge unitaire. Tous les appareils USB fonctionnent initialement à basse puissance par défaut. Les appareils basse puissance ou grande puissance opérant à basse puissance sont limités à 500 µA de courant de veille. Si l'appareil est configuré pour fonctionner à grande puissance et activé en tant que source de réveil à distance, il peut appeler jusqu'à 2,5 mA lorsqu'il se trouve en état de veille. Lors de la comptabilisation du courant de veille, le courant à partir du VBUS qui traverse les résistances de rappel à la source et de rappel à la masse doit être inclus. Les hubs alimentés par bus configurés peuvent aussi consommer un maximum de 2,5 mA, avec 500 µA alloués à chaque port externe disponible et le reste disponible sur le hub et ses fonctions internes. Si un hub

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

n'est pas configuré, il fonctionne en tant qu'appareil basse puissance et doit limiter son courant de veille à 500 µA.

Lorsqu'il se trouve en état interrompu, un appareil peut appeler brièvement plus de courant que la moyenne. L'amplitude de la variation brusque du courant ne peut pas dépasser l'allocation de puissance de l'appareil de 100 mA (ou 500 mA). Un maximum de 1 seconde est autorisé pour un intervalle de moyennage. Le courant moyen ne peut pas excéder la limite de courant de veille moyen (ICCSH ou ICCSL, voir le Tableau 7-7) au cours de tout intervalle de 1 seconde (TSUSAVG1). Le profil de la variation brusque du courant est restreint afin que la réponse transitoire de la source électrique (qui peut être une source électrique de compensation) ne soit pas accablante. Le front montant de la variation brusque de courant ne doit pas être supérieur à 100 mA/µs. Les ports orientés en aval doivent être capables d'absorber le pic de variation brusque de courant de 500 mA et de satisfaire aux exigences de chute de tension définies pour le courant d'appel au cours de la connexion dynamique (voir Section 7.2.4.1). La Figure 7-48 illustre un exemple de profil typique pour un intervalle de moyennage. Si l'alimentation vers la résistance de rappel à la source sur D+/D- est dérivée du VBUS, le courant de veille n'est jamais à zéro, car les résistances de rappel à la source et à la masse appellent toujours du courant.



Légende

Anglais	Français
Edge rate must not exceed 100mA/µs	Le débit de bord ne doit pas dépasser 100 mA/µs
Current Spike	Variation brusque du courant
Time	Temps
Averaging Interval	Intervalle de moyennage

Figure 7-48. Profil de moyennage de courant suspendu type

Les appareils sont responsables de la gestion de la réduction de tension du bus due aux effets inductifs et résistants du câble. Lorsqu'un hub se trouve en état interrompu, il doit toujours être en mesure de fournir le courant maximum par port (une charge unitaire de courant par port pour les hubs alimentés par bus et cinq charges unitaires pour les hubs autoalimentés). Cela est nécessaire pour pendre en charge les appareils capables de réveil à distance qui se mettent sous tension alors que le reste du système est toujours interrompu. Ces appareils, lorsqu'ils ont été activés pour le réveil à distance, doivent entraîner la reprise de signalisation en amont dans les 10 ms suivant le début de l'appel de courant de non veille plus élevé. Les appareils qui ne sont pas capables de réveil à distance doivent appeler du courant plus élevé lorsqu'ils ne sont pas interrompus.

Lorsque des appareils se réveillent, soit d'eux-mêmes (réveil à distance) ou en voyant la reprise de la signalisation, ils doivent limiter le courant d'appel sur VBUS. Le statisme maximal cible dans le hub VBUS est de 330 mV. L'appareil doit disposer de suffisamment de capacité de dérivation à bord ou une séquence de mise sous tension contrôlée, de manière à ce que le courant appelé du hub n'excède pas, à tout moment, la fonctionnalité maximale de courant du port pendant le réveil de l'appareil.

7.2.4 Connexion et déconnexion dynamiques

L'acte de connexion ou de déconnexion d'un hub ou d'une fonction ne doit pas affecter la fonctionnalité d'un autre appareil placé sur d'autres segments du réseau. La déconnexion d'une fonction arrête la transaction entre cette fonction et l'hôte. Toutefois, le hub auquel

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

cette fonction a été connectée récupère de cette condition et alerte l'hôte de la déconnexion du port.

7.2.4.1 Limitation de courant d'appel

Lorsqu'une fonction ou un hub est connecté sur le réseau, il dispose d'une certaine quantité de capacité à bord, entre le VBUS et la masse. En outre, le régulateur sur l'appareil peut fournir du courant à sa capacité de dérivation de sortie et à la fonction, sitôt que la puissance électrique est appliquée. Par conséquent, si aucune mesure n'est prise pour éviter cela, une hausse de courant peut surgir dans l'appareil qui peut ramener le VBUS sur le hub à son niveau d'exploitation minimum. Des courants d'appel peuvent également surgir lorsqu'une fonction grande puissance est commutée en mode grande puissance. Ce problème doit être résolu en limitant le courant d'appel et en fournissant suffisamment de capacités à chaque hub pour empêcher que le courant fourni aux autres ports sorte des limites de tolérance. Une motivation supplémentaire de limiter le courant d'appel est de minimiser les décharges électrostatiques du contact et prolonger ainsi la durée de vie du contact du connecteur.

Le statisme maximum dans le VBUS du hub est de 330 mV ou d'environ 10 % de la variation du signal nominal à partir de la fonction. Afin de répondre à cette exigence, les conditions suivantes doivent être satisfaites:

- La charge maximale (CRPB) qui peut être placée à l'extrémité en aval d'un câble est de 10 µF en parallèle avec 44 Ω. La capacité de 10 µF représente tout condensateur de découplage connecté directement à travers les lignes du VBUS dans la fonction, plus tout effet capacitif visible à travers le régulateur placé dans l'appareil. La résistance 44 Ω représente une charge unitaire de courant appelé par l'appareil au cours de la connexion.
- Si une capacité de découplage plus importante est requise dans l'appareil, ce dernier doit donc incorporer une certaine forme de limitation de courant de surcharge VBUS, de telle sorte qu'elle satisfasse aux caractéristiques de la charge ci-dessus.
- Les lignes électriques du VBUS du port orienté en aval du hub doivent être dérivées (CHPB) avec pas moins de 120 µF de capacité ESR basse par hub. Il convient que les méthodes de dérivation normalisées soient utilisées pour minimiser l'inductance et la résistance entre les condensateurs de contournement et les connecteurs afin de réduire le statisme. Il convient que les condensateurs de contournement disposent d'un facteur de dissipation bas, afin de permettre le découplage à des fréquences plus élevées.

Le port orienté en amont d'un hub est également nécessaire afin de pouvoir satisfaire aux exigences ci-dessus. Par ailleurs, un hub alimenté par bus doit fournir une limitation de surcharge supplémentaire, sous forme de circuit de démarrage souple, lorsqu'il active électriquement ses ports orientés en aval.

Un appareil grande puissance alimenté par bus qui effectue une commutation entre une configuration à plus basse puissance vers une configuration à plus haute puissance ne doit pas entraîner un statisme > 330 mV sur le VBUS au niveau de son hub en amont. L'appareil peut satisfaire à cela en garantissant que les modifications dans la charge capacitive qu'il présente n'excèdent pas 10 µF.

Les broches de signaux sont protégées contre les courants excessifs durant la connexion dynamique, en étant encastrées dans le connecteur, de manière à ce que les broches électriques établissent le premier contact. Cela garantit que les rails conducteurs pour l'appareil en aval sont référencés avant que les broches de signaux établissent un contact. En outre, les circuits d'acheminement des signaux sont en état de haute impédance pendant la connexion afin qu'aucun courant ne s'écoule pour les niveaux de signaux normalisés.

7.2.4.2 Déconnexion dynamique

Lorsqu'un appareil est déconnecté du réseau, avec du courant s'écoulant dans le câble, l'inductance du câble entraîne l'apparition d'une importante tension de retour à l'extrémité ouverte du câble de l'appareil. Cette tension de retour n'est pas destructrice. Des mesures de dérivation correctes sur les ports du hub suppriment tout bruit couplé. La gamme de

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

fréquences de ce bruit est inversement dépendante de la longueur du câble, et peut aller jusqu'à un maximum de 60 MHz pour un câble d'un mètre. Cela exige des condensateurs de découplage de plus ou moins basse capacité et de très faible inductance sur chaque connecteur du port du hub. Le bruit généré par la tension de retour est également diminué par la capacité de dérivation sur l'extrémité du câble de l'appareil. Une certaine capacité minimum doit également être présente sur l'extrémité du câble de l'appareil, afin de garantir que la tension de retour inductive sur l'extrémité ouverte du câble n'entraîne pas une inversion de la polarité de la tension sur l'appareil. Un minimum de 1,0 µF est recommandé pour la dérivation à travers le VBUS.

7.3 Couche physique

Les spécifications de la couche physique sont décrites dans les sous-sections suivantes.

7.3.1 Exigences réglementaires

Il convient que tous les appareils USB soient conçus de manière à satisfaire aux exigences réglementaires applicables.

7.3.2 Synchronisation du bus/caractéristiques électriques

Tableau 7-7. Caractéristiques électriques CC

Paramètre	Symbole	Conditions	Min.	Max.	Unités
Tension d'alimentation:					
Port grande puissance	VBUS	Note 2, Section 7.2.1	4,75	5,25	V
Port basse puissance	VBUS	Note 2, Section 7.2.1	4,40	5,25	V
Courant d'alimentation:					
Port du hub grande puissance (à la sortie)	ICCPRT	Section 7.2.1	500		mA
Port du hub basse puissance (à la sortie)	ICCUPT	Section 7.2.1	100		mA
Fonction grande puissance (à l'entrée)	ICCHPF	Section 7.2.1		500	mA
Fonction basse puissance (à l'entrée)	ICCLPF	Section 7.2.1		100	mA
Fonction/hub non configurée (à l'entrée)	ICCINIT	Section 7.2.1.4		100	mA
Appareil grande puissance interrompu	ICCSH	Section 7.2.3; note 15		2,5	mA
Appareil basse puissance interrompu	ICCSL	Section 7.2.3		500	µA
Niveaux d'entrée pour la basse vitesse/vitesse intégrale:					
Elevé (conduit)	Vih	Note 4, Section 7.1.4	2,0		V
Elevé (flottant)	VIHZ	Note 4, Section 7.1.4	2,7	3,6	V
Faible	Vil	Note 4, Section 7.1.4		0,8	V
Sensibilité d'entrée différentielle	VDI	|(D+)-(D-)|; Figure 7-19; note 4	0,2		V
Mode commun différentiel Plage	VCM	Inclut la gamme VDI; Figure 7-19; note 4	0,8	2,5	V

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 7-7. Caractéristiques électriques CC (suite)

Paramètre	Symbole	Conditions	Min.	Max.	Unités
Niveaux d'entrée pour la grande vitesse:					
Seuil de détection du silencieux à grande vitesse (amplitude du signal différentiel)	VHSSQ	Section 7.1.7.2 (la spécification se réfère à l'amplitude du signal différentiel)	100	150	mV
Seuil de détection de déconnexion à grande vitesse (amplitude du signal différentiel)	VHSDSC	Section 7.1.7.2 (la spécification se réfère à l'amplitude du signal différentiel)	525	625	mV
Niveaux de signalisation d'entrée différentielle à grande vitesse		Section 7.1.7.2 Spécifié par les modèles de diagramme en œil			
Gamme de tension du mode commun de signalisation des données à grande vitesse (directive pour le récepteur)	VHSCM	Section 7.1.4.2	-50	500	mV
Niveaux de sortie pour la basse vitesse/vitesse intégrale:					
Faible	VOL	Note 4, 5, Section 7.1.1	0,0	0,3	V
Elevé (Conduit)	VOH	Note 4, 6, Section 7.1.1	2,8	3,6	V
SE1	VOSE1	Section 7.1.1	0,8		V
Tension de croisement du signal de sortie	VCRS	Mesuré comme dans la Figure 7-8; note 10	1,3	2,0	V
Niveaux de sortie pour la grande vitesse:					
Niveau de repos à grande vitesse	VHSOI	Section 7.1.7.2	-10,0	10,0	mV
Signalisation des données à grande vitesse (haute)	VHSOH	Section 7.1.7.2	360	440	mV
Signalisation des données à grande vitesse (basse)	VHSOL	Section 7.1.7.2	-10,0	10,0	mV
Niveau J de gazouillis (tension différentielle)	VCHIRPJ	Section 7.1.7.2	700	1 100	mV
Niveau K de Chirp (tension différentielle)	VCHIRPK	Section 7.1.7.2	-900	-500	mV
Capacité de découplage:					
Capacité de dérivation du port orienté en aval (par hub)	CHPB	VBUS à GND, Section 7.2.4.1	120		µF
Capacité de dérivation du port orienté en amont	CRPB	VBUS à GND; note 9, Section 7.2.4.1	1,0	10,0	µF
Capacité d'entrée pour la basse vitesse/vitesse intégrale:					
Port orienté en aval	CIND	Note 2; Section 7.1.6.1		150	pF
Port orienté en amont (sans câble)	CINUB	Note 3; Section 7.1.6.1		100	pF
Capacité de contrôle de débit du bord de l'émetteur-récepteur	CEDGE	Section 7.1.6.1		75	pF
Impédance d'entrée pour la grande vitesse:					
Spéc. TDR pour la terminaison à grande vitesse		Section 7.1.6.2			

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 7-7. Caractéristiques électriques CC (suite)

Paramètre	Symbole	Conditions	Min.	Max.	Unités
Terminaisons:					
Résistance de rappel à la source du bus sur le port orienté en amont	RPU	1,5 kΩ ± 5 % Section 7.1.5	1,425	1,575	kΩ
Résistance de rappel à la masse du bus sur le port orienté en aval	RPD	15 kΩ ± 5 % Section 7.1.5	14,25	15,75	kΩ
Impédance d'entrée exclusive au rappel à la source/à la masse (pour la basse vitesse/vitesse intégrale)	ZINP	Section 7.1.6	300		kΩ
Tension de terminaison pour le rappel à la source du port orienté en amont (RPU)	VTERM	Section 7.1.5	3,0	3,6	V
Terminaisons à grande vitesse:					
Tension de terminaison à grande vitesse	VHSTERM	Section 7.1.6.2	-10	10	mV

Tableau 7-8. Caractéristiques électriques de source grande vitesse

Paramètre	Symbole	Conditions	Min.	Max.	Unités
Caractéristiques du pilote:					
Temps de montée (10 % - 90 %)	THSR	Section 7.1.2	500		ps
Temps de chute (10 % - 90 %)	THSF	Section 7.1.2	500		ps
Exigences de la forme d'onde du pilote		Spécifié par les modèles de diagramme en œil dans la Section 7.1.2			
Résistance de sortie du pilote (qui sert également de terminaison à grande vitesse)	ZHSDRV	Section 7.1.1.1	40,5	49,5	Ω
Synchronisation de l'horloge:					
Taux de données à grande vitesse	THSDRAT	Section 7.1.11	479,760	480,240	Mb/s
Intervalle de microtrame	THSFRAM	Section 7.1.12	124,9375	125,062 5	µs
Intervalle de microtrame consécutif Différence	THSRFI	Section 7.1.12		4 temps bit à grande vitesse	
Synchronisation des données à grande vitesse:					
Gigue de la source de données		La gigue de la source et du récepteur spécifiée par les modèles du diagramme en œil dans la Section 7.1.2.2			
Tolérance de gigue du récepteur					

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 7-9. Caractéristiques électriques de source vitesse intégrale

Paramètre	Symbole	Conditions	Min.	Max.	Unités
Caractéristiques du pilote:					
Temps de montée	TFR	Figure 7-8; Figure 7-9	4	20	ns
Temps de chute	TFF	Figure 7-8; Figure 7-9	4	20	ns
Correspondance des temps de montée et de chute différentielles	TFRFM	(TFR/TFF) note 10, Section 7.1.2	90	111,11	%
Résistance de sortie pour un pilote qui n'est pas en mesure d'opérer à grande vitesse	ZDRV	Section 7.1.1.1	28	44	Ω
Synchronisation de l'horloge:					
Débit de données à vitesse intégrale pour les hubs et appareils en mesure de fonctionner à grande vitesse	TFDRATHS	Débit binaire moyen, Section 7.1.11	11,9940	12,0060	Mb/s
Débit de données à vitesse intégrale pour les appareils qui ne sont pas en mesure de fonctionner à grande vitesse	TFDRATE	Débit binaire moyen, Section 7.1.11	11,9700	12,0300	Mb/s
Intervalle de trame	TFRAME	Section 7.1.12	0,9995	1,0005	ms
Gigue des intervalles de trame consécutifs	TRFI	Aucun réglage de l'horloge Section 7.1.12		42	ns
Synchronisation des données à vitesse intégrale:					
Total de gigue de la source (y compris la tolérance de fréquence): Vers la transition suivante Pour les transitions jumelées	 TDJ1 TDJ2	Note 7, 8, 12, 10; Mesuré comme dans la Figure 7-49;	 -3,5 -4	 3,5 4	 ns ns
Gigue de la source pour la transition différentielle vers la transition SE0	TFDEOP	Note 8; Figure 7-50; note 11	-2	5	ns
Gigue du récepteur: Vers la transition suivante Pour les transitions jumelées	 TJR1 TJR2	Note 8; Figure 7-51	 -18.5 -9	 18.5 9	 ns ns
Intervalle SE0 source de l'EOP	TFEOPT	Figure 7-50	160	175	ns
Intervalle SE0 du récepteur de l'EOP	TFEOPR	Note 13; Section 7.1.13.2; Figure 7-50	82		ns
Largeur de l'intervalle SE0 au cours de la transition différentielle	TFST	Section 7.1.4		14	ns

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 7-10. Caractéristiques électriques de source basse vitesse

Paramètre	Symbole	Conditions	Min.	Max.	Unités
Caractéristiques du pilote:					
Temps de transition: Temps de montée Temps de chute	 TLR TLF	 Mesuré comme à la Figure 7-8	 75 75	 300 300	 ns ns
Correspondance des temps de montée et de chute	TLRFM	(TLR/TLF) note 10	80	125	%
Port orienté en amont (avec câble, basse vitesse uniquement)	CLINUA	Note 1; Section 7.1.6	200	450	pF
Synchronisation de l'horloge:					
Débit de données à basse vitesse pour les hubs en mesure de fonctionner à grande vitesse	TLDRATHS	Section 7.1.11	1,49925	1,50075	Mb/s
Débit de données à basse vitesse pour les appareils qui ne sont pas en mesure de fonctionner à grande vitesse	TLDRATE	Section 7.1.11	1,4775	1,5225	Mb/s
Synchronisation des données à basse vitesse:					
Total de gigue source du port orienté en amont (y compris la tolérance de fréquence): Vers la transition suivante Pour les transitions jumelées	 TUDJ1 TUDJ2	Note 7, 8; Figure 7-49	 -95 -150	 95 150	 ns ns
Gigue source du port orienté en amont pour la transition différentielle vers la transition SE0	TLDEOP	Note 8; Figure 7-50; note 11	-40	100	ns
Gigue du récepteur différentiel du port orienté en amont: Vers la transition suivante Pour les transitions jumelées	 TDJR1 TDJR2	Note 8; Figure 7-51	 -75 -45	 75 45	 ns ns
Total de gigue source du port orienté en aval (y compris la tolérance de fréquence): Vers la transition suivante Pour les transitions jumelées	 TDDJ1 TDDJ2	Note 7, 8; Figure 7-49	 -25 -14	 25 14	 ns ns
Gigue source du port orienté en aval pour la transition différentielle vers la transition SE0		Note 8; Figure 7-50; note 11			ns
Gigue de récepteur différentiel du port orienté en aval: Vers la transition suivante Pour les transitions jumelées	 TUJR1 TUJR2	Note 8; Figure 7-50	 -152 -200	 152 200	 ns ns
Intervalle SE0 source de l'EOP	TLEOPT	Figure 7-50	1,25	1,50	µs
Intervalle SE0 du récepteur de l'EOP	TLEOPR	Note 13; Section 7.1.13.2; Figure 7-50	670		ns
Largeur de l'intervalle SE0 au cours de la transition différentielle	TLST	Section 7.1.4		210	ns

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 7-11. Caractéristiques électriques du hub/répéteur

Paramètre	Symbole	Conditions	Min.	Max.	Unités
Caractéristiques du hub à vitesse intégrale (comme mesuré au niveau des connecteurs):					
Caractéristiques du pilote: (Voir Tableau 7-9)		Port orienté en amont et ports orientés en aval configurés sous vitesse intégrale			
Délai des données différentielles du hub: (avec câble) (sans câble)	 THDD1 THDD2	Note 7, 8 Figure 7-52A Figure 7-52B		 70 44	 ns ns
Gigue du pilote différentiel du hub: (câble inclus) Vers la transition suivante Pour les transitions jumelées	 THDJ1 THDJ2	Note 7, 8; Figure 7-52, Section 7.1.14	 -3 -1	 3 1	 ns ns
Distorsion de la largeur du bit de données après SOP	TFSOP	Note 8; Figure 7-52	-5	5	ns
Délai de l'EOP de hub lié à THDD	TFEOPD	Note 8; Figure 7-53	0	15	ns
Dérivation de largeur de sortie de l'EOP de hub	TFHESK	Note 8; Figure 7-53	-15	15	ns
Caractéristiques du hub à basse vitesse (comme mesuré au niveau des connecteurs):					
Caractéristiques du pilote: (Voir Tableau 7-10)		Ports orientés en aval configurés sous basse vitesse			
Délai des données différentielles du hub	TLHDD	Note 7, 8; Figure 7-52		300	ns
Gigue du pilote différentiel du hub (câble inclus): Port orienté en aval: Vers la transition suivante Pour les transitions jumelées Port orienté en amont: Vers la transition suivante Pour les transitions jumelées	 TLDHJ1 TLDHJ2 TLUHJ1 TLUHJ2	Note 7, 8; Figure 7-52	 -45 -15 -45 -45	 45 15 45 45	 ns ns ns ns
Distorsion de la largeur du bit de données après SOP	TLSOP	Note 8; Figure 7-52	-60	60	ns
Délai de l'EOP de hub lié à THDD	TLEOPD	Note 8; Figure 7-53	0	200	ns
Dérivation de largeur de sortie de l'EOP de hub	TLHESK	Note 8; Figure 7-53	-300	+300	ns
Caractéristiques du hub à grande vitesse (comme mesuré au niveau des connecteurs):					
Caractéristiques du pilote: (Voir Tableau 7-8)		Port orienté en amont et ports orientés en aval configurés sous grande vitesse			
Délai des données du hub (sans câble):	THSHDD	Section 7.1.14.2		36 temps bit à grande vitesse + 4 ns	
Gigue des données du hub:		Spécifié par les diagrammes en œil dans la Section 7.1.2.2			

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Gamme de variations du délai de hub:	THSHDV	Section 7.1.14.2		5 temps bit à grande vitesse	

Tableau 7-12. Caractéristiques du câble (Note 14)

Paramètre	Symbole	Conditions	Min	Max	Unités
Chute de tension du VBUS pour les câbles détachables	VBUSD	Section 7.2.2		125	mV
Chute de tension GND (pour tous les câbles)	VGNDD	Section 7.2.2		125	mV
Impédance différentielle du câble (vitesse intégrale/grande vitesse)	ZO	(90 Ω ± 15 %);	76,5	103,5	Ω
Impédance différentielle du câble de mode commun (vitesse intégrale/grande vitesse)	ZCM	(30 Ω ± 30 %);	21,0	39,0	Ω
Délai du câble (un sens)		Section 7.1.16			
Vitesse intégrale/grande vitesse	TFSCBL			26	ns
Basse vitesse	TLSCBL			18	ns
Dérivation de câble	TSKEW	Section 7.1.3		100	ps
Capacité du contact débranché	CUC	Section 6.7		2	pF
Affaiblissement du câble		Spécifié par le tableau et le graphique dans la Section 7.1.17			

Note 1: Mesuré au niveau de la fiche.

Note 2: Mesuré au niveau du réceptacle A.

Note 3: Mesuré au niveau du réceptacle B.

Note 4: Mesuré au niveau du connecteur A ou B.

Note 5: Mesuré avec un RL de 1.425 kΩ à 3,6 V.

Note 6: Mesuré avec un RL de 14,25 kΩ au GND.

Note 7: Différence de synchronisation entre les signaux de données différentiels.

Note 8: Mesuré au point de croisement des signaux de données différentiels.

Note 9: La spécification de charge maximale correspond à la charge capacitive effective maximum autorisée qui satisfait à la chute de 330 mV du VBUS cible.

Note 10: Sauf la première transition à partir de l'état de repos.

Note 11: Il convient que les deux transitions correspondent à une séparation de temps bit (nominal).

Note 12: Pour les deux transitions de signalisation différentielle.

Note 13: Doit l'accepter en tant qu'EOP valide.

Note 14: La capacité à extrémité unique de D+ ou D- correspond à la capacité de D+/D- pour tous les autres conducteurs et le cas échéant le blindage du câble. C'est-à-dire, afin de pouvoir mesurer ensemble la capacité à extrémité unique de D+, D- court, VBUS, GND et de la ligne blindée et pour mesurer la capacité de D+ pour tous les autres conducteurs.

Note 15: Pour les appareils à grande puissance (pas les hubs) lors qu'ils sont activés pour le réveil à distance.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 7-13. Synchronisations d'évènement d'un hub

Description de l'évènement	Symbole	Conditions	Min	Max	Unit
Temps pour détecter un évènement de connexion du port orienté en aval	TDCNN	Section 11.5 et Section 7.1.7.3			
Hub réveillé			2,5	2 000	µs
Hub interrompu			2,5	12 000	µs
Temps pour détecter un évènement de déconnexion au niveau du port orienté en aval du hub	TDDIS	Section 7.1.7.3	2	2,5	µs
Durée de la conduite de la reprise vers un port en aval; uniquement à partir d'un hub de contrôle	TDRSMDN	Nominal; Section 7.1.7.7 et Section 11.5	20		ms
Temps pour la détection de la reprise en aval jusqu'à la rediffusion	TURSM	Section 7.1.7.7		1,0	ms
Durée de la conduite de réinitialisation pour un port orienté en aval	TDRST	Uniquement pour une requête SetPortFeature (PORT_RESET); Section 7.1.7.5 et Section 11.5	10	20	ms
Durée globale de la conduite de réinitialisation pour un port orienté en aval, un hub racine	TDRSTR	Uniquement pour les hubs racines; Section 7.1.7.5	50		ms
Intervalle maximum entre les segments de réinitialisation utilisés pour créer TDRSTR	TRHRSI	Uniquement pour les hubs racines; chaque impulsion de réinitialisation doit être de la longueur de TDRST; Section 7.1.7.5		3	ms
Temps pour la détection d'un K long, à partir de l'aval	TURLK	Section 11.6	2,5	100	µs
Temps pour la détection d'un SE0 long, à partir de l'aval	TURLSE0	Section 11.6	2,5	10 000	µs
Durée de la répétition SE0 en amont (pour les répéteurs basse vitesse/vitesse intégrale)	TURPSE0	Section 11.6		23	Temps bit FS
Durée de l'envoi SE0 en amont après l'EOF1 (pour les répéteurs basse vitesse/vitesse intégrale)	TUDEOP	Section 11.6 *en option*		2	Temps bit FS
Délai interpaquet (pour la grande vitesse) pour les paquets voyageant dans le même sens	THSIPDSD	Section 7.1.18.2	88		temps bit
Délai interpaquet (pour la grande vitesse) pour les paquets voyageant dans un sens opposé	THSIPDOD	Section 7.1.18.2	8		temps bit
Délai interpaquet pour la réponse de l'appareil/du hub racine doté d'un câble détachable, pour la grande vitesse	THSRSPIPD1	Section 7.1.18.2		192	temps bit

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 7-13. Synchronisations d'évènement d'un hub (suite)

Description de l'évènement	Symbole	Conditions	Min	Max	Unité
Protocoles d'établissement d'une liaison de réinitialisation:					
Temps pendant lequel un gazouillis J ou un gazouillis K doit être détecté en continu (filtré) par le hub ou l'appareil au cours de l'établissement d'une liaison de réinitialisation	TFILT	Section 7.1.7.5	2,5		µs
Temps après la fin du gazouillis K d'appareil au cours duquel un hub doit commencer à entraîner le premier gazouillis K dans la séquence de gazouillis du hub	TWTDCH	Section 7.1.7.5		100	µs
Temps pendant lequel chaque gazouillis J ou K individuel dans la séquence de gazouillis est conduit en aval par le hub, au cours de la réinitialisation	TDCHBIT	Section 7.1.7.5	40	60	µs
Temps avant la fin de la réinitialisation au cours duquel un hub doit terminer sa séquence de gazouillis en aval	TDCHSE0	Section 7.1.7.5	100	500	µs

Tableau 7-14. Synchronisations d'évènement d'un appareil

Paramètre	Symbole	Conditions	Min	Max	Unités
Temps à partir duquel la puissance interne adaptée à l'appareil tire D+/D- au-delà de VIHZ (min) (connexion de signalisation)	TSIGATT	Figure 7-29		100	ms
Intervalle antirebond fourni par le logiciel du système USB après connexion	TATTDB	Figure 7-29		100	ms
Temps maximum pendant lequel un appareil peut appeler du courant >courant de veille lorsque le bus se trouve en continu dans l'état de repos	T2SUSP	Section 7.1.7.6		10	ms
Durée maximale de l'intervalle de moyennage interrompu	TSUSAVGI	Section 7.2.3		1	s
Période de bus inexploité avant que l'appareil puisse lancer une reprise	TWTRSM	L'appareil doit être activé pour le réveil à distance Section 7.1.7.5	5		ms
Durée de la conduite de la reprise en amont	TDRSMUP	Section 7.1.7.7	1	15	ms
Temps de récupération sur reprise	TRSMRCY	Fourni par le logiciel du système USB; Section 7.1.7.7	10		ms
Temps pour détecter une réinitialisation à partir de l'amont pour les appareils qui ne sont pas en mesure de fonctionner à grande vitesse	TDETRST	Section 7.1.7.5	2,5	10 000	µs
Temps de récupération sur réinitialisation	TRSTRCY	Section 7.1.7.5		10	ms
Délai interpaquet (pour la basse vitesse/vitesse intégrale)	TIPD	Section 7.1.18	2		temps bit
Délai interpaquet pour la réponse de l'appareil doté d'un câble détachable, pour la basse vitesse/vitesse intégrale	TRSPIPD1	Section 7.1.18		6,5	temps bit
Délai interpaquet pour la réponse de l'appareil doté d'un câble captif, pour la basse vitesse/vitesse intégrale	TRSPIPD2	Section 7.1.18		7,5	temps bit

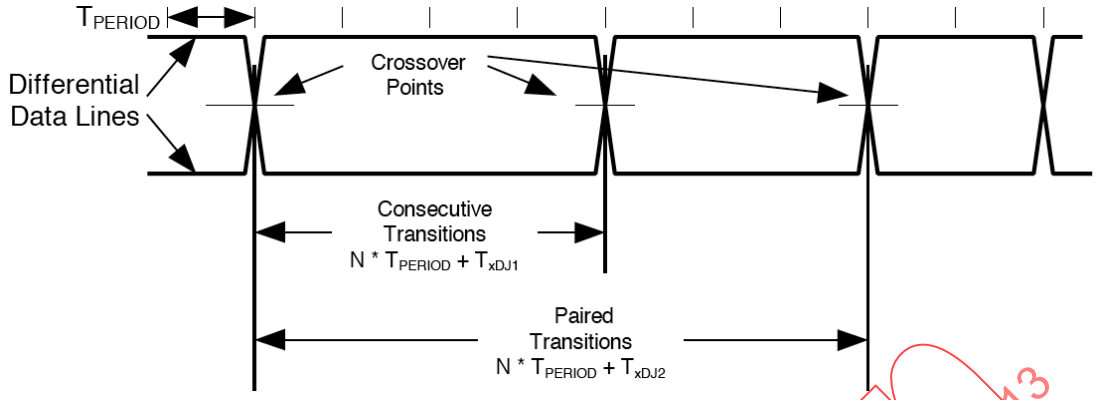
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 7-14. Synchronisations d'évènement d'un appareil (suite)

Paramètre	Symbole	Conditions	Min	Max	Unités
Temps pour l'achèvement de SetAddress()	TDSETADDR	Section 9.2.6.3		50	ms
Temps pour l'achèvement d'une requête normalisée sans données	TDRQCMPLTND	Section 9.2.6.4		50	ms
Temps pour livrer la première donnée et les données ultérieures (sans la dernière) pour une requête normalisée	TDRETDATA1	Section 9.2.6.4		500	ms
Temps pour livrer la dernière donnée pour une requête normalisée	TDRETDATAN	Section 9.2.6.4		50	ms
Délai interpaquet pour la réponse de l'appareil doté d'un câble captif, (grande vitesse)	THSRSPIPD2	Section 7.1.18.2		192 temps bit + 52 ns	
Temps pour l'achèvement de SetAddress()	TDSETADDR	Section 9.2.6.3		50	ms
Temps pour l'achèvement d'une requête normalisée sans données	TDRQCMPLTND	Section 9.2.6.4		50	ms
Protocoles d'établissement d'une liaison de réinitialisation:					
Temps pendant lequel un appareil interrompu, capable de fonctionner à grande vitesse doit observer un SE0 continu avant de débuter l'établissement d'une liaison de détection à grande vitesse	TFILTSE0	Section 7.1.7.5	2,5		µs
Temps pendant lequel un appareil capable de fonctionner à grande vitesse et opérant en mode à vitesse intégrale non interrompu doit patienter après le démarrage d'un SE0 avant de débuter l'établissement d'une liaison de détection à grande vitesse	TWTRSTFS	Section 7.1.7.5	2,5	3 000	µs
Temps pendant lequel un appareil capable de fonctionner à grande vitesse et opérant à grande vitesse doit patienter après le démarrage d'un SE0 avant de revenir en mode à vitesse intégrale	TWTREV	Section 7.1.7.5	3,0	3,125	ms
Temps pendant lequel un appareil doit patienter après être revenu en mode à vitesse intégrale, avant de contrôler l'état du bus pour un SE0 et débuter l'établissement d'une liaison de détection à grande vitesse	TWTRSTHS	Section 7.1.7.5	100	875	µs
Durée minimum d'un gazouillis K en provenance d'un appareil capable de fonctionner à grande vitesse dans le protocole de réinitialisation	TUCH	Section 7.1.7.5	1,0		ms
Temps après le démarrage d'un SE0 pendant lequel il faut qu'un appareil capable de fonctionner à grande vitesse achève son gazouillis K dans le protocole de réinitialisation.	TUCHEND	Section 7.1.7.5		7,0	ms
Temps entre la détection d'un gazouillis en aval et l'entrée en état à grande vitesse	TWTHS	Section 7.1.7.5		500	µs
Temps après la fin d'un gazouillis en amont pendant lequel un appareil revient à son état à vitesse intégrale par défaut, si aucun gazouillis en aval n'est détecté	TWTFS	Section 7.1.7.5	1,0	2,5	ms

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

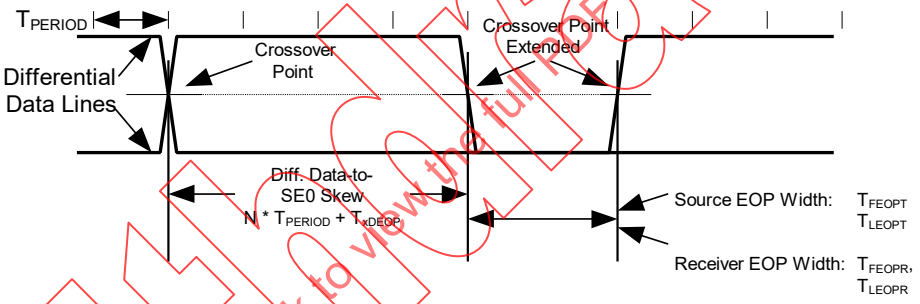
7.3.3 Formes d'ondes de synchronisation



Légende

Anglais	Français
Differential Data Lines	Lignes de données différentielles
Crossover Points	Points de croisement
Consecutive Transitions	Transitions consécutives
Paired Transitions	Transitions jumelées

Figure 7-49. Gigue de données différentielle à basse vitesse/vitesse intégrale

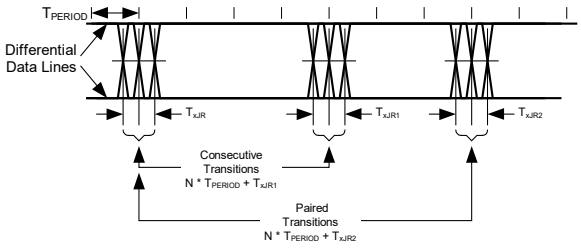


Légende

Anglais	Français
Differential Data Lines	Lignes de données différentielles
Crossover Points	Points de croisement
Crossover Point Extended	Points de croisement élargis
Diff. Data-to-SE0 Skew	Dérivation différentielle données à SE0
Source EOP Width	Largeur de source d'EOP
Receiver EOP Width	Largeur de récepteur d'EOP

Figure 7-50. Dérivation de transition différentiel-à-EOP et largeur EOP pour basse vitesse/vitesse intégrale

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



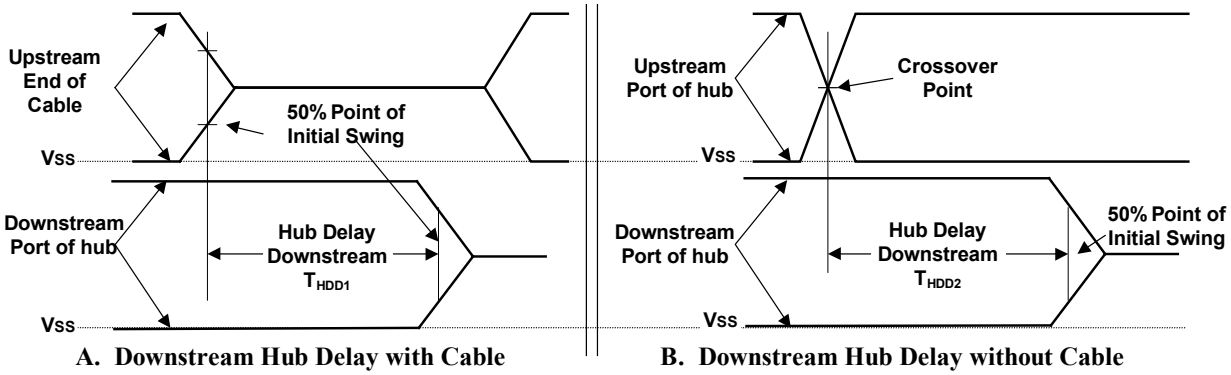
Légende

Anglais	Français
Differential Data Lines	Lignes de données différentielles
Consecutive Transitions	Transitions consécutives
Paired Transitions	Transitions jumelées

Figure 7-51. Tolérance de gigue du récepteur à basse vitesse/vitesse intégrale

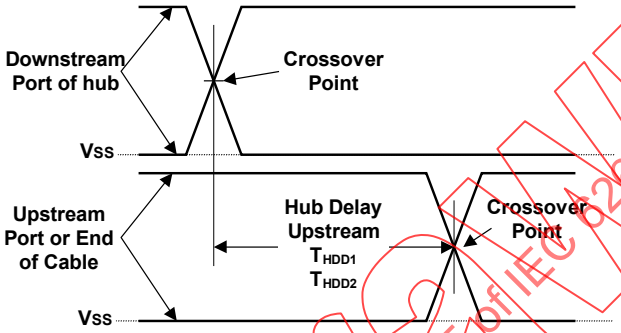
TPERIOD correspond au débit de données du récepteur qui peut contenir la gamme définie dans la Section 7.1.11.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



A. Downstream Hub Delay with Cable

B. Downstream Hub Delay without Cable



C. Upstream Hub Delay with or without Cable

Hub Differential Jitter:

$T_{HDJ1} = T_{HDDx}(J) - T_{HDDx}(K)$ or $T_{HDDx}(K) - T_{HDDx}(J)$ Consecutive Transitions

$T_{HDJ2} = T_{HDDx}(J) - T_{HDDx}(J)$ or $T_{HDDx}(K) - T_{HDDx}(K)$ Paired Transitions

Bit after SOP Width Distortion (same as data jitter for SOP and next J transition):

$T_{FSOP} = T_{HDDx}$(next J) - T_{HDDx}(SOP)

Low-speed timings are determined in the same way for:

T_{LHDD}, T_{LDHJ1}, T_{LDJH2}, T_{LUHJ1}, T_{LUJH2}, and T_{LSOP}

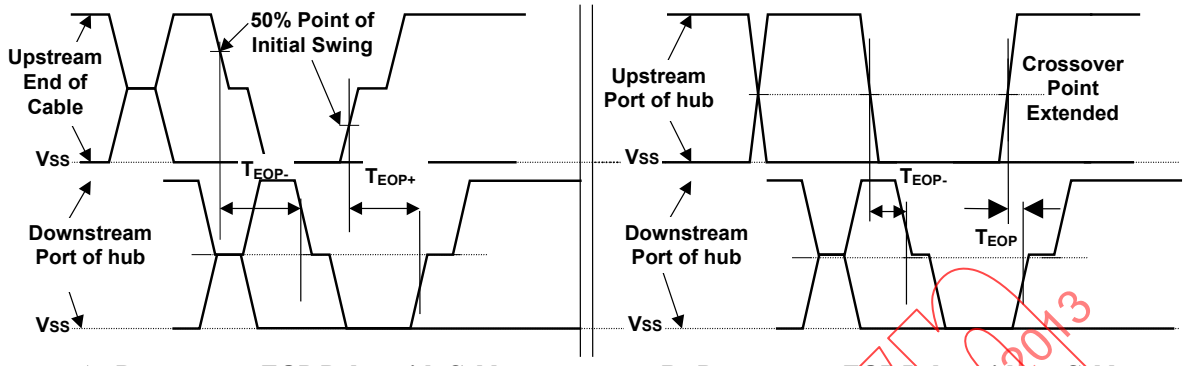
Légende

Anglais	Français
Upstream End of Cable	Extrémité en amont du câble
50% Point of Initial Swing	50 % du point de la variation initiale
Upstream Port of hub	Port en amont du hub
Crossover Point	Point de croisement
Downstream Port of hub	Port en aval du hub
Hub Delay Downstream T_{HDD1}	Délai de hub en aval T_{HDD1}
Hub Delay Downstream T_{HDD2}	Délai de hub en aval T_{HDD2}
Downstream Hub Delay with Cable	Délai de hub en aval avec câble
Downstream Hub Delay without Cable	Délai de hub en aval sans câble
Upstream Port or End of Cable	Port en amont ou extrémité de câble
Hub Delay Upstream	Délai de hub en amont
Upstream Hub Delay with or without Cable	Délai de hub en amont avec ou sans câble
Hub Differential Jitter	Gigue différentielle du hub
Consecutive Transitions	Transitions consécutives
Paired Transitions	Transitions jumelées
Bit after Width Distortion (same as data jitter for SOP and next J transition)	Bit après distorsion de la largeur (identique à la gigue de données du SOP et la transition J suivante)
Low-speed timings are determined in the same way for:	La synchronisation à basse vitesse est déterminée de la même manière pour:

Figure 7-52. Délai différentiel du hub, gigue différentielle et distorsion SOP pour la basse vitesse/vitesse intégrale

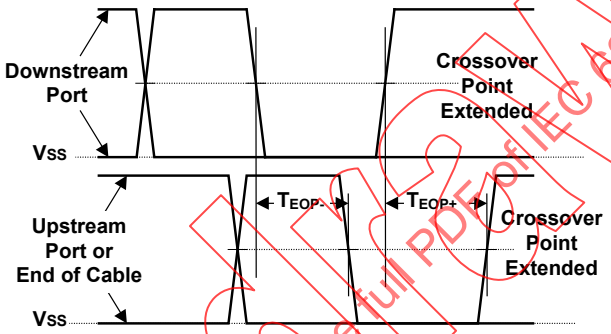
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Les points de mesure référencés dans la Figure 7-52 et la Figure 7-53 sont spécifiés dans la Figure 7-38.



A. Downstream EOP Delay with Cable

B. Downstream EOP Delay without Cable



C. Upstream EOP Delay with or Without Cable

EOP Delay:

$T_{FEOPD} = T_{EOPy} - T_{HDDx}$

(T_{EOPy} means that this equation applies to T_{EOP-} and T_{EOP+})

EOP Skew:

$T_{FHESK} = T_{EOP+} - T_{EOP-}$

Low-speed timings are determined in the same way for:

T_{LEOPD} and T_{LHESK}

Légende

Anglais	Français
Upstream End of Cable	Extrémité en amont du câble
50% Point of Initial Swing	50 % du point de la variation initiale
Upstream Port of hub	Port en amont du hub
Crossover Point Extended	Point de croisement élargi
Downstream Port of hub	Port en aval du hub
Downstream EOP Delay with Cable	Délai de l'EOP en aval avec câble
Downstream EOP Delay without Cable	Délai de l'EOP en aval sans câble
Downstream Port	Port en aval
Upstream Port or End of Cable	Port en amont ou extrémité du câble
Upstream EOP Delay with or without Cable	Délai de l'EOP en amont avec ou sans câble

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
EOP Delay	Délai de l'EOP
T_{EOPY} means that this equation applies to T_{EOP} and T_{EOP+}	T_{EOPY} signifie que cette équation s'applique à T_{EOP} et T_{EOP+}
EOP Skew	Dérivation de l'EOP
Low-speed timings are determined in the same way for:	La synchronisation à basse vitesse est déterminée de la même manière pour:
T_{LEOPD} and T_{LHESK}	T_{LEOPD} et T_{LHESK}

Figure 7-53. Délai EOP du hub et dérivation EOP pour basse vitesse/vitesse intégrale

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8 Chapitre 8
Couche de protocole

Ce chapitre présente une vue de bas en haut du protocole USB, en débutant par les définitions de champ et de paquet. Cette introduction est suivie par une description des formats de transaction de paquet pour les différents types de transaction. Le contrôle du flux de la couche de liaison et la récupération en cas d'anomalie au niveau de la transaction sont également couverts. Le chapitre termine par une discussion sur la synchronisation de la relance, le bruit diaphonique, la récupération de l'activité en cas de perte du bus et le protocole PING à grande vitesse.

8.1 Ordonnancement des octets/bits

Les bits sont tout d'abord diffusés sur le bit du bus le moins significatif (LSb), puis par le LSb suivant, jusqu'au bit le plus significatif (MSb), en dernier. Dans les schémas suivants, les paquets sont affichés de manière à ce que les bits et champs individuels soient représentés (dans un ordre de lecture de gauche à droite) de la façon dont ils se déplacent à travers le bus.

De multiples champs d'octets dans les descripteurs, requêtes et réponses normalisés sont interprétés en tant que petite file indienne et déplacés dans cet ordre sur le bus, c'est-à-dire de LSB à MSB.

8.2 Champ SYNC

Tous les paquets débutent par un champ de synchronisation (SYNC) qui correspond à une séquence codée qui génère une densité de bord de transition maximale. Elle est utilisée par le circuit d'entrée pour aligner les données entrantes sur l'horloge locale. Un SYNC à partir d'un émetteur initial est défini sur huit bits de longueur pour la vitesse intégrale/basse vitesse et sur 32 bits pour la grande vitesse. Les champs SYNC reçus peuvent être plus courts que ceux décrits au Chapitre 7. SYNC sert uniquement de mécanisme de synchronisation et n'est pas présenté dans les schémas de paquet suivants (voir la Section 7.1.10). Les deux derniers bits dans les champs SYNC correspondent à un marqueur qui est utilisé pour identifier la fin du champ SYNC, et par voie de conséquence, le début du PID.

8.3 Formats des champs de paquets

Les formats de champs pour les paquets de jetons, de données et d'établissements de liaison sont décrits dans la section suivante. Les définitions des bits de paquets sont affichées dans un format de données non codées. Les effets du codage NRZI et de l'insertion binaire ont été supprimés par souci de clarté. Tous les paquets disposent de délimiteurs de début et de fin de paquets distincts. Le délimiteur du début de paquet (SOP) fait partie du champ SYNC et le délimiteur de fin de paquet (EOP) est décrit dans le chapitre 7.

8.3.1 Champ de l'identificateur de paquets

Un identificateur de paquets (PID) suit immédiatement le champ SYNC sur chaque paquet USB. Un PID comprend un champ de type de paquet à quatre bits, suivi par un champ de contrôle des quatre bits, comme illustré dans la Figure 8-1. Le PID indique le type de paquets et, par voie de conséquence, le format du paquet et le type de détection des erreurs appliqué au paquet. Le champ de contrôle des quatre bits du PID garantit un décodage fiable du PID, de manière à ce que le reste du paquet soit interprété de manière correcte. Le champ de contrôle du PID est généré en effectuant un complément à un du champ de type de paquets. Une erreur de PID surgit si les quatre bits de contrôle du PID ne correspondent pas à un complément de leurs bits respectifs d'identification du paquet.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

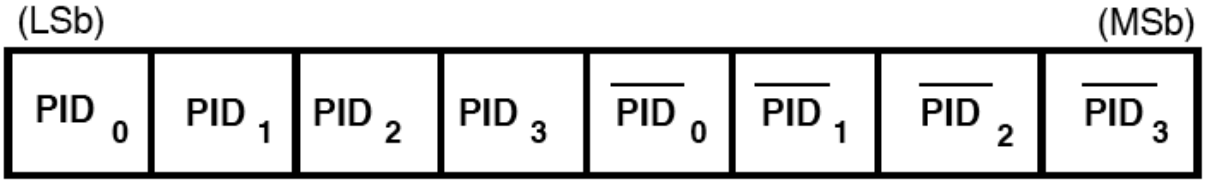


Figure 8-1. Format PID

L'hôte et toutes les fonctions doivent effectuer un décodage complet de tous les champs PID reçus. Tout PID reçu avec un champ de contrôle défaillant ou qui décode sur une valeur non définie est considéré comme corrompu et doit, ainsi que le reste du paquet, être ignoré par le récepteur du paquet. Si une fonction reçoit un PID valide contraire pour un type de transaction ou pour un sens qu'elle ne prend pas en charge, la fonction ne doit pas répondre. Par exemple, un point d'extrémité uniquement IN doit ignorer un jeton OUT. Les types de PID, de codages et de descriptions sont répertoriés dans le le Tableau 8-1.

Tableau 8-1. Types de PID

Type de PID	Nom de PID	PID<3:0>*	Description
jeton	OUT	0001B	Adresse + nombre de points d'extrémité dans la transaction d'hôte à fonction
	IN	1001B	Adresse + nombre de points d'extrémité dans la transaction de fonction à hôte
	SOF	0101B	Marqueur de début de trame et numéro de trame
	SETUP	1101B	Adresse + nombre de points d'extrémité dans la transaction d'hôte à fonction pour SETUP à une canalisation de contrôle
Données	DATA0	0011B	PID pair de paquet de données
	DATA1	1011B	PID impair de paquet de données
	DATA2	0111B	Paquet de données à PID haute vitesse, transaction isochrone de bande supérieure dans une microtrame (voir la Section 5.9.2 pour en savoir plus)
	MDATA	1111B	Paquet de données à PID haute vitesse pour les transactions divisées isochrones de bande supérieure (voir les Sections 5.9.2, 11.20 et 11.21 pour en savoir plus)
Etablissement de liaison	ACK	0010B	Le récepteur accepte le paquet de données exempt d'erreur
	NAK	1010B	L'appareil récepteur ne peut pas accepter des données ou l'appareil émetteur ne peut pas envoyer des données
	STALL	1110B	Le point d'extrémité est stoppé ou une demande de canalisation de contrôle n'est pas prise en charge
	NYET	0110B	Pas encore de réponse de la part du récepteur (voir les Sections 8.5.1 et 11.17-11.21)
Spécial	PRE	1100B	(Jeton) Préambule émis par l'hôte. Active le trafic sur le bus en aval pour les appareils à basse vitesse.
	ERR	1100B	(Etablissement de liaison) Erreur d'établissement de liaison pour la transaction divisée (réutilise la valeur PRE)
	SPLIT	1000B	(Jeton) Jeton de transaction divisée à haute vitesse (voir la Section 8.4.2)
	PING	0100B	(Jeton) Sonde de contrôle de débit à haute vitesse pour un volume/point d'extrémité de contrôle (voir la Section 8.5.1)
	Réservé	0000B	PID réservé

Note: Les bits de PID sont indiqués dans l'ordre du bit de poids fort (MSb). Quand ils sont envoyés sur l'USB, le bit le plus à droite (bit 0) est envoyé en premier.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Les PID sont divisés en quatre groupes de codage: jeton, données, établissement de liaison et spécial avec les deux premiers bits de PID transmis (PID<0:1>) indiquant le groupe. Cela représente la distribution des codes de PID.

8.3.2 Champs d'adresse

Les points d'extrémité de fonction sont traités avec deux champs: le champ d'adresse de la fonction et le champ de point d'extrémité. Une fonction doit décoder entièrement les champs d'adresse et de point d'extrémité. Le repliement spectral de l'adresse ou du point d'extrémité vie n'est pas admis, et le jeton est ignoré en cas d'incohérence sur l'un des champs. En cas d'accès à des points d'extrémité non initialisés, le jeton est également ignoré.

8.3.2.1 Champ d'adresse

Le champ d'adresse de la fonction (ADDR) spécifie la fonction grâce à son adresse, qui est soit la source soit la destination d'un paquet de données, en fonction de la valeur du PID du jeton. Comme indiqué dans la Figure, 128 adresses au total sont spécifiées en tant que ADDR<6:0>. Le champ ADDR est spécifié pour les jetons IN, SETUP et OUT, et le jeton spécial PING et SPLIT. Par définition, chaque valeur ADDR définit une seule fonction. A la réinitialisation et au démarrage, l'adresse d'une fonction se règle à une valeur de zéro et doit être programmée par l'hôte pendant le processus d'énumération. L'adresse de fonction zéro est réservée en tant qu'adresse par défaut et ne peut être assignée à aucune autre utilisation.

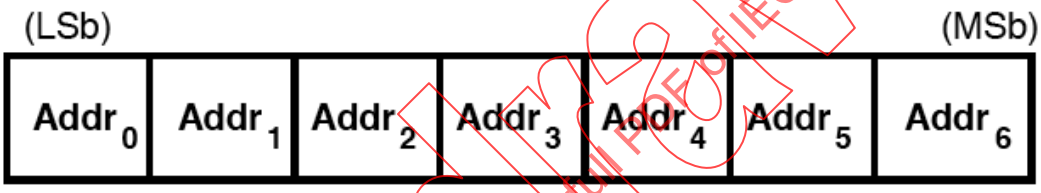


Figure 8-2. Champ ADDR

8.3.2.2 Champ de point d'extrémité

Un champ de point d'extrémité à quatre bits (ENDP) supplémentaire, indiqué dans la Figure 8-3 permet de traiter de manière plus souple des fonctions où plus d'un point d'extrémité est nécessaire. Sauf pour l'adresse zéro de point d'extrémité, les nombres de points d'extrémité sont spécifiques à la fonction. Le champ de point d'extrémité est défini pour les jetons IN, SETUP et OUT, et le jeton spécial PING. Toutes les fonctions doivent prendre en charge une canalisation de contrôle au nombre zéro de point d'extrémité (la canalisation de contrôle par défaut). Les appareils à basse vitesse prennent en charge un maximum de trois canalisations par fonction: une canalisation de contrôle à un nombre zéro de point d'extrémité plus deux canalisations supplémentaires (soit deux canalisations de contrôle, une canalisation de contrôle et un point d'extrémité d'interruption, soit deux points d'extrémité d'interruption). Les fonctions à vitesse intégrale et à haute vitesse peuvent prendre en charge jusqu'à 16 points d'extrémité IN et OUT.

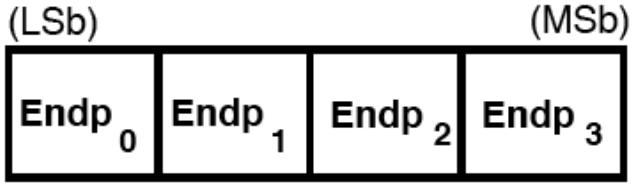


Figure 8-3. Champ de point d'extrémité

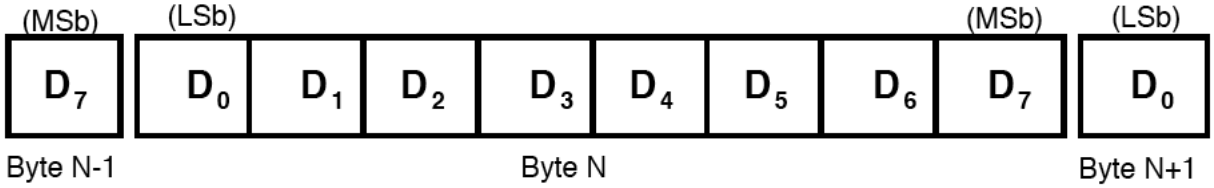
8.3.3 Champ du nombre de trames

Le champ du nombre de trames est un champ à 11 bits que l'hôte incrémente trame par trame. Le champ du nombre de trames se réinitialise après avoir atteint une valeur maximale de 7FFH et n'est envoyé que dans des jetons SOF au début de chaque (micro)trame.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8.3.4 Champs de données

Le champ de données peut aller de zéro à 1 024 octets et doit être un nombre entier d'octets. La Figure 8-4 indique le format pour plusieurs octets. Les bits de données au sein de chaque octet sont d'abord sortis du bit de poids faible (LSb).



Légende

Anglais	Français
Byte	Octet

Figure 8-4. Format du champ de données

La taille des paquets de données varie selon le type de transfert, comme décrit dans le chapitre 5.

8.3.5 Contrôles de redondance cyclique

Les contrôles de redondance cyclique (CRC) servent à protéger tous les champs non PID dans les paquets de jetons et de données. Dans ce contexte, ces champs sont considérés comme des champs protégés. Le PID n'est pas inclus dans le contrôle CRC d'un paquet contenant un CRC. Tous les CRC sont générés sur leurs champs respectifs dans l'émetteur avant l'insertion de bits. De manière similaire, les CRC sont décodés dans le récepteur après la suppression des bits insérés. Les CRC des paquets de jetons et de données couvrent à 100 % toutes les erreurs de bit simple et double. On considère qu'un CRC échoué indique qu'un ou plusieurs champs protégés sont corrompus et fait que le récepteur ignore ces champs, et la plupart du temps tout le paquet.

Pour la génération et la vérification du CRC, les registres à décalage dans le générateur et le vérificateur sont classés dans un motif tout à un. Pour chaque bit de données envoyé ou reçu, le bit de poids le plus fort du reste actuel est XORé avec le bit de données puis le reste est déplacé à gauche d'un bit et le bit de poids le plus faible se met à zéro. Si le résultat de ce XOR est un, alors le reste est XORé avec le polynôme générateur.

Quand le dernier bit du champ vérifié est envoyé, le CRC dans le générateur est inversé et envoyé d'abord au vérificateur du bit de poids fort (MSb). Quand le dernier bit du champ vérifié est reçu par le vérificateur et si aucune erreur ne s'est produite, le reste est égal au reliquat du polynôme.

Il existe une erreur CRC si le reste de la somme de contrôle calculé à la fin de la réception d'un paquet ne correspond pas au reliquat.

Les exigences d'insertion de bits doivent être satisfaites pour le CRC, et cela inclut la nécessité d'insérer un zéro à la fin d'un CRC si les six bits précédents ont tous été faits.

8.3.5.1 CRC de jetons

Un champ de CRC à cinq bits est fourni pour les jetons et couvre les champs ADDR et ENDP des jetons IN, SETUP et OUT ou le champ d'horodatage d'un jeton SOF. Les jetons spéciaux PING et SPLIT incluent également un champ de CRC à cinq bits. Le polynôme générateur est:

$$G(X) = X^5 + X^2 + 1$$

Le motif binaire qui représente ce polynôme est 00101B. Si tous les bits de jetons sont reçus sans erreur, le reliquat à cinq bits au niveau du récepteur est 01100B.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8.3.5.2 CRC de données

Le CRC de données est un polynôme à 16 bits appliqué au champ de données d'un paquet de données. Le polynôme générateur est:

$$G(X) = X^{16} + X^{15} + X^2 + 1$$

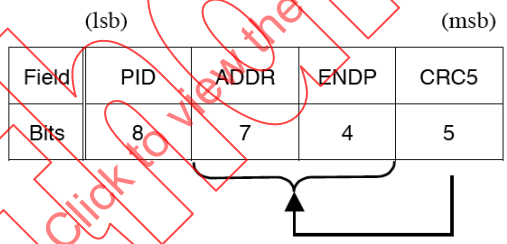
Le motif binaire qui représente ce polynôme est 1000000000000101B. Si toutes les données et bits de CRC sont reçus sans erreur, le reliquat à 16 bits est 1000000000001101B.

8.4 Formats du paquet

Cette section montre des formats de paquet pour les paquets de jeton, de données et d'établissement de liaison. Les champs au sein d'un paquet sont indiqués dans ces figures dans l'ordre où les bits sont sortis sur le bus.

8.4.1 Paquets de jetons

La Figure 8-5 montre les formats des champs pour un paquet de jetons. Un jeton est constitué d'un PID, précisant soit le type de paquet IN, OUT ou SETUP et les champs ADDR et ENDP. Le paquet de jetons spécial PING a également les mêmes champs qu'un paquet de jetons. Pour les transactions OUT et SETUP, les champs d'adresse et de point d'extrémité identifient de manière unique le point d'extrémité qui reçoit le paquet de données ultérieures. Pour les transactions IN, ces champs identifient de manière unique à quel point d'extrémité il convient de transmettre un paquet de données. Pour les transactions PING, ces champs identifient de manière unique quel point d'extrémité répond avec un paquet d'établissement de liaison. Seul l'hôte peut émettre des paquets de données. Un PID IN définit une transaction de données d'une fonction vers l'hôte. Des PID OUT et SETUP définissent des transactions de données de l'hôte vers une fonction. Un PID PING définit une transaction d'établissement de liaison de la fonction vers l'hôte.



Légende

Anglais	Français
Field	Champ

Figure 8-5. Format du jeton

Les paquets de données ont un CRC à cinq bits qui couvre les champs d'adresse et de point d'extrémité comme indiqué ci-dessus. Le CRC ne couvre pas le PID, qui a son propre champ de vérification. Les paquets de jetons et SOF sont délimités par un EOP après trois octets de données de champ de paquet. Si un paquet se décode comme un jeton ou un SOF autrement valide, mais ne se termine pas par un EOP après trois octets, il doit être considéré comme invalide et ignoré par le récepteur.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

8.4.2 Paquet de jetons spécial de transaction divisée

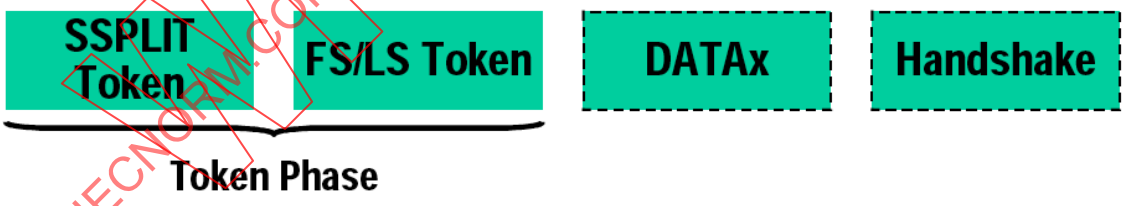
USB définit un jeton spécial pour les transactions divisées: SPLIT. Il s'agit d'un paquet de jetons à 4 octets comparé à d'autres paquets normaux de jeton à 3 octets. Le paquet de jetons de transaction divisée fournit des types supplémentaires de transactions avec des informations supplémentaires spécifiques à la transaction. Le jeton de transaction divisée sert à prendre en charge les transactions divisées entre le contrôleur d'hôte qui communique avec un hub fonctionnant à haute vitesse avec des appareils à vitesse intégrale/basse vitesse vers certains de ses ports face à l'aval. Il y a deux transactions divisées définies qui utilisent le jeton spécial SPLIT: une transaction divisée de début (SSPLIT) et une transaction divisée de fin (CSPLIT). Un champ dans le jeton spécial SPLIT, décrit dans les sections suivantes, indique la transaction divisée spécifique.

8.4.2.1 Transactions de scission

Une transaction divisée à haute vitesse est utilisée uniquement entre le contrôleur d'hôte et un hub quand des appareils à vitesse intégrale/basse vitesse sont raccordés au hub. Cette transaction divisée à haute vitesse est utilisée pour initier une transaction à vitesse intégrale/basse vitesse via le hub et un certain point d'extrémité de l'appareil à vitesse intégrale/basse vitesse. La transaction divisée à haute vitesse permet également de récupérer du hub l'état d'avancement de la transaction à vitesse intégrale/basse vitesse. Cette approche permet au contrôleur d'hôte de démarrer une transaction à vitesse intégrale/basse vitesse via une transaction à haute vitesse puis de continuer avec d'autres transactions à haute vitesse sans devoir attendre que la transaction à vitesse intégrale/basse vitesse se poursuivre/se termine à la vitesse la plus lente. Voir le chapitre 11 pour en savoir plus sur les diagrammes d'états et les définitions des transactions divisées.

Une transaction divisée à haute vitesse se compose de deux parties: une transaction divisée de début et une transaction divisée de fin. Les transactions divisées sont uniquement définies pour être utilisées entre le contrôleur d'hôte et un hub. Aucun autre appareil à haute vitesse ou à vitesse intégrale/basse vitesse n'utilise jamais les transactions divisées.

La Figure 8-6 montre les paquets composant une transaction divisée de début générique. La phase de jeton comprend deux paquets: le jeton spécial SPLIT et un jeton vitesse intégrale/basse vitesse. Selon le sens du transfert de données et si un établissement de liaison est défini pour le type de transaction, la phase de jeton est facultativement suivie par un paquet de données et un paquet d'établissement de liaison. Les transactions divisées de début peuvent être constituées de 2, 3 ou 4 paquets comme déterminé par le type de transfert spécifique et le sens des données.



Légende

Anglais	Français
SSPLIT Token	Jeton SSPLIT
FS/LS Token	Jeton FS/LS
Handshake	Etablissement de liaison
Token phase	Phase de jeton

Figure 8-6. Paquets dans une transaction divisée de début

La Figure 8-7 montre les paquets composant une transaction divisée de fin générique. La phase de jeton comprend deux paquets: le jeton spécial SPLIT et un jeton vitesse intégrale/basse vitesse. Un paquet de données ou d'établissement de liaison suit les paquets de phase de jeton dans la fin de transaction divisée selon le sens de transfert des données et le type de transaction spécifique. Les transactions divisées de fin peuvent être constituées de 2 ou 3 paquets comme déterminé par le type de transfert spécifique et le sens des données.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

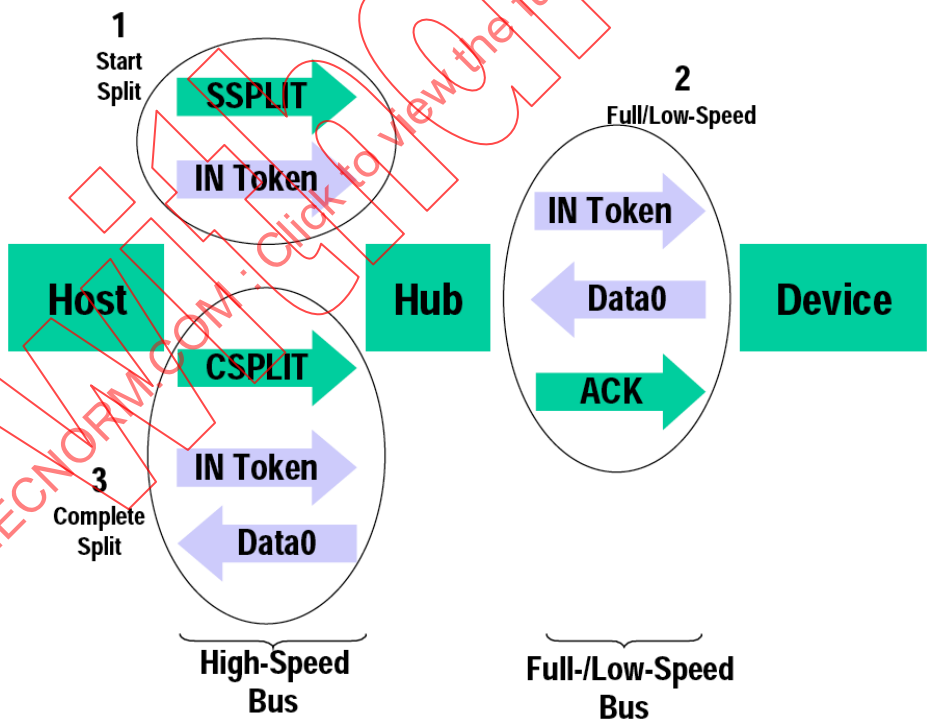


Légende

Anglais	Français
CSPLIT Token	Jeton CSPLIT
FS/LS Token	Jeton FS/LS
Handshake	Etablissement de liaison
Token Phase	Phase de jeton

Figure 8-7. Paquets dans une transaction divisée de fin

Les résultats d'une transaction divisée sont retournés par une transaction divisée de fin. La Figure 8-8 montre cette "conversion" conceptuelle pour un exemple de type de transfert IN d'interruption. L'hôte émet un début divisé (indiqué par 1) au hub puis il peut procéder à d'autres transactions à haute vitesse. Suite au début divisé, le hub émet un jeton IN vitesse intégrale/basse vitesse un peu plus tard (indiqué par 2). L'appareil répond au jeton IN (dans cet exemple) avec un paquet de données et le hub répond à l'appareil avec un établissement de liaison. Et enfin, l'hôte émet quelque temps plus tard une fin de transaction divisée (indiqué par 3) pour récupérer les données fournies par l'appareil. A noter que dans l'exemple, le hub a fourni l'établissement de liaison vitesse intégrale/basse vitesse (ACK dans cet exemple) au point d'extrémité de l'appareil avant la fin de transaction divisée, et la fin de transaction divisée n'a pas fourni d'établissement de liaison haute vitesse au hub.



Légende

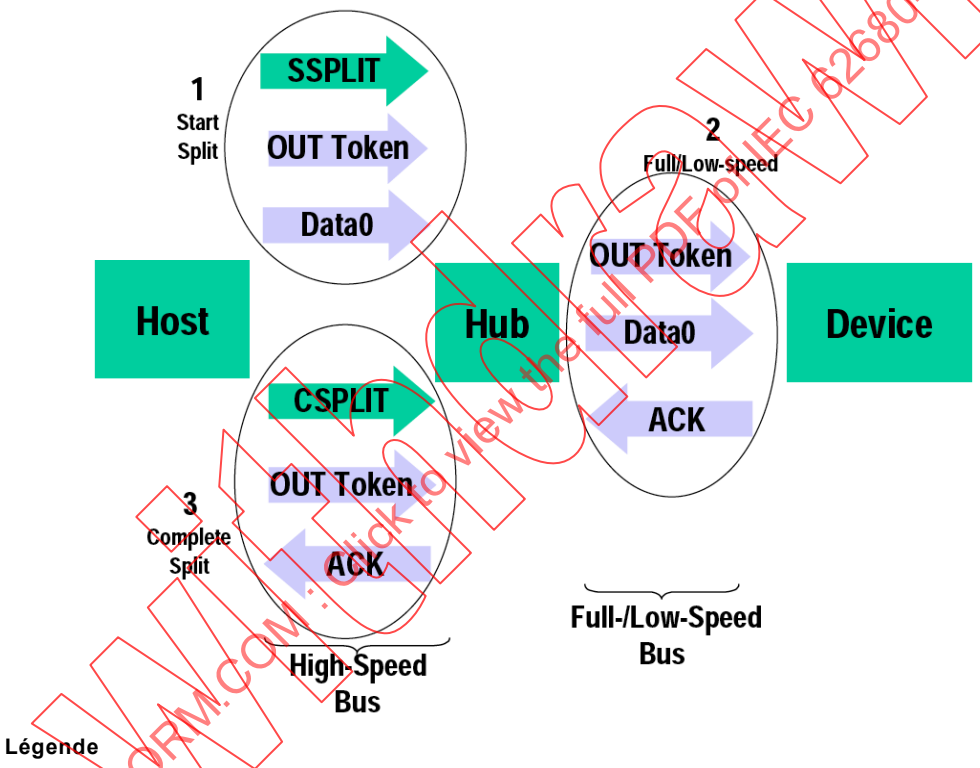
Anglais	Français
Start Split	Division de début
IN Token	Jeton IN
Full/Low-Speed	Vitesse intégrale/basse vitesse
Host	Hôte
Hub	Hub
IN Token	Jeton IN

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Anglais	Français
Device	Appareil
Complete Split	Division complète
High-Speed Bus	Bus à grande vitesse
Full/Low-Speed Bus	Bus à vitesse intégrale/basse vitesse

Figure 8-8. Relation entre la transaction IN d'interruption et la transaction divisée à haute vitesse

Une transaction OUT normale en vitesse intégrale/basse vitesse est, de la même façon, "convertie" conceptuellement en des transactions divisées de début et de fin. La Figure 8-9 montre cette "conversion" pour un exemple de type de transfert OUT d'interruption. L'hôte émet une transaction divisée de début qui est constituée d'un jeton spécial SSPLIT, un jeton OUT et d'un paquet DATA. Le hub émet parfois ensuite le jeton OUT et le paquet DATA sur le bis à vitesse intégrale/basse vitesse. L'appareil répond avec un établissement de liaison. Parfois plus tard, l'hôte émet la transaction divisée de fin et le hub répond avec les résultats (soit les données à vitesse intégrale/basse vitesse soit l'établissement de liaison) fournis par l'appareil.



Légende

Anglais	Français
Start Split	Division de début
OUT Token	Jeton IN
Full/Low-Speed	Vitesse intégrale/basse vitesse
Host	Hôte
Hub	Hub
OUT Token	Jeton OUT
Device	Appareil
Complete Split	Division complète
High-Speed Bus	Bus à grande vitesse
Full/Low-Speed Bus	Bus à vitesse intégrale/basse vitesse

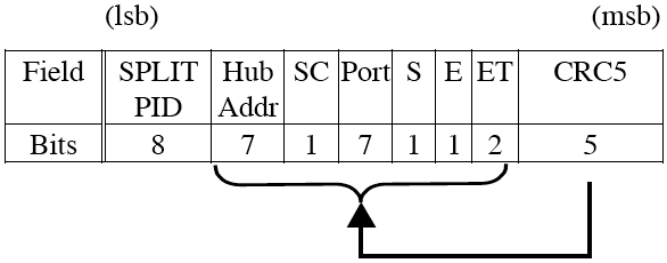
Figure 8-9. Relation entre la transaction OUT d'interruption et la transaction OUT divisée à haute vitesse

Les deux sections suivantes décrivent les champs qui composent les paquets de jetons détaillés divisés de début et de fin. Les Figures 8-10 et 8-12 indiquent les champs dans le paquet de jetons de transaction divisée. Le jeton spécial SPLIT suit le format de jeton général et commence avec un champ PID (après un SYNC) et se termine avec un champ CRC5 (et

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

EOP). Les paquets de jetons divisés de début et de fin font 4 octets de long. Les transactions SPLIT doivent provenir uniquement de l'hôte. Le jeton divisé de début est défini dans la Section 8.4.2.2 et le jeton divisé de fin dans la Section 8.4.2.3.

8.4.2.2 Jeton de transaction divisée de début



Légende

Anglais	Français
Field	Champ

Figure 8-10. Jeton de transaction divisée de début (SSPLIT)

Le champ d'adresse du hub contient l'adresse de l'appareil USB du hub prenant en charge l'appareil vitesse intégrale/basse vitesse spécifié pour cette transaction vitesse intégrale/basse vitesse. Ce champ a la même définition que la définition du champ ADDR dans la Section 8.3.2.1.

Un paquet de jetons spécial SPLIT avec le champ SC (début/fin) réglé à zéro indique qu'il s'agit d'une transaction divisée de début (SSPLIT).

Le champ Port contient le numéro du port du hub cible auquel la transaction vitesse intégrale/basse vitesse est destinée. Comme indiqué sur la Figure 8-11, 128 ports au total sont spécifiés en tant que PORT<6:0>. L'hôte doit régler correctement le champ de port pour la mise en application de hubs TT simples et multiples. Une mise en application de hub TT simple peut ignorer le champ de port.

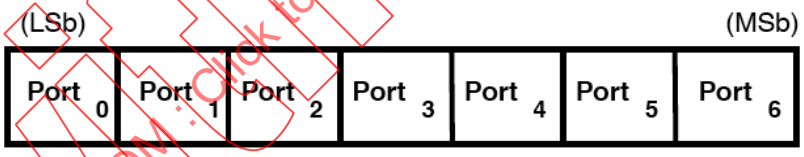


Figure 8-11. Champ de port

Le champ S (vitesse) spécifie la vitesse pour cette transaction d'interruption ou de contrôle comme suit:

- 0: vitesse intégrale
- 1: basse vitesse

Pour les débuts divisés IN/OUT en masse et IN isochrones, le champ S doit être réglé sur zéro. Pour les débuts divisés IN/OUT en masse/de contrôle, IN/OUT d'interruption et IN isochrones, le champ E doit être réglé sur zéro.

Pour les débuts divisés OUT isochrones vitesse intégrale, le champ S[1] (début) et E (fin) spécifie comment les données utiles haute-vitesse correspondent aux données pour un paquet de données vitesse intégrale comme indiqué dans le Tableau 8-2.

[1] Le bit S peut être réutilisé pour ces encodages car les transactions isochrones ne doivent pas être à basse vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 8-2. Encodage de poursuite de données utiles OUT isochrones

S	E	Relation entre les données haute vitesse et vitesse intégrale
0	0	Les données haute vitesse sont au milieu des données utiles vitesse intégrale.
0	1	Les données haute vitesse sont à la fin des données utiles vitesse intégrale.
1	0	Les données haute vitesse sont au début des données utiles vitesse intégrale.
1	1	Les données haute vitesse sont toutes les données utiles vitesse intégrale.

Les transactions divisées de début OUT isochrones utilisent ces encodages pour permettre au hub de détecter divers cas d'erreur due à l'absence de réception de transactions divisées de début pour un point d'extrémité avec des données utiles exigeant plusieurs débuts divisés. Par exemple, de grandes données utiles vitesse intégrale peuvent exiger trois transactions divisées de début: un début divisé/début, un début divisé/intermédiaire et un début divisé/fin. Si l'une de ces transactions n'est pas reçue par le hub, soit il ignore la transaction vitesse intégrale (si le début divisé/début n'est pas reçu), soit il force une erreur pour la transaction vitesse intégrale correspondante (si l'une des deux transactions n'est pas reçue). D'autres conditions d'erreur peuvent être détectées en ne recevant pas de début divisé pendant une microtrame.

Le champ ET (type de point d'extrémité) spécifie le type de point d'extrémité de la transaction vitesse intégrale/basse vitesse comme indiqué dans le Tableau 8-3.

Tableau 8-3. Valeurs de types de points d'extrémité dans le jeton spécial divisé

Valeur ET (msb:lsb)	Type de point d'extrémité
00	Contrôle
01	Isochrone
10	Volume
11	Interruption

Ce champ dit au hub quel diagramme d'états de transaction divisée il doit utiliser pour cette transaction vitesse intégrale/basse vitesse.

Les informations sur l'adresse de l'appareil vitesse intégrale/basse vitesse et le nombre de points d'extrémité sont contenues dans le paquet de jetons normal qui suit le paquet de jetons spécial SPLIT.

8.4.2.3 Jeton de transaction divisée de fin

(lsb) (msb)

Field	SPLIT PID	Hub Addr	SC	Port	S	U	ET	CRC5
Bits	8	7	1	7	1	1	2	5

Légende

Anglais	Français
Field	Champ
Hub Addr	Adresse du hub
Bits	Bits
Port	Port

Figure 8-12. Jeton de transaction divisée de fin (CSPLIT)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

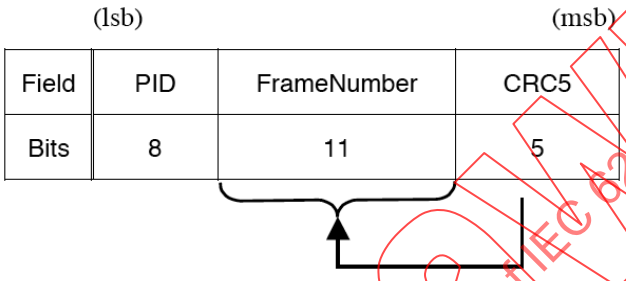
Un paquet de jetons spécial SPLIT avec le champ SC réglé à un indique qu'il s'agit d'une transaction divisée de fin (CSPLIT).

Le bit U est réservé/inutilisé et doit être réinitialisé à zéro (0B).

Les autres champs du paquet de jetons divisé de fin ont les mêmes définitions que ceux du paquet de jeton divisé de début.

8.4.3 Paquets de début de trame

Les paquets de début de trame (SOF) sont émis par l'hôte à une vitesse nominale de un toutes les 1,00 ms ± 0,0005 ms pour un bus vitesse intégrale et 125 µs ±0,0625 µs pour un bus haute vitesse. Les paquets SOF sont constitués d'un PID indiquant le type de paquet suivi d'un champ de nombre de trames à 11 bits comme représenté à la Figure 8-13.



Légende

Anglais	Français
Field	Champ
Bits	Bits

Figure 8-13. Paquet de SOF

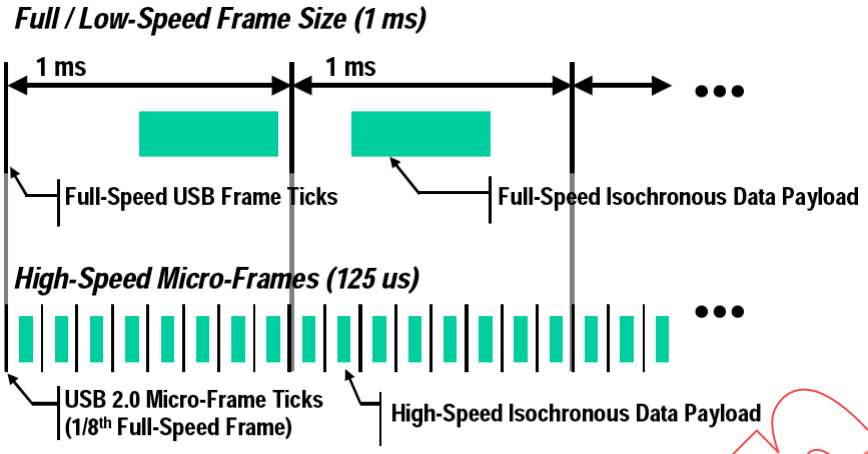
Le jeton de SOF comprend la transaction uniquement à jeton qui distribue un marqueur de SOF et accompagnant le nombre de trames précisément aux intervalles réguliers correspondant au début de chaque trame. Toutes les fonctions haute vitesse et vitesse intégrale, y compris les hubs, reçoivent le paquet de SOF. Le jeton de SOF ne fait générer de paquet de retour par aucune fonction réceptrice; on ne peut donc pas garantir qu'un SOF est transmis à une fonction donnée. Le paquet de SOF fournit deux informations sur la synchronisation. Une fonction est informée qu'un SOF a eu lieu quand elle détecte le PID du SOF. Les fonctions sensibles à la synchronisation des trames, qui ne nécessitent pas de suivre le nombre de trames (par ex.: un hub fonctionnant à vitesse intégrale), nécessitent uniquement de décoder le PID du SOF, elles peuvent ignorer le nombre de trames et son CRC. Si une fonction nécessite de suivre le nombre de trames, elle doit comprendre le PID ainsi que l'horodatage. Les appareils à vitesse intégrale qui n'ont pas de besoin particulier pour les informations de synchronisation de bus peuvent ignorer le paquet de SOF.

8.4.3.1 Trames et microtrames USB

USB définit une durée de trame de 1 ms à vitesse intégrale indiquée par un paquet de début de trame (SOF) à chaque période de 1 ms avec des tolérances à la gigue définies. USB définit également une microtrame à haute vitesse avec une durée de trame de 125 µs avec des tolérances à la gigue liées (voir le chapitre 7). Les paquets de SOF sont générés (par le contrôleur d'hôte ou le compilateur de transaction de hub) toutes les 1 ms pour les liaisons à vitesse intégrale. Les paquets de SOF sont également générés après les sept périodes suivantes de 125 µs pour les liaisons à haute vitesse.

La Figure 8-14 montre la relation entre les microtrames et les trames.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

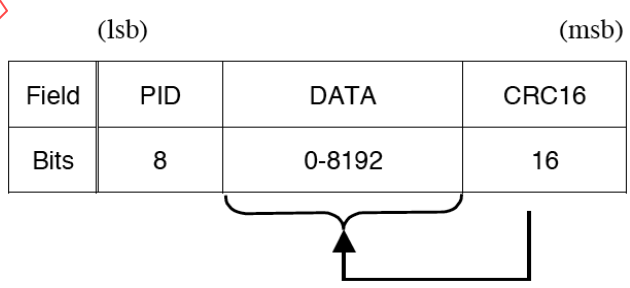
Anglais	Français
Full/Low-Speed Frame Size (1 ms)	Dimension de trame à vitesse intégrale/basse vitesse
Full-Speed USB Frame Ticks	Coches de trame USB à vitesse intégrale
Full-Speed Isochronous Data Payload	Données utiles isochrones à vitesse intégrale
High-Speed Micro-Frames (125 µs)	Microtrames à grande vitesse (125 µs)
USB 2.0 Micro-Frame Ticks (1/8th Full-Speed Frame)	Coches de microtrame USB 2.0 (1/8ème de trame à vitesse intégrale)
High-Speed Isochronous Data Payload	Données utiles isochrones à grande vitesse

Figure 8-14. Relation entre les trames et les microtrames

Les appareils à haute vitesse voient un paquet de SOF avec le même nombre de trames huit fois (toutes les 125 µs) pendant chaque période de 1 ms. Si nécessaire, un appareil à haute vitesse peut déterminer localement un "nombre" de microtrames particulier en détectant le SOF qui avait un nombre de trames différent que le SOF précédent et en le traitant comme une microtrame ayant le numéro zéro. Les sept SOF suivants avec le même nombre de trames peuvent être traités comme les microtrames 1 à 7.

8.4.4 Paquets de données

Un paquet de données est constitué d'un PID, un champ de données contenant zéro ou davantage d'octets de données, et un CRC comme représenté à la Figure 8-15. Il existe quatre types de paquets de données, identifiés par différents PID: DATA0, DATA1, DATA2 et MDATA. Deux PID du paquet de données (DATA0 et DATA1) sont définis pour prendre en charge la synchronisation de basculement des données (voir la Section 8.6). Les quatre PID de données sont tous utilisés dans le séquencement du PID de données pour les points d'extrémité isochrones haute vitesse de bande passante supérieure (voir la Section 5.9). Ces PID de données (MDATA, DATA0 et DATA1) sont utilisés dans les transactions divisées (voir les Sections 11.17 à 11.21).



Légende

Anglais	Français
Field	Champ
Bits	Bits

Figure 8-15. Format du paquet de données

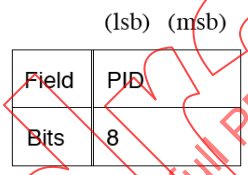
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Les données doivent toujours être envoyées en nombres entiers d'octets. Le CRC des données est calculé uniquement sur le champ de données dans le paquet et n'inclut pas le PID, qui a son propre champ de vérification.

La taille de données utiles maximum permise pour les appareils basse vitesse est de 8 octets. La taille de données utiles maximale permise pour les appareils vitesse intégrale est de 1 023 octets. La taille de données utiles maximale permise pour les appareils haute vitesse est de 1 024 octets.

8.4.5 Paquets d'établissement de liaison

Les paquets d'établissement de liaison, comme indiqué sur la Figure 8-16, sont constitués uniquement d'un PID. Les paquets d'établissement de liaison servent à signaler le statut d'une transaction de données et peut renvoyer des valeurs indiquant une réception réussie des données, une acceptation ou un rejet de commande, un contrôle du débit et des conditions d'arrêt. Seuls les types de transaction prenant en charge le contrôle de débit et peuvent renvoyer les établissements de liaison. Les établissements de liaison sont toujours retournés dans la phase d'établissement de liaison d'une transaction et peuvent être retournés, à la place des données dans la phase de données. Les paquets d'établissement de liaison sont délimités par un EOP après un octet de champ de paquet. Si un paquet se décode comme un établissement de liaison autrement valide, mais ne se termine pas par un EOP après un octet, il doit être considéré comme invalide et ignoré par le récepteur.



Légende

Anglais	Français
Field	Champ
Bits	Bits

Figure 8-16. Paquet poignée de mains

Il existe quatre types de paquets d'établissement de liaison et un paquet d'établissement de liaison spécial:

- **ACK** indique que le paquet de données a été reçu sans insertion de bit ni erreur de CRC sur le champ de données et que le PID de données a été reçu correctement. ACK peut être émis soit quand les bits de séquence correspondent et le récepteur peut accepter des données soit quand les bits de séquence ne correspondent pas et l'envoyeur et le récepteur doivent se resynchroniser ensemble (voir la Section 8.6 pour les détails). Un établissement de liaison ACK est applicable uniquement aux transactions dans lesquelles les données ont été transmises et où un établissement de liaison est escompté. ACK peut être retourné par l'hôte pour les transactions IN et par une fonction pour les transactions OUT, SETUP ou PING.
- **NAK** indique qu'une fonction n'a pas réussi à accepter des données provenant de l'hôte (OUT) ou qu'une fonction n'a pas de données à transmettre à l'hôte (IN). NAK peut être retourné uniquement par les fonctions dans la phase de données des transactions IN ou la phase d'établissement de liaison des transactions OUT ou PING. L'hôte ne peut jamais émettre de NAK. NAK sert à des fins de contrôle de débit à indiquer qu'une fonction ne peut provisoirement pas émettre ou recevoir des données, mais peut au final le faire sans nécessiter l'intervention de l'hôte.
- **STALL** est retourné par une fonction en réponse à un jeton IN ou après la phase de données d'un OUT ou en réponse à une transaction PING (voir les Figures 8-30 et 8-38). STALL indique qu'une fonction ne parvient pas à émettre ou à recevoir des données, ou qu'une demande de canalisation de contrôle n'est pas prise en charge. Le statut d'une fonction après avoir renvoyé un STALL (pour tout point d'extrémité sans le point

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

d'extrémité par défaut) n'est pas défini. Il n'est en aucun cas admis que l'hôte renvoie un STALL.

L'établissement de liaison STALL est utilisé par un appareil à une ou deux occasions distinctes. Dans le premier cas, intitulé "calage fonctionnel", la caractéristique *Halt* associée au point d'extrémité est réglée. (La caractéristique *Halt* est spécifiée dans le chapitre 9 du présent document.) Un cas particulier de calage fonctionnel est le "calage commandé". Le calage commandé a lieu quand l'hôte règle explicitement la caractéristique *Halt* du point d'extrémité, comme détaillé dans le chapitre 9. Une fois qu'un point d'extrémité de fonction est arrêté, la fonction doit continuer à renvoyer STALL jusqu'à ce que la condition causant l'arrêt ait été acquittée par l'intervention de l'hôte.

Le deuxième cas, intitulé "calage de protocole", est détaillé dans la Section 8.5.3. Le calage de protocole est propre aux canalisations de contrôle. Le calage de protocole diffère du calage fonctionnel dans la signification et la durée. Un STALL de protocole est renvoyé pendant le stade de données ou de statut d'un transfert de contrôle, et la condition de STALL se termine au début du transfert de contrôle suivant (Setup). La suite de cette section se rapporte au cas général d'un calage fonctionnel.

- **NYET** est un établissement de liaison uniquement à haute vitesse qui est renvoyé dans deux circonstances. Il est renvoyé par un point d'extrémité haute vitesse dans le cadre du protocole PING décrit plus loin dans ce chapitre. NYET peut également être renvoyé par un hub en réponse à une transaction divisée quand la transaction vitesse intégrale/basse vitesse n'est pas encore terminée ou le hub n'a autrement pas réussi à prendre en charge la transaction divisée. Voir le chapitre 11 pour de plus amples détails.
- **ERR** est un établissement de liaison uniquement à haute vitesse qui est renvoyé pour permettre à un hub haute vitesse de signaler une erreur sur un bus vitesse intégrale/basse vitesse. Il est renvoyé uniquement par un hub haute vitesse dans le cadre du protocole de transaction divisée. Voir le chapitre 11 pour de plus amples détails.

8.4.6 Réponses d'établissement de liaison

Les fonctions d'émission et de réception doivent renvoyer les établissements de liaison selon un ordre de priorité détaillé dans les Tableaux 8-4 à 8-6. Tous les établissements de liaison ne sont pas permis, en fonction du type de transaction et de l'établissement de liaison s'il est émis par une fonction ou l'hôte. A noter que si une erreur se produit pendant l'émission du jeton à la fonction, la fonction ne répond pas avec aucun paquet tant que le jeton suivant n'est pas reçu et décodé avec succès.

8.4.6.1 Réponse d'une fonction à des transactions IN

Le Tableau 8-4 indique les réponses possibles qu'une fonction peut donner en réponse à un jeton IN. Si la fonction ne parvient pas à envoyer des données en raison d'un arrêt ou d'une condition de contrôle de débit, elle émet respectivement un établissement de liaison STALL ou NAK. Si la fonction peut envoyer les données, elle le fait. Si le jeton reçu est corrompu, la fonction ne renvoie aucune réponse.

Tableau 8-4. Réponse d'une fonction à des transactions IN

Jeton reçu corrompu	Caractéristique d'arrêt de point d'extrémité de la fonction Tx	La fonction peut émettre des données	Mesure prise
Oui	Inutile	Inutile	Ne renvoyer aucune réponse
Non	Réglée	Inutile	Emettre un établissement de liaison STALL
Non	Non réglée	Non	Emettre un établissement de liaison NAK
Non	Non réglée	Oui	Emettre un paquet de données

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

8.4.6.2 Réponse d'une fonction à des transactions IN

Le Tableau 8-5 indique la réponse d'un hôte à une transaction IN. L'hôte peut renvoyer uniquement un type d'établissement de liaison: ACK. Si l'hôte reçoit un paquet de données corrompues, il élimine les données et n'émet aucune réponse. Si l'hôte ne peut pas accepter des données provenant d'une fonction (à cause de problèmes tels qu'un engorgement de la mémoire tampon interne), cette condition est considérée comme une erreur et l'hôte ne renvoie aucune réponse. Si l'hôte peut accepter des données et si le paquet de données est reçu sans erreur, l'hôte accepte les données et émet un établissement de liaison ACK.

Tableau 8-5. Réponse de l'hôte à des transactions IN

Paquet de données corrompu	L'hôte peut accepter des données	Etablissement de liaison renvoyé par l'hôte
Oui	N/A	Eliminer les données, ne pas renvoyer de réponse
Non	Non	Eliminer les données, ne pas renvoyer de réponse
Non	Oui	Accepter les données, émettre un ACK

8.4.6.3 Réponse d'une fonction à des transactions OUT

Les réponses d'établissement de liaison pour une transaction OUT sont données dans le Tableau 8-6. En supposant que le jeton se décode avec succès, une fonction, sur réception d'un paquet de données, peut renvoyer n'importe lequel des trois types d'établissement de liaison. Si le paquet de données était corrompu, la fonction ne renvoie aucun établissement de liaison. Si le paquet de données a été reçu sans erreur et si le point d'extrémité de la fonction est arrêté, la fonction renvoie STALL. Si la transaction maintient la synchronisation des bits de séquence et si une incohérence est détectée (voir la Section 8.6 pour les détails), alors la fonction renvoie ACK et élimine les données. Si la fonction peut accepter les données et a reçu les données sans erreur, elle renvoie ACK. Si la fonction ne peut pas accepter le paquet de données pour des raisons de contrôle de débit, elle renvoie NAK.

Tableau 8-6. Réponses d'une fonction à des transactions OUT par ordre de priorité

Paquet de données corrompu	Caractéristique d'arrêt du récepteur	Les bits de séquence correspondent	La fonction peut accepter des données	Etablissement de liaison renvoyé par la fonction
Oui	N/A	N/A	N/A	Néant
Non	Réglée	N/A	N/A	STALL
Non	Non réglée	Non	N/A	ACK
Non	Non réglée	Oui	Oui	ACK
Non	Non réglée	Oui	Non	NAK

8.4.6.4 Réponse d'une fonction à une transaction SETUP

SETUP définit un type particulier de transaction de données de l'hôte vers la fonction qui permet à l'hôte d'initialiser les bits de synchronisation de point d'extrémité avec ceux de l'hôte. Sur réception d'un jeton SETUP, une fonction doit accepter les données. Une fonction peut ne pas répondre à un jeton SETUP avec STALL ou NAK, et la fonction réceptrice doit accepter le paquet de données qui suit le jeton SETUP. Si un point d'extrémité non-contrôle reçoit un jeton SETUP, il doit ignorer la transaction et ne renvoyer aucune réponse.

8.5 Séquences des paquets de transactions

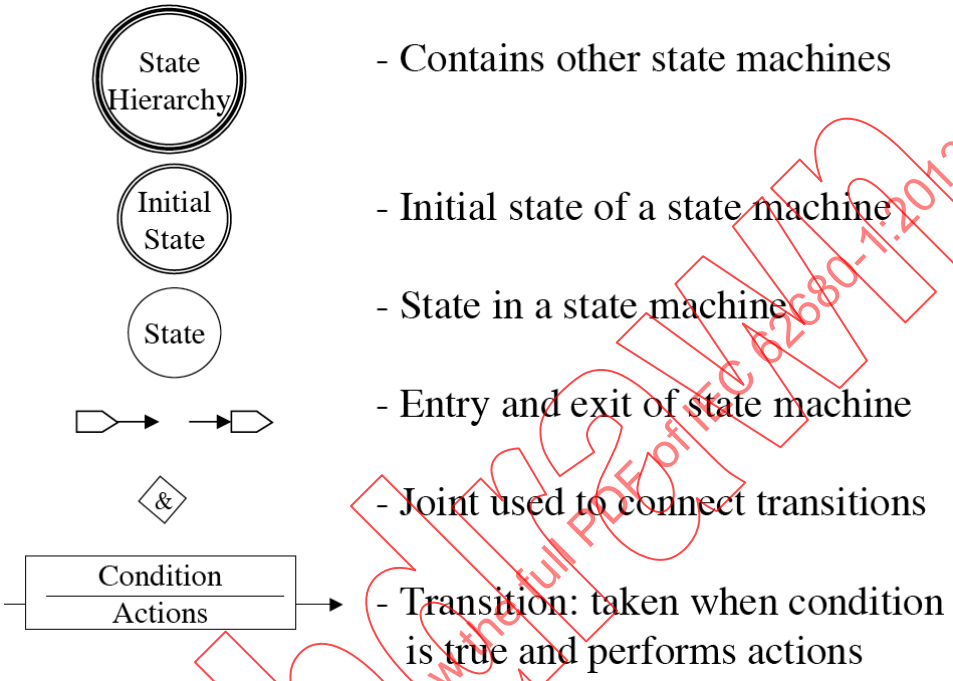
Les paquets qui comprennent une transaction varient selon le type de point d'extrémité. Il existe quatre types de point d'extrémité: volume, contrôle, interruption et isochrone.

Un contrôleur d'hôte et un appareil nécessitent chacun différents diagrammes d'états pour séquencer correctement chaque type de transaction. Les figures dans les sections suivantes

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

montrent des diagrammes d'états qui définissent le séquencement correct des paquets au sein d'une transaction de chaque type. Il convient de ne pas prendre les schémas comme une mise en œuvre exigée, mais pour spécifier le comportement requis.

La Figure 8-17 montre la légende des diagrammes d'états. Un cercle avec une bordure de trois lignes indique une référence à un autre diagramme d'états (hiérarchique). Un cercle avec une bordure de deux lignes indique un état initial. Un cercle avec une bordure d'une ligne représente un état simple.



Légende

Anglais	Français
State Hierarchy	Hiérarchie d'état
Contains other state machines	Contient les autres diagrammes d'états
Initial State	Etat initial
Initial state of state machine	Etat initial du diagramme d'états
State	Etat
State in a state machine	Etat dans un diagramme d'états
Entry and exit of state machine	Entrée et sortie du diagramme d'états
Joint used to connect transitions	Joint utilisé pour connecter les transitions
Conditions/Actions	Conditions/Actions
Transition: taken when condition is true and performs actions	Transition: prise lorsque la condition est vraie et effectue les actions

Figure 8-17. Légende des diagrammes d'états

Les formes de "languette" avec des flèches sont (respectivement dans la légende) l'entrée ou la sortie vers/venant du diagramme d'états. L'entrée/la sortie se rapportent à un autre état dans un diagramme d'états à un niveau supérieur dans la hiérarchie des diagrammes d'états.

Un diamant (joint) sert à rassembler plusieurs transitions en un point commun. Un joint permet une transition à une seule entrée avec des transitions à plusieurs sorties ou à plusieurs entrées et une transition à une seule sortie. Toutes les conditions sur les transitions d'une voie impliquant un joint doivent être vraies pour que la voie soit prise. Une voie est simplement une séquence de transitions impliquant un ou plusieurs joints.

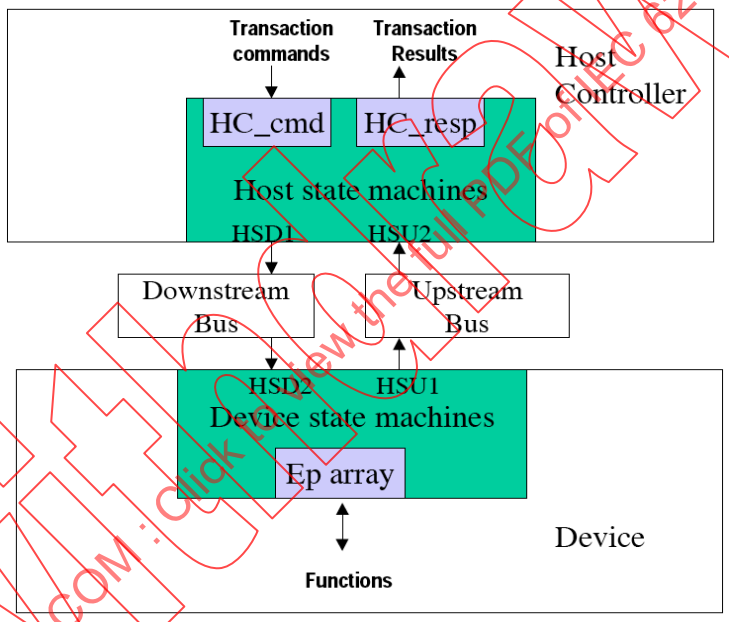
Une transition est marquée avec un bloc avec une ligne au milieu séparant la condition (supérieure) et les actions (inférieures). Il est nécessaire que la condition soit vraie pour prendre la transition. La syntaxe pour les actions et conditions est VHDL. Les actions sont effectuées si la transition est prise. Un cercle inclut un nom en gras et facultativement une ou plusieurs actions qui sont effectuées à l'entrée dans l'état.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Le contrôleur d'hôte et les diagrammes d'états de l'appareil sont dans un contexte qui est représenté à la Figure 8-18. Le contrôleur d'hôte détermine la prochaine transaction à lancer pour un point d'extrémité et émet une commande (HC_cmd) vers les diagrammes d'états du contrôleur d'hôte. En conséquence, les diagrammes d'états du contrôleur d'hôte émettent un ou plusieurs paquets qui se déplacent sur le bus en aval (HSD1).

L'appareil reçoit ces paquets venant du bus (HSD2), il réagit au paquet reçu et interagit avec sa ou ses fonctions via l'état du point d'extrémité correspondant (dans EP_array). Ensuite l'appareil peut répondre avec un paquet sur les bus en amont (HSU1). Les diagrammes d'états du contrôleur d'hôte peuvent recevoir un paquet provenant du bus (HSU2) et rendre un résultat de la transaction au contrôleur d'hôte (HC_resp). Ce qui détermine quels paquets sont envoyés sur le bus est le type de transfert pour le point d'extrémité et quelle activité de bus les diagrammes d'états observent.

Les diagrammes d'états sont présentés sous forme hiérarchique. La Figure 8-19 montre les diagrammes d'états de niveau supérieur pour le contrôleur d'hôte. Les transactions non divisées sont présentées dans la suite de ce chapitre. Les diagrammes d'états de transactions divisées (HC_Do_start et HC_Do_complete) sont décrits et indiqués dans le chapitre 11.



Légende

Anglais	Français
Transition commands	Commandes de transition
Transaction Results	Résultats de transaction
Host Controller	Contrôleur hôte
Host state machines	Diagrammes d'états de l'hôte
Downstream Bus	Bus en aval
Upstream Bus	Bus en amont
Device state machines	Diagrammes d'états d'appareil
Functions	Fonctions
Device	Appareil

Figure 8-18. Vue d'ensemble du contexte d'un diagramme d'états

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

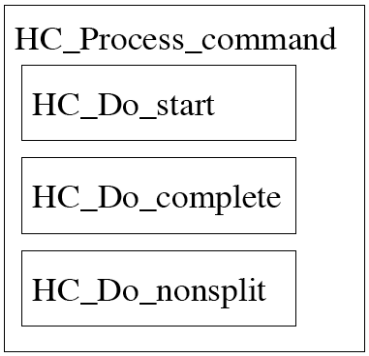


Figure 8-19. Vue d'ensemble de la hiérarchie des diagrammes d'états pour une transaction de niveau supérieur du contrôleur d'hôte

Les diagrammes d'états du contrôleur d'hôte se situent dans le contrôleur d'hôte. Le contrôleur d'hôte fait émettre des paquets en aval (marqués HSD1) et il reçoit des paquets en amont (marqués HSU2).

Les diagrammes d'états de l'appareil sont situés dans l'appareil. L'appareil fait émettre des paquets en amont (marqués HSU1) et il reçoit des paquets en aval (marqués HSD2).

Le contrôleur d'hôte a des commandes qui lui disent quelle transaction émettre après un point d'extrémité. Le contrôleur d'hôte suit les transactions pour plusieurs points d'extrémité. Les diagrammes d'états du contrôleur d'hôte séquencent pour déterminer ce que le contrôleur d'hôte a besoin de faire après le point d'extrémité actuel. L'appareil a un état pour chacun de ses points d'extrémité. Les diagrammes d'états de l'appareil séquencent pour déterminer quelle réaction l'appareil a pour une transaction.

L'annexe inclut des déclarations qui ont servi à construire les diagrammes d'états et peuvent être utiles pour comprendre d'autres détails des diagrammes d'états. Il existe plusieurs procédures et fonctions de pseudocode pour les conditions et actions. L'annexe comprend également leurs descriptions simples.

La Figure 8-20 montre un aperçu de la hiérarchie globale des diagrammes d'états pour le contrôleur d'hôte pour des types de transactions non divisées. La Figure 8-21 montre la hiérarchie des diagrammes d'états de l'appareil. Les diagrammes d'états communs aux types de fins de vie sont présentés d'abord. Le type de point d'extrémité le plus bas spécifique aux diagrammes d'états est présenté dans chaque section de type de point d'extrémité suivante.

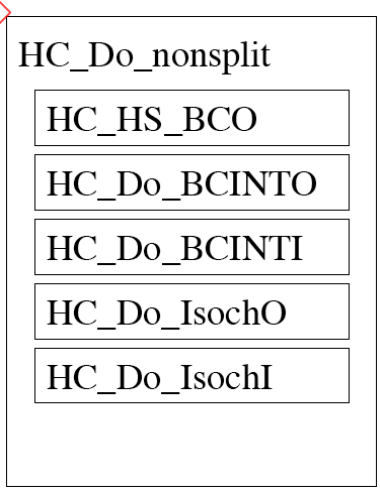


Figure 8-20. Vue d'ensemble de la hiérarchie des diagrammes d'états pour une transaction non divisée du contrôleur d'hôte

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

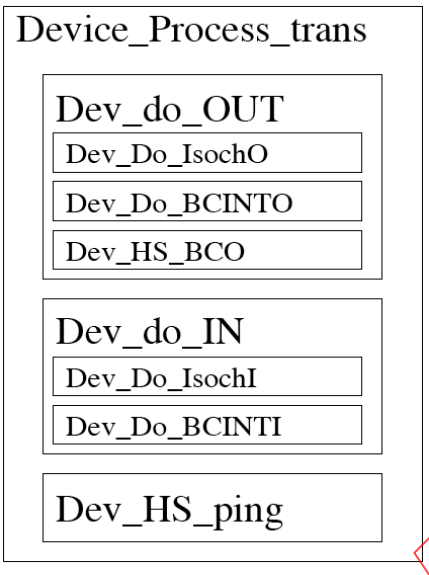
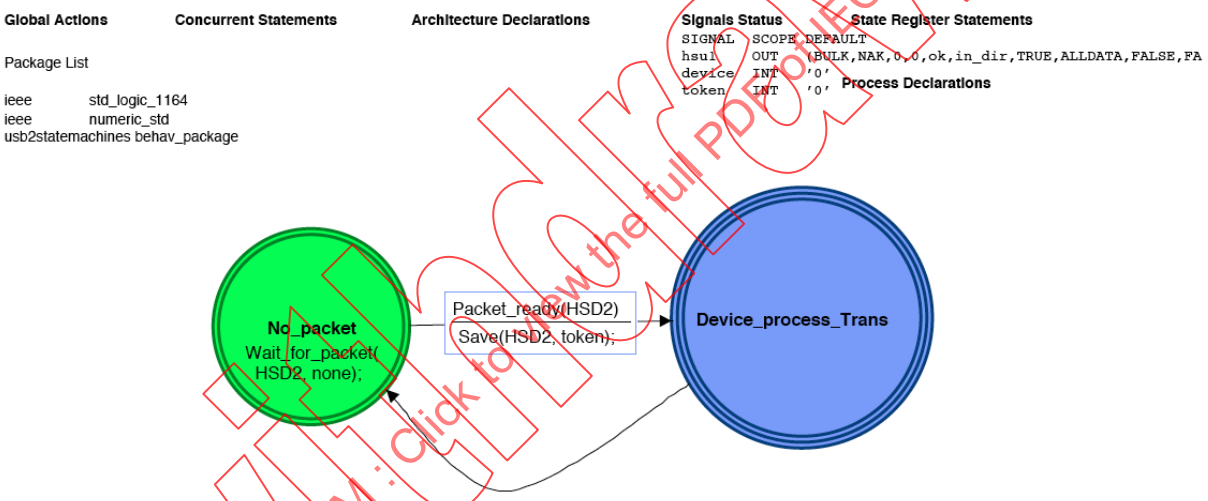


Figure 8-21. Vue d'ensemble de la hiérarchie des diagrammes d'états pour une transaction de l'appareil

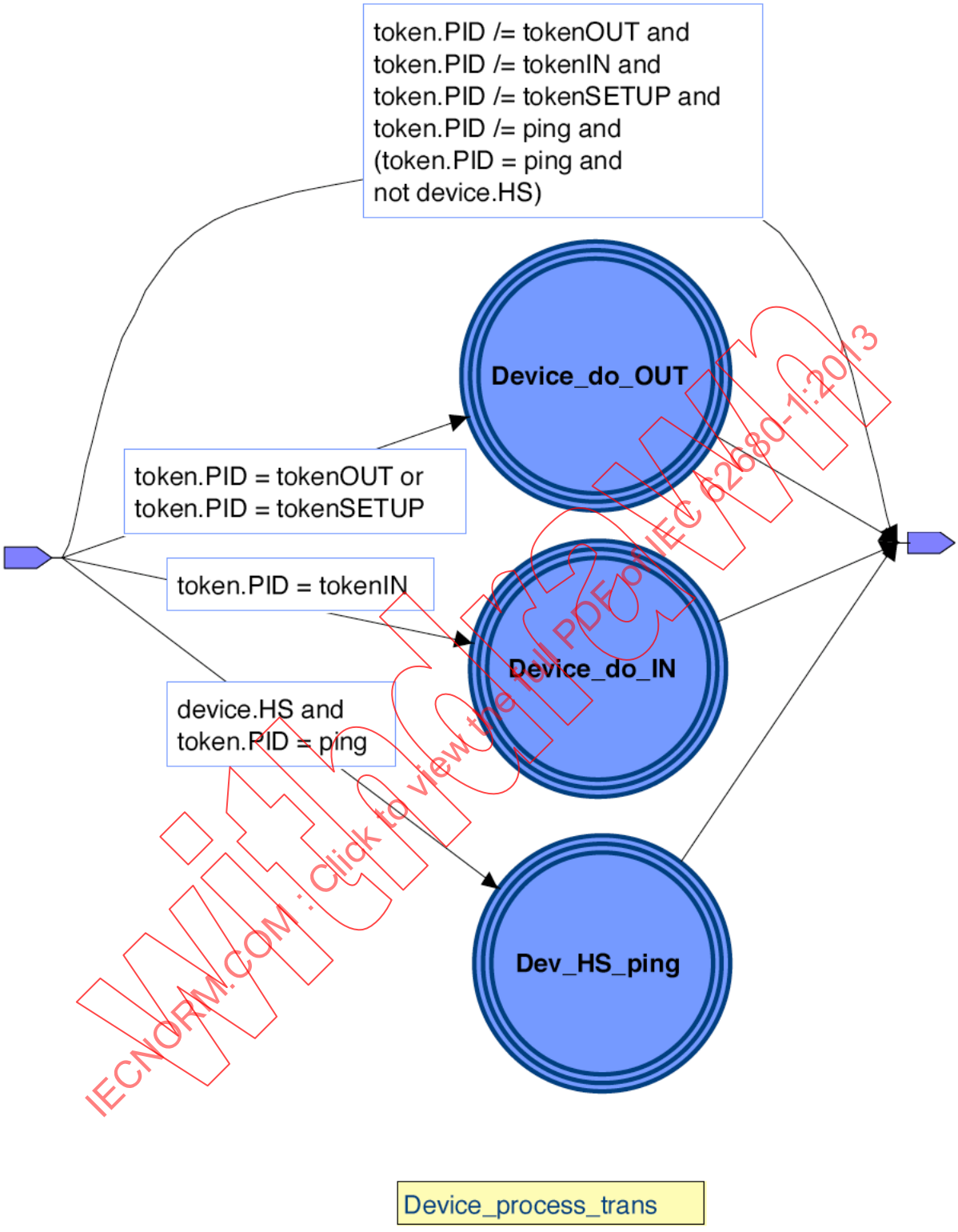


Légende

Anglais	Français
Global Actions	Actions globales
Concurrent Statements	Instructions concurrentes
Architecture Declarations	Déclarations d'architecture
Signals Status	Etat des signaux
State Register Statements	Instructions du registre d'états
Package List	Liste de l'ensemble
Process Declarations	Déclarations de processus

Figure 8-22. Diagramme d'états de niveau supérieur de l'appareil

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Figure 8-23. Diagramme d'états Device_process_Trans

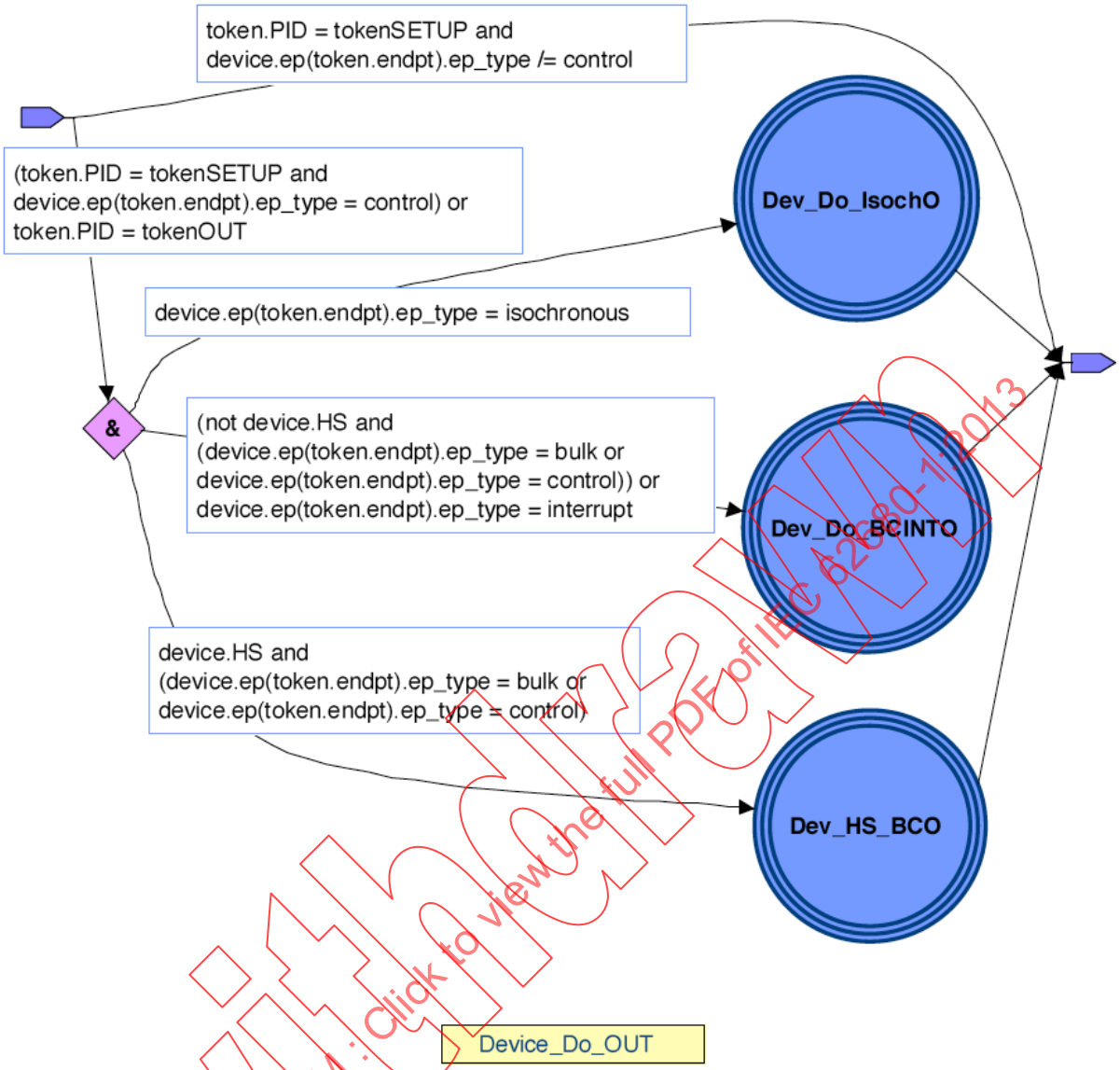
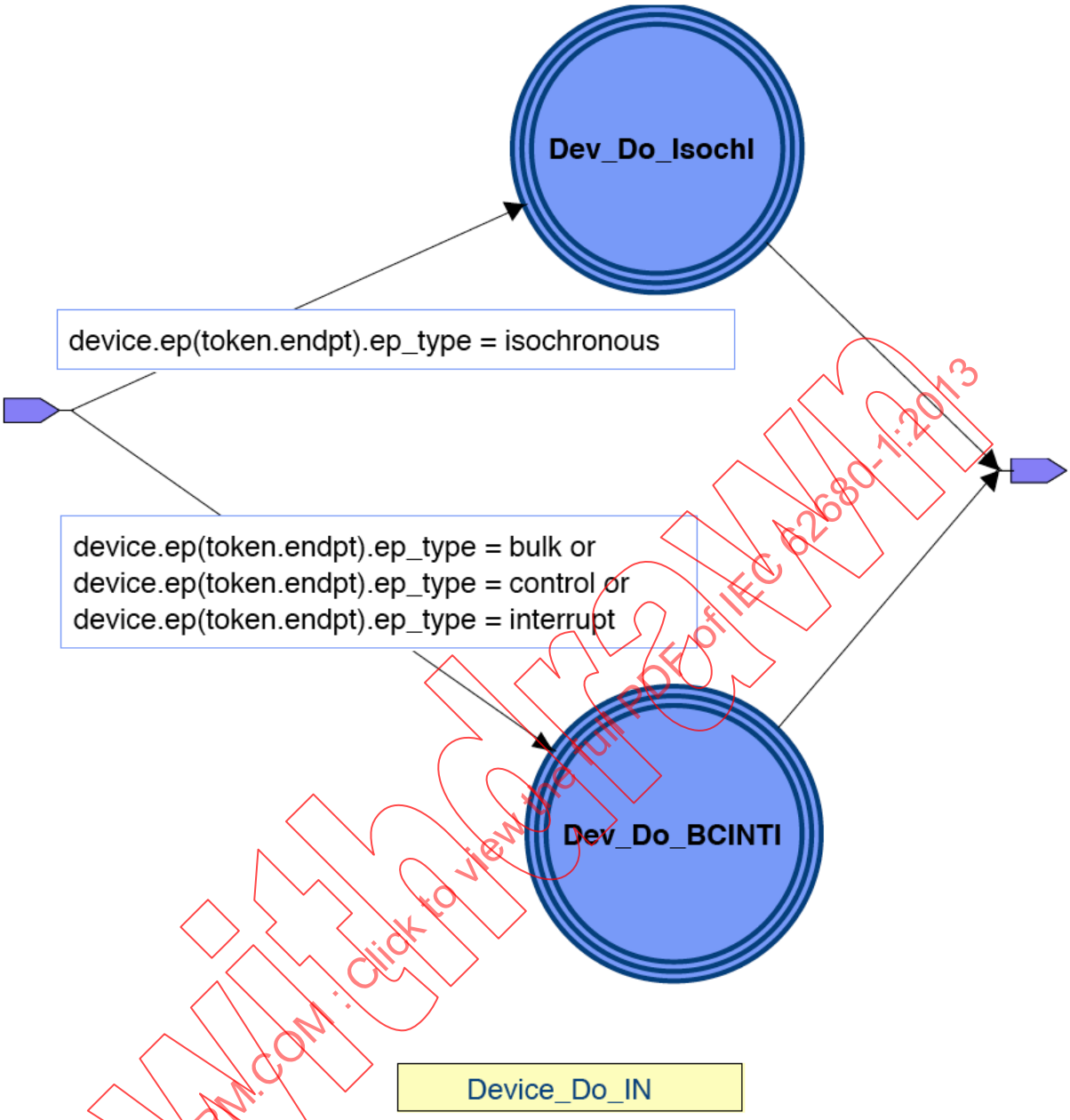


Figure 8-24. Diagramme d'états Dev_do_OUT

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

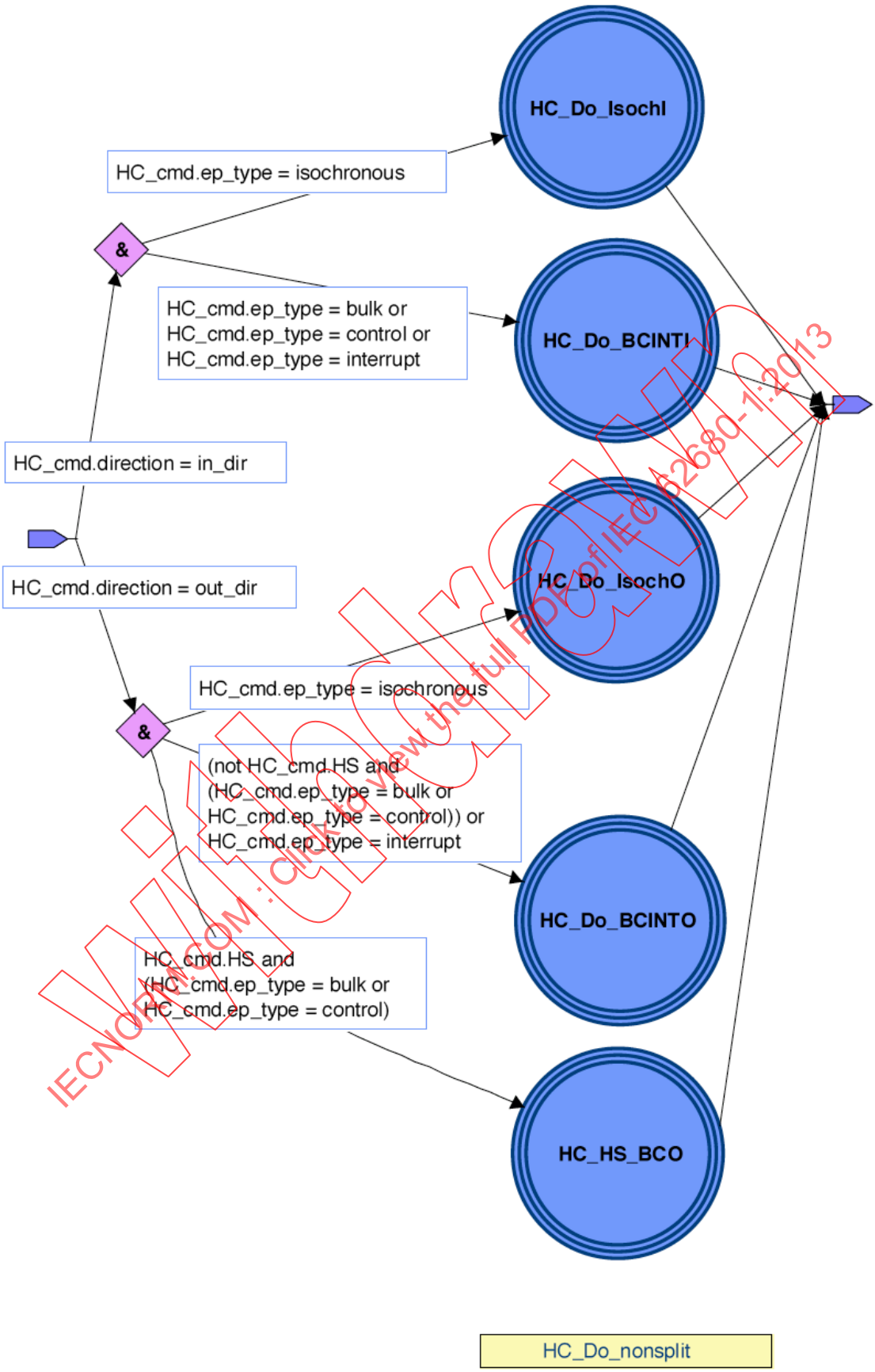


Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Légende

Anglais	Français
device.ep(token.endpt).ep_type /= isochronous	device.ep(token.endpt).ep_type /= isochrone
(device.ep(token.endpt).ep_type = bulk or device.ep(token.endpt).ep_type = control)) or device.ep(token.endpt).ep_type = interrupt	(device.ep(token.endpt).ep_type = en masse ou device.ep(token.endpt).ep_type = contrôle)) ou device.ep(token.endpt).ep_type = interruption

Figure 8-25. Diagramme d'états Dev_do_IN



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
HC_cmd.ep_type = bulk or HC_cmd.ep_type = control or HC_cmd.ep_type = interrupt	HC_cmd.ep_type = en masse ou HC_cmd.ep_type = contrôle ou HC_cmd.ep_type = interruption
(not HC_cmd.HS and (HC_cmd.ep_type = bulk or HC_cmd.ep_type = control)) or HC_cmd.ep_type = interrupt	(not HC_cmd.HS et (HC_cmd.ep_type = en masse ou HC_cmd.ep_type = contrôle)) ou HC_cmd.ep_type = interruption
HC_cmd.HS and (HC_cmd.ep_type = bulk or HC_cmd.ep_type = control)	HC_cmd.HS et (HC_cmd.ep_type = en masse ou HC_cmd.ep_type = contrôle)
HC_cmd.ep_type = isochronous	HC_cmd.ep_type = isochrone

Figure 8-26. Diagramme d'états HC_Do_nonsplit

8.5.1 Limitation de NAK par un contrôle de débit Ping

Les appareils à vitesse intégrale/basse vitesse peuvent avoir des points d'extrémité en masse/de contrôle qui prennent du temps à traiter leurs données, et donc qui répondent à des transactions OUT par un établissement de liaison NAK. Cette réponse d'établissement de liaison indique que le point d'extrémité n'a pas accepté les données, car elle n'avait pas d'espace pour les données. Le contrôleur d'hôte devrait réessayer la transaction par la suite quand le point d'extrémité a de l'espace disponible. Malheureusement, d'ici que le point d'extrémité fasse un NAK, la majorité du temps de bus à vitesse intégrale/basse vitesse pour la transaction a été utilisée. Cela signifie que le bus à vitesse intégrale/basse vitesse a une faible utilisation quand il y a une fréquence élevée de transactions OUT NAK.

Les appareils à haute vitesse doivent prendre en charge un mécanisme NAK amélioré pour OUT de volume ainsi que les points d'extrémité et transactions de contrôle. Les points d'extrémité de contrôle doivent prendre en charge ce protocole pour une transaction OUT dans les stades de données et de statuts. Le stade de mise en œuvre de commande ne doit pas prendre en charge le protocole PING.

Ce mécanisme permet également à l'appareil de dire au contrôleur d'hôte s'il y a ou non suffisamment d'espace de point d'extrémité pour la prochaine transaction OUT. Si le point d'extrémité de l'appareil n'a pas d'espace, le contrôleur d'hôte peut choisir de remettre à plus tard une tentative de transaction pour ce point d'extrémité et plutôt essayer une autre transaction. Cela peut conduire à une meilleure utilisation du bus. Ce mécanisme évite d'utiliser le temps de bus pour envoyer des données jusqu'à ce que le contrôleur d'hôte sache que le point d'extrémité a de l'espace pour les données.

Le contrôleur d'hôte interroge le point d'extrémité de l'appareil à haute vitesse avec un jeton spécial PING. Le paquet de jetons spécial PING est un paquet de jetons normal comme représenté à la Figure 8-5. Le point d'extrémité répond au PING avec un établissement de liaison NAK ou ACK.

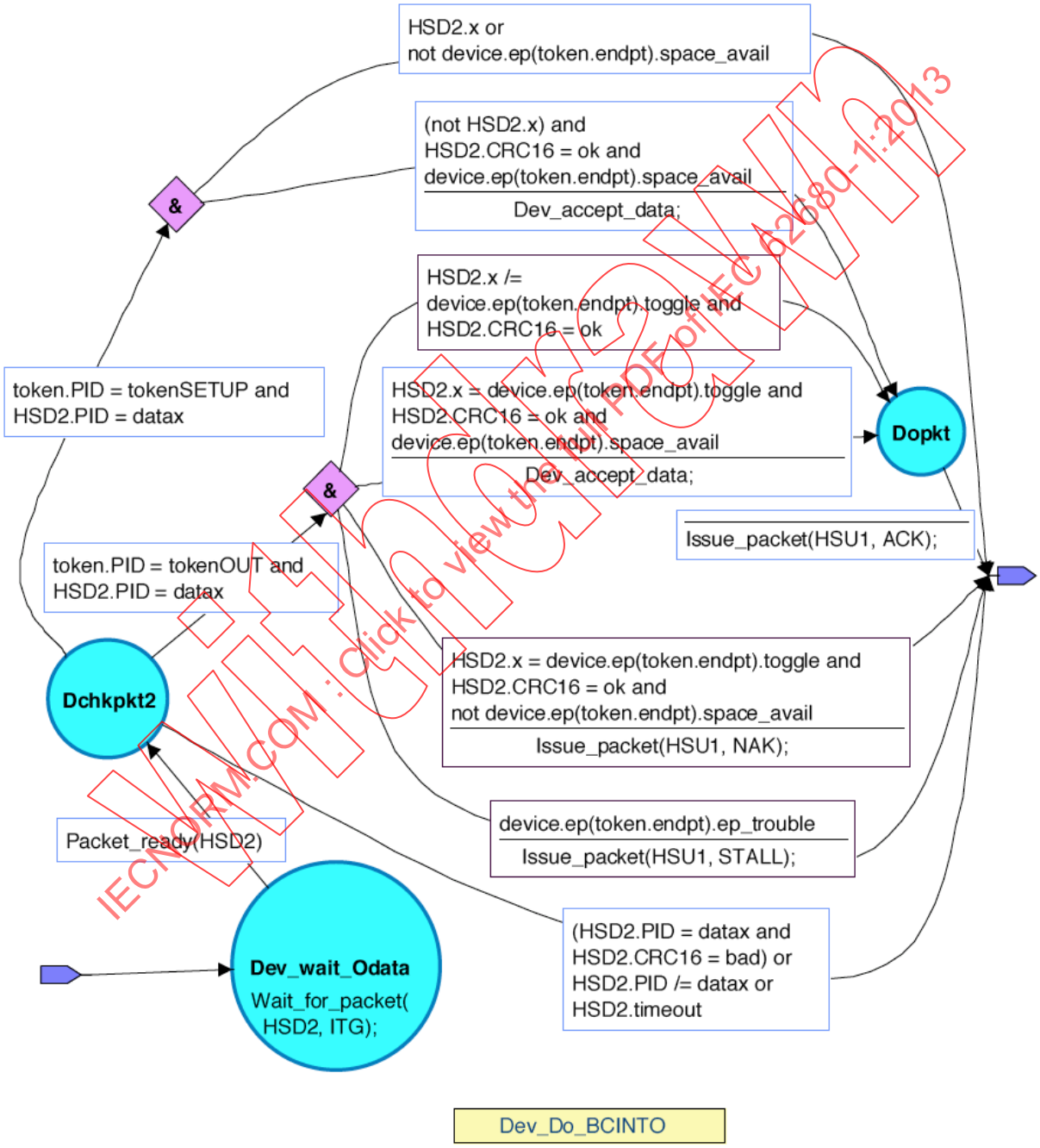
Un établissement de liaison NAK indique que le point d'extrémité n'a pas l'espace pour des données utiles *wMaxPacketSize*. Le contrôleur d'hôte retente le PING plus tard pour réinterroger le point d'extrémité. Un appareil peut répondre à un PING avec un NAK pendant de longues durées. Une réponse NAK n'est pas une raison de retirer une demande de transfert pour un contrôleur d'hôte. Si un appareil répond avec un NAK dans une (micro)trame, le contrôleur d'hôte peut choisir d'émettre la prochaine transaction dans le *bInterval* suivant spécifié pour le point d'extrémité. Cependant, l'appareil doit être préparé à recevoir des PING en tant que transactions séquentielles, par ex. un immédiatement après l'autre.

Un établissement de liaison ACK indique que le point d'extrémité a de l'espace pour des données utiles *wMaxPacketSize*. Le contrôleur d'hôte doit générer une transaction OUT avec une phase DATA comme la prochaine transaction pour le point d'extrémité. Le contrôleur d'hôte peut générer d'autres transactions vers d'autres appareils ou points d'extrémité avant la transaction OUT/DATA pour ce point d'extrémité.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Si le point d'extrémité répond à la transaction OUT/DATA avec un établissement de liaison ACK, cela signifie que le point d'extrémité a réussi à accepter les données et a de la place pour d'autres données utiles *wMaxPacketSize*. Le contrôleur d'hôte continue avec des transactions OUT/DATA (elles ne doivent pas nécessairement être des transactions sur le bus) tant qu'il a des transactions à générer.

Si au lieu de cela le point d'extrémité répond à la transaction OUT/DATA avec un établissement de liaison NYET, cela signifie que le point d'extrémité a accepté les données, mais n'a pas de la place pour d'autres données utiles *wMaxPacketSize*. Le contrôleur d'hôte doit réutiliser un jeton PING jusqu'à ce que le point d'extrémité indique avoir de l'espace.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
HSD2.x or Not device.ep(token.endpt).space_avail	HSD2.x ou Not device.ep(token.endpt).space_avail
(not HSD2.x) and	(not HSD2.x) et

Anglais	Français
HSD2.CRC16 = ok and device.ep(token.endpt).space_avail	HSD2.CRC16 = ok et device.ep(token.endpt).space_avail
HSD2.x /= device.ep(token.endpt).toggle and HSD2.CRC16 = ok	HSD2.x /= device.ep(token.endpt).toggle et HSD2.CRC16 = ok
token.PID = tokenSETUP and HSD2.PID = datax	token.PID = tokenSETUP et HSD2.PID = datax
token.PID = tokenOUT and HSD2.PID = datax	token.PID = tokenOUT et HSD2.PID = datax
HSD2.x = device.ep(token.endpt).toggle and HSD2.CRC16 = ok and Not device.ep(token.endpt).space_avail	HSD2.x = device.ep(token.endpt).toggle et HSD2.CRC16 = ok et Not device.ep(token.endpt).space_avail
(HSD2.PID = datax and HSD2.CRC16 = bad) or HSD2.PID /= datax or HSD2.timeout	(HSD2.PID = datax et HSD2.CRC16 = bad) ou HSD2.PID /= datax ou HSD2.timeout
HSD2.x = device.ep(token.endpt).toggle and HSD2.CRC16 = ok and device.ep(token.endpt).space_avail	HSD2.x = device.ep(token.endpt).toggle et HSD2.CRC16 = ok et device.ep(token.endpt).space_avail

Figure 8-27. Diagramme d'états de ping de commande/OUT en masse à haute vitesse d'hôte

8.5.1.1 Réponses NAK à OUT/DATA pendant le protocole PING

Le point d'extrémité peut également répondre à une transaction OUT/DATA par un établissement de liaison NAK. Cela signifie que le point d'extrémité n'a pas accepté les données et n'a pas assez d'espace pour des données utiles *wMaxPacketSize* pour le moment. Le contrôleur d'hôte doit réutiliser un jeton PING jusqu'à ce que le point d'extrémité indique avoir de l'espace.

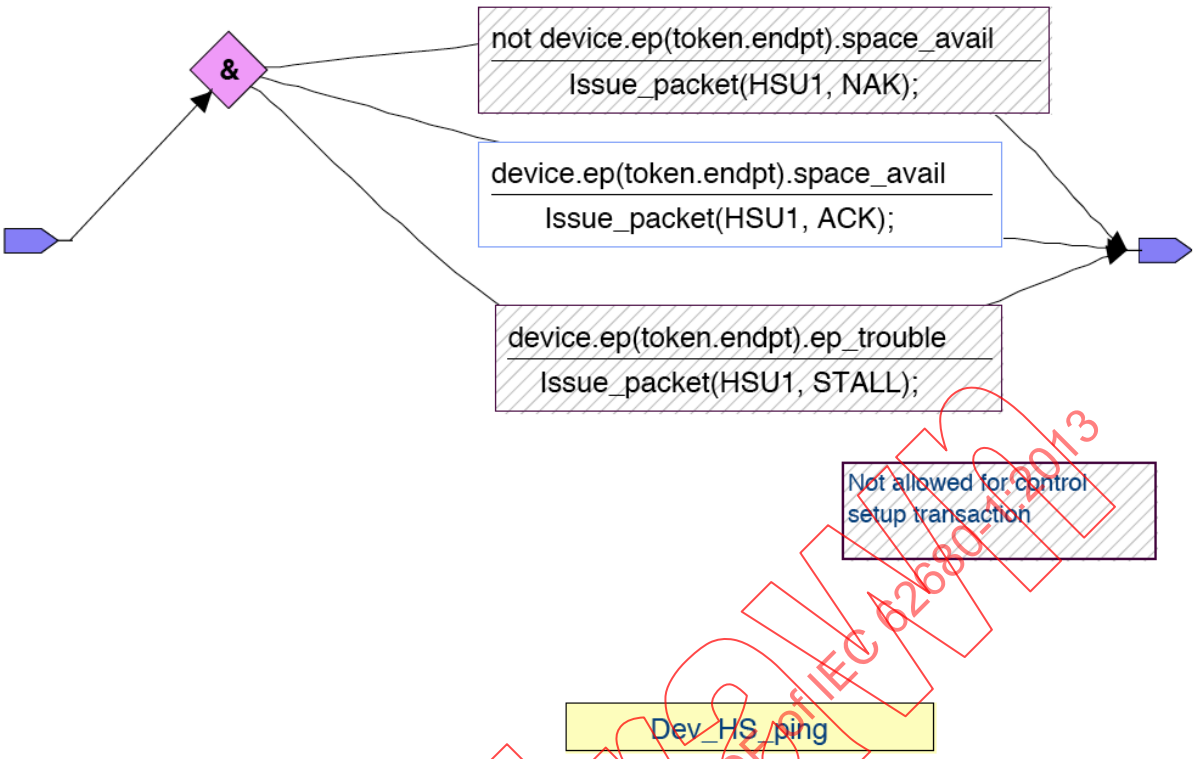
Une réponse NAK est attendue comme une occurrence inhabituelle. Un point d'extrémité en masse/de commande à haute vitesse doit spécifier sa cadence NAK maximum dans son descripteur de point d'extrémité. Le point d'extrémité peut faire NAK au maximum une fois par période *bInterval*. Un NAK suggère que le point d'extrémité a répondu à un précédent OUT ou PING par un établissement de liaison inapproprié, ou que le point d'extrémité a transité vers un état où il ne pouvait pas (provisoirement) accepter des données. Un point d'extrémité peut utiliser un *bInterval* de zéro pour indiquer qu'il ne NAK jamais. Un point d'extrémité doit toujours pouvoir accepter un PING provenant de l'hôte, même s'il ne NAK jamais.

Si le délai a expiré après la phase de données, l'hôte doit réutiliser un jeton PING. A noter qu'une transition revenant à l'état PING n'influe pas sur l'état de basculement des données de la phase de données de la transaction.

La Figure 8-27 indique le diagramme d'états du contrôleur d'hôte pour les interactions et transitions entre les jetons PING et OUT/DATA et les établissements de liaisons ACK, NAK et NYET autorisés pour le mécanisme PING.

La Figure 8-29 indique le diagramme d'états de point d'extrémité de l'appareil pour PING basé sur l'espace de mémoire tampon que le point d'extrémité a de disponible.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

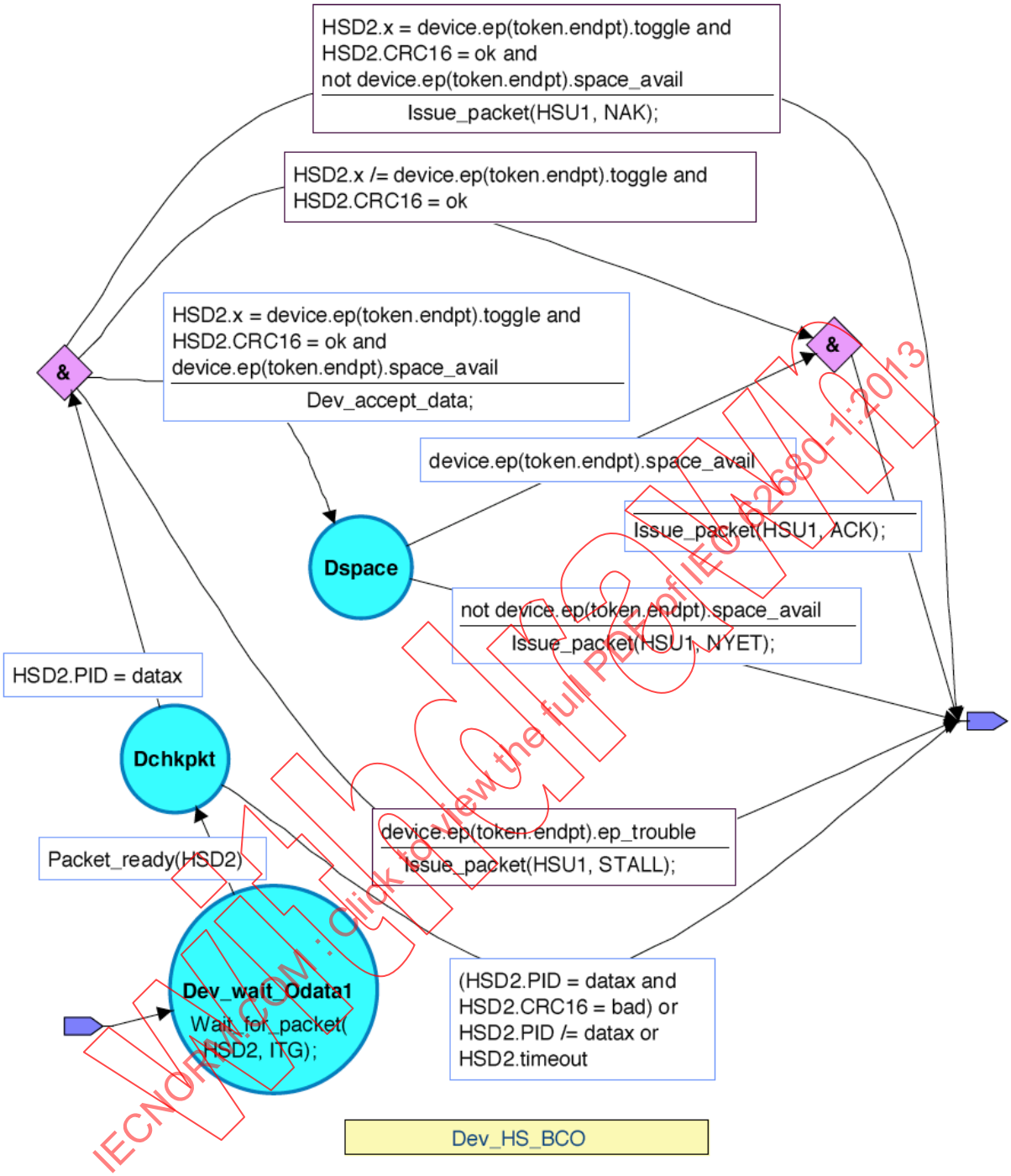


Légende

Anglais	Français
Not allowed for control setup transaction	Non autorisé pour la transaction de configuration de contrôle

Figure 8-28. Diagramme d'états Dev_HS_ping

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
HSD2.x = device.ep(token.endpt).toggle and HSD2.CRC16 = ok and not device.ep(token.endpt).space_avail	HSD2.x = device.ep(token.endpt).toggle et HSD2.CRC16 = ok et not device.ep(token.endpt).space_avail
HSD2.x /= device.ep(token.endpt).toggle and HSD2.CRC16 = ok	HSD2.x /= device.ep(token.endpt).toggle et HSD2.CRC16 = ok
HSD2.x = device.ep(token.endpt).toggle and HSD2.CRC16 = ok and device.ep(token.endpt).ep_trouble	HSD2.x = device.ep(token.endpt).toggle et HSD2.CRC16 = ok et device.ep(token.endpt).ep_trouble
(HSD2.PID = datax and HSD2.CRC16 = bad) or HSD2.PID /= datax or HSD2.timeout	(HSD2.PID = datax et HSD2.CRC16 = bad) ou HSD2.PID /= datax ou HSD2.timeout

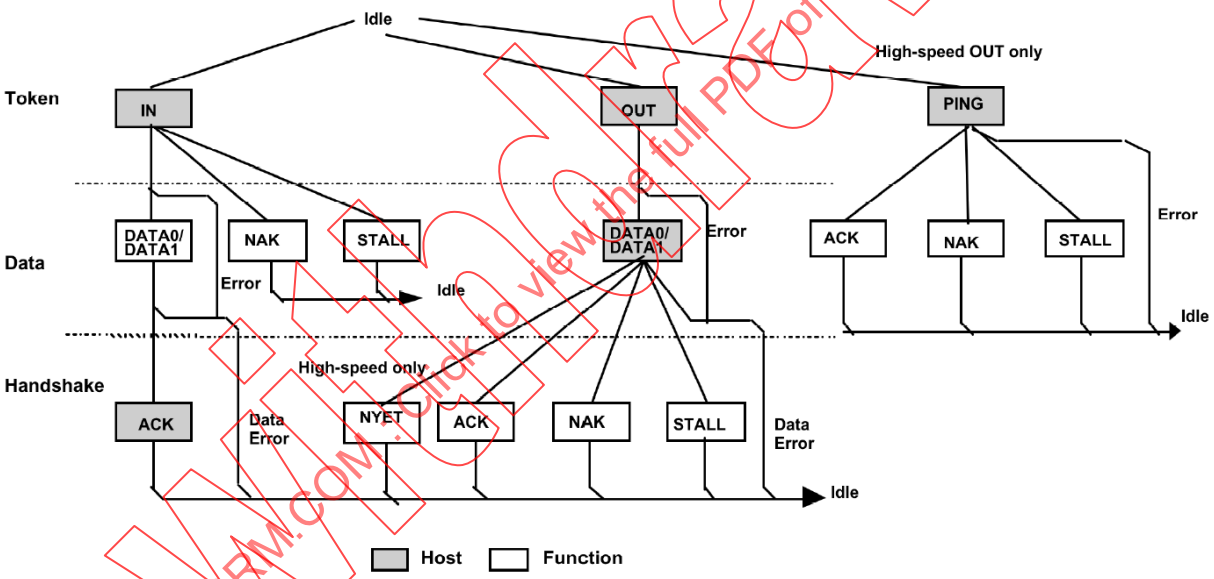
Figure 8-29. Diagramme d'états de commande/OUT de volume à haute vitesse d'appareil

Les appareils/points d'extrémité à vitesse intégrale/basse vitesse ne doivent pas prendre en charge le protocole PING. Les contrôleurs d'hôte ne doivent pas prendre en charge le protocole PING pour les appareils à vitesse intégrale/basse vitesse.

Note: Le protocole PING n'est pas non plus inclus dans la définition du protocole des transactions divisées. Certaines transactions divisées ont un contrôle de débit équivalent sans utiliser le PING. D'autres transactions divisées ne bénéficient pas du PING comme défini. Dans tous les cas, les transactions divisées qui peuvent renvoyer un établissement de liaison NAK ont de petites données utiles dont il convient qu'elles aient un impact mineur sur le bus à haute vitesse. Les hubs doivent prendre en charge le PING sur leur point d'extrémité de contrôle, mais le PING n'est pas défini pour les transactions divisées qui sont utilisées pour communiquer avec des appareils vitesse intégrale/basse vitesse pris en charge par un hub.

8.5.2 Transactions en masse

Les types de transactions en masse se caractérisent par la capacité de garantir une transmission de données sans erreur entre l'hôte et une fonction grâce à une détection d'erreur et des nouvelles tentatives. Les transactions en masse utilisent une transaction en trois phases constituée de paquets de jeton, de données et d'établissement de liaison comme représenté à la Figure 8-30. Dans certaines conditions de contrôle de débit et d'arrêt, la phase de données peut être remplacée par un établissement de liaison, ce qui donne une transaction en deux phases dans laquelle aucune donnée n'est transmise. Les paquets PING et NYET doivent être utilisés uniquement avec des appareils fonctionnant à haute vitesse.



Légende

Anglais	Français
Token	Jeton
High-speed OUT only	Haute vitesse OUT uniquement
Data	Données
Error	Erreur
Idle	Inactif
Handshake	Etablissement de liaison
High-speed only	Grande vitesse uniquement
Data Error	Erreur de données
Host	Hôte
Function	Fonction

Figure 8-30. Format des transactions en masse

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Quand l'hôte est prêt à recevoir les données en masse, il émet un jeton IN. Le point d'extrémité de la fonction répond soit en renvoyant un paquet de données, soit si elle ne peut pas renvoyer des données un établissement de liaison NAK ou STALL. NAK indique que la fonction ne peut pas renvoyer des données pour l'instant, alors que STALL indique que le point d'extrémité est arrêté définitivement et nécessite une intervention logicielle du système par USB. Si l'hôte reçoit un paquet de données valide, il répond avec un établissement de liaison ACK. Si l'hôte détecte une erreur en recevant des données, il ne renvoie pas de paquet d'établissement de liaison à la fonction.

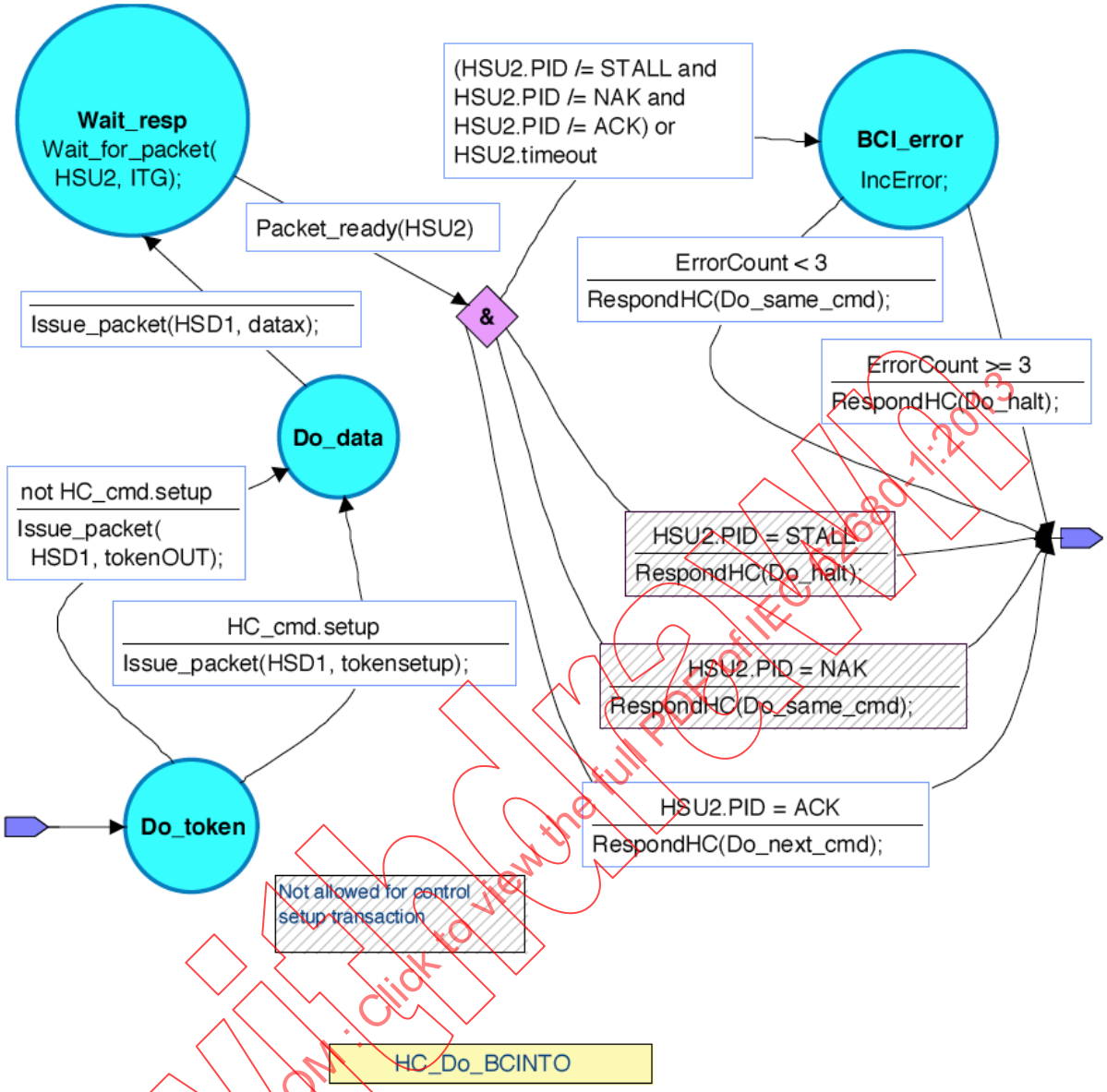
Quand l'hôte est prêt à transmettre des données en masse, il émet d'abord un paquet de jetons OUT suivi par un paquet de données (ou un paquet de jetons spécial PING, voir la Section 8.5.1). Si les données sont reçues sans erreur par la fonction, il renvoie un établissement de données sur trois (ou quatre y compris NYET, pour un appareil fonctionnant à haute vitesse):

- ACK indique que le paquet de données a été reçu sans erreur, informe l'hôte qu'il peut envoyer le prochain paquet dans la séquence.
- NAK indique que les données ont été reçues sans erreur, mais qu'il convient que l'hôte renvoie les données, car la fonction était dans un état provisoire qui l'empêchait d'accepter les données (par ex.: mémoire tampon pleine).
- Si le point d'extrémité était arrêté, STALL est renvoyé pour indiquer qu'il convient que l'hôte ne réessaie pas la transmission, car il y a une condition d'erreur sur la fonction.

Si le paquet de données a été reçu avec une erreur de CRC d'insertion de bit, aucun établissement de liaison n'est renvoyé.

Les Figures 8-31 et 8-32 montrent les diagrammes d'états de l'appareil respectivement pour les transactions à vitesse intégrale/basse vitesse OUT en masse, de contrôle et d'interruption. Les Figures 8-27, 8-28 et 8-29 montrent les diagrammes d'états pour des transactions à haute vitesse. Les Figures 8-33 et 8-34 montrent les diagrammes d'états de l'hôte et de l'appareil respectivement pour les transactions IN en masse, de contrôle et d'interruption.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

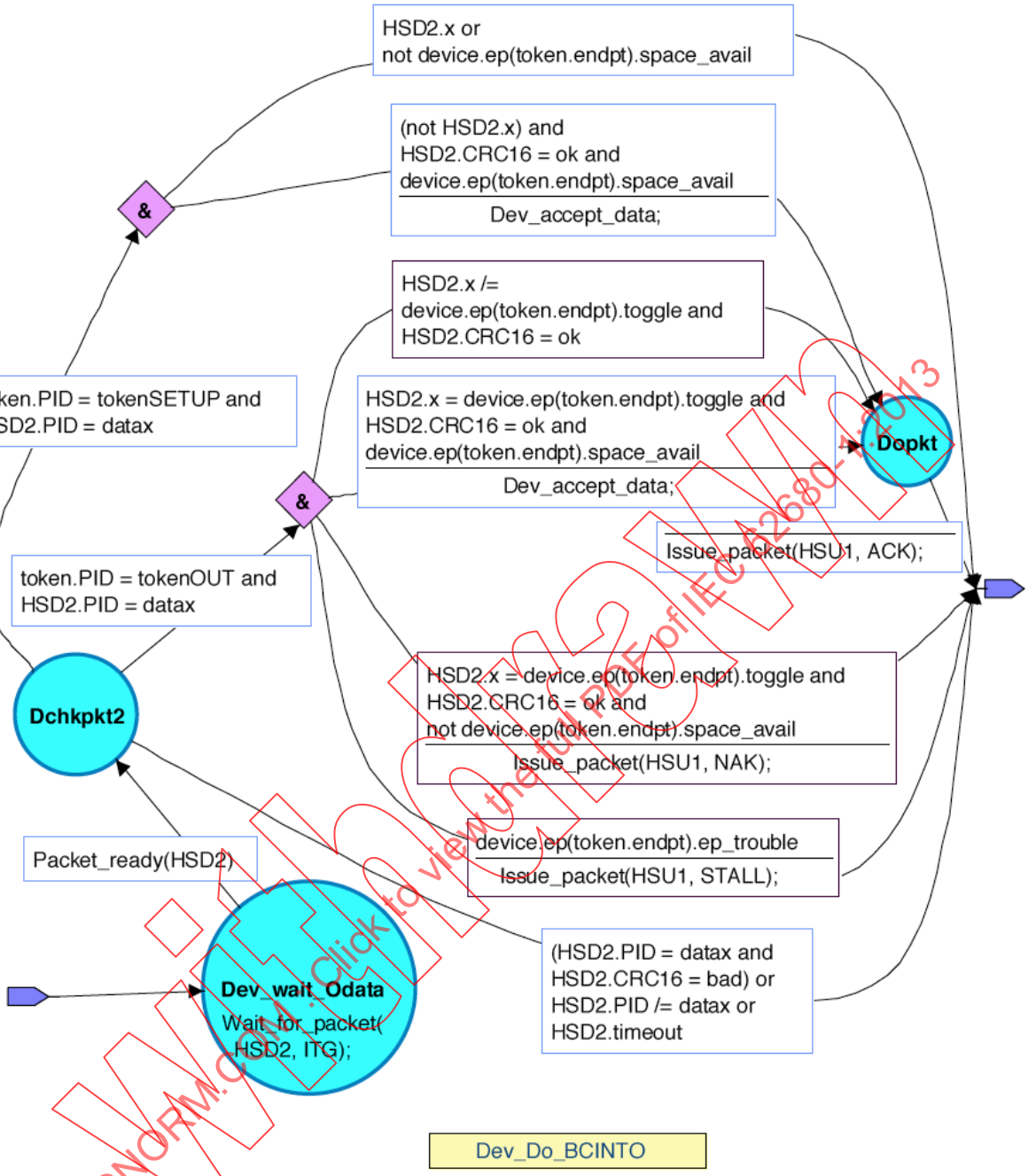


Légende

Anglais	Français
(HSU2.PID /= STALL and HSU2.PID /= NAK and HSU2.PID /= ACK) or HSU2.timeout	(HSU2.PID /= STALL et HSU2.PID /= NAK et HSU2.PID /= ACK) ou HSU2.timeout
Not allowed for control setup transaction	Non autorisé pour la transaction de configuration de contrôle

Figure 8-31. Diagrammes d'états de l'hôte pour les transactions OUT de masse/contrôle/interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



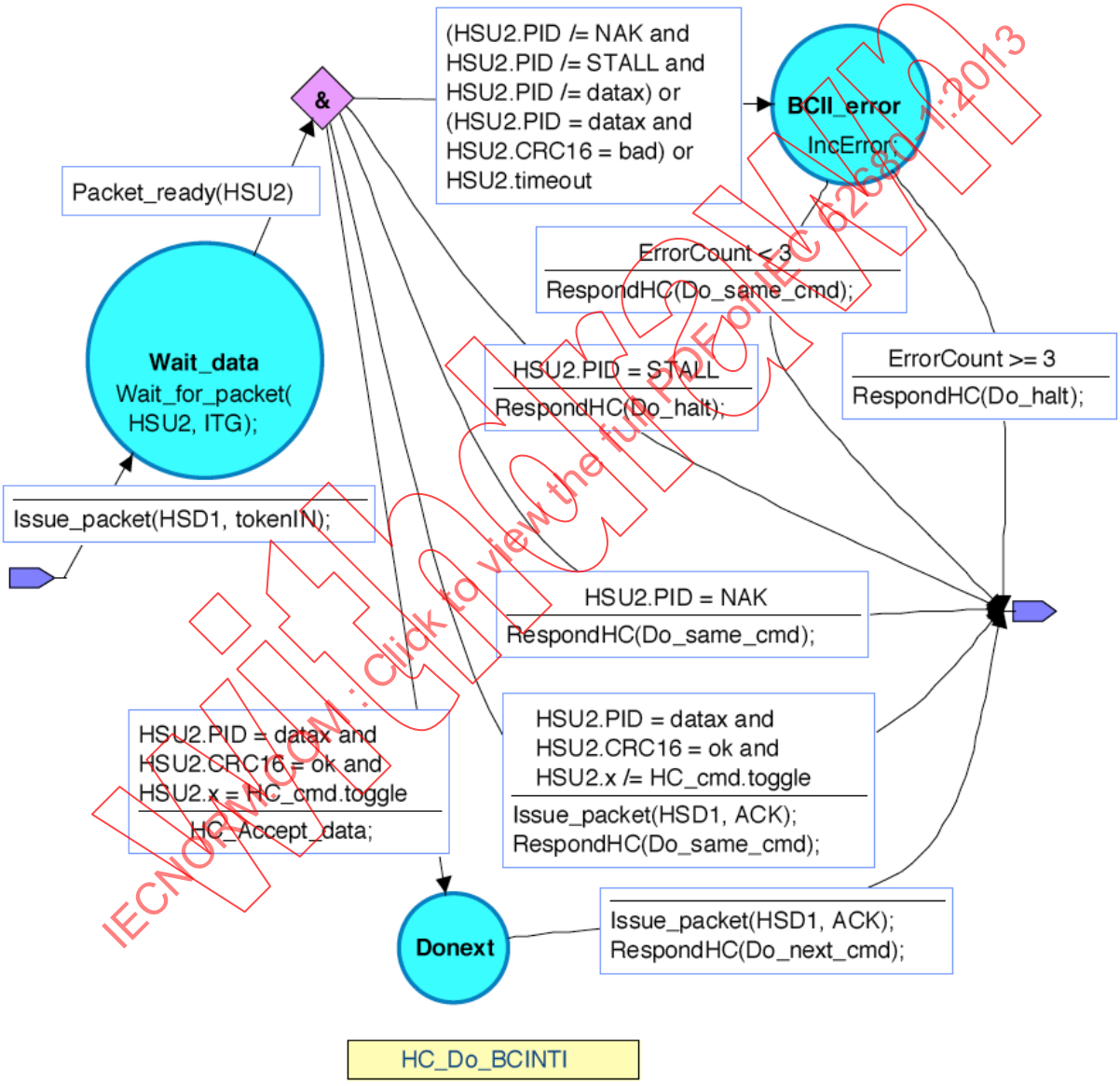
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
HSD2.x or not device.ep(token.endpt).space_avail	HSD2.x ou not device.ep(token.endpt).space_avail
(not HSD2.x) and HSD2.CRC16 = ok and device.ep(token.endpt).space_avail	(not HSD2.x) et HSD2.CRC16 = ok et device.ep(token.endpt).space_avail
HSD2.x /= device.ep(token.endpt).toggle and HSD2.CRC16 = ok	HSD2.x /= device.ep(token.endpt).toggle et HSD2.CRC16 = ok
token.PID = tokenSETUP and HSD2.PID = datax	token.PID = tokenSETUP et HSD2.PID = datax
token.PID tokenOUT and HSD2.PID = datax	token.PID tokenOUT et HSD2.PID = datax
HSD2.x/= device.ep(token.endpt).toggle and HSD2.CRC16 = ok and device.ep(token.endpt).space_avail	HSD2.x/= device.ep(token.endpt).toggle et HSD2.CRC16 = ok and device.ep(token.endpt).space_avail

Anglais	Français
token.PID = tokenOUT and HSD2.PID = datax	token.PID = tokenOUT et HSD2.PID = datax
HSD2.x = device.ep(token.endpt).toggle and HSD2.CRC16 = ok and not device.ep(token.endpt).space_avail	HSD2.x = device.ep(token.endpt).toggle et HSD2.CRC16 = ok et not device.ep(token.endpt).space_avail
(HSD2.PID = datax and HSD2.CRC16 = bad or HSD2.PID /= datax or HSD2.timeout	(HSD2.PID = datax et HSD2.CRC16 = bad ou HSD2.PID /= datax ou HSD2.timeout

Figure 8-32. Diagrammes d'états de l'appareil pour les transactions OUT de masse/contrôle/interruption



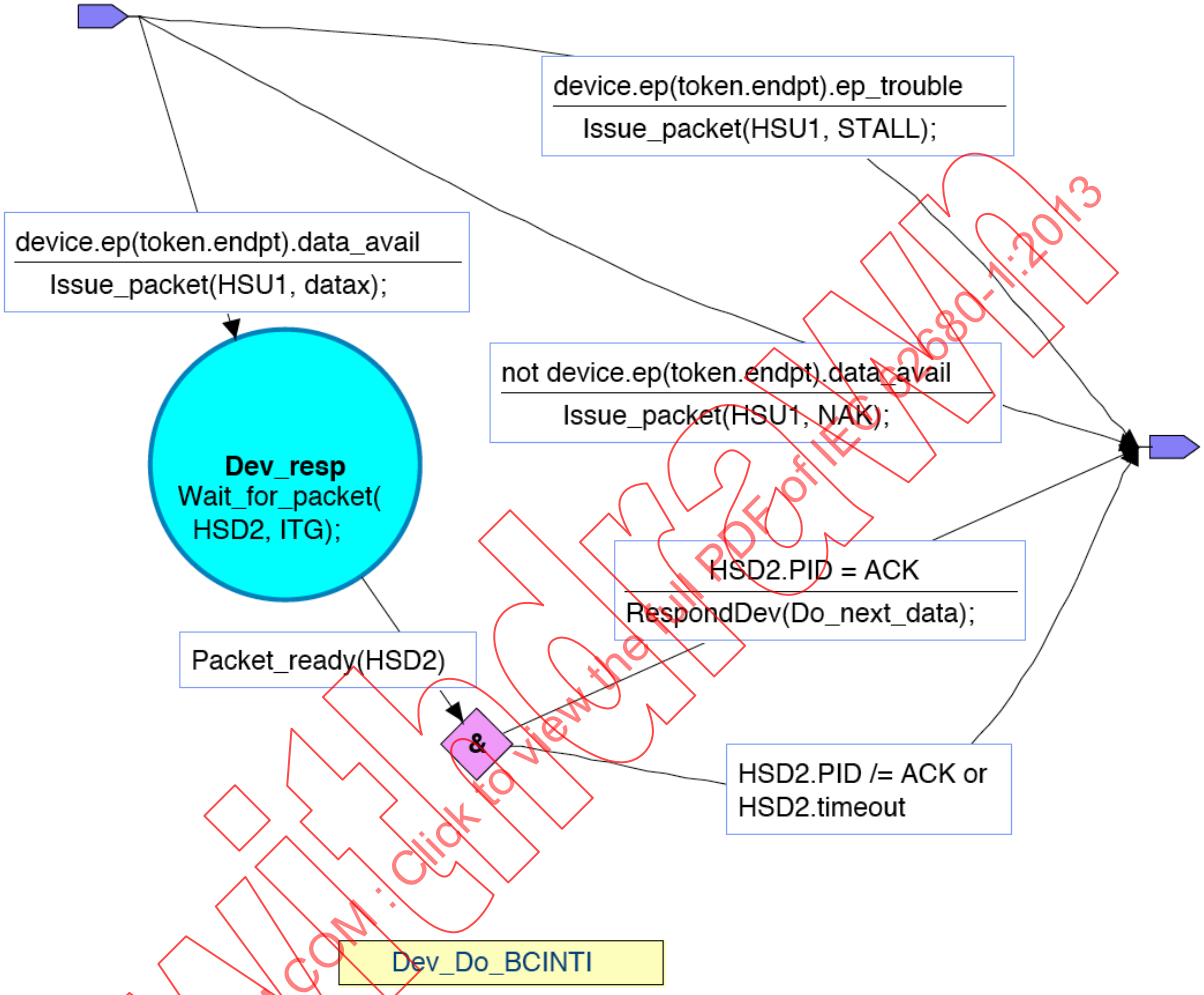
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
(HSU2.PID /= NAK and HSU2.PID /= STALL and HSU2.PID /= datax) or (HSU2.PID = datax and HSU2.CRC16 =bad) or HSU2.timeout	(HSU2.PID /= NAK et HSU2.PID /= STALL et HSU2.PID /= datax) ou (HSU2.PID = datax et HSU2.CRC16 =bad) ou HSU2.timeout
HSU2.PID = datax and HSU2.CRC16 = ok and	HSU2.PID = datax et HSU2.CRC16 = ok et

Anglais	Français
HSU2.x = HC_cmd.toggle	HSU2.x = HC_cmd.toggle
HSU2.PID datax and HSU2.CRC16 = ok and HSU2.x /= HC_cmd.toggle	HSU2.PID datax et HSU2.CRC16 = ok et HSU2.x /= HC_cmd.toggle

Figure 8-33. Diagrammes d'états de l'hôte pour les transactions IN de masse/contrôle/interruption



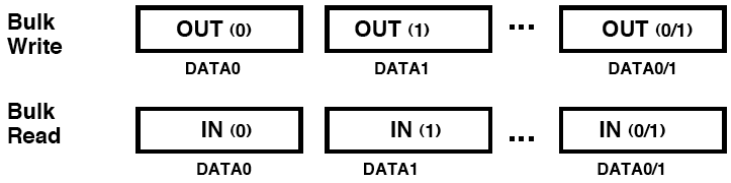
Légende

Anglais	Français
HSD2.PID /= ACK or HSD2.timeout	HSD2.PID /= ACK ou HSD2.timeout

Figure 8-34. Diagrammes d'états de l'appareil pour les transactions IN de masse/contrôle/interruption

La Figure 8-35 montre l'usage des bits de séquence et du PID de données pour la lecture et l'écriture en masse. La synchronisation des paquets de données se fait grâce aux bits de basculement de séquence de données et les PID DATA0/DATA1. Une séquence de basculement de point d'extrémité en masse est initialisée à DATA0 quand le point d'extrémité rencontre un évènement de configuration (les évènements de configuration sont expliqués dans les Sections 9.1.1.5 et 9.4.5). Le basculement de données sur un point d'extrémité n'est PAS initialisé en conséquence directe d'un transfert de paquet de données court ou le retrait d'un IRP.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

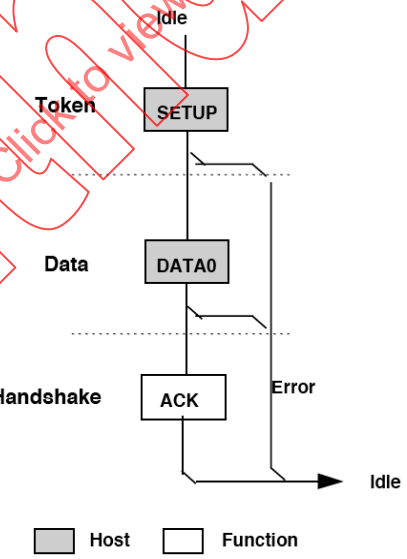
Anglais	Français
Bulk Write	Ecriture en masse
Bulk Read	Lecture en masse

Figure 8-35. Lecture et écriture en masse

L'hôte initialise toujours la première transaction d'un transfert de bus au PID DATA0 avec un évènement de configuration. La deuxième transaction utilise un PID DATA1, et les transferts de données successifs alternent pour le reste du transfert en masse. Le paquet de données transmet des basculements sur réception d'un ACK, et le récepteur bascule sur réception et acceptation d'un paquet de données valide (voir la Section 8.6).

8.5.3 Transferts de contrôle

Les transferts de contrôle ont au moins deux étapes de transaction: Configuration (Setup) et statut (Status). Un transfert de contrôle peut facultativement contenir une étape de données entre les étapes de configuration et de statut. Pendant l'étape de configuration, une transaction SETUP est utilisée pour transmettre des informations au point d'extrémité de contrôle d'une fonction. Les transactions SETUP sont similaires en format d'OUT, mais utilisent un SETUP plutôt qu'un PID OUT. La Figure 8-36 indique le format d'une transaction SETUP. Un SETUP utilise toujours un PID DATA0 pour le champ de données de la transaction SETUP. La fonction recevant un SETUP doit accepter les données SETUP et répondre avec ACK, et si les données sont corrompues elle élimine les données et ne renvoie pas d'établissement de liaison.



Légende

Anglais	Français
Idle	Inactif
Token	Jeton
Data	Données
Handshake	Etablissement de liaison
Error	Erreur
Idle	Inactif
Host	Hôte
Function	Fonction

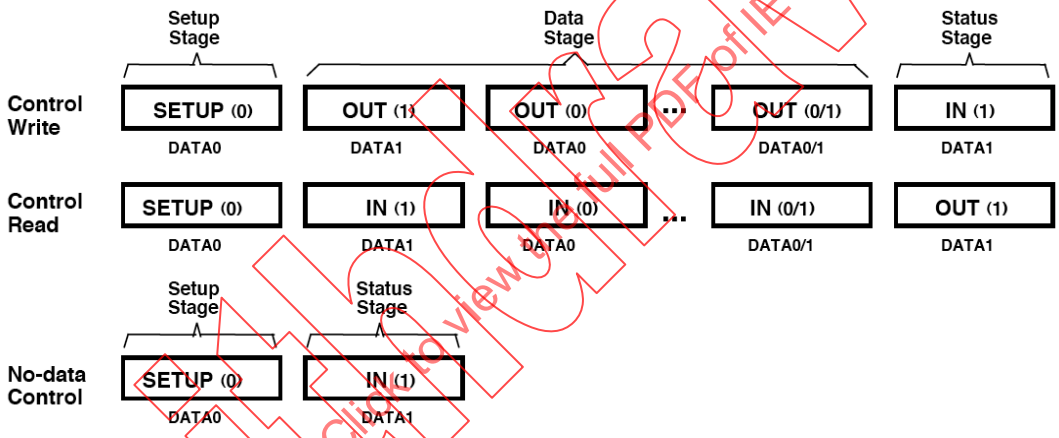
Figure 8-36. Transaction SETUP de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

L'étape de données, si présente, d'un transfert de contrôle consiste en une ou plusieurs transactions IN ou OUT et suit les mêmes règles de protocole que les transferts en masse. Toutes les transactions dans l'étape de données doivent être dans le même sens (c'est-à-dire toutes des IN ou toutes des OUT). La quantité de données à envoyer pendant l'étape de données et son sens sont spécifiés pendant l'étape de configuration. Si la quantité de données dépasse la taille du paquet de données prénégociée, les données sont envoyées en plusieurs transactions (IN ou OUT) qui contiennent la taille maximum de paquet. Toutes les données restantes sont envoyées comme reliquat dans la dernière transaction.

L'étape de statut d'un transfert de contrôle est la dernière transaction dans la séquence. Les transactions de l'étape de statut suivent la même séquence de protocole que les transactions en masse. L'étape de statut pour les appareils fonctionnant à haute vitesse inclut également le protocole PING. Une étape de statut s'identifie par un changement de sens du débit de données par rapport à l'étape précédente et utilise toujours un PID DATA1. Si, par exemple l'étape de données est consiste en des OUT, le statut est une simple transaction IN. Si la séquence de contrôle n'a pas d'étape de données, alors elle consiste en une étape de configuration suivie d'une étape de statut constitué d'une transaction IN.

La Figure 8-37 indique l'ordre d'une transaction, la valeur des bits de séquence de données et les types de PID de données pour les séquences de lecture et d'écriture de contrôle. Les bits de séquence sont indiqués entre in parenthèses.



Légende

Anglais	Français
Setup Stage	Etape de configuration
Data Stage	Etape de données
Status Stage	Etape d'état
Control Write	Ecriture de contrôle
Control Read	Lecture de contrôle
No-data Control	Contrôle des données nulles

Figure 8-37. Séquences de lecture et d'écriture de contrôle

Quand un établissement de liaison STALL est envoyé à un point d'extrémité de contrôle dans les étapes de données ou de statut d'un transfert de contrôle, un établissement de liaison STALL doit être renvoyé pour tous les accès réussis à ce point d'extrémité jusqu'à ce qu'un PID SETUP soit reçu. Le point d'extrémité n'est pas tenu de renvoyer un établissement de liaison STALL après avoir reçu un PID SETUP ultérieur. Pour le point d'extrémité par défaut, si un établissement de liaison ACK est renvoyé pour la transaction SETUP, l'hôte escompte que le point d'extrémité se soit automatiquement rétablie de la condition qui a causé le STALL et le point d'extrémité doit fonctionner normalement.

8.5.3.1 Rapports de résultats

L'étape de statut signale à l'hôte le résultat des étapes précédentes de configuration et de données du transfert. Trois résultats peuvent être renvoyés:

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- La séquence de commande a été exécutée avec succès.
- Echec de la séquence de commande.
- La fonction est encore occupée à exécuter la commande.

L'envoi de rapports de statut va toujours dans le sens de la fonction vers l'hôte. Le Tableau 8-7 résume le type de réponses requises pour chacun. Les transferts d'écriture de contrôle renvoient les informations de statut dans la phase de données de la transaction de l'étape de statut. Les transferts de lecture de contrôle renvoient les informations de statut dans la phase d'établissement de liaison d'une transaction d'étape de statut, après que l'hôte a émis un paquet de données de longueur zéro pendant la phase de données précédentes.

Tableau 8-7. Réponses de l'étape de statut

Réponse de statut	Transfert d'écriture de contrôle (envoyé pendant la phase de données)	Transfert de lecture de contrôle (envoyé pendant la phase d'établissement de liaison)
La fonction s'exécute	Paquet de données de longueur zéro	Etablissement de liaison ACK
La fonction a une erreur	Etablissement de liaison STALL	Etablissement de liaison STALL
La fonction est occupée	Etablissement de liaison NAK	Etablissement de liaison NAK

Pour les lectures de contrôle, l'hôte doit envoyer un jeton OUT ou un jeton spécial PING (pour un appareil fonctionnant à haute vitesse) à la canalisation de contrôle pour initier l'étape de statut. L'hôte peut envoyer uniquement un paquet de données de longueur zéro pendant cette phase, mais la fonction peut accepter toute longueur de paquet comme requête de statut valide. La réponse d'établissement de liaison de la canalisation à ce paquet de données indique le statut actuel. NAK indique que la fonction est encore en train de traiter la commande et qu'il convient que l'hôte continue l'étape de statut. ACK indique que la fonction a exécuté la commande et qu'elle est prête à accepter une nouvelle commande. STALL indique que la fonction rencontre une erreur qui l'empêche d'exécuter la commande.

Pour l'écriture de contrôle, l'hôte envoie un jeton IN à la canalisation de contrôle pour initier l'étape de statut. La fonction répond avec un paquet d'établissement de liaison ou de données de longueur zéro pour indiquer sont statut actuel. NAK indique que la fonction est encore en train de traiter la commande et qu'il convient que l'hôte continue l'étape de statut, un renvoi d'un paquet de longueur zéro indique une exécution normale de la commande, et STALL indique que la fonction ne peut pas exécuter la commande. La fonction attend que l'hôte réponde au paquet de données dans l'étape de statut avec ACK. Si la fonction ne reçoit pas ACK, elle reste dans l'étape de statut de la commande et continue de renvoyer le paquet de données de longueur zéro tant que l'hôte continue d'envoyer des jetons IN.

Pendant une étape de données, si une canalisation de commande reçoit davantage de données ou si elle doit renvoyer davantage de données qu'indiqué auparavant dans l'étape de configuration (voir la Section 8.5.3.2), il convient qu'elle renvoie STALL. Si une canalisation de contrôle renvoie STALL pendant l'étape de données, il n'y a pas d'étape de statut pour ce transfert de contrôle.

8.5.3.2 Etape de données à longueur variable

Une canalisation de contrôle peut avoir une phase de données à longueur variable dans laquelle l'hôte demande davantage de données que dans la structure de données spécifiée. Quand l'intégralité de la structure de données est renvoyée à l'hôte, il convient que la fonction indique que l'étape de données est terminée en renvoyant un paquet qui est plus court que *MaxPacketSize* pour la canalisation. Si la structure de données est un multiple exact de MaxPacketSize pour la canalisation, la fonction renvoie un paquet de longueur zéro pour indiquer la fin de l'étape de données.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8.5.3.3 Gestion des erreurs sur la dernière transaction de données

Si l'établissement de données ACK sur une transaction IN est corrompu, la fonction et l'hôte ne concluent temporairement pas identiquement quant à la réussite de la transaction. Si la transaction est suivie d'un autre IN, le mécanisme de relance de basculement détecte l'incohérence et résout l'erreur. Si ACK était sur le dernier IN d'une étape de données, le mécanisme de relance de basculement ne peut pas être utilisé et un système alternatif doit être utilisé.

L'hôte qui a reçu les données du dernier avec succès IN envoie ACK. Plus tard, l'hôte émet un jeton OUT pour commencer l'étape de statut du transfert. Si la fonction n'a pas reçu l'ACK qui terminait l'étape de données, la fonction interprète le début de l'étape de statut comme une vérification que l'hôte a bien reçu les données. L'écriture de contrôle n'a pas cette ambigüité. Si un établissement de liaison ACK sur un OUT se corrompt, l'hôte ne passe à l'étape de statut, mais réessaie les dernières données. Une analyse détaillée de la politique de relance est présentée dans la Section 8.6.4.

8.5.3.4 Etablissements de liaison STALL renvoyés par les canalisations de contrôle

Les canalisations de contrôle ont la capacité unique de renvoyer un établissement de liaison STALL en raison des problèmes de fonction dans les transferts de contrôle. Si l'appareil ne parvient pas à exécuter une commande, il renvoie un STALL dans les étapes de données et/ou de statut du transfert de contrôle. Contrairement au cas de calage fonctionnel, le calage de protocole n'indique pas d'erreur avec l'appareil. La condition de STALL de protocole dure jusqu'à la réception du prochain SETUP, et la fonction renvoie STALL en réponse à une transaction IN ou OUT sur la canalisation jusqu'à ce que la transaction SETUP soit reçue. En général, le calage de protocole indique que la demande ou ses paramètres ne sont pas compris par l'appareil et il donc fournit un mécanisme pour étendre les demandes USB.

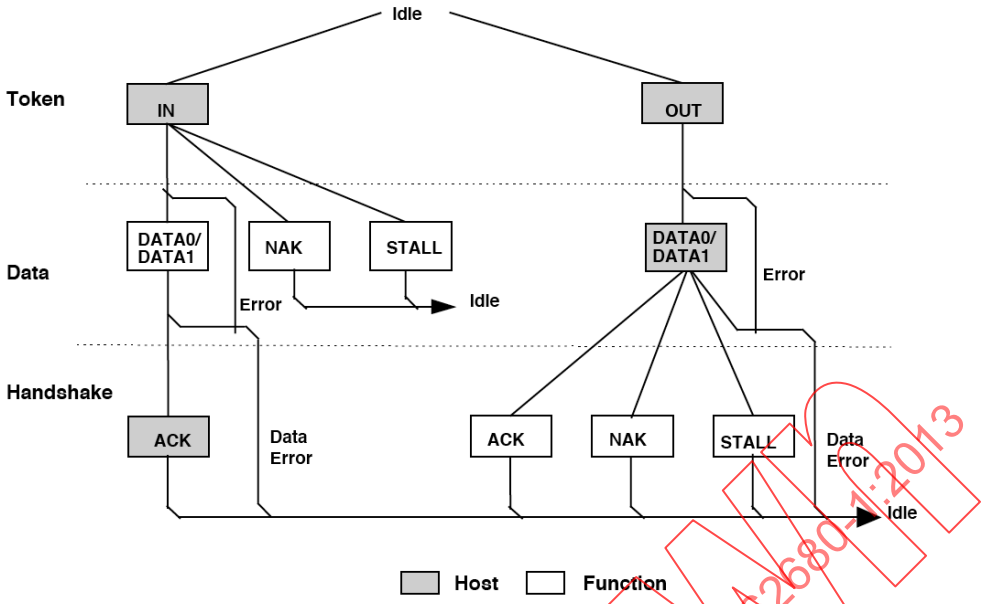
Une canalisation de contrôle peut également prendre en charge un calage fonctionnel, mais ce n'est pas recommandé. Il s'agit d'un cas dégénératif, car un calage fonctionnel sur une canalisation de contrôle indique qu'il a perdu la capacité de communiquer avec l'hôte. Si la canalisation de contrôle ne prend pas en charge le calage fonctionnel, alors il doit posséder une caractéristique *Halt*, que l'hôte peut régler ou effacer. Le chapitre 9 explique en détail comment traiter le cas particulier d'une caractéristique *Halt* sur une canalisation de contrôle. Un appareil bien conçu associe toutes ses fonctions et caractéristiques Halt avec des points d'extrémité non contrôle. Il convient que les canalisations de contrôle soient réservées pour desservir les demandes USB.

8.5.4 Transactions d'interruption

Les transactions d'interruption peuvent consister en des transferts IN ou OUT. Sur réception d'un jeton IN, une fonction peut renvoyer des données, NAK ou STALL. Si un point d'extrémité n'a pas de nouvelles informations d'interruption à renvoyer (c.-à-d. qu'aucune interruption n'est en cours), la fonction renvoie un établissement de liaison NAK lors de la phase de données. Si la caractéristique Halt est définie pour le point d'extrémité d'interruption, la fonction renvoie un établissement de liaison STALL. Si une interruption est en cours, la fonction renvoie les informations d'interruption comme un paquet de données. L'hôte, en réponse à la réception du paquet de données, émet soit un établissement de liaison ACK si les données sont reçues sans erreur soit ne renvoie pas d'établissement de liaison si le paquet de données a été reçu corrompu. La Figure 8-38 indique le format d'une transaction d'interruption.

La Section 5.9.1 contient des informations supplémentaires sur les points d'extrémité d'interruption de bande supérieure à haute vitesse. De tels points d'extrémité utilisent plusieurs transactions dans une microtrame comme défini dans cette section. Chaque transaction pour un point d'extrémité de bande passante supérieure suit le format de transaction indiqué dans la Figure 8-38.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Token	Jeton
Idle	Inactif
Data	Données
Error	Erreur
Idle	Inactif
Handshake	Etablissement de liaison
Data Error	Erreur de données
Host	Hôte
Function	Fonction

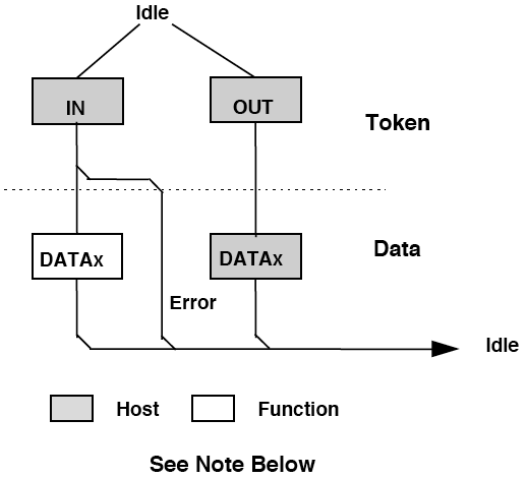
Figure 8-38. Format des transactions d'interruption

Quand un point d'extrémité utilise le mécanisme de transfert d'interruption pour des données d'interruption réelles, le protocole de basculement des données doit être suivi. Cela permet à la fonction de savoir que l'hôte a reçu les données et la condition d'évènement peut être effacée. Cette livraison "garantie" d'évènements permet à la fonction d'envoyer uniquement les informations d'interruption tant que l'hôte les reçoit plutôt que de devoir envoyer les données d'interruption à chaque fois que la fonction est interrogée et tant que le logiciel du système USB efface la condition d'interruption. Si un point d'extrémité d'interruption est utilisé en mode de basculement, tout évènement de configuration sur le point d'extrémité l'initialise au PID DATA0 et elle se comporte comme les transactions en masse indiquées sur la Figure 8-35.

8.5.5 Transactions isochrones

Les transactions isochrones ont une phase de jeton et de données, mais pas de phase d'établissement de liaison, comme représenté à la Figure 8-39. L'hôte émet un jeton IN ou OUT suivi par la phase de données dans laquelle le point d'extrémité (pour les IN) ou l'hôte (pour les OUT) transmet des données. Les transactions isochrones ne prennent pas en charge une phase d'établissement de liaison ou une fonctionnalité de relance.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Idle	Inactif
Token	Jeton
Data	Données
Error	Erreur
Idle	Inactif
Host	Hôte
Function	Fonction
See note Below	Voir note ci-dessous

Figure 8-39. Format des transactions isochrones

Note: Il convient qu'un appareil ou un contrôleur d'hôte à vitesse intégrale puisse accepter des PID DATA0 ou DATA1 dans les paquets de données. Il convient qu'un appareil ou un contrôleur d'hôte à vitesse intégrale envoie uniquement PID DATA0 dans les paquets de données. Un contrôleur d'hôte à haute vitesse doit pouvoir accepter et envoyer des PID DATA0, DATA1, DATA2 ou MDATA dans les paquets de données. Un appareil à haute vitesse avec au plus 1 transaction par microtrame doit envoyer uniquement des PID DATA0 dans les paquets de données. Un appareil à haute vitesse avec des points d'extrémité de bande passante supérieure (par ex.: un ayant plus de 1 transaction par microtrame) doit pouvoir accepter et/ou envoyer des PID DATA0, DATA1, DATA2 ou MDATA dans les paquets de données.

Les transactions isochrones à vitesse intégrale ne prennent pas en charge le séquencement de basculement. Les transactions isochrones à haute vitesse avec une seule transaction par microtrame ne prennent pas en charge le séquencement de basculement. Les transactions isochrones à haute vitesse de bande passante supérieure prennent en charge le séquencement des PID de données (voir la Section 5.9.1 pour plus de détails).

Les Figures 8-40 et 8-41 montrent les diagrammes d'états de l'hôte et de l'appareil respectivement pour les transactions OUT isochrones. Les Figures 8-42 et 8-43 montrent les diagrammes d'états de l'hôte et de l'appareil respectivement pour les transactions IN isochrones.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

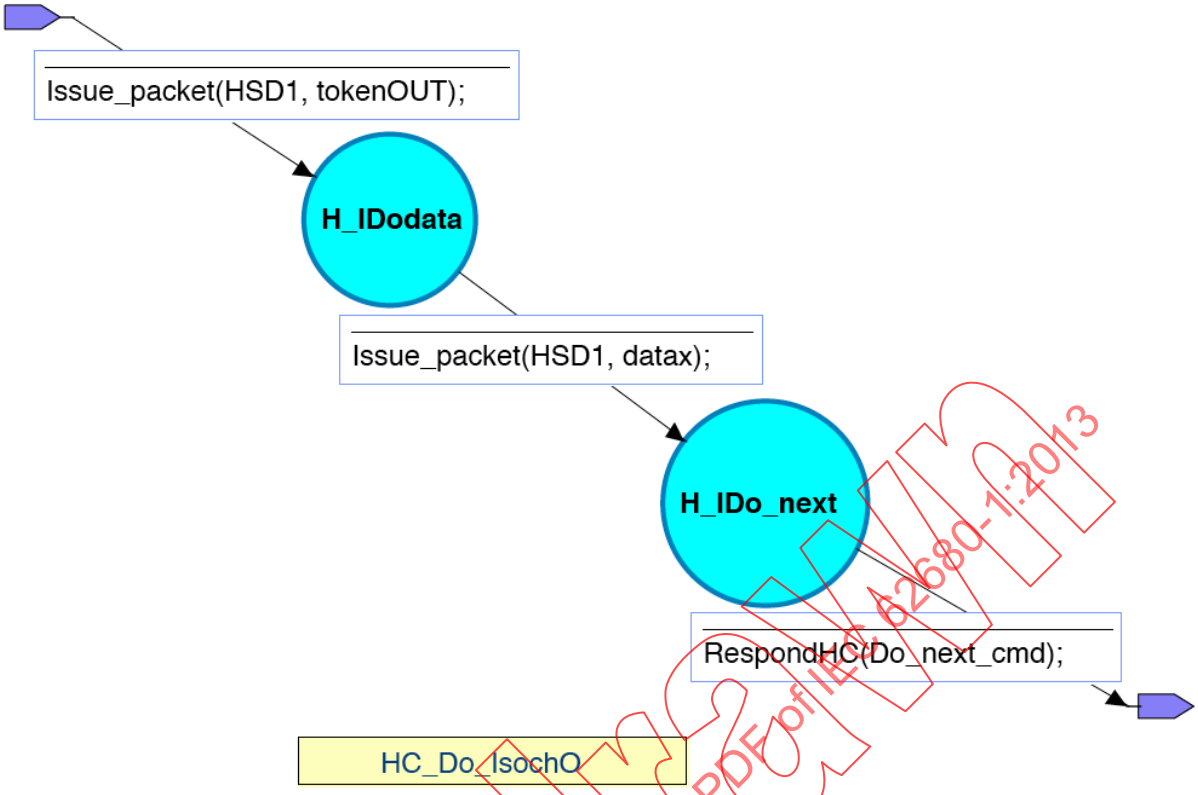
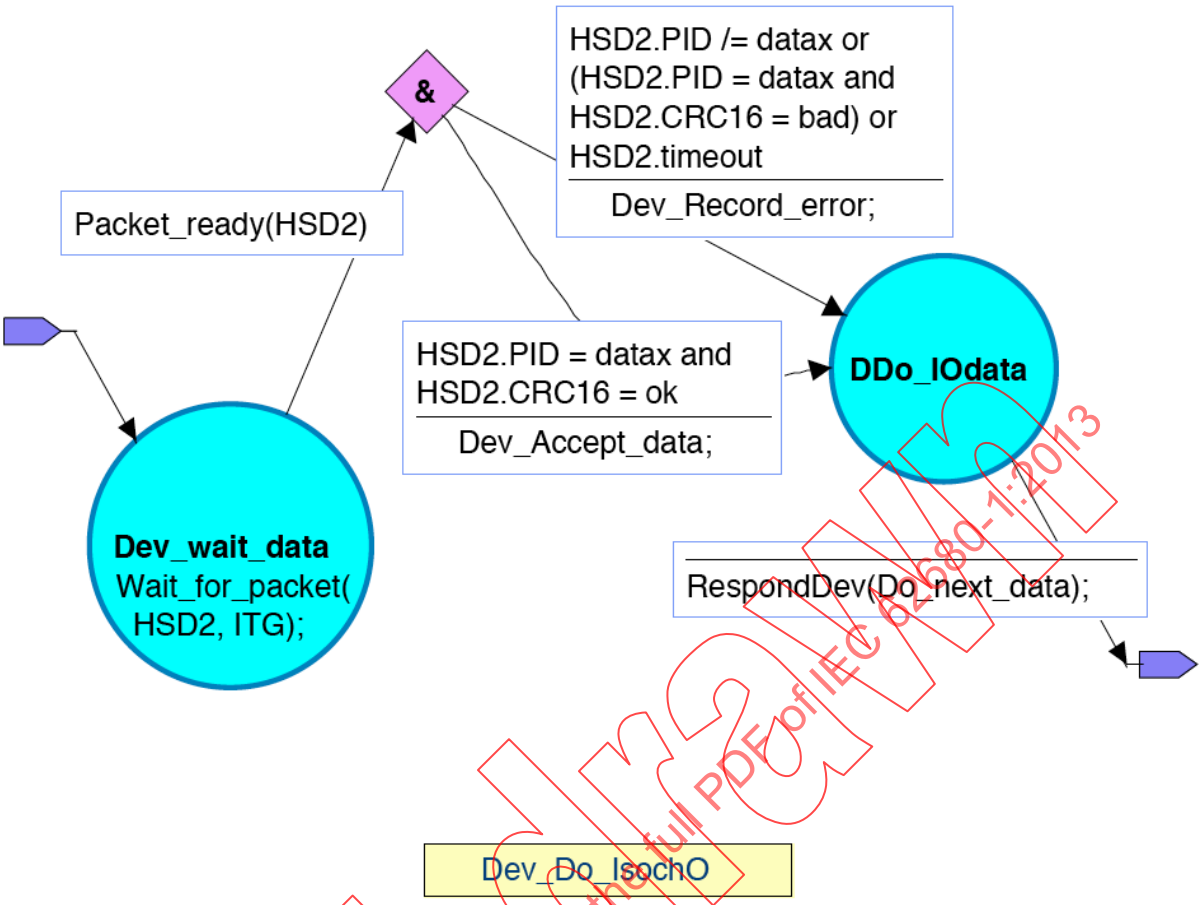


Figure 8-40. Diagrammes d'états de l'hôte pour les transactions OUT isochrones

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

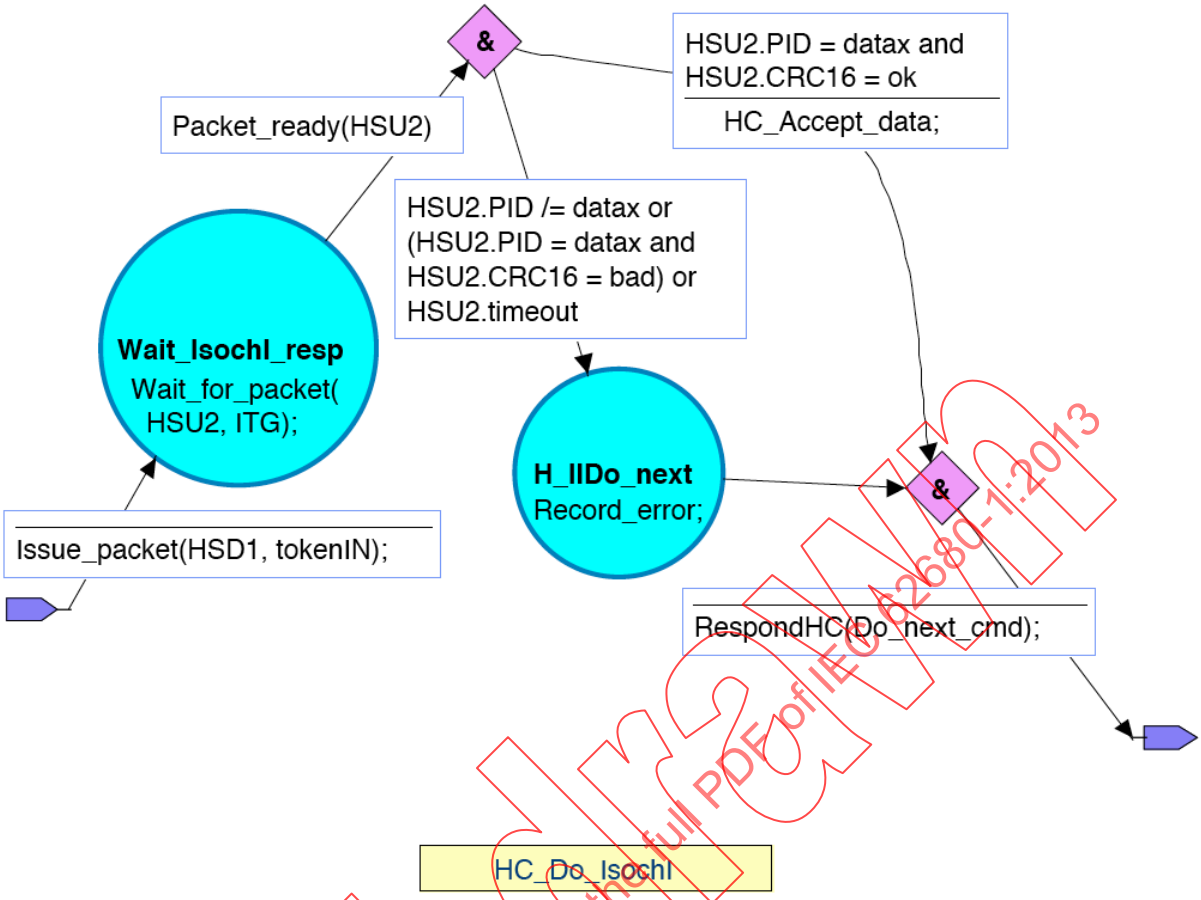


Légende

Anglais	Français
HSD2.PID /= datax or (HSD2.PID = datax and HSD2.CRC16 = bad) or HSD2.timeout	HSD2.PID /= datax ou (HSD2.PID = datax et HSD2.CRC16 = bad) ou HSD2.timeout
HSD2.PID = datax and HSD2.CRC16 = ok	HSD2.PID = datax et HSD2.CRC16 = ok

Figure 8-41. Diagrammes d'états de l'appareil pour les transactions OUT isochrones

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
HSU2.PID = datax and HSU2.CRC16 = ok	HSU2.PID = datax et HSU2.CRC16 = ok
HSU2.PID /= datax or HSU2.PID /= datax and HSU2.CRC16 = bad) or HSU2.timeout	HSU2.PID /= datax ou HSU2.PID /= datax et HSU2.CRC16 = bad) ou HSU2.timeout

Figure 8-42. Diagrammes d'états de l'hôte pour les transactions IN isochrones

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

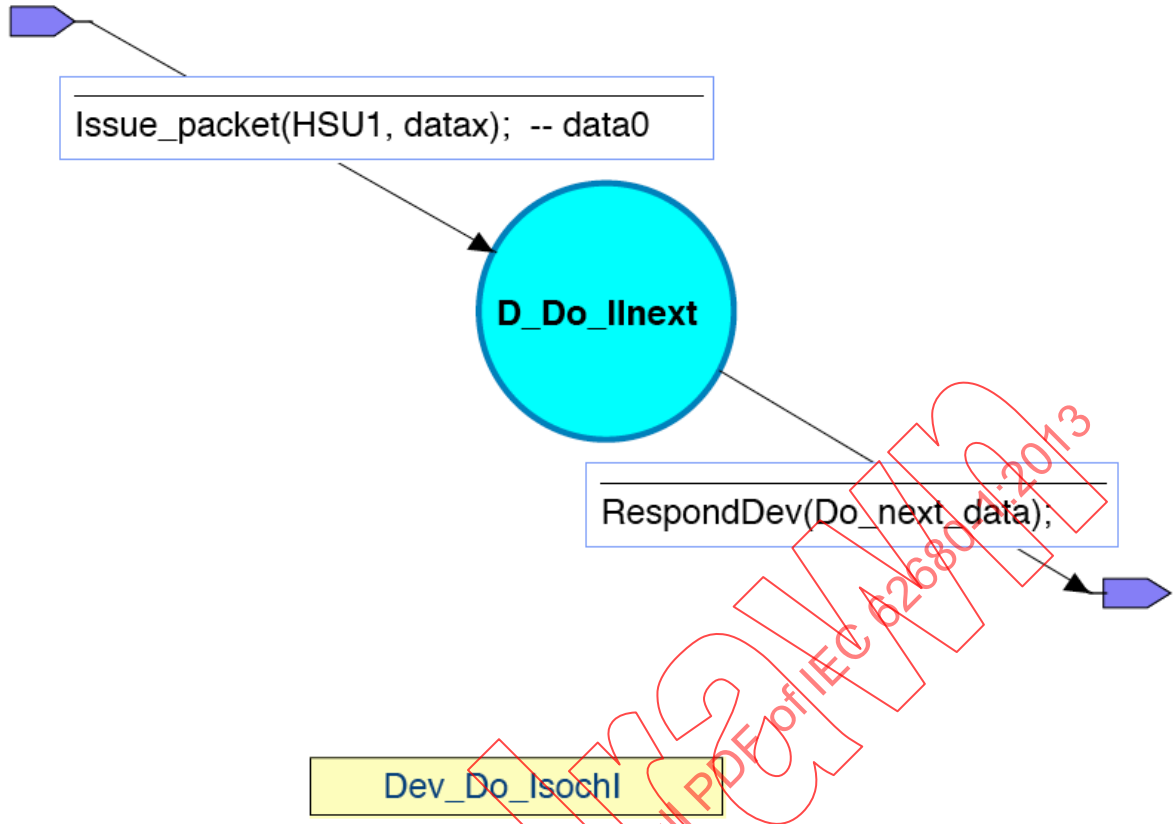


Figure 8-43. Diagrammes d'états de l'appareil pour les transactions IN isochrones

8.6 Synchronisation et relance de basculement des données

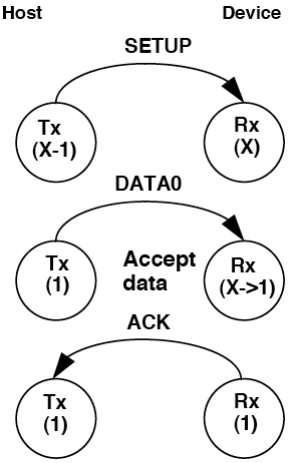
L'USB fournit un mécanisme pour garantir la synchronisation des séquences de données entre l'émetteur et le récepteur des données dans les différentes transactions. Ce mécanisme fournit un moyen de garantir que la phase d'établissement de liaison d'une transaction a été interprétée correctement par l'émetteur ainsi que le récepteur. La synchronisation s'obtient grâce à des PID DATA0 et DATA1 et des bits de séquence de basculement des données séparés pour l'émetteur et le récepteur des données. Les bits de séquence du récepteur basculent uniquement quand le récepteur peut accepter des données et reçoit un paquet de données sans erreur avec le bon PID de données. Les bits de séquence de l'émetteur basculent uniquement quand l'émetteur des données reçoit un établissement de liaison ACK valide. Les bits de séquence de l'émetteur et du récepteur des données doivent être synchronisés au début d'une transaction. Le mécanisme de synchronisation utilisé varie avec le type de transaction. La synchronisation de basculement des données n'est pas prise en charge pour les transferts isochrones.

Les diagrammes d'états contenus dans ce chapitre et le chapitre 11 décrivent la synchronisation de basculement des données sous une forme plus compacte. Au lieu d'identifier explicitement DATA0 et DATA1, il utilise une valeur "DATAx" pour représenter les PID DATA0 et/ou DATA1. Dans certains cas où le PID de données spécifiques est important, une autre variable marquée "x" ayant la valeur 0 pour DATA0 et 1 pour DATA1 est utilisée.

Les points d'extrémité d'interruption et isochrones de bande passante supérieure et à haute vitesse prennent en charge une technique de synchronisation des données similaire, mais différente appelée séquencement de PID des données. Cette technique est utilisée à la place de la synchronisation de basculement des données. La Section 5.9.1 définit le séquencement de PID des données.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

8.6.1 Initialisation par jeton SETUP



Légende

Anglais	Français
Host	Hôte
Device	Appareil
Accept Data	Accepter les données

Figure 8-44. Initialisation de SETUP

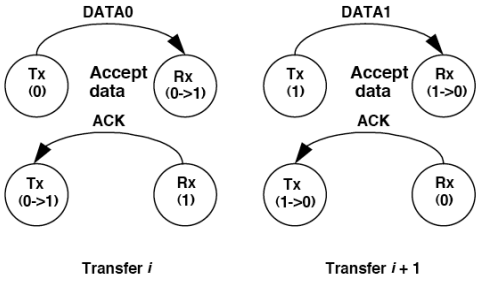
Les transferts de contrôle utilisent le jeton SETUP pour initialiser les bits de séquence de l'hôte et de la fonction. La Figure 8-44 montre l'hôte qui émet un paquet SETUP à une fonction suivi d'une transaction OUT. Les chiffres dans les cercles représentent les bits de séquence de l'émetteur et du récepteur. La fonction doit accepter les données et renvoyer un ACK. Quand la fonction accepte la transaction, elle doit régler son bit de séquence de manière à ce que les bits de séquence de l'hôte et de la fonction soient égaux à celui à la fin de la transaction SETUP.

8.6.2 Transactions de données réussies

La Figure 8-45 montre le cas de deux transactions réussies. Pour l'émetteur de données, cela signifie qu'il bascule son bit de séquence sur réception d'un ACK. Le récepteur bascule son bit de séquence uniquement s'il reçoit un paquet de données valide et si le PID de données du paquet correspond à la valeur actuelle de son bit de séquence. L'émetteur bascule son bit de séquence uniquement après qu'il a reçu un ACK d'un paquet de données.

Lors de chaque transaction, le récepteur compare le bit de séquence de l'émetteur (encodé dans le PID du paquet de données comme DATA0 ou DATA1) avec son bit de séquence du récepteur. Si les données ne peuvent pas être acceptées, le récepteur doit émettre un NAK et les bits de séquence de l'émetteur ainsi que du récepteur restent inchangés. Si les données peuvent être acceptées et si le bit de séquence du récepteur correspond au bit de séquence du PID, alors les données sont acceptées et le bit de séquence bascule. Les transactions à deux phases dans lesquelles il n'y a pas de paquet de données laissent inchangés les bits de séquence de l'émetteur et du récepteur.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



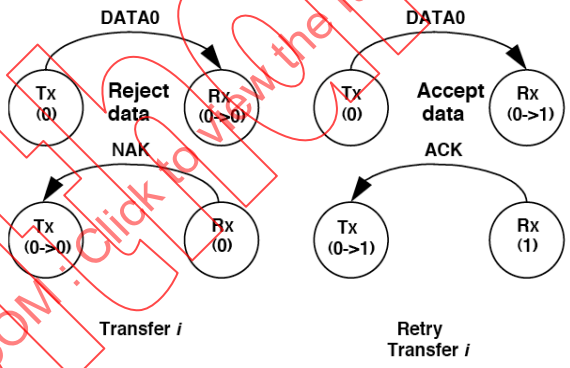
Légende

Anglais	Français
Accept Data	Accepter les données
Transfer *i*	Transférer *i*
Transfer *i* + 1	Transférer *i* + 1

Figure 8-45. Transactions consécutives

8.6.3 Données corrompues ou non acceptées

Si les données ne peuvent pas être acceptées ou si le paquet de données reçu est corrompu, alors le récepteur émet un établissement de liaison NAK ou STALL, ou bien une expiration de délai, selon les circonstances, et le récepteur ne bascule pas son bit de séquence. La Figure 8-46 montre le cas où une transaction est NAKée puis relancée. Tout établissement de liaison non ACK ou expiration de délai génère un comportement similaire de relance. L'émetteur n'ayant reçu d'établissement de liaison ACK, ne bascule pas son bit de séquence. Par conséquent, une transaction de paquet de données qui échoue laisse synchronisés et non basculés les bits de séquence de l'émetteur et du récepteur. La transaction est relancée, et si elle réussit elle fait basculer les bits de séquence de l'émetteur et du récepteur.



Légende

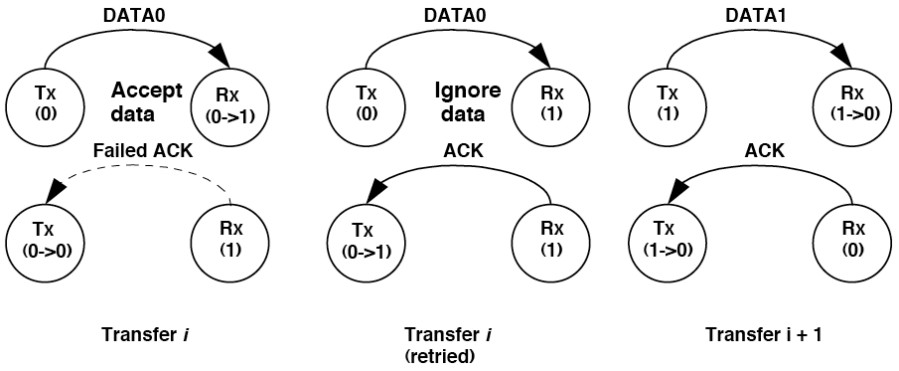
Anglais	Français
Reject Data	Rejeter les données
Accept data	Accepter les données
Transfer *i*	Transférer *i*
Retry Transfer *i*	Réessayer le transfert *i*

Figure 8-46. Transaction NAKée avec relance

8.6.4 Etablissement de liaison ACK corrompu

L'émetteur est le seul et le dernier agent à savoir avec certitude si une transaction est réussie, car il reçoit un établissement de liaison ACK. Un établissement de liaison ACK perdu ou corrompu peut mener à une perte temporaire de synchronisation entre l'émetteur et le récepteur comme représenté à la Figure 8-47. Ici l'émetteur émet un paquet de données valide, que le récepteur acquiert avec succès, mais l'établissement de liaison ACK est corrompu.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Accept Data	Accepter les données
Ignore Data	Ignorer les données
Failed ACK	ACK échoué
Transfer *i*	Transférer *i*
Transfer *i* (retried)	Transférer *i* (relancé)
Transfer *i* + 1	Transférer *i* + 1

Figure 8-47. Etablissement de liaison ACK corrompu avec relance

A la fin de la transaction *i*, il y a une perte temporaire de cohérence entre l'émetteur et le récepteur, comme le prouve l'inadéquation entre leurs bits de séquence respectifs. Le récepteur a reçu les bonnes données, mais l'émetteur ne sait pas s'il a réussi à envoyer les données. Sur la prochaine transaction, l'émetteur renvoie les données précédentes en utilisant le PID DATA0 précédent. Le bit de séquence du récepteur et le PID de données ne correspondent pas, donc le récepteur sait qu'il a préalablement accepté ces données. Par conséquent, il élimine le paquet de données entrant et ne bascule pas son bit de séquence. Le récepteur émet ensuite un ACK, qui fait que l'émetteur considère la transaction retentée comme réussie. La réception d'un ACK fait que l'émetteur bascule son bit de séquence. Au début de la transaction *i*+1, les bits de séquence ont basculé et sont resynchronisés.

L'émetteur des données doit garantir que tout paquet de données retenté est identique (longueur et contenu identiques) à celui envoyé dans la transaction initiale. Si l'émetteur des données ne peut pas émettre la quantité identique de données que celle du paquet de données initial, en raison de problèmes tels qu'une condition de perte de données de la mémoire tampon, alors il doit annuler la transaction en générant une violation d'insertion de bits pour la vitesse intégrale/basse vitesse. Une erreur pour la haute vitesse doit être forcée en prenant le CRC actuellement calculé et en le complétant avant de l'émettre. Cela cause une erreur détectable au niveau du récepteur et garantit qu'un paquet partiel n'est pas interprété comme un bon paquet. Il convient que l'émetteur n'essaie pas de forcer une erreur au niveau du récepteur en envoyant un mauvais CRC constant connu. Le récepteur peut interpréter un mauvais paquet combiné avec un "mauvais" CRC comme un bon paquet.

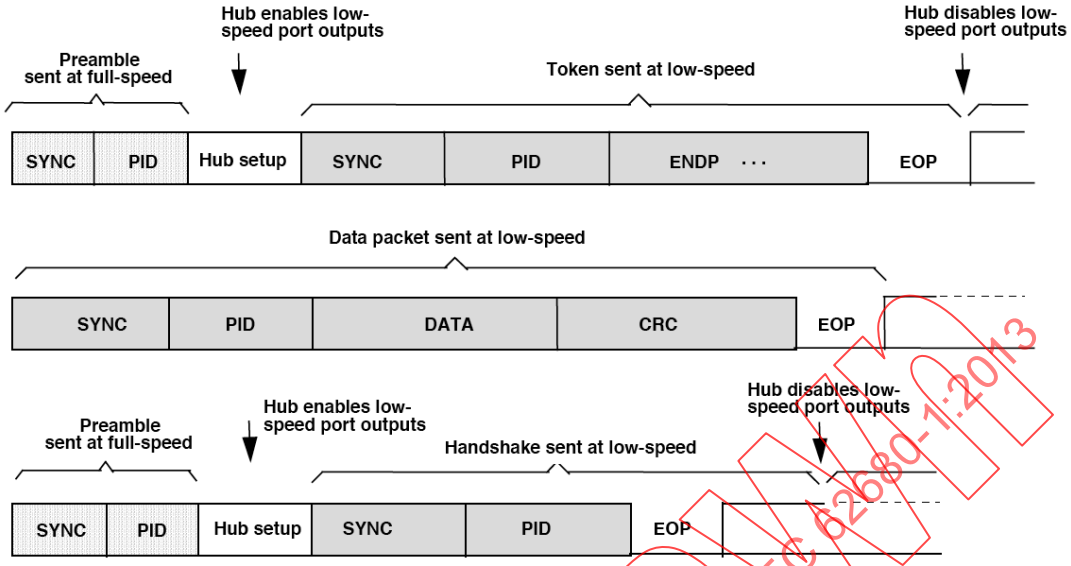
8.6.5 Transactions à basse vitesse

L'USB prend en charge des signaux à trois vitesses: des signaux à haute vitesse à 480 Mo/s, des signaux à vitesse intégrale à 12,0 Mo/s et des signaux à basse vitesse à 1,5 Mo/s. Les hubs isolent les signaux à haute vitesse des environnements des signaux à vitesse intégrale/basse vitesse.

Au sein d'un environnement de signaux à vitesse intégrale/basse vitesse, les hubs désactivent le trafic sur le bus en aval vers tous les ports où sont raccordés des appareils à basse vitesse pendant les signaux en aval à vitesse intégrale. Cela est nécessaire pour des raisons d'interférence électromagnétique (EMI) et pour prévenir la possibilité qu'un appareil à basse vitesse puisse mal interpréter un paquet à vitesse intégrale en aval quand il traité vers lui.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

La Figure 8-48 montre une transaction basse vitesse IN dans laquelle l'hôte (ou le TT) émet un jeton et un établissement de liaison, et reçoit un paquet de données.



Légende

Anglais	Français
Preamble sent at full-speed	Préambule envoyé à vitesse intégrale
Hub enables low-speed port outputs	Le hub active les sorties de port à basse vitesse
Token sent at low-speed	Jeton envoyé à basse vitesse
Hub disables low-speed port outputs	Le hub désactive les sorties de port à basse vitesse
Data packet sent at low-speed	Paquet de données envoyé à basse vitesse
Handshake sent at low-speed	Etablissement de liaison envoyé à basse vitesse
Hub setup	Configuration du hub

Figure 8-48. Transaction à basse vitesse

Tous les paquets en aval émis vers des appareils en aval dans un environnement de signaux à vitesse intégrale/basse vitesse nécessitent un préambule. Les préambules ne sont jamais utilisés dans un environnement de signaux à haute vitesse. Le préambule consiste en un SYNC suivi d'un PID PRE, tous deux envoyés à vitesse intégrale. Les hubs doivent comprendre le PID PRE, tous les autres appareils USB peuvent l'ignorer et le traiter comme non défini. A la fin du PID du préambule, l'hôte (ou TT) met le bus en état d'attente pendant au moins la durée d'un bit à vitesse intégrale. Cette période d'attente du bus s'appelle l'intervalle de configuration du hub et dure au moins quatre fois la durée d'un bit à vitesse intégrale. Pendant cet intervalle de configuration de hub, les hubs doivent mettre leurs ports à vitesse intégrale et basse vitesse dans leurs états d'attente respectifs. Les hubs doivent être prêts à répéter les signaux à basse vitesse sur les ports à basse vitesse avant la fin de l'intervalle de configuration du hub. Les règles de connectivité à basse vitesse sont résumées ci-dessous:

1. Les appareils à basse vitesse sont identifiés pendant le processus de connexion, et les ports du hub auxquels ils sont connectés sont identifiés comme étant à basse vitesse.
2. Tous les paquets à basse vitesse en aval doivent être précédés d'un préambule (envoyé à vitesse intégrale), qui active les mémoires tampons de sortie sur les ports du hub à basse vitesse.
3. Les mémoires tampons de sortie des ports du hub à basse vitesse se désactivent sur réception de l'EOP et ne se réactivent pas tant qu'un PID de préambule ne soit détecté.
4. La connectivité en amont ne dépend pas du fait qu'un port de hub soit à vitesse intégrale ou à basse vitesse.

Les signaux à basse vitesse commencent avec l'hôte (ou TT) qui émet un SYNC à basse vitesse, suivi du reste du paquet. La fin du paquet est identifiée par une fin de paquet (EOP),

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

auquel moment tous les hubs désactivent la connectivité et tous les ports auxquels sont connectés des appareils à basse vitesse. Les hubs ne changent pas les ports pour les signaux en amont, les ports à basse vitesse restent activés dans le sens vers l'amont pour les signaux à basse et à vitesse intégrale.

Des transactions à faible vitesse et à grande vitesse maintiennent un degré élevé de similitude. Cependant, la signalisation à faible vitesse a en effet certaines limites qui incluent:

- La charge utile de données est limitée à huit bits maximum.
- Seuls les types d'interruption et de commande de transferts sont supportés.
- Des appareils à faible vitesse ne reçoivent pas le paquet SOF.

8.7 Détection et récupération des erreurs

L'USB permet une communication de bout en bout fiable en présence d'erreurs sur la couche de signalisation physique. Cela inclut la possibilité de détecter de manière fiable la grande majorité d'erreurs éventuelles et de récupérer les erreurs sur une base de transaction. Les transactions de commande, par exemple, nécessitent un degré élevé de fiabilité des données; elles supportent une intégrité des données de bout en bout, en utilisant la détection d'erreurs et la relance. Les transactions isochrones, en vertu de leur largeur de bande et de leurs exigences de latence, ne permettent pas de relances et doivent tolérer une incidence plus élevée des erreurs non corrigées.

8.7.1 Catégories de taux d'erreur de paquets

L'USB utilise trois mécanismes de détection d'erreur: Violations remplissage bit, bits de contrôle PID et CRC. Les violations remplissage bit sont définies dans la Section 7.1.9. Les erreurs PID sont définies à la Section 8.3.1. Les erreurs CRC sont définies à la Section 8.3.5.

A l'exception du jeton SOF, tout paquet qui est reçu corrompu entraîne que le récepteur l'ignore et rejette toutes données ou d'autres informations de champ qui provenaient du paquet. Le Tableau 8-8 énumère les mécanismes de détection d'erreur, les types de paquets auxquels ils s'appliquent, et la réponse appropriée du récepteur du paquet.

Tableau 8-8. Types d'erreur de paquet

Champ	Erreur	Action
PID	Contrôle PID, Remplissage bit	Ignorer paquet
Adresse	Remplissage paquet, Adresse CRC	Ignorer jeton
Numéro cadre	Remplissage paquet, Numéro cadre CRC	Ignorer champ numéro cadre
Data	Remplissage paquet, Données CRC	Rejeter données

8.7.2 Temps d'exécution de bus

Ni l'appareil ni l'hôte n'envoient l'indication qu'un paquet reçu a subi une erreur. Cette absence de reconnaissance positive est considérée comme étant l'indication qu'il y a eu une erreur. Comme conséquence de cette méthode de rapport d'erreur, l'hôte et la fonction USB doivent garder une trace du temps écoulé à partir du moment où l'émetteur a fini d'envoyer un paquet jusqu'à ce qu'il ait commencé à recevoir un paquet en réponse. Ce temps est mentionné comme temps d'exécution du bus. Les appareils et les hôtes nécessitent des minuteurs d'exécution pour mesurer ce temps.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

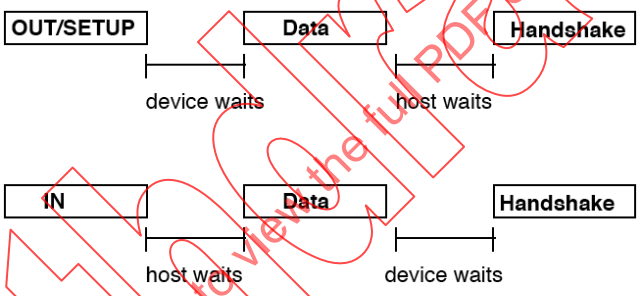
Pour les transactions à vitesse intégrale/faible vitesse, le minuteur commence à compter sur la transition SEO-vers-"J" du signal d'échantillonnage de Fin de paquet et s'arrête de compter à la détection de la transition Pause-vers-"K". Pour les transactions à grande vitesse, le minuteur commence à compter quand les lignes de données repassent au niveau du silencieux et s'arrête de compter quand les lignes de données quittent le niveau du silencieux.

Le temps d'exécution du bus de l'appareil est défini par le temps de propagation aller-retour du pire des cas auquel s'ajoute le temps de réponse maximum de l'appareil (voir Sections 7.1.18 et 7.1.19 pour les temps d'exécution spécifiques de bus). En cas de non-réception de réponse dans cette temporisation du pire des cas, l'émetteur considère alors que la transmission du paquet a échoué.

La temporisation est utilisée et interprétée comme un état d'erreur de transaction pour de nombreux types de transferts. Si l'hôte souhaite indiquer un état d'erreur pour une transaction par une temporisation, il doit attendre le temps d'exécution complet du bus avant d'émettre le jeton suivant, afin de s'assurer que tous les appareils en aval sont temporisés.

Tel qu'indiqué à la Figure 8-49, l'appareil utilise son minuteur d'exécution de bus entre les phases jeton et données ou données et établissement de liaison. L'hôte utilise son minuteur entre les phases données et établissement de liaison ou jeton et données.

Si l'hôte reçoit un paquet de données corrompues, il peut demander un temps d'attente supplémentaire avant d'envoyer le jeton suivant. Cet intervalle d'attente supplémentaire garantit que l'hôte gère correctement les fins de paquet incorrectes.



Légende

Anglais	Français
Data	Données
Handshake	Etablissement de liaison
device waits	appareil en attente
host waits	hôte en attente

Figure 8-49. Utilisation de minuteurs d'exécution de bus

8.7.3 Fin de paquet incorrecte

Les fins de paquet incorrectes doivent être gérées d'une manière qui garantisse que le paquet actuellement en cours de progression soit achevé avant que l'hôte ou tout autre appareil essaie de transmettre un nouveau paquet. Si un tel évènement se produisait, cela constituerait une collision de bus et aurait la capacité de corrompre jusqu'à deux transactions consécutives. La détection d'une fin de paquet incorrecte se fonde sur le fait qu'un paquet dans lequel une fin de paquet incorrecte a été insérée apparaît comme paquet tronqué avec un échec de CRC. (Les 16 derniers bits du paquet de données ont une très faible probabilité d'apparaître comme un CRC correct.)

L'hôte et les appareils gèrent différemment les situations de fins de paquet incorrectes. Quand un appareil reçoit un paquet de données corrompues, il n'émet pas de réponse et attend que l'hôte envoie le jeton suivant. Ce système garantit que l'appareil n'essaye pas de rétablir une liaison tandis que l'hôte peut toujours être en train de transmettre un paquet de données. Si une fin de paquet incorrecte s'est produite, le paquet de données de l'hôte se termine par la suite, et l'appareil peut détecter le jeton suivant. Si un appareil émet un paquet de données qui est corrompu par une fin de paquet incorrecte, l'hôte ignore le paquet et ne

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

procède pas à l'établissement de la liaison. L'appareil, attendant un établissement de liaison de la part de l'hôte, temporise la transaction.

Si l'hôte reçoit un paquet de données corrompu, à haute/faible vitesse, il suppose qu'une fin de paquet incorrecte peut s'être produite et attend une période de 16 bits pour voir s'il existe un trafic subséquent en amont. Si aucune transition de bus n'est détectée dans l'intervalle de 16 bits et que le bus reste en état de Pause, l'hôte peut émettre le jeton suivant.

Sinon, l'hôte attend que l'appareil ait fini d'envoyer le reste de son paquet à haute/faible vitesse. Attendre une période de 16 bits garantit deux conditions:

- La première condition est de s'assurer que l'appareil a terminé d'envoyer son paquet. Cela est garanti par un intervalle de temporisation (sans transitions de bus) supérieur au pire cas d'intervalle de remplissage de paquet de six bits.
- La seconde condition est que le minuteur d'exécution du bus de l'appareil de transmission doit absolument expirer.

Noter que l'intervalle de temporisation est sensible à la vitesse de transaction. Pour des transactions à vitesse intégrale, l'hôte doit attendre des périodes de bits à vitesse intégrale; pour des transactions à faible vitesse, il doit attendre des périodes de bits à faible vitesse.

Si l'hôte reçoit un paquet de données corrompu, à grande vitesse, il ignore toute donnée jusqu'à ce que les lignes de données repassent au niveau du silencieux avant d'émettre le jeton suivant. Pour des transactions à grande vitesse, il n'est pas nécessaire que l'hôte attende pendant une période de temps additionnelle (au-delà de la durée d'intervalle inter-transaction normale) après que les lignes de données soient repassées au niveau du silencieux.

Si l'hôte reçoit un paquet de données avec un CRC valide, il suppose que le paquet est complet et qu'il ne requiert aucune période de temps supplémentaire (au-delà de la durée d'intervalle inter-transaction normale) pour émettre le jeton suivant.

8.7.4 Récupération suite à un murmure et une perte d'activité

L'USB doit pouvoir détecter et récupérer de l'état qui le laisse attendre indéfiniment une fin de paquet à vitesse intégrale/faible vitesse ou qui laisse le bus autre que dans état de Pause à la fin d'une (micro)trame.

- La perte d'activité à vitesse intégrale/faible vitesse (LOA) se caractérise par un SOP suivi par un manque d'activité du bus (le bus reste entraîné vers un "J" ou un "K") et aucune fin de paquet à la fin d'un cadre.
- Le murmure confus à vitesse intégrale/faible vitesse se caractérise par un SOP suivi par la présence d'une activité de bus après la fin d'un cadre.
- Le murmure confus/LOA à grande vitesse se caractérise par les lignes de données se trouvant à un niveau non silencieux à la fin d'une microtrame.

Le LOA et le murmure confus ont la possibilité soit d'interbloquer le bus soit de retarder le commencement du (micro)cadre suivant. Aucun des deux états n'est acceptable, et il faut éviter que les deux ne se produisent. Comme le composant USB responsable de la commande de la connectivité, les hubs sont responsables de la détection et de la reprise du murmure confus/LOA. On évite que tous les appareils USB qui ne réussissent pas à achever leur transmission à la fin d'un (micro)cadre ne transmettent après la fin d'un (micro)cadre, le hub le plus proche désactivant le port auquel est attaché l'appareil en cause. Les détails du mécanisme de reprise du murmure confus/LOA du hub apparaissent à la Section 11.2.5.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9 Chapitre 9
Cadre d'appareils

Un appareil USB peut être séparé en trois couches:

- La couche inférieure est une interface de bus qui transmet et reçoit des paquets.
- La couche intermédiaire gère les données de routine entre l'interface de bus et divers points d'extrémité sur l'appareil. Un point d'extrémité est le consommateur ou fournisseur ultime de données. Il peut être considéré comme source ou récepteur des données.
- La couche supérieure est la fonctionnalité prévue par l'appareil de bus en série, par exemple une souris ou une interface ISDN.

Ce chapitre décrit les attributs et opérations courants de la couche intermédiaire d'un appareil USB. Ces attributs et opérations sont utilisés par les parties spécifiques à la fonction de l'appareil afin de communiquer par l'interface de bus et au final avec l'hôte.

9.1 Etats des appareils USB

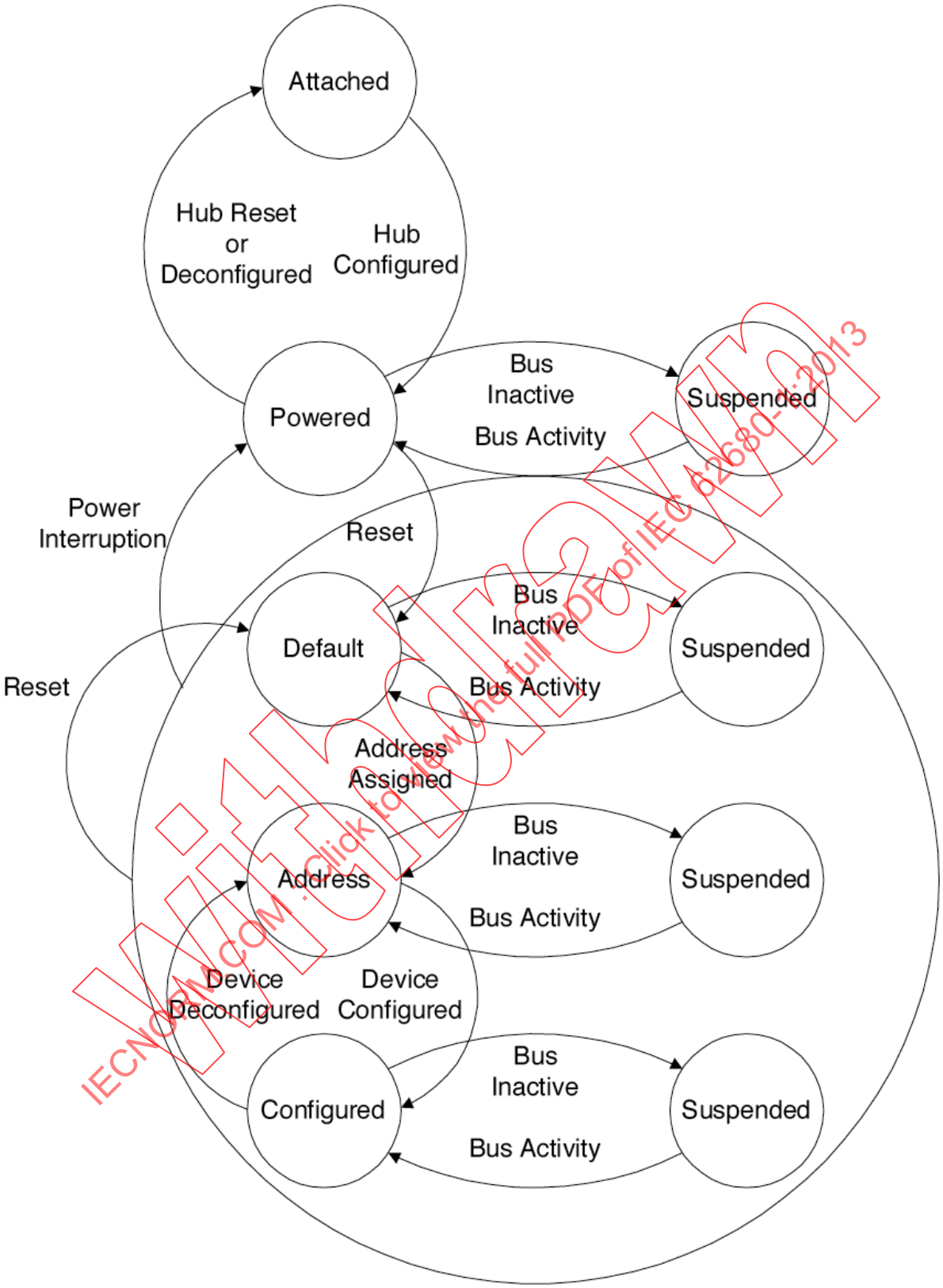
Un appareil USB a plusieurs états possibles. Certains de ces états sont visibles pour l'USB et l'hôte, tandis que d'autres sont internes à l'appareil USB. Cette section décrit ces états.

9.1.1 Etats des appareils visibles

Cette section décrit les états d'appareils USB qui sont visibles de l'extérieur (voir Figure 9-1). Le Tableau 9-1 résume les états d'appareils visibles.

Note: Les appareils USB effectuent une opération de réinitialisation en réponse au signal de réinitialisation sur le port faisant face en aval. Quand le signal de réinitialisation est terminé, l'appareil USB est réinitialisé.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Attached	Connecté
Hub Reset or Deconfigured	Hub réinitialisé ou déconfiguré
Hub Configured	Hub configuré
Powered	Alimenté

Anglais	Français
Bus Inactive	Bus inactif
Bus Activity	Activité de bus
Suspended	Suspendu
Power Interruption	Interruption de l'alimentation
Reset	Réinitialiser
Default	Défaut
Address Assigned	Adresse attribuée
Address	Adresse
Device Deconfigured	Appareil déconfiguré
Device Configured	Appareil configuré
Configured	Configuré

Figure 9-1. Diagramme d'états des appareils

Tableau 9-1. Etats des appareils visibles

Attaché	Alimenté	Par défaut	Adresse	Configuré	Suspendu	Etat
Non	--	--	--	--	--	L'appareil n'est pas attaché à l'USB. Les autres attributs ne sont pas significatifs.
Oui	Non	--	--	--	--	L'appareil est attaché à l'USB, mais pas alimenté. Les autres attributs ne sont pas significatifs.
Oui	Oui	Non	--	--	--	L'appareil est attaché à l'USB et alimenté, mais n'a pas été réinitialisé.
Oui	Oui	Oui	Non	--	--	L'appareil est attaché à l'USB et alimenté et n'a pas été réinitialisé, mais n'a pas été assigné à une adresse unique. L'appareil répond à l'adresse par défaut.
Oui	Oui	Oui	Oui	Non	--	L'appareil est attaché à l'USB, alimenté, a été réinitialisé, et une adresse unique d'appareil a été assignée. L'appareil n'est pas configuré.
Oui	Oui	Oui	Oui	Oui	Non	L'appareil est attaché à l'USB, alimenté, a été réinitialisé, a une adresse unique, est configuré et n'est pas suspendu. L'hôte peut maintenant utiliser la fonction prévue par l'appareil.
Oui	Oui	--	--	--	Oui	L'appareil est, au minimum, relié à l'USB et il est alimenté et n'a pas constaté d'activité du bus pendant 3 ms. Il peut également avoir une adresse unique et être configuré pour utilisation. Cependant, comme l'appareil est suspendu, l'hôte peut ne pas utiliser la fonction de l'appareil.

9.1.1.1 Attaché

Un appareil USB peut être attaché à ou détaché de l'USB. L'état d'un appareil USB quand il est détaché de l'USB n'est pas défini par ces spécifications. Ces spécifications n'abordent que les opérations et attributs requis une fois que l'appareil est attaché.

9.1.1.2 Alimenté

Les appareils USB peuvent acquérir une alimentation à partir d'une source externe et/ou à partir de l'USB par le hub auquel ils sont attachés. Les appareils USB à alimentation externe sont nommés "autoalimentés". Bien que les appareils autoalimentés puissent déjà être alimentés avant d'être attachés à l'USB, ils ne sont pas considérés comme étant en état Alimenté sauf s'ils sont attachés à l'USB et que VBUS est appliqué à ce service.

Un appareil peut supporter à la fois des configurations autoalimentées et alimentées par le bus. Certaines configurations d'appareil supportent l'une ou l'autre source d'alimentation. D'autres configurations d'appareil peuvent être disponibles uniquement si l'appareil est autoalimenté. Les appareils signalent leur fonctionnalité de source d'alimentation par le descripteur de configuration. La source de courant actuelle est signalée dans le cadre du

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

statut de l'appareil. Les appareils peuvent changer leur source d'alimentation à tout moment, par exemple en passant d'une autoalimentation à une alimentation par bus. Si une configuration peut supporter les deux modes d'alimentation, la puissance maximum signalée pour cette configuration est la puissance maximum que l'appareil retire du VBUS dans un mode ou l'autre. L'appareil doit observer ce maximum, sans tenir compte de son mode. Si une configuration supporte uniquement un mode d'alimentation et que la source d'alimentation de l'appareil change, l'appareil perd sa configuration et son adresse actuelles et repasse en état Alimenté. Si un appareil est autoalimenté et que sa configuration actuelle nécessite plus de 100 mA, l'appareil passe alors en mode alimenté par le bus et il doit repasser à l'état Adresse. Les hubs autoalimentés qui utilisent VBUS pour alimenter le Contrôleur de hub sont autorisés à rester dans l'état configuré dans le cas de perte de l'alimentation. Voir la Section 11.13 pour plus de détails.

Un port hub doit être alimenté afin de détecter les changements de statut du port, incluant l'attachement et le détachement. Les hubs alimentés par bus, ne fournissent pas d'alimentation an aval sauf s'ils sont configurés et à ce point fournissent une alimentation tel qu'autorisé par leur configuration et source d'alimentation. Un appareil USB doit être en mesure d'être traité dans un délai spécifié à partir du moment où l'alimentation est appliquée initialement (voir le Chapitre 7). Une fois qu'une connexion à un port a été détectée, l'hôte peut activer le port, qui réinitialise également l'appareil connecté au port.

9.1.1.3 Défaut

Une fois l'appareil alimenté, il ne doit répondre à aucune transaction de bus jusqu'à ce qu'il ait reçu une réinitialisation du bus. Après réception d'une réinitialisation, l'appareil est ensuite adressable à l'adresse par défaut.

Quand le processus de réinitialisation est terminé, l'appareil USB fonctionne à la vitesse correcte (c'est-à-dire basse/vitesse intégrale/grande vitesse). La sélection de vitesse pour la basse vitesse et la vitesse intégrale est déterminée par les résistances du raccordement de l'appareil. Un appareil qui peut réaliser une opération à grande vitesse détermine s'il fonctionne à grande vitesse dans le cadre du processus de réinitialisation (voir Chapitre 7 pour plus de détails).

Un appareil qui peut réaliser une opération à grande vitesse doit se réinitialiser avec succès à vitesse intégrale quand il se trouve dans un environnement électrique qui fonctionne à peine vitesse. Après la réinitialisation correcte de l'appareil, l'appareil doit également répondre avec succès aux requêtes du descripteur de l'appareil et de la configuration et retourner les informations appropriées. L'appareil peut ou non être capable de prendre en charge sa fonctionnalité prévue quand il fonctionne à vitesse intégrale.

9.1.1.4 Adresse

Tous les appareils USB utilisent l'adresse par défaut quand ils sont alimentés au départ ou après que l'appareil a été réinitialisé. L'hôte assigne une adresse unique à chaque appareil USB après attachement ou réinitialisation. Un appareil USB conserve son adresse assignée tout en étant suspendu.

Un appareil USB répond aux requêtes sur son canal par défaut selon que l'appareil est actuellement assigné à une unique adresse ou s'il utilise l'adresse par défaut.

9.1.1.5 Configuré

Avant qu'une fonction d'appareil USB puisse être utilisée, l'appareil doit être configuré. A partir de la perspective de l'appareil, une configuration implique un traitement correct d'une requête de SetConfiguration() avec une valeur de configuration de non-zéro. La configuration d'un appareil ou la modification d'un autre paramètre fait que les valeurs d'état et de configuration associées aux points d'extrémité dans les interfaces affectées doivent être définies à leurs valeurs par défaut. Cela inclut le réglage de la bascule des données à tout point d'extrémité utilisant des bascules de données vers les data0 de valeur.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9.1.1.6 Suspendu

Afin de maintenir la puissance, les appareils USB entrent automatiquement en état Suspendu lorsque l'appareil n'a observé aucun trafic de bus pendant une période spécifiée (voir le Chapitre 7). Lorsqu'il est suspendu, l'appareil USB maintient tout statut interne, y compris son adresse et sa configuration.

Tous les appareils doivent être suspendus si l'activité du bus n'a pas été constatée pendant la durée spécifiée au Chapitre 7. Les appareils connectés doivent être prêts à passer en mode suspendu dès qu'ils sont alimentés, qu'on leur ait attribué ou non une adresse sans défaut ou qu'ils aient été ou non configurés. L'activité du bus peut cesse du fait de l'hôte entrant dans son propre mode suspendu. En outre, un appareil USB doit également entrer en état Suspendu quand le port du hub auquel il est attaché est désactivé. Cela est mentionné comme une suspension sélective.

Un appareil USB quitte le mode suspendu quand il y a une activité de bus. Un appareil USB peut également demander à l'hôte de quitter le mode suspendu ou la suspension sélective en utilisant un signal électrique pour indiquer la restauration à distance. La capacité d'un appareil à signaler une restauration à distance est optionnelle. Si un appareil USB peut effectuer un signal de restauration à distance, l'appareil doit supporter la capacité de l'hôte à activer et à désactiver cette fonctionnalité. Quand l'appareil est réinitialisé, le signal de restauration à distance doit être désactivé.

9.1.2 Enumération de bus

Quand un appareil USB est attaché à ou enlevé de l'USB, l'hôte utilise un processus connu comme énumération de bus pour identifier et gérer les modifications nécessaires de l'état de l'appareil. Quand un appareil USB est attaché à un port alimenté, les actions suivantes sont entreprises:

1. Le hub auquel l'appareil USB est maintenant attaché informe l'hôte de l'évènement par une réponse sur son canal de changement d'état (voir la Section 11.12.3 pour plus d'informations). A ce point, l'appareil USB se trouve en état Alimenté et le port auquel il est attaché est désactivé.
2. L'hôte détermine la nature exacte du changement en demandant au hub.
3. Maintenant que l'hôte connaît le port auquel le nouvel appareil a été attaché, l'hôte attend alors pendant au moins 100 ms pour permettre l'achèvement du processus d'insertion et pour que l'alimentation de l'appareil devienne stable. L'hôte émet alors une activation de port et réinitialise la commande vers ce port. Voir la Section 7.1.7.5 pour la séquence des évènements et les minutages de connexion par la réinitialisation de l'appareil.
4. Le hub effectue le traitement de réinitialisation requis pour ce port (voir Section 11.5.1.5). Lorsque le signal de réinitialisation est libéré, le port a alors été activé. L'appareil USB est maintenant en état par Défaut et ne peut retirer plus de 100 mA à partir du VBUS. Tous ses registres et états ont été réinitialisés et il répond à l'adresse par défaut.
5. L'hôte assigne une adresse unique à l'appareil USB, en déplaçant l'appareil vers l'état d'adresse.
6. Avant que l'appareil USB ne reçoive une adresse unique, son canal de commande par défaut est toujours accessible par l'adresse par défaut. L'hôte lit le descripteur de l'appareil pour déterminer quelle taille de charge utile de données maximum actuelles ce canal par défaut de l'appareil USB peut utiliser.
7. L'hôte lit les informations de configuration à partir de l'appareil en lisant chaque configuration zéro vers *n*-1, où *n* est le nombre de configurations. Ce processus peut prendre plusieurs millisecondes pour se terminer.
8. Sur la base des informations de configuration et sur la manière dont l'appareil USB est utilisé, l'hôte assigne une valeur de configuration à l'appareil. L'appareil est maintenant en état Configuré et tous les points d'extrémité de cette configuration ont

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

pris en charge leurs caractéristiques décrites. L'appareil USB peut maintenant tirer le volume d'alimentation VBUS décrit dans son descripteur pour la configuration sélectionnée. A partir du point de vue de l'appareil, il est maintenant prêt à être utilisé.

Quand l'appareil USB est enlevé, le hub envoie à nouveau une notification à l'hôte. Détacher un appareil désactive le port auquel il avait été attaché. A réception de la notification de détachement, l'hôte met à jour ses informations topologiques locales.

9.2 Opérations génériques de appareils USB

Tous les appareils USB supportent un ensemble commun d'opérations. Cette section décrit ces opérations.

9.2.1 Connexion et déconnexion dynamiques

Les appareils USB peuvent être attachés et enlevés à tout moment. Le hub qui assure le point ou port d'attachement est responsable du rapport de tout changement dans l'état du port.

L'hôte active le port hub où l'appareil est attaché, lors de la détection d'un attachement, qui a également pour effet de réinitialiser l'appareil. Un appareil USB réinitialisé a les caractéristiques suivantes:

- Répond à l'adresse USB par défaut
- N'est pas configuré
- N'est pas initialement suspendu

Quand un appareil est enlevé d'un port hub, le hub désactive le port où l'appareil a été attaché et notifie l'hôte de l'enlèvement.

9.2.2 Attribution d'adresses

Quand un appareil USB est attaché, l'hôte est responsable de l'attribution d'une adresse unique à l'appareil. Cela est effectué après que l'appareil ait été réinitialisé par l'hôte, et que le port hub où l'appareil est attaché a été activé.

9.2.3 Configuration

Un appareil USB doit être configuré avant que sa(ses) fonction(s) puisse(nt) être utilisée(s). L'hôte est responsable de la configuration d'un appareil USB. L'hôte requête typiquement des informations de configuration à partir de l'appareil USB pour déterminer les fonctionnalités de l'appareil.

Dans le cadre du processus de configuration, l'hôte définit la configuration de l'appareil et, si nécessaire sélectionne les autres paramètres appropriés pour les interfaces.

Dans une configuration simple, un appareil peut supporter plusieurs interfaces. Une interface est un ensemble associé de points d'extrémité qui présentent une caractéristique ou fonction de l'appareil vers l'hôte. Le protocole utilisé pour communiquer avec cet ensemble associé de points d'extrémité et le but de chaque point d'extrémité dans l'interface peut être spécifié dans le cadre d'une classe d'appareils ou d'une définition spécifique au fournisseur.

En outre, une interface dans une configuration peut avoir d'autres paramètres qui redéfinissent le nombre de caractéristiques des points d'extrémité associés. Si c'est le cas, l'appareil doit supporter la requête GetInterface() pour signaler l'autre paramètre actuel pour l'interface spécifiée et la requête SetInterface() pour sélectionner l'autre paramètre pour l'interface spécifiée.

Dans chaque configuration, chaque descripteur d'interface contient des champs qui identifient le numéro d'interface et l'autre paramètre. Les interfaces sont numérotées de zéro à un moins le nombre d'interfaces concurrentes supportées par la configuration. D'autres paramètres allant de zéro à un moins le nombre d'autres paramètres pour une interface spécifique. Le paramètre par défaut quand un appareil est configuré au départ est un autre paramètre zéro.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

En support aux pilotes d'appareils d'adaptation qui peuvent gérer un groupe associé d'appareils USB, les descripteurs de l'appareil et de l'interface contiennent les champs *Class*, *SubClass*, et *Protocol*. Ces champs sont utilisés pour identifier la(les) fonction(s) fournies par un appareil USB et les protocoles utilisés pour communiquer avec la(les) fonction(s) sur l'appareil. Un code de classe est assigné à un groupe d'appareils associés qui a été identifié comme une partie d'une Spécification de classe USB. Une classe d'appareils peut en outre être sous-divisée en sous-classes et, dans une classe ou sous-classe, un code de protocole peut définir la manière dont le Logiciel Hôte communique avec l'appareil.

Note: L'attribution de classe, sous-classe et codes de protocole doit être coordonnée, mais se situe au-delà du domaine d'application de cette spécification.

9.2.4 Transfert de données

Les données peuvent être transférées entre un point d'extrémité d'un appareil USB et l'hôte dans une des quatre manières. Voir Chapitre 5 pour la définition des quatre types de transferts. Un certain nombre de points d'extrémité peut être utilisé pour différents types de transferts de données avec différents autres paramètres. Cependant, une fois qu'un autre paramètre est sélectionné (incluant le paramètre par défaut d'une interface), le point d'extrémité d'un appareil USB utilise uniquement une méthode de transfert de données jusqu'à ce qu'un autre paramètre différent soit sélectionné.

9.2.5 Gestion de l'énergie

La gestion de l'alimentation sur les appareils USB implique les problèmes décrits dans les sections suivantes.

9.2.5.1 Budgétisation alimentation

L'alimentation du bus USB est une ressource limitée. Au cours de l'énumération de l'appareil, un hôte évalue les exigences d'alimentation de l'appareil. Si les exigences d'alimentation d'une configuration spécifique dépassent la puissance disponible pour l'appareil, le Logiciel Hôte ne doit pas sélectionner la configuration.

Les appareils USB doivent limiter la puissance qu'ils consomment à partir du VBUS vers une charge unitaire ou moins jusqu'à ce qu'ils soient configurés. Les appareils suspendus, qu'ils soient ou non configurés, doivent limiter leur consommation électrique de bus comme le définit le Chapitre 7. Selon les fonctionnalités d'alimentation du port auquel est connecté l'appareil, un appareil USB peut être capable d'extraire jusqu'à cinq charges unitaires du VBUS après la configuration.

9.2.5.2 Restauration à distance

La restauration à distance permet à un appareil USB suspendu de signaler un hôte qui peut également être suspendu. Elle informe l'hôte qu'il convient qu'il reprenne à partir de son mode suspendu si nécessaire, et fournisse l'évènement externe qui a fait que l'appareil USB suspendu a signalé l'hôte. Un appareil USB signale sa capacité à supporter la restauration à distance dans un descripteur de configuration. Si un appareil supporte une restauration à distance, il doit également permettre la fonctionnalité d'être activé et désactivé en utilisant les requêtes de l'USB normalisées.

La restauration à distance est accomplie en utilisant un signal électrique décrit à la Section 7.1.7.7.

9.2.6 Traitement des requêtes

A l'exception des requêtes SetAddress() (voir Section 9.4.6), un appareil peut commencer à traiter une requête dès que l'appareil renvoie l'ACK suivant la Configuration. Il est prévu que l'appareil "achève" le traitement de la requête avant qu'il ne permette à la phase d'Etat de s'achever avec succès. Certaines requêtes débutent des opérations qui prennent plusieurs millisecondes pour être terminées. Pour des requêtes comme celle-ci, la classe d'appareil est requise pour définir une méthode autre que l'achèvement de phase d'Etat afin d'indiquer que l'opération a été achevée. Par exemple, une réinitialisation sur un port hub prend au moins

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10 ms pour s'achever. Le SetPortFeature(PORT_RESET) (voir Chapitre 11) requête des "achèvements" quand la réinitialisation sur le port est commencée. L'achèvement de l'opération de réinitialisation est signalé que le changement de l'état du port est défini pour indiquer que le port est maintenant activé. Cette technique évite à l'hôte de devoir demander constamment un achèvement quand on sait que la requête nécessite une assez longue période de temps.

9.2.6.1 Temps de traitement de la requête

Il est prévu que tous les appareils gèrent les requêtes dans les délais. L'USB définit une limite supérieure de 5 secondes comme limite supérieure pour toute commande à traiter. Cette limite n'est pas applicable dans tous les cas. Les limites sont décrites dans les sections suivantes. Il convient de noter que les limites données ci-dessous sont prévues d'englober une vaste plage de mises en application. Si tous les appareils d'un système USB ont utilisé le temps alloué maximum pour le traitement de la requête, l'expérience de l'utilisateur en souffre. Pour cette raison, il convient que les mises en application s'efforcent de terminer les requêtes en temps voulu, dans un délai aussi court que possible.

9.2.6.2 Temps de récupération de la Réinitialisation /Reprise

Après la réinitialisation ou la reprise d'un port, il est attendu que le Logiciel du système de l'USB fournisse un intervalle de "récupération" de 10 ms avant que l'appareil attaché au port ne réponde aux transferts de données. L'appareil peut ignorer tout transfert de données au cours de l'intervalle de récupération.

Après la fin de l'intervalle de récupération (mesuré à partir de la fin de la réinitialisation ou la fin d'une Fin de paquet à la fin du signal de reprise), l'appareil doit accepter les transferts de données à tout moment.

9.2.6.3 Traitement réglage d'adresse

Après l'intervalle de récupération de la réinitialisation/reprise, si un appareil reçoit une requête de SetAddress(), l'appareil doit pouvoir achever le traitement de la requête et doit pouvoir achever avec succès la phase d'Etat de la requête dans les 50 ms. Dans le cas d'une requête de SetAddress(), la phase d'Etat est achevée avec succès quand l'appareil envoie le paquet d'Etat de longueur zéro ou quand l'appareil voit l'ACK en réponse au paquet de données de la phase d'Etat.

Après l'achèvement réussi de la phase d'Etat, on permet à l'appareil un intervalle de récupération SetAddress() de 2 ms. A la fin de cet intervalle, l'appareil doit pouvoir accepter les paquets Setup adressés à la nouvelle adresse. De même, à la fin de l'intervalle de récupération, l'appareil ne doit pas répondre aux jetons envoyés à l'ancienne adresse (sauf si, bien sûr, l'ancienne et la nouvelle adresse sont les mêmes).

9.2.6.4 Requêtes normalisées de l'appareil

Pour les requêtes normalisées d'appareil qui ne nécessitent pas de phase de Données, un appareil doit pouvoir terminer la requête et pouvoir terminer avec succès la phase d'Etat de la requête dans les 50 ms de la requête. Cette limite s'applique aux requêtes vers l'appareil, l'interface ou le point d'extrémité.

Pour les requêtes normalisées d'appareil qui nécessitent un transfert de phase de données vers l'hôte, l'appareil doit pouvoir retourner le premier paquet de données vers l'hôte dans les 500 ms de la réception de la requête. Pour des paquets de données subséquents, si c'est le cas, l'appareil doit pouvoir les retourner dans les 500 ms de l'achèvement réussi de la transmission du paquet précédent. L'appareil doit ensuite pouvoir achever avec succès la phase d'état dans les 50 ms après le retour du dernier paquet de données.

Pour les requêtes normalisées d'appareil qui nécessitent un transfert de phase de données vers l'appareil, la limite de 5 secondes s'applique. Cela signifie que l'appareil doit être capable d'accepter tous les paquets de données provenant de l'hôte et achever avec succès la phase d'Etat si l'hôte fournit les données au débit maximum que l'appareil peut accepter.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Les temps entre les paquets introduits par l'hôte s'ajoutent au temps autorisé pour que l'appareil termine la requête.

9.2.6.5 Requêtes spécifiques à la classe

Sauf spécification contraire dans le document de classe, toutes les requêtes spécifiques de classe doivent satisfaire aux limites de temps pour des requêtes normalisées d'appareil. Si un document de classe fournit une exemption, l'exemption ne peut être spécifiée que sur une base de requête par requête.

Un document de classe peut demander qu'un appareil réponde plus rapidement que ce qui est spécifié dans cette section. Une réponse plus rapide peut être requise pour des requêtes normalisées et spécifiques à une classe.

9.2.6.6 Descripteurs dépendants de la vitesse

Un appareil capable d'opération à grande vitesse peut fonctionner soit à vitesse intégrale soit à grande vitesse. L'appareil connaît toujours sa vitesse d'opération puisqu'il doit gérer correctement ses émetteurs-récepteurs dans le cadre du traitement de la réinitialisation (voir Chapitre 7 pour plus de détails sur la réinitialisation). Un appareil fonctionne également à une vitesse unique après avoir achevé la séquence de réinitialisation. En particulier, il n'y a pas de commutation de vitesse au cours d'une opération normale. Cependant, un appareil capable de grande vitesse peut avoir des configurations qui dépendent de la vitesse. C'est-à-dire qu'il peut avoir certaines configurations qui ne sont possibles qu'en fonctionnant à grande vitesse ou certaines qui ne sont possibles qu'une fonctionnant à vitesse intégrale. Des appareils capables de grande vitesse doivent être en mesure de signaler leurs configurations dépendantes de la vitesse.

Un appareil capable de grande vitesse répond avec des informations du descripteur qui sont valides pour la vitesse de fonctionnement actuelle. Par exemple, quand des descripteurs de configuration sont demandés à un appareil, il ne retourne que ceux de la vitesse de fonctionnement actuelle (par exemple, vitesse intégrale). Cependant, il doit exister une manière de déterminer les fonctionnalités pour une opération à grande et à vitesse intégrale.

Deux descripteurs permettent à un appareil capable de grande vitesse de signaler des informations de configuration sur l'autre vitesse de fonctionnement. Il s'agit du descripteur (other_speed) device_qualifier et du descripteur other_speed_configuration, qui sont récupérés par l'hôte via la requête getDesciptor avec les valeurs de type de descripteur correspondantes.

Note: Ces descripteurs ne sont pas récupérés sauf si l'hôte émet explicitement les requêtes GetDescriptor correspondantes. Si ces deux requêtes ne sont pas émises, l'appareil apparaît simplement comme étant un appareil à simple vitesse.

Les appareils qui sont capables de grande vitesse doivent régler le numéro de version dans le champ *bcdUSB* de leurs descripteurs sur 0200H. Cela indique que ces appareils prennent en charge les requêtes other_speed définies par USB 2.0. Il convient qu'un appareil dont les numéros de version de descripteur sont inférieurs à 0200H entraîne une réponse d'erreur de requête (voir section suivante) s'il reçoit ces requêtes other_speed. Il convient qu'un appareil USB 1.x (c'est-à-dire un appareil avec une version de descripteur d'appareil inférieure à 0200H) n'émette pas les requêtes other_speed.

9.2.7 Erreur de requête

Quand une requête est reçue par un appareil qui n'est pas défini pour l'appareil, est inapproprié pour les paramètres actuels de l'appareil, ou a des valeurs qui ne sont pas compatibles avec la requête, alors une Erreur de requête existe. L'appareil traite l'erreur de requête en renvoyant un STALL PID en réponse à la transaction suivante de phase de données ou dans la phase d'Etat du message. Il est préférable que le STALL PID soit renvoyé à la prochaine transaction d'étape de données, puisque cela évite une activité de bus inutile.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

9.3 Requêtes de appareils USB

Tous les appareils USB répondent aux demandes provenant de l'hôte sur le Canal de commande par défaut de l'appareil. Ces requêtes sont réalisées en utilisant des transferts de commande. La requête et les paramètres de la requête sont envoyés à l'appareil dans le paquet Setup. L'hôte est chargé d'établir les valeurs transmises dans les champs énoncés au Tableau 9-2. Chaque paquet Setup comporte huit octets.

Tableau 9-2. Format des données Configuration

Décalage	Champ	Taille	Valeur	Description
0	*bmRequestType*	1	Bitmap	Caractéristiques de la requête: D7: Direction du transfert de données 0 = Host-to-device 1 = Device-to-host D6…5: Type 0 = Normalisé 1 = Classe 2 = Fournisseur 3 = Réservé D4…0: Destinataire 0 = Appareil 1 = Interface 2 = Point d'extrémité 3 = Autre 4…31 = Réservé
1	*bRequest*	1	Valeur	Requête spécifique (voir Tableau 9-3)
2	*wValue*	2	Valeur	Champ de longueur de mot qui varie selon la requête
4	*wIndex*	2	Indice ou décalage	Champ de longueur de mot qui varie selon la requête; utilisé typiquement pour passer un indice ou un décalage
6	*wLength*	2	Count	Nombre d'octets à transférer s'il y a une phase de données

9.3.1 Type bmRequest

Ce champ bitmappé identifie les caractéristiques de la requête spécifique. En particulier, ce champ identifie la direction du transfert de données dans la seconde phase du transfert de commande. L'état du bit *Direction* est ignoré si le champ *wLength* est zéro, signifiant qu'il n'y a pas de phase de données.

La spécification USB définit une série de requêtes normalisées que tous les appareils doivent supporter. Ces requêtes sont énumérées au Tableau 9-3. En outre, une classe d'appareil peut définir des requêtes supplémentaires. Un fournisseur d'appareil peut également définir des requêtes supportées par l'appareil.

Les requêtes peuvent être dirigées vers l'appareil, une interface sur l'appareil ou un point d'extrémité spécifique sur un appareil. Ce champ spécifie également le destinataire prévu de la requête. Lorsqu'une interface ou un point d'extrémité est spécifiée, le champ *wIndex* identifie l'interface ou le point d'extrémité.

9.3.2 bRequest

Ce champ spécifie la requête particulière. Les bits *Type* dans le champ *bmRequestType* modifient la signification de ce champ. Cette spécification définit les valeurs pour le champ *bRequest* uniquement quand les bits sont réinitialisés à zéro, indiquant une requête normalisée (voir Tableau 9-3).

9.3.3 wValue

Les contenus de ce champ varient selon la requête. Il est utilisé pour passer un paramètre vers l'appareil, spécifique pour la requête.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

9.3.4 wIndex

Les contenus de ce champ varient selon la requête. Il est utilisé pour passer un paramètre vers l'appareil, spécifique pour la requête.

Le champ *wIndex* est souvent utilisé pour des requêtes pour spécifier un point d'extrémité ou une interface. La Figure 9-2 montre le format de *wIndex* quand il est utilisé pour spécifier un point d'extrémité.

D7	D6	D5	D4	D3	D2	D1	D0
Direction	Réservé (Réinitialisation à zéro)			Numéro de point d'extrémité			
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Réservé (Réinitialisation à zéro)							

Figure 9-2. Format wIndex lors de la spécification d'un point d'extrémité

Le bit de *Direction* est réglé sur zéro pour indiquer le point d'extrémité OUT avec le *Numéro de point d'extrémité* spécifié et sur un pour indiquer le point d'extrémité IN. Dans le cas d'un canal de commande, il convient que la requête ait le bit de *Direction* réglé sur zéro, mais l'appareil peut accepter toute valeur du bit de *Direction*.

La Figure 9-3 montre le format de *wIndex* quand il est utilisé pour spécifier un point d'extrémité.

D7	D6	D5	D4	D3	D2	D1	D0
Numéro d'interface							
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Réservé (Réinitialisation à zéro)							

Figure 9-3. Format wIndex lors de la spécification d'une interface

9.3.5 wLength

Ce champ spécifie la longueur des données transférées au cours de la seconde phase du transfert de commande. La direction du transfert de données (host-to-device ou device-to-host) est indiquée par le bit de *Direction* du champ *bmRequestType*. Si ce champ est zéro, il n'y a pas de phase de transfert de données.

Sur une requête d'entrée, un appareil ne doit jamais retourner plus de données que ce qui est indiqué par la valeur *wLength*; il peut en retourner moins. Sur une requête de sortie, *wLength* indique toujours le volume exact de données à envoyer par l'hôte. Le comportement de l'appareil est indéfini s'il convient que l'hôte envoie plus de données que ce qui est spécifié dans *wLength*.

9.4 Requêtes normalisées de l'appareil

Cette section décrit les requêtes normalisées de l'appareil définies pour tous les appareils USB. Le Tableau 9-3 expose les requêtes normalisées de l'appareil, tandis que les Tableaux 9-4 et 9-5 fournissent les codes de requête normalisés et les types de descripteur, respectivement.

Les appareils USB doivent répondre aux requêtes d'appareil normalisées, même si l'appareil n'a pas encore été assigné à une adresse ou n'a pas été configuré.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 9-3. Requêtes normalisées de l'appareil

bmRequestType	bRequest	wValue	wIndex	wLength	Data
00000000B 00000001B 00000010B	CLEAR_FEATURE	Sélecteur de caractéristiques	Point d'extrémité d'interface zéro	Zéro	Aucun
10000000B	GET_CONFIGURATION	Zéro	Zéro	Un	Valeur de configuration
10000000B	GET_DESCRIPTOR	Type de descripteur et indice de descripteur	Zéro ou ID langage	Longueur descripteur	Descripteur
10000001B	GET_INTERFACE	Zéro	Interface	Un	Autre interface
10000000B 10000001B 10000010B	GET_STATUS	Zéro	Point d'extrémité interface zéro	Deux	Statut Appareil, Interface ou point d'extrémité
00000000B	SET_ADDRESS	Adresse de l'appareil	Zéro	Zéro	Aucun
00000000B	SET_CONFIGURATION	Valeur de configuration	Zéro	Zéro	Aucun
00000000B	SET_DESCRIPTOR	Type de descripteur et indice de descripteur	Zéro ou ID langage	Longueur descripteur	Descripteur
00000000B 00000001B 00000010B	SET_FEATURE	Sélecteur de caractéristiques	Point d'extrémité d'interface zéro	Zéro	Aucun
00000001B	SET_INTERFACE	Autre paramètre	Interface	Zéro	Aucun
10000010B	SYNCH_FRAME	Zéro	Point d'extrémité	Deux	Numéro de trame

Tableau 9-4. Codes de requêtes normalisées

bRequest	Valeur
GET_STATUS	0
CLEAR_FEATURE	1
Réservé pour utilisation ultérieure	2
SET_FEATURE	3
Réservé pour utilisation ultérieure	4
SET_ADDRESS	5
GET_DESCRIPTOR	6
SET_DESCRIPTOR	7
GET_CONFIGURATION	8
SET_CONFIGURATION	9
GET_INTERFACE	10
SET_INTERFACE	11
SYNCH_FRAME	12

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 9-5. Types de descripteurs

Types de descripteurs	Valeur
DEVICE	1
CONFIGURATION	2
STRING	3
INTERFACE	4
ENDPOINT	5
DEVICE_QUALIFIER	6
OTHER_SPEED_CONFIGURATION	7
INTERFACE_POWER[1]	8

Le descripteur INTERFACE_POWER est défini dans la version actuelle du document *USB Interface Power Management Specification*.

Les sélecteurs de caractéristiques sont utilisés lors de l'activation ou du réglage des caractéristiques, telles que la restauration à distance, spécifique à un appareil, à une interface ou à un point d'extrémité. Les valeurs pour les sélecteurs de caractéristiques sont données au Tableau 9-6.

Tableau 9-6. Sélecteur de caractéristiques normalisées

Sélecteur de caractéristiques	Destinataire	Valeur
device_remote_wakeup	Appareil	1
endpoint_HALT	point d'extrémité	0
TEST_MODE	Appareil	2

Si une requête non supportée ou invalide est faite vers un appareil USB, l'appareil répond en renvoyant STALL dans la phase de Données ou d'Etat de la requête. Si l'appareil détecte l'erreur dans la phase d'installation, il est préférable que l'appareil renvoie STALL lors de la phase antérieure de Données ou d'Etat. La réception d'une requête non supportée ou invalide n'entraîne PAS le réglage de la caractéristique *Halt* optionnelle sur le canal de commande. Si pour quelque raison que ce soit, l'appareil devient incapable de communiquer par son Canal de commande par défaut du fait d'un état d'erreur, l'appareil doit être réinitialisé pour effacer l'état et redémarrer le Canal de commande par défaut.**ClearFeature**

Cette requête est utilisée pour effacer ou désactiver une caractéristique spécifique.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
00000000B 00000001B 00000010B		Sélecteur de caractéristiques	Point d'extrémité interface zéro	Zéro	Aucun

Les valeurs de sélecteur de caractéristiques dans wValue doivent être adaptées au destinataire. Seules les valeurs du sélecteur de caractéristiques de l'appareil peuvent être utilisées quand le destinataire est un appareil, seules les valeurs du sélecteur de caractéristiques d'interface peuvent être utilisées quand le destinataire est une interface, et seules les valeurs du sélecteur de caractéristiques du point d'extrémité peuvent être utilisées quand le destinataire est un point d'extrémité.

Voir le Tableau 9-6 pour une définition des valeurs de sélecteur de caractéristiques énoncées et pour quels destinataires.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Une requête ClearFeature() qui fait référence à une caractéristique qui ne peut être effacée, qui n'existe pas ou qui fait référence à une interface ou à un point d'extrémité qui n'existe pas, fait que l'appareil répond avec une Erreur de requête.

Si *wLength* est non nul, le comportement de l'appareil n'est pas spécifié.

Etat Default: Le comportement de l'appareil lors de la réception de cette requête tandis que l'appareil est en état Default n'est pas spécifié.

Etat Address: La requête est valide quand l'appareil est en état Address; des références aux interfaces ou aux points d'extrémité autres que le zéro du point d'extrémité doivent faire que l'appareil répond avec une Erreur de requête.

Etat Configured: Cette requête est valide quand l'appareil est en état Configured.

Note: La caractéristique Test_Mode ne peut être effacée par la requête ClearFeature().

9.4.2 Get Configuration

Cette requête renvoie la valeur de configuration actuelle de l'appareil.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
10000000B	GET_CONFIGURATION	Zéro	Zéro	Un	Valeur de configuration

Si la valeur renvoyée est zéro, l'appareil n'est pas configuré.

Si *wValue*, *wIndex*, ou *wLength* ne sont pas tels que spécifiés ci-dessus, alors le comportement de l'appareil n'est pas spécifié.

Etat Default: Le comportement de l'appareil lors de la réception de cette requête tandis que l'appareil est en état Default n'est pas spécifié.

Etat Address: La valeur zéro doit être renvoyée.

Etat Configured: La *bConfigurationValue* non nulle de la configuration actuelle doit être renvoyée.

9.4.3 Get Descriptor

Cette requête renvoie le descripteur spécifié si le descripteur existe.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
10000000B	GET_DESCRIPTOR	Type de descripteur et indice de descripteur	Zéro ou ID langage (voir Section 9.6.7)	Longueur descripteur	Descripteur

Le champ wValue spécifie le type de descripteur dans l'octet supérieur (voir le Tableau 9-5) ainsi que l'index du descripteur dans l'octet inférieur. L'indice du descripteur est utilisé pour sélectionner un descripteur spécifique (uniquement pour des descripteurs de chaîne et configuration) quand plusieurs descripteurs du même type sont mis en place dans un appareil. Par exemple, un appareil peut mettre en place plusieurs descripteurs de configuration. Pour d'autres descripteurs normalisés qui peuvent être récupérés par une demande GetDescriptor(), un indice de descripteur de zéro doit être utilisé. La plage de valeurs utilisée pour un indice de descripteurs va de 0 à un moins le nombre de descripteurs de ce type mis en place par l'appareil.

Le champ wIndex spécifie l'ID de langage pour les descripteurs de chaîne ou est réinitialisé à zéro pour d'autres descripteurs. Le champ wLength spécifie le nombre d'octets à renvoyer. Si le descripteur est plus long que le champ wLength, seuls les octets initiaux du descripteur sont renvoyés. Si le descripteur est plus court que le champ wLength, l'appareil indique l'extrémité du transfert de commande en envoyant un paquet court quand d'autres données sont demandées. Un paquet court est défini comme un paquet plus court que la taille de la charge utile ou comme un paquet de données de longueur zéro (Chapitre 5).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

La requête normalisée vers un appareil supporte trois types de descripteurs: appareil (également device_qualifier), configuration (également other_speed_configuration), et chaîne. Un appareil capable de grande vitesse supporte le descripteur device_qualifier pour renvoyer des informations sur l'appareil pour la vitesse à laquelle il ne fonctionne pas (incluant *wMaxPacketSize* pour le point d'extrémité par défaut et le nombre de configurations pour l'autre vitesse). Other_speed_configuration renvoie des informations dans la même structure qu'un descripteur de configuration, mais pour une configuration si l'appareil fonctionnait à l'autre vitesse. Une requête pour un descripteur de configuration renvoie le descripteur de configuration, tous les descripteurs d'interface et les descripteurs de point d'extrémité pour toutes les interfaces d'une simple requête. Le premier descripteur d'interface suit le descripteur de configuration. Les descripteurs de point d'extrémité pour la première interface suivent le premier descripteur d'interface. S'il y a d'autres interfaces, leur descripteur d'interface et descripteurs de point d'extrémité suivent les descripteurs de point d'extrémité de la première interface. Les descripteurs spécifiques à une classe et/ou spécifiques au fournisseur suivent les descripteurs normalisés qu'ils prolongent ou modifient.

Tous les appareils doivent fournir un descripteur d'appareil et au moins un descripteur de configuration. Si un appareil ne prend pas en charge un descripteur demandé, il répond avec une Erreur de requête.

Etat Default: Cette requête est valide quand l'appareil est en état Default.

Etat Address: Cette requête est valide quand l'appareil est en état Address.

Etat Configured: Cette requête est valide quand l'appareil est en état Configured.

9.4.4 Get Interface

Cette requête renvoie l'autre paramètre sélectionné pour l'interface spécifiée.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
10000001B	GET_INTERFACE	Zéro	Interface	Un	Autre paramètre

Certains appareils USB ont des configurations avec des interfaces qui ont des paramètres mutuellement exclusifs. Cette requête permet à l'hôte de déterminer l'autre paramètre actuellement sélectionné.

Si *wValue* ou *wLength* ne sont pas tels que spécifiés ci-dessus, alors le comportement de l'appareil n'est pas spécifié.

Si l'interface spécifiée n'existe pas, alors l'appareil répond avec une Erreur de requête.

Etat Default: Le comportement de l'appareil lors de la réception de cette requête tandis que l'appareil est en état Default n'est pas spécifié.

Etat Address: Une réponse Erreur de requête est donnée par l'appareil.

Etat Configured: Cette requête est valide quand l'appareil est en état Configured.

9.4.5 Get Status

Cette requête renvoie l'état pour le destinataire spécifié.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
10000000B 10000001B 10000010B	GET_STATUS	Zéro	Point d'extrémité interface zéro	Deux	Statut Appareil, Interface ou point d'extrémité

Les bits *Recipient* du champ bmRequestType précisent le destinataire souhaité. Les données renvoyées sont l'état actuel du destinataire spécifié.

Si *wValue* ou *wLength* ne sont pas tels que spécifiés ci-dessus, ou si *wIndex* est non zéro pour une requête d'état d'appareil, alors le comportement de l'appareil n'est pas spécifié.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Si une interface ou un point d'extrémité est spécifié(e) et n'existe pas, alors l'appareil répond avec une Erreur de requête.

Etat Default: Le comportement de l'appareil lors de la réception de cette requête tandis que l'appareil est en état Default n'est pas spécifié.

Etat Address: Si une interface ou un point d'extrémité autre que le zéro du point d'extrémité est spécifié, l'appareil répond avec une Erreur de requête.

Etat Configured: Si une interface ou un point d'extrémité qui n'existe pas est spécifié, l'appareil répond avec une Erreur de requête.

Une requête GetStatus() vers un appareil renvoie les informations de la Figure 9-4.

D7	D6	D5	D4	D3	D2	D1	D0
Réservé (Réinitialisation à zéro)						Restauration à distance	Autoalimenté
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Réservé (Réinitialisation à zéro)							

Figure 9-4. Informations renvoyées à un appareil par une requête GetStatus()

Le champ Self Powered indique si l'appareil est actuellement autoalimenté. Si D0 est réglé sur zéro, l'appareil est autoalimenté. Si D0 est réglé sur un, l'appareil est autoalimenté. Le champ Self Powered peut ne pas être modifié par les requêtes SetFeature() ou ClearFeature().

Le champ Remote Wakeup indique si l'appareil est actuellement activé pour demander la restauration à distance. Le mode par défaut pour les appareils qui supportent la restauration à distance est désactivé. Si D1 est réglé sur zéro, la capacité de l'appareil à signaler la restauration à distance est désactivée. Si D1 est réglé sur un, la capacité de l'appareil à signaler la restauration à distance est activée. Le champ Remote Wakeup peut être modifié par les requêtes SetFeature() ou ClearFeature() en utilisant le sélecteur de caractéristiques device_remote_wakeup. Ce champ est réglé sur zéro quand l'appareil est réinitialisé.

Une requête GetStatus() vers une interface renvoie les informations de la Figure 9-5.

D7	D6	D5	D4	D3	D2	D1	D0
Réservé (Réinitialisation à zéro)							
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Réservé (Réinitialisation à zéro)							

Figure 9-5. Informations renvoyées à une interface par une requête GetStatus()

Une requête GetStatus() vers un point d'extrémité renvoie les informations de la Figure 9-6.

D7	D6	D5	D4	D3	D2	D1	D0
Réservé (Réinitialisation à zéro)							Arrêt
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
Réservé (Réinitialisation à zéro)							

Figure 9-6. Informations renvoyées à un point d'extrémité par une requête GetStatus()

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

La caractéristique *Halt* doit être mise en place de tous les types de point d'extrémité interruption et en masse. Si le point d'extrémité est actuellement arrêté, la caractéristique Halt est réglée sur un. Sinon; elle est réglée sur zéro. La caractéristique Halt peut être réglée de manière optionnelle avec la requête SetFeature(ENDPOINT_HALT). Quand il est réglé par la requête Setfeature(), le point d'extrémité montre le même comportement d'arrêt que si le champ a été réglé par un état du matériel. Si l'état entraînant l'arrêt a été levé, l'effacement de la caractéristique *Halt* au moyen d'une requête ClearFeature(ENDPOINT_HALT) fait que le point d'extrémité ne renvoie plus STALL. Pour les points d'extrémité utilisant le basculement de données, qu'un point d'extrémité possède ou non l'ensemble de caractéristiques *Halt*, une requête ClearFeature(ENDPOINT_HALT) donne toujours lieu au basculement de données réinitialisé sur data0. La caractéristique *Halt* est réinitialisée sur zéro soit après une demande SetConfiguration() ou SetInterface(), même si la configuration ou l'interface requise est identique à la configuration ou à l'interface actuelle.Il n'est ni nécessaire ni recommandé que la caractéristique *Halt* soit mise en place pour le Canal de commande par défaut. Cependant, des appareils peuvent régler la caractéristique *Halt* du Canal de commande par défaut afin de refléter un état d'erreur fonctionnel. Si la caractéristique est réglée sur un, l'appareil renvoie STALL dans les phases Data et Status de chaque requête normalisée vers le canal sauf les requêtes GetStatus(), SetFeature() et ClearFeature(). Il n'est pas nécessaire pour l'appareil de renvoyer STALL pour les requêtes spécifiques à une classe et spécifiques à un fournisseur.

9.4.6 Set Address

Cette requête règle l'adresse de l'appareil pour tous les futurs accès de l'appareil.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
00000000B	SET_ADDRESS	Adresse de l'appareil	Zéro	Zéro	Aucun

Le champ wValue spécifie l'adresse de l'appareil à utiliser pour tous les accès subséquents.

Tel que mentionné ailleurs, les requêtes peuvent en fait entraîner jusqu'à trois phases. Dans la première phase, le paquet Installation est envoyé vers l'appareil. Dans la deuxième phase optionnelle, les données sont transférées entre l'hôte et l'appareil. Dans la dernière phase, l'état est transféré entre l'hôte et l'appareil. La direction des données et de l'état dépend soit de l'hôte qui envoie des données vers l'appareil, soit de l'appareil qui envoie des données vers l'hôte. Le transfert de la phase Status est toujours dans la direction opposée de la phase Data. S'il n'y a pas de phase Data, la phase Status va de l'appareil vers l'hôte.

Les phases suivant le paquet Setup initial supposent la même adresse d'appareil que le paquet Setup. L'appareil USB ne change pas son adresse d'appareil jusqu'à ce que la phase Status de cette requête soit achevée avec succès. Noter qu'il y a une différence entre cette requête et toutes les autres requêtes. Pour toutes les autres requêtes, l'opération indiquée doit être terminée avant la phase Status.

Si l'adresse de l'appareil spécifié est supérieure à 127, ou si *wIndex* ou *wLength* sont non nuls, alors le comportement de l'appareil n'est pas spécifié.

La réponse de l'appareil à SetAddress() avec une valeur 0 est indéfinie.

Etat Default: Si l'adresse spécifiée est non zéro, l'appareil doit passer à l'état Address; sinon, l'appareil reste en état Default (ce n'est pas un état d'erreur).

Etat Address: Si l'adresse spécifiée est zéro, l'appareil doit passer à l'état Default; sinon l'appareil reste en état Address mais utilise l'adresse nouvellement spécifiée.

Etat Configured: Le comportement de l'appareil lors de la réception de cette requête tandis que l'appareil est en état Configured n'est pas spécifié.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

9.4.7 Set Configuration

Cette requête règle la configuration de l'appareil.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
00000000B	SET_CONFIGURATION	Valeur de configuration	Zéro	Zéro	Aucun

L'octet le plus bas du champ wValue précise la configuration souhaitée. Cette valeur de configuration doit être zéro ou doit correspondre à une valeur de configuration à partir d'un descripteur de configuration. Si la valeur de configuration est zéro, l'appareil est placé en état Address. L'octet le plus haut du champ *wValue* est réservé.

Si *wIndex*, *wLength*, ou l'octet le plus haut de *wValue* est non nul, alors le comportement de cette requête n'est pas spécifié.

Etat Default: Le comportement de l'appareil lors de la réception de cette requête tandis que l'appareil est en état Default n'est pas spécifié.

Etat Address: Si la valeur de configuration spécifiée est zéro, alors l'appareil reste en état Address. Si la valeur de configuration spécifiée correspond à la valeur de configuration à partir d'un descripteur de configuration, alors cette configuration est sélectionnée et l'appareil passe à l'état Configured. Sinon l'appareil répond avec une Erreur de requête.

Etat Configured: Si la valeur de configuration spécifiée est zéro, alors l'appareil passe à l'état Address. Si la valeur de configuration spécifiée correspond à la valeur de configuration à partir d'un descripteur de configuration, alors cette configuration est sélectionnée et l'appareil reste en état Configured. Sinon l'appareil répond avec une Erreur de requête.

9.4.8 Set Descriptor

Cette requête est optionnelle et peut être utilisée pour mettre à jour des descripteurs existants ou de nouveaux descripteurs peuvent être ajoutés.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
00000000B	SET_DESCRIPTOR	Type de descripteur et indice de descripteur	ID langage (voir la Section 9.6.7) ou zéro	Longueur descripteur	Descripteur

Le champ wValue spécifie le type de descripteur dans l'octet supérieur (voir le Tableau 9-5) ainsi que l'index du descripteur dans l'octet inférieur. L'indice du descripteur est utilisé pour sélectionner un descripteur spécifique (uniquement pour des descripteurs de chaîne et configuration) quand plusieurs descripteurs du même type sont mis en place dans un appareil. Par exemple, un appareil peut mettre en place plusieurs descripteurs de configuration. Pour d'autres descripteurs normalisés qui peuvent être réglés par une demande SetDescriptor(), un indice de descripteur de zéro doit être utilisé. La plage de valeurs utilisée pour un indice de descripteurs va de 0 à un moins le nombre de descripteurs de ce type mis en place par l'appareil.

Le champ wIndex spécifie l'ID de langage pour les descripteurs de chaîne ou est réinitialisé à zéro pour d'autres descripteurs. Le champ wLength spécifie le nombre d'octets à transférer de l'hôte jusqu'à l'appareil.

Les seules valeurs permises pour le type de descripteur sont les types d'appareil, configuration et descripteur de chaîne.

Si cette requête n'est pas supportée, l'appareil répond avec une Erreur de requête.

Etat Default: Le comportement de l'appareil lors de la réception de cette requête tandis que l'appareil est en état Default n'est pas spécifié.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Etat Address: Si elle est supportée, cette requête est valide quand l'appareil est en état Address.

Etat Configured: Si elle est supportée, cette requête est valide quand l'appareil est en état Configured.

9.4.9 Set Feature

Cette requête est utilisée pour régler ou activer une caractéristique spécifique.

Type bmRequest	bRequest	wValue	wIndex		wLength	Données
00000000B 00000001B 00000010B	SET_FEATURE	Sélecteur de caractéristiques	Sélecteur d'essais	Point d'extrémité interface zéro	Zéro	Aucun

Les valeurs de sélecteur de caractéristiques dans wValue doivent être adaptées au destinataire. Seules les valeurs du sélecteur de caractéristiques de l'appareil peuvent être utilisées quand le destinataire est un appareil, seules les valeurs du sélecteur de caractéristiques d'interface peuvent être utilisées quand le destinataire est une interface, et seules les valeurs du sélecteur de caractéristiques du point d'extrémité peuvent être utilisées quand le destinataire est un point d'extrémité.

Voir le Tableau 9-6 pour une définition des valeurs de sélecteur de caractéristiques énoncées et pour quels destinataires.

La caractéristique TEST_MODE est uniquement définie pour un destinataire d'appareil (c'est-à-dire bmRequestType = 0) et l'octet inférieur de wIndex doit être nul. La définition de la caractéristique TEST_MODE fait passer le port orienté en amont de l'appareil en mode essai. L'appareil répond avec une erreur de requête si la requête contient un sélecteur d'essai invalide. La transition vers le mode d'essai doit être terminée avant 3 ms suivant l'achèvement de l'étape d'état de la requête. La transition vers le mode essai d'un port en aval ne doit se produire qu'après la phase d'état de la requête. L'alimentation vers l'appareil doit être amenée de manière à quitter le mode d'essai d'un port en aval d'un appareil. Voir la Section 7.1.20 pour les définitions de chaque mode d'essai. Un appareil doit prendre en charge la caractéristique TEST_MODE lorsqu'il est dans les états haute vitesse Default, Address ou Configured.

Une requête SetFeature() qui fait référence à une caractéristique qui ne peut être réglée, ou qui n'existe pas, fait qu'un STALL doit être renvoyé à la phase Status de la requête.

Tableau 9-7. Sélecteurs de mode d'essai

Valeur	Description
00H	Réservé
01H	Test_J
02H	Test_K
03H	Test_SE0_NAK
04H	Test_Paquet
05H	Test_Force_Activer
06H-3FH	Réservé pour sélecteurs d'essai normalisés
3FH-BFH	Réservé
C0H-FFH	Réservé pour modes d'essai spécifiques au fournisseur.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Si le sélecteur de caractéristiques est *TEST_MODE*, alors l'octet le plus significatif de *wIndex* est utilisé pour spécifier le mode d'essai spécifique. Le destinataire d'un SetFeature(TEST_MODE…) doit être l'appareil; c'est-à-dire que l'octet inférieur de *wIndex* doit être nul et que *bmRequestType* doit être réglé sur zéro. L'alimentation de l'appareil doit être amenée de manière à quitter le mode d'essai. Les valeurs pour les sélecteurs de mode d'essai valides sont données au Tableau 9-7. Voir la Section 7.1.20 pour plus d'informations sur les modes d'essai spécifiques.

Si wLength est non zéro, alors le comportement de l'appareil n'est pas spécifié.

Si un point d'extrémité ou une interface est spécifié et n'existe pas, alors l'appareil répond avec une Erreur de requête.

Etat Default: Un appareil doit être capable d'accepter une requête SetFeature(TEST_MODE, TEST_SELECTOR) lorsqu'il est en état Default. Le comportement de l'appareil pour d'autres requêtes SetFeature tandis que l'appareil est en état Default n'est pas spécifié.

Etat Address: Si une interface ou un point d'extrémité autre que le zéro du point d'extrémité est spécifié, l'appareil répond avec une Erreur de requête.

Etat Configured: Cette requête est valide quand l'appareil est en état Configured.

9.4.10 Set Interface

Cette requête permet à l'hôte de sélectionner un autre paramètre pour l'interface spécifiée.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
00000001B	SET_INTERFACE	Autre paramètre	Interface	Zéro	Aucun

Certains appareils USB ont des configurations avec des interfaces qui ont des paramètres mutuellement exclusifs. Cette requête permet à l'hôte de sélectionner l'autre paramètre souhaité. Si un appareil ne supporte qu'un paramètre par défaut pour l'interface spécifiée, alors un STALL peut être renvoyé dans la phase Etat de la requête. Cette requête ne peut être utilisée pour modifier l'ensemble d'interfaces configurées (la requête SetConfiguration() doit être utilisée à la place).

Si l'interface ou l'autre paramètre n'existe pas, alors l'appareil répond avec une Erreur de requête. Si wLength est non zéro, alors le comportement de l'appareil n'est pas spécifié.

Etat Default: Le comportement de l'appareil lors de la réception de cette requête tandis que l'appareil est en état Default n'est pas spécifié.

Etat Address: L'appareil doit répondre avec une Erreur de requête.

Etat Configured: Cette requête est valide quand l'appareil est en état Configured.

9.4.11 Synch Frame

Cette requête est utilisée pour régler puis signaler une trame de synchronisation d'un point d'extrémité.

Type bmRequest	bRequest	wValue	wIndex	wLength	Données
10000010B	SYNCH_FRAME	Zéro	Point d'extrémité	Deux	Numéro de trame

Quand un point d'extrémité supporte des transferts isochrones, le point d'extrémité peut également demander que des transferts de trame varient en taille selon un modèle spécifique. L'hôte et le point d'extrémité doivent s'accorder sur quelle trame débute le modèle de répétition. Le numéro de trame dans lequel a commencé le modèle est renvoyé à l'hôte.

Si un appareil à grande vitesse supporte la requête Synch Frame, il doit se synchroniser lui-même de manière interne vers la microtrame zéro et avoir une notion de temps de trame classique. Seul le numéro de trame est utilisé pour synchroniser et est signalé par le point d'extrémité de l'appareil (c'est-à-dire pas de numéro de microtrame). Le point d'extrémité doit synchroniser vers la microtrame zéro.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Cette valeur est utilisée uniquement pour des transferts de données isochrones en utilisant une synchronisation de modèle implicite. Si *wValue* est nul ou *wLength* n'est pas égal à deux, alors le comportement de l'appareil n'est pas spécifié.

Si le point d'extrémité spécifié ne supporte pas cette requête, l'appareil répond avec une Erreur de requête.

Etat Default: Le comportement de l'appareil lors de la réception de cette requête tandis que l'appareil est en état Default n'est pas spécifié.

Etat Address: L'appareil doit répondre avec une Erreur de requête.

Etat Configured: Cette requête est valide quand l'appareil est en état Configured.

9.5 Descripteurs

Les appareils USB signalent leurs attributs en utilisant les descripteurs. Un descripteur est une structure de données avec un format défini. Chaque descripteur commence avec un champ de largeur d'octet qui contient le nombre total d'octets dans le descripteur suivi par un champ de largeur d'octet qui identifie le type de descripteur.

L'utilisation de descripteurs permet un stockage concis des attributs des configurations individuelles puisque chaque configuration peut réutiliser des descripteurs ou parties de descripteurs à partir d'autres configurations qui ont les mêmes caractéristiques. De cette manière, les descripteurs s'apparentent à des enregistrements de données individuels dans une base de données relationnelle.

Si appropriés, les descripteurs contiennent des références à des descripteurs de chaîne qui fournissent des informations affichables décrivant un descripteur sous forme lisible par l'humain. L'inclusion des descripteurs de chaîne est optionnelle. Cependant, les champs de référence dans les descripteurs sont obligatoires. Si un appareil ne supporte pas les descripteurs de chaîne, les champs de référence de chaîne doivent être réinitialisés à zéro pour indiquer qu'aucun descripteur de chaîne n'est disponible.

Si un descripteur retourne avec une valeur dans son champ de longueur inférieure à celle définie par cette spécification, le descripteur est invalide et il convient que l'hôte le rejette. Si le descripteur retourne avec une valeur dans son champ de longueur qui est supérieure à celle définie par cette spécification, les octets supplémentaires sont ignorés par l'hôte, mais le descripteur suivant est placé avec la longueur retournée plutôt que la longueur prévue.

Un appareil peut retourner des descripteurs spécifiques à la classe ou au fournisseur de deux manières:

1. Si les descripteurs spécifiques à la classe ou au fournisseur utilisent le même format que les descripteurs normalisés (par exemple, commencent avec un octet de longueur et suivis par un octet de type), ils doivent être retournés entrelacés avec des descripteurs normalisés dans les informations de configuration renvoyées par une requête GetDescriptor(Configuration). Dans ce cas, les descripteurs spécifiques à la classe ou au fournisseur doivent suivre un descripteur normalisé associé qu'ils modifient ou prolongent.
2. Si les descripteurs spécifiques à la classe ou au fournisseur sont indépendants des informations de configuration ou utilisent un format non normalisé, une demande GetDescriptor() spécifiant le type et l'indice de descripteur spécifique à la classe ou au fournisseur peut être utilisée pour récupérer le descripteur à partir de l'appareil. Une spécification de classe ou de fournisseur définit la manière appropriée de récupérer ces descripteurs.

9.6 Définitions de descripteur USB normalisé

Les descripteurs normalisés définis dans cette spécification peuvent uniquement être modifiés ou prolongés avec révision de la Spécification USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Note: Une extension du descripteur de point d'extrémité normalisé USB 1.0 a été publiée dans le document Device Class Specification for Audio Devices Revision 1.0. (Révision 1.0 de la spécification des classes d'appareils des appareils audio). Il s'agit de la seule extension définie en dehors de la spécification USB autorisée. De futures révisions des Spécifications USB prolongent le descripteur du point d'extrémité normalisé de manière à ne pas entrer en conflit avec l'extension définie dans Audio Device Class Specification Revision 1.0 (Révision 1.0 des spécifications de classe d'appareils audio).

9.6.1 Appareil

Un descripteur d'appareil décrit des informations générales sur un appareil USB. Il inclut des informations qui s'appliquent globalement aux configurations de l'appareil et de tous les appareils. Un appareil USB n'a qu'un descripteur d'appareil.

Un appareil capable de grande vitesse qui a des informations d'appareil différentes pour la vitesse intégrale/la grande vitesse doit aussi avoir un descripteur device_qualifier (voir 9.6.2).

Le descripteur DEVICE d'un appareil capable de grande vitesse a un numéro de version de 2.0 (0200H). Si l'appareil est uniquement à vitesse intégrale ou uniquement à grande vitesse, ce numéro de version indique qu'il répond correctement à une requête pour le descripteur device_qualifier (c'est-à-dire qu'il répond avec une erreur de requête).

Le champ *bcdUSB* contient un numéro de version BCD. La valeur du champ *bcdUSB* est 0xJJMN pour la version JJ.M.N (JJ - numéro de version majeur, M - numéro de version mineur, N - numéro de version sous-mineur), par exemple, la version 2.1.3 est représentée avec une valeur de 0x0213 et la version 2.0 est représentée avec une valeur de 0x0200.

bNumConfigurations indique le nombre de configurations à la vitesse de fonctionnement actuelle. Les configurations pour l'autre vitesse d'opération ne sont pas incluses dans le comptage. S'il y a des configurations spécifiques de l'appareil pour des vitesses spécifiques, le champ *bNumConfigurations* ne reflète que le nombre de configurations pour une vitesse unique, et non le nombre total de configurations pour les deux vitesses.

Si l'appareil fonctionne à grande vitesse, le champ *bMaxPacketSize0* doit être 64 indiquant un paquet maximum de 64 octets. L'opération à grande vitesse ne permet pas d'autres tailles maximales de paquet pour le point d'extrémité de commande (point d'extrémité 0).

Tous les appareils USB ont un Canal de commande par défaut. La taille de paquet maximum d'un Canal de commande par défaut d'un appareil est décrite dans le descripteur de l'appareil. Des points d'extrémité spécifiques à une configuration et leur(s) interface(s) sont décrit(s) dans le descripteur de configuration. Une configuration et son(ses) interface(s) n'incluent pas de descripteur de point d'extrémité pour le Canal de commande par défaut. A part la taille maximum du paquet, les caractéristiques du Canal de commande par défaut sont définies par ces spécifications et sont les mêmes pour tous les appareils USB.

Le champ bNumConfigurations identifie le nombre de configurations prises en charge par l'appareil. Le Tableau 9-8 illustre le descripteur d'appareil normalisé.

Tableau 9-8. Descripteur d'appareil normalisé

Décalage	Champ	Taille	Valeur	Description
0	*bLength*	1	Numéro	Taille du descripteur en octets
1	*bDescriptorType*	1	Constant	Type de descripteur DEVICE
2	*bcdUSB*	2	BCD	Numéro de version de spécification USB en décimal codé binaire (c'est-à-dire, 2.10 est égal à 210H). Ce champ identifie la version de la spécification USB à laquelle l'appareil et ses descripteurs se conforment.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 9-8. Descripteur d'appareil normalisé (suite)

Décalage	Champ	Taille	Valeur	Description
4	*bDeviceClass*	1	Classe	Code de classe (attribué par l'USB-IF). Si ce champ est réinitialisé à zéro, chaque interface d'une même configuration spécifie ses propres informations de classe et les différentes interfaces fonctionnent de façon indépendante. Si ce champ est réglé sur une valeur entre 1 et FEH, l'appareil prend en charge différentes spécifications de classe sur différentes interfaces et les interfaces peuvent ne pas fonctionner de manière indépendante. Cette valeur identifie la définition de classe utilisée pour les interfaces agrégées. Si ce champ est réglé sur FFH, la classe de l'appareil est spécifique au vendeur.
5	*bDeviceSubClass*	1	Sous-classe	Code de sous-classe (attribué par l'USB-IF). Ces codes sont qualifiés par la valeur du champ bDeviceClass. Si le champ bDeviceClass est réinitialisé à zéro, ce champ doit également être réinitialisé à zéro. Si le champ bDeviceClass n'est pas réglé sur FFH, toutes les valeurs sont réservées pour l'attribution par l'USB-IF.
6	*bDeviceProtocol*	1	Protocole	Code de protocole (attribué par l'USB-IF). Ces codes sont qualifiés par la valeur des champs bDeviceClass et bDeviceSubClass. Si un appareil prend en charge des protocoles spécifiques à la classe sur une base d'appareil, par opposition à une classe d'interface, ce code identifie les protocoles que l'appareil utilise tel que défini par la spécification de la classe d'appareil. Si ce champ est réinitialisé à zéro, l'appareil n'utilise pas de protocole spécifique à la classe sur une base d'appareil. Cependant, il peut utiliser des protocoles spécifiques à la classe sur une base d'interface. Si ce champ est réglé sur FFH, l'appareil utilise un protocole spécifique au vendeur sur une base d'appareil.
7	*bMaxPacketSize0*	1	Numéro	Taille maximum de paquet pour un point d'extrémité zéro (seuls 8, 16, 32 ou 64 sont valides)
8	*idVendor*	2	ID	ID du vendeur (attribué par l'USB-IF)
10	*idProduct*	2	ID	ID du produit (attribué par le fabricant).
12	*bcdDevice*	2	BCD	Numéro de version de l'appareil en décimal codé binaire
14	*iManufacturer*	1	Indice	Indice du descripteur de chaîne de caractères décrivant le fabricant
15	*iProduct*	1	Indice	Indice du descripteur de chaîne de caractères décrivant le produit
16	*iSerialNumber*	1	Indice	Indice du descripteur de chaîne de caractères décrivant le numéro de série de l'appareil
17	*bNumConfigurations*	1	Numéro	Nombre de configurations possibles

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

9.6.2 Device_Qualifier

Le descripteur device_qualifier décrit les informations concernant un appareil à haute vitesse qui sont susceptibles de changer si l'appareil fonctionne à une autre vitesse. Par exemple, si l'appareil fonctionne actuellement à vitesse intégrale, le descripteur device_qualifier retourne les informations concernant la manière dont il devrait fonctionner à haute vitesse et vice-versa. Le Tableau 9-9 illustre les champs du descripteur device_qualifier.

Tableau 9-9. Device_Qualifier Descriptor

Décalage	Champ	Taille	Valeur	Description
0	*bLength*	1	Numéro	Taille du descripteur
1	*bDescriptorType*	1	Constant	Type du qualificatif d'appareil
2	*bcdUSB*	2	BCD	Numéro de version de la spécification USB (par exemple, 0200H pour V2.00)
4	*bDeviceClass*	1	Classe)	Code de classe
5	*bDeviceSubClass*	1	Sous-classe	Code de sous-classe
6	*bDeviceProtocol*	1	Protocole	Code de protocole
7	*bMaxPacketSize0*	1	Numéro	Taille de paquet maximum pour une autre vitesse
8	*bNumConfigurations*	1	Numéro	Nombre de configurations à d'autres vitesses
9	*bReserved*	1	Zéro	Réservé à un usage ultérieur, doit être réglé sur zéro

Les champs vendeur, produit, appareil, fabricant et numéro de série du descripteur d'appareil normalisé ne sont pas inclus dans ce descripteur étant donné que ces informations sont constantes pour un appareil pour toutes les vitesses prises en charge. Le numéro de version de ce descripteur doit être au moins égal à 2.0 (0200H).

L'hôte accède au descripteur à l'aide de la requête GetDescriptor(). Le type de descripteur dans la requête GetDescriptor() est réglé sur device_qualifier (voir Tableau 9-5).

Si un appareil fonctionnant uniquement à vitesse intégrale (avec un numéro de version de descripteur d'appareil égal à 0200H) reçoit une requête GetDescriptor() pour un device_qualifier, il doit répondre avec une erreur de requête. L'hôte ne doit pas effectuer de requête pour un autre descripteur other_speed_configuration à moins qu'il ne retrouve tout d'abord avec succès le descripteur device_qualifier.

9.6.3 Configuration

Le descripteur de configuration décrit les informations concernant une configuration spécifique à l'appareil. Le descripteur contient un champ bConfigurationValue avec une valeur qui, lorsqu'elle est utilisée comme paramètre de la requête SetConfiguration(), oblige l'appareil à assumer la configuration décrite.

Le descripteur décrit le nombre d'interfaces fournies par la configuration. Chaque interface peut fonctionner de manière indépendante. Par exemple, un appareil ISDN peut être configuré avec deux interfaces, chacune fournissant des canaux bidirectionnels de 64 Kb/s qui ont des sources ou des destinataires de données séparés sur l'hôte. Une autre configuration peut présenter l'appareil comme une interface unique, combinant les deux canaux en un canal bidirectionnel de 128 Kb/s.

Lorsque l'hôte demande le descripteur de configuration, tous les descripteurs d'interface et de point d'extrémité associés sont retournés (voir la Section 9.4.3).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Un appareil USB dispose d'un ou plusieurs descripteurs de configuration. Chaque configuration dispose d'une ou plusieurs interfaces et chaque interface dispose de zéro ou plusieurs points d'extrémité. Un point d'extrémité n'est pas partagé entre les interfaces dans le cadre d'une seule configuration, à moins que le point d'extrémité ne soit utilisé par des réglages alternatifs de la même interface. Les points d'extrémité peuvent être partagés par plusieurs interfaces qui font partie de différentes configurations sans cette restriction.

Une fois configurés, les appareils peuvent prendre en charge des réglages limités de la configuration. Si une interface particulière a des réglages alternatifs, l'un d'entre eux peut être sélectionné après la configuration. Le Tableau 9-10 illustre le descripteur de configuration normalisé.

Tableau 9-10. Descripteur de configuration normalisé

Décalage	Champ	Taille	Valeur	Description
0	*bLength*	1	Numéro	Taille du descripteur en octets
1	*bDescriptorType*	1	Constant	Type de descripteur CONFIGURATION
2	*wTotalLength*	2	Numéro	Longueur totale des données retournées par cette configuration. Inclut la longueur combinée de tous les descripteurs (configuration, interface, point d'extrémité et spécifique à la classe ou au vendeur) retournés par cette configuration.
4	*bNumInterfaces*	1	Numéro	Nombre d'interfaces prises en charge par cette configuration
5	*bConfigurationValue*	1	Numéro	Valeur qui doit être utilisée comme argument de la requête SetConfiguration() pour la sélection de cette configuration
6	*iConfiguration*	1	Indice	Indice du descripteur de chaîne de caractères décrivant cette configuration
7	*bmAttributes*	1	Bitmap	Caractéristiques de configuration D7: réservé (réglé sur un) D6: autoalimenté D5: activation à distance D4...0: réservé (réinitialisé à zéro) D7 est réservé et doit être réglé sur zéro pour des raisons historiques. Une configuration d'appareil qui utilise l'alimentation d'un bus et d'une source locale indique une valeur non nulle dans le champ *bMaxPower* pour indiquer la quantité d'alimentation de bus requise et règle D6. La source d'alimentation actuelle durant l'exécution peut être déterminée à l'aide de la requête GetStatus(DEVICE) (voir Section 9.4.5). Si une configuration d'appareil prend en charge l'activation à distance, D5 est réglé sur un.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 9-10. Descripteur de configuration normalisé (suite)

Décalage	Champ	Taille	Valeur	Description
8	*bMaxPower*	1	mA	Consommation énergétique maximum de l'appareil USB à partir du bus avec cette configuration spécifique lorsque l'appareil est complètement opérationnel. Exprimée en unités 2 mA (c'est-à-dire, 50 = 100 mA). Note: une configuration d'appareil signale si la configuration est alimentée par le bus ou autoalimentée. L'état de l'appareil indique si l'appareil est actuellement autoalimenté. Si un appareil est déconnecté de sa source d'alimentation externe, il met à jour l'état de l'appareil pour indiquer qu'il n'est plus autoalimenté. Un appareil peut ne pas augmenter sa consommation d'énergie provenant du bus lorsqu'il perd sa source d'alimentation externe, en plus de la quantité indiquée par sa configuration. Si un appareil peut continuer à fonctionner lorsqu'il est déconnecté de sa source d'alimentation externe, il continue à le faire. Si l'appareil ne peut plus fonctionner, il fait échouer les opérations qu'il ne peut plus prendre en charge. Le logiciel système USB peut déterminer la cause du dysfonctionnement en vérifiant l'état et en notant la perte de la source d'alimentation de l'appareil.

9.6.4 Other_Speed_Configuration

Le descripteur other_speed_configuration illustré au Tableau 9-11 décrit une configuration d'un appareil à haute vitesse s'il fonctionne à une autre vitesse possible. La structure other_speed_configuration est identique à celle d'un descripteur de configuration.

Tableau 9-11. Descripteur Other_Speed_Configuration

Décalage	Champ	Taille	Valeur	Description
0	*bLength*	1	Numéro	Taille du descripteur
1	*bDescriptorType*	1	Constant	Type Other_speed_Configuration
2	*wTotalLength*	2	Numéro	Longueur totale des données retournées
4	*bNumInterfaces*	1	Numéro	Nombre d'interfaces prises en charge par cette configuration de vitesse
5	*bConfigurationValue*	1	Numéro	Valeur qui doit être utilisée pour la sélection de la configuration
6	*iConfiguration*	1	Indice	Indice du descripteur de chaîne de caractères
7	*bmAttributes*	1	Bitmap	Identique au descripteur de configuration
8	*bMaxPower*	1	mA	Identique au descripteur de configuration

L'hôte accède au descripteur à l'aide de la requête GetDescriptor(). Le type de descripteur dans la requête GetDescriptor() est réglé sur other_speed_configuration (voir Tableau 9-5).

9.6.5 Interface

Le descripteur d'interface décrit une interface spécifique à une seule configuration. Une configuration fournit une ou plusieurs interfaces, chacune avec zéro ou plusieurs descripteurs de point d'extrémité décrivant un ensemble unique de points d'extrémité dans le cadre de la configuration. Lorsqu'une configuration prend en charge plusieurs interfaces, les descripteurs de point d'extrémité d'une interface particulière suivent le descripteur d'interface dans les données retournées par la requête GetConfiguration(). Un descripteur d'interface est toujours retourné comme faisant partie d'un descripteur de configuration. Les descripteurs d'interface ne peuvent pas avoir un accès direct à une requête GetDescriptor() ou SetDescriptor().

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Une interface peut inclure des réglages alternatifs qui permettent la modification des points d'extrémité et/ou de leurs caractéristiques après la configuration de l'appareil. Le réglage par défaut d'une interface est toujours un réglage alternatif à zéro. La requête SetInterface() est utilisée pour sélectionner un réglage alternatif ou retourner le réglage par défaut. La requête GetInterface() retourne le réglage alternatif sélectionné.

Le réglage alternatif permet la modification d'une partie de la configuration de l'appareil pendant que les autres interfaces continuent à fonctionner. Si une ou plusieurs interfaces d'une configuration ont subi des réglages alternatifs, un descripteur d'interface séparé et ses points d'extrémité associés sont inclus dans chaque réglage.

Si une configuration d'appareil a pris en charge une seule interface avec deux réglages alternatifs, le descripteur de configuration ne serait suivi d'un descripteur d'interface avec les champs *bInterfaceNumber* et *bAlternateSetting* réglés à zéro, puis des descripteurs de point d'extrémité de ce réglage, suivis d'un autre descripteur d'interface et des descripteurs de point d'extrémité associés. Le champ *bInterfaceNumber* du deuxième descripteur d'interface devrait également être réglé à zéro, mais le champ *bAlternateSetting* du deuxième descripteur d'interface serait réglé sur un.

Si une interface utilise un point d'extrémité zéro, aucun descripteur de point d'extrémité ne suit le descripteur d'interface. Dans ce cas, le champ *bNumEndpoints* doit être réglé sur zéro.

Un descripteur d'interface n'inclut jamais un point d'extrémité zéro dans le nombre de points d'extrémité. Le Tableau 9-12 illustre le descripteur d'interface normalisé.

Tableau 9-12. Descripteur d'interface normalisé

Décalage	Champ	Taille	Valeur	Description
0	*bLength*	1	Numéro	Taille du descripteur en octets
1	*bDescriptorType*	1	Constant	Type de descripteur INTERFACE
2	*bInterfaceNumber*	1	Numéro	Nombre d'interfaces Sync Valeur à base zéro identifiant l'indice de l'ensemble d'interfaces concurrentes prises en charge par cette configuration.
3	*bAlternateSetting*	1	Numéro	Valeur utilisée pour sélectionner ce réglage alternatif pour l'interface identifiée dans le champ précédent
4	*bNumEndpoints*	1	Numéro	Nombre de points d'extrémité utilisés par cette interface (excepté le point d'extrémité zéro). Si cette valeur est égale à zéro, cette interface utilise uniquement le canal de contrôle par défaut.
5	*bInterfaceClass*	1	Classe	Code de classe (attribué par l'USB-IF). Une valeur égale à zéro est réservée pour une normalisation ultérieure. Si ce champ est réglé sur FFH, la classe de l'interface est spécifique au vendeur. Toutes les autres valeurs sont réservées pour l'attribution par l'USB-IF.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 9-12. Descripteur d'interface normalisé (suite)

Décalage	Champ	Taille	Valeur	Description
6	*bInterfaceSubClass*	1	Sous-classe	Code de sous-classe (attribué par l'USB-IF). Ces codes sont qualifiés par la valeur du champ bInterfaceClass. Si le champ bInterfaceClass est réinitialisé à zéro, ce champ doit également être réinitialisé à zéro. Si le champ bInterfaceClass n'est pas réglé sur FFH, toutes les valeurs sont réservées pour l'attribution par l'USB-IF.
7	*bInterfaceProtocol*	1	Protocole	Code de protocole (attribué par l'USB). Ces codes sont qualifiés par la valeur des champs bInterfaceClass et bInterfaceSubClass. Si une interface prend en charge des requêtes spécifiques à la classe, ce code identifie les protocoles que l'appareil utilise tel que défini par la spécification de la classe de l'appareil. Si ce champ est réinitialisé à zéro, l'appareil n'utilise pas de protocole spécifique à la classe sur cette interface. Si ce champ est réglé sur FFH, l'appareil utilise un protocole spécifique au vendeur pour cette interface.
8	*iInterface*	1	Indice	Indice du descripteur de chaîne de caractères décrivant cette interface

9.6.6 Point d'extrémité

Chaque point d'extrémité utilisé pour une interface dispose de son propre descripteur. Ce descripteur contient les informations requises par l'hôte pour déterminer les exigences en matière de bande passante de chaque point d'extrémité. Un descripteur de point d'extrémité est toujours retourné dans le cadre des informations de configuration retournées par une requête GetDescriptor(Configuration). Un descripteur de point d'extrémité ne peut pas avoir d'accès direct à une requête GetDescriptor() ou SetDescriptor(). Il n'y a jamais de descripteur de point d'extrémité pour un point d'extrémité zéro. Le Tableau 9-13 illustre le descripteur de point d'extrémité normalisé.

Tableau 9-13. Descripteur de point d'extrémité normalisé

Décalage	Champ	Taille	Valeur	Description
0	*bLength*	1	Numéro	Taille du descripteur en octets
1	*bDescriptorType*	1	Constant	Type de descripteur ENDPOINT
2	*bEndpointAddress*	1	Point d'extrémité	L'adresse du point d'extrémité de l'appareil USB décrit par ce descripteur. L'adresse est codée de la façon suivante: Bit 3...0: le numéro du point d'extrémité Bit 6...4: réservé, réinitialisé à zéro Bit 7: Direction, ignorée pour les points d'extrémité de contrôle 0 = point d'extrémité OUT 1 = point d'extrémité IN

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 9-13. Descripteur de point d'extrémité normalisé (suite)

Décalage	Champ	Taille	Valeur	Description
3	*bmAttributes*	1	Bitmap	Ce champ décrit les attributs du point d'extrémité lorsqu'il est configuré à l'aide du champ *bConfigurationValue*. Bits 1..0: type de transfert 00 = Contrôle 01 = Isochrone 10 = Bloc 11 = Interruption Si le point d'extrémité n'est pas isochrone, les bits 5..2 sont réservés et doivent être réglés sur zéro. S'il est isochrone, ils sont définis comme suit: Bits 3..2: Type de synchronisation 00 = Aucune synchronisation 01 = Asynchrone 10 = Adaptatif 11 = Synchrone Bits 5..4: Type d'utilisation 00 = Point d'extrémité de données 01 = Point d'extrémité de rétroaction 10 = Point d'extrémité de données à rétroaction implicite 11 = Réservé Voir le Chapitre 5 pour plus d'informations. Tous les autres bits sont réservés et doivent être réinitialisés à zéro. Les bits réservés doivent être ignorés par l'hôte.
4	*wMaxPacketSize*	2	Numéro	La taille maximum de paquet que ce point d'extrémité est en mesure d'envoyer ou de recevoir lorsque cette configuration est sélectionnée. Pour les points d'extrémité isochrones, cette valeur est utilisée pour réserver le temps de bus dans la planification, requis pour les charges utiles de données par (micro)trame. Sur une base continue, le canal peut en fait utiliser une quantité de bande passante inférieure à celle réservée. L'appareil signale, si nécessaire, la bande passante réellement utilisée par l'intermédiaire de ses mécanismes normaux et non définis pour l'USB. Pour tous les points d'extrémité, les bits 10..0 spécifient la taille maximum de paquet (en octets). Pour les points d'extrémité isochrones et d'interruption à vitesse intégrale: les bits 12..11 spécifient le nombre d'opportunités de transaction supplémentaires par microtrame: 00 = Aucune (1 transaction par microtrame) 01 = 1 supplémentaire (2 par microtrame) 10 = 2 supplémentaires (3 par microtrame) 11 = Réservée Les bits 15..13 sont réservés et doivent être réglés sur zéro. Voir le Chapitre 5 pour plus d'informations.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 9-13. Descripteur de point d'extrémité normalisé (suite)

Décalage	Champ	Taille	Valeur	Description
6	*bInterval*	1	Numéro	Intervalle pour le point d'extrémité d'interrogation des transferts de données. Exprimé en trames ou microtrames en fonction de la vitesse de fonctionnement de l'appareil (c'est-à-dire, soit 1 milliseconde, soit 125 µs unités) Pour les points d'extrémité isochrones à vitesse intégrale/haute vitesse, cette valeur doit se trouver dans l'intervalle compris entre 1 et 16. La valeur *bInterval* est utilisée comme exposant d'une valeur $2^{bInterval-1}$; par exemple, une valeur *bInterval* égale à 4 est équivalente à une période égale à 8 (2^{4-1}). Pour les points d'extrémité d'interruption à vitesse intégrale/faible vitesse, la valeur de ce champ peut varier de 1 à 255. Pour les points d'extrémité à haute vitesse, la valeur *bInterval* est utilisée comme exposant d'une valeur $2^{bInterval-1}$; par exemple, une valeur *bInterval* égale à 4 est équivalente à une période égale à 8 (2^{4-1}). Cette valeur doit être comprise entre 1 et 16. Pour les points d'extrémité OUT en bloc/de contrôle, la valeur *bInterval* doit spécifier le débit NAK maximum du point d'extrémité. Une valeur égale à zéro indique que le point d'extrémité ne répond jamais avec un NAK. D'autres valeurs indiquent au moins 1 NAK pour chaque nombre *bInterval* de microtrames. Cette valeur doit être comprise dans l'intervalle allant de 0 à 255. Voir la description des périodes au Chapitre 5 pour de plus amples détails.

Le champ *bmAttributes* fournit des informations concernant le type de transfert du point d'extrémité (bits 1..0) et le type de synchronisation (bits 3..2). De plus, les bits du type d'utilisation (bits 5..4) indiquent si c'est un point d'extrémité utilisé pour les transferts de données normaux (bits 5..4=00B), s'il est utilisé pour acheminer les informations de rétroaction explicite d'un ou plusieurs points d'extrémité de données (bits 5..4=01B) ou si c'est un point d'extrémité qui sert également de point de données implicite pour un ou plusieurs points d'extrémité (bits 5..4=10B). Les bits 5..2 ne sont significatifs que pour les points d'extrémité isochrones et doivent être réinitialisés à zéro pour tous les autres types de transfert.

Si le point d'extrémité est utilisé comme point d'extrémité de rétroaction explicite (bits 5..4=01B), le type de transfert doit être isochrone (bits1..0 = 01B) et le type de synchronisation doit être réglé sur Aucune synchronisation (bits 3..2=00B).

Un point d'extrémité de rétroaction (explicite ou implicite) doit être associé à un (ou plusieurs) point(s) d'extrémité de données isochrone(s) au(x)quel(s) il fournit un service de rétroaction. L'association repose sur la correspondance du numéro de point d'extrémité. Un point d'extrémité de rétroaction est toujours dirigé dans la direction opposée à celles du(des) point(s) d'extrémité au(x)quel(s) il offre des services. Si plusieurs points d'extrémité de données doivent être desservis par le même point d'extrémité de rétroaction, les points d'extrémité de données doivent être triés par ordre croissant, mais ils ne doivent pas nécessairement avoir des numéros de point d'extrémité consécutifs. Le premier point d'extrémité de données et le point d'extrémité de rétroaction doivent avoir le même numéro de point d'extrémité (et une direction opposée). Cela permet de s'assurer qu'un point d'extrémité de données peut identifier de façon unique son point d'extrémité de rétroaction en recherchant le premier point d'extrémité de rétroaction qui dispose d'un numéro de point d'extrémité inférieur ou égal à son propre numéro de point d'extrémité.

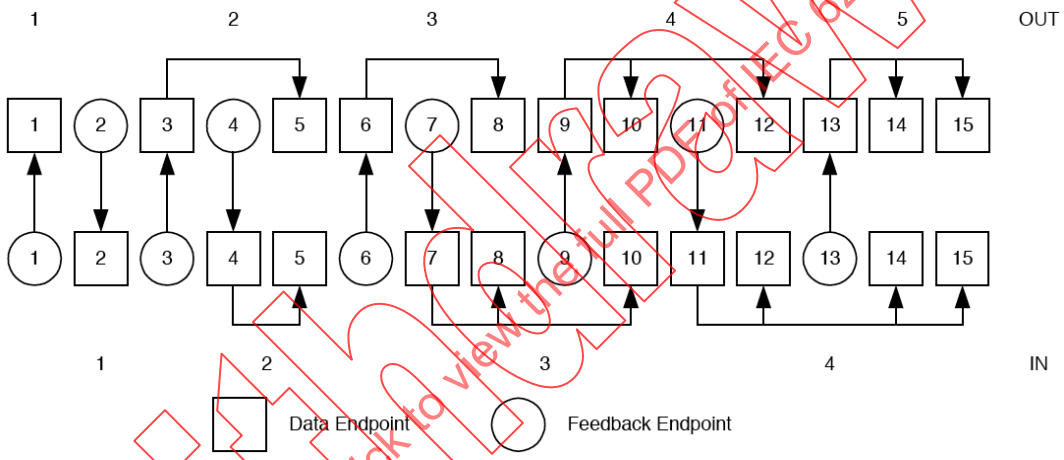
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Exemple: soit le cas extrême qui nécessite cinq groupes de points d'extrémité OUT isochrones et asynchrones et, en même temps, quatre groupes de points d'extrémité IN isochrones et adaptatifs. Chaque groupe nécessite un point d'extrémité de rétroaction et les groupes sont composés selon la Figure 9-7.

Groupe OUT	Nombre de points d'extrémité OUT	Groupe IN	Nombre de points d'extrémité IN
1	1	6	1
2	2	7	2
3	2	8	3
4	3	9	4
5	3		

Figure 9-7. Exemple de numéros de point d'extrémité de rétroaction

Les numéros de point d'extrémité peuvent être associés comme l'illustre la Figure 9-8.



Légende

Anglais	Français
Data Endpoint	Point d'extrémité de données
Feedback Endpoint	Point d'extrémité de rétroaction

Figure 9-8. Exemple de relations entre points d'extrémité de rétroaction

Les points d'extrémité isochrones et d'interruption à haute vitesse utilisent les bits 12..11 de la valeur *wMaxPacketSize* pour spécifier plusieurs transactions pour chaque microtrame spécifiée par la valeur *bInterval.* Si les bits 12..11 de *wMaxPacketSize* sont égaux à zéro, la taille maximum de paquet du point d'extrémité peut être toute valeur autorisée (tel que défini au Chapitre 5). Si les bits 12..11 du champ *wMaxPacketSize* ne sont pas égaux à zéro (0), les valeurs autorisées pour les bits 10..0 du champ *wMaxPacketSize* sont limités selon le Tableau 9-14.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 9-14. Valeurs wMaxPacketSize autorisées pour différents nombres de transactions par microtrame

Bits 12..11 de *wMaxPacketSize*	Valeur des bits 10..0 autorisées de *wMaxPacketSize*
00	1 - 1 024
01	513 - 1 024
10	683 - 1 024
11	S/O; réservé

Pour les points d'extrémité OUT en bloc/de contrôle, le champ *bInterval* est uniquement utilisé à des fins de conformité; le contrôleur de l'hôte ne doit pas changer son comportement en fonction de la valeur de ce champ.

9.6.7 Chaîne

Les descripteurs de chaîne de caractères sont optionnels. Comme énoncé antérieurement, si un appareil ne prend pas en charge les descripteurs de chaîne de caractères, toutes les références qui y font référence dans le cadre des descripteurs d'appareil, de configuration et d'interface doivent être réinitialisées à zéro.

Les descripteurs de chaîne de caractères utilisent les codages UNICODE tel que défini par *The Unicode Standard, Worldwide Character Encoding, Version 3.0*, The Unicode Consortium, Addison-Wesley Publishing Company, Reading, Massachusetts (URL: http://www.unicode.com). Les chaînes de caractères d'un appareil USB peuvent prendre en charge plusieurs langues. Lorsqu'un descripteur de chaîne de caractères est demandé, le demandeur spécifie la langue désirée à l'aide d'un ID de langue sur seize bits (LANGID) défini par l'USB-IF. La liste des LANGID USB actuellement définis peut être disponible à l'adresse http://www.usb.org/developers/docs.html. La chaîne de caractères d'indice zéro dans toutes les langues retourne un descripteur de chaîne de caractères qui contient un ensemble de codes LANGID sur deux octets pris en charge par l'appareil. Le Tableau 9-15 illustre l'ensemble de codes LANGID. Un appareil USB peut omettre tous les descripteurs de chaîne de caractères. Les appareils USB qui omettent tous les descripteurs de chaîne de caractères ne doivent pas retourner un ensemble de codes LANGID.

L'ensemble de codes LANGID ne se termine pas par NULL. La taille de l'ensemble (en octets) est calculée en soustrayant deux de la valeur du premier octet du descripteur.

Tableau 9-15. Descripteur de chaîne de caractères zéro, spécifiant les langues prises en charge par l'appareil

Décalage	Champ	Taille	Valeur	Description
0	*bLength*	1	N+2	Taille du descripteur en octets
1	*bDescriptorType*	1	Constant	Type de descripteur STRING
2	*wLANGID[0]*	2	Numéro	LANGID de code zéro
…	…	…	…	…
N	*wLANGID[x]*	2	Numéro	LANGID de code *x*

Le descripteur de chaîne de caractères UNICODE (décrit dans le Tableau 9-16) ne se termine pas par la valeur NULL. La longueur de la chaîne de caractères est calculée en soustrayant deux de la valeur du premier octet du descripteur.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 9-16. Descripteur de chaîne de caractères UNICODE

Décalage	Champ	Taille	Valeur	Description
0	*bLength*	1	Numéro	Taille du descripteur en octets
1	*bDescriptorType*	1	Constant	Type de descripteur STRING
2	*bString*	N	Numéro	Chaîne de caractères codée en Unicode

9.7 Définitions de classe d'appareil

Tous les appareils doivent prendre en charge les requêtes et les définitions de descripteur décrites dans le présent chapitre. La plupart des appareils fournissent des requêtes supplémentaires et, si possible, des descripteurs pour les extensions spécifiques à l'appareil. En outre, les appareils peuvent proposer des services étendus qui sont communs à un groupe d'appareils. Afin de définir une classe d'appareils, les informations suivantes doivent être fournies pour définir de façon complète l'apparence et le comportement de la classe d'appareil.

9.7.1 Descripteurs

Si la classe nécessite une définition spécifique des descripteurs normalisés, la définition de la classe doit inclure ces exigences dans la définition de la classe. De plus, si la classe définit un ensemble étendu de descripteurs normalisés, ces derniers doivent être entièrement définis dans la définition de la classe. Les définitions de descripteur étendues doivent respecter l'approche utilisée pour les descripteurs normalisés; par exemple, tous les descripteurs doivent commencer par un champ de longueur.

9.7.2 Interface(s) et utilisation des points d'extrémité

Si une classe d'appareils est normalisée, les interfaces utilisées par les appareils, y compris la manière dont les points d'extrémité sont utilisés, doivent être incluses dans la définition de la classe d'appareil. Les appareils peuvent étendre une définition de classe avec des caractéristiques propriétaires tant que celles-ci satisfont à la définition de base de la classe.

9.7.3 Requêtes

Toutes les requêtes spécifiques à la classe doivent être définies.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

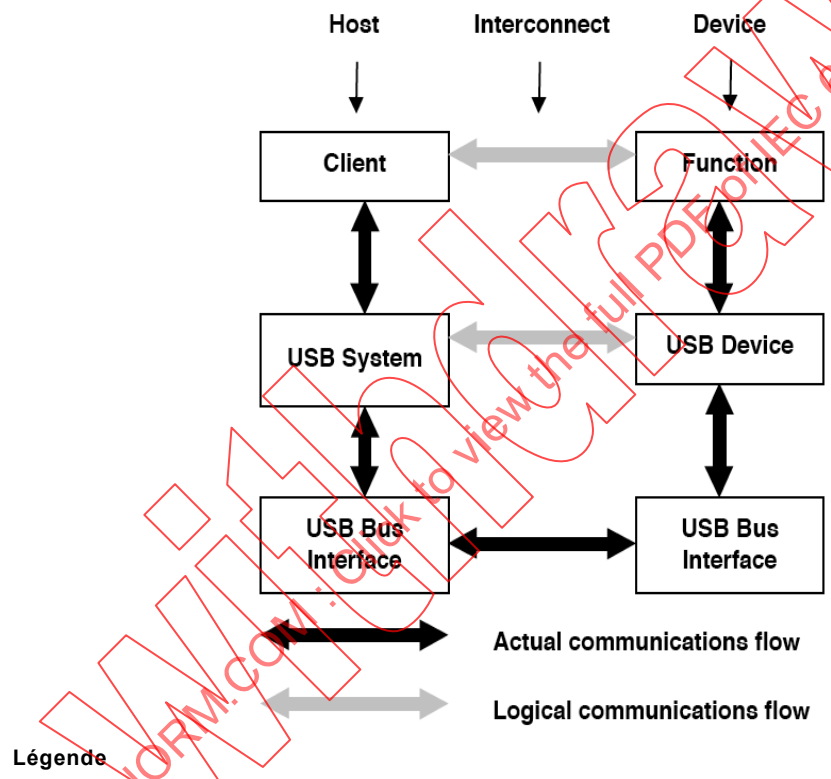
10 Chapitre 10
Hôte USB: Matériel et logiciel

L'interconnexion USB prend en charge le trafic des données entre un hôte et un appareil USB. Ce chapitre décrit les interfaces hôtes nécessaires pour faciliter la communication USB entre un client logiciel, résidant sur l'hôte, et une fonction mise en œuvre sur un appareil. La mise en œuvre décrite dans ce chapitre n'est pas requise. Cette mise en œuvre est fournie à titre d'exemple pour illustrer le comportement du système hôte attendu par un appareil USB. Un système hôte peut fournir une mise en œuvre logicielle hôte différente tant qu'un appareil USB connaît le même comportement hôte.

10.1 Présentation de l'hôte USB

10.1.1 Présentation

L'organigramme et les interrelations de base du modèle de communication USB sont illustrés dans la Figure 10-1.



Légende

Anglais	Français
Host	Hôte
Interconnect	Interconnexion
Device	Appareil
Client	Client
Function	Fonction
USB System	Système USB
USB Device	Appareil USB
USB Bus Interface	Interface bus USB
Actual communications flow	Flux de communication réel
Logical communications flow	Flux de communication logique

Figure 10-1. Modèle de communication inter-couche

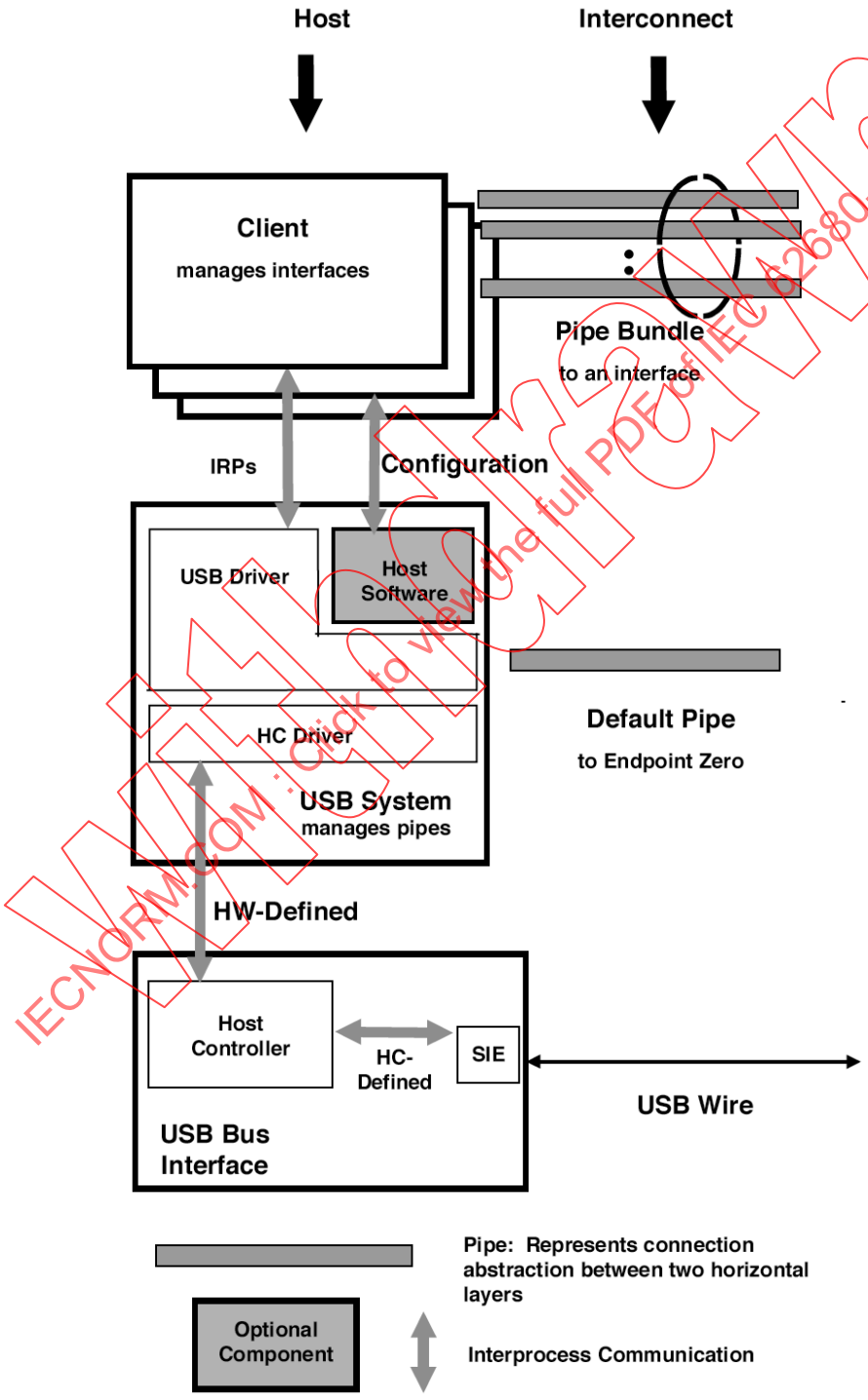
L'hôte et l'appareil sont divisés en couches distinctes illustrées dans la Figure 10-1. Les flèches verticales indiquent la communication actuelle sur l'hôte. Les interfaces correspondantes sur l'appareil sont spécifiques à la mise en œuvre. Toutes les communications entre l'hôte et l'appareil ont finalement lieu sur le câble USB physique. Il existe cependant des interfaces hôte-appareil logiques entre chaque couche horizontale. Ces

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

communications, entre le logiciel client résidant sur l'hôte et la fonction fournie par l'appareil, sont caractérisées par un contrat basé sur les besoins de l'application utilisant actuellement l'appareil et les fonctionnalités fournies par l'appareil.

Cette interaction client-fonction crée les exigences pour toutes les couches sous-jacentes et leurs interfaces.

Ce chapitre décrit ce modèle du point de vue de l'hôte et de ses couches. La Figure 10-2 illustre, en fonction de la vue générale présentée au Chapitre 5, le point de vue de l'hôte de sa communication avec l'appareil.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host	Hôte
Interconnect	Interconnexion
Client manages interfaces	Le client gère les interfaces
Pipe Bundle to an interface	Faisceau de canaux vers une interface
IRPs	IRP
Configuration	Configuration
USB Driver	Pilote USB
Host Software	Logiciel hôte
HC Driver	Pilote HC
USB System manages pipes	Le système USB gère les canaux
Default Pipe To Endpoint Zero	Canal par défaut vers point d'extrémité zéro
HW-Defined	Défini au niveau matériel
Host Controller	Contrôleur hôte
HC-Defined	Défini au niveau HC
SIE	SIE
USB Bus Interface	Interface bus USB
USB Wire	Câble USB
Optional Component	Composant en option
Pipe: Represents connection abstraction between two horizontal layers	Canal: Représente l'abstraction de connexion entre deux couches horizontales
Interprocess Communication	Communication inter-processus

Figure 10-2. Communications avec l'hôte

Il y a un seul hôte pour chaque USB. Les principales couches d'un hôte incluent:

- Interface bus USB
- Système USB
- Client

L'interface bus USB traite les interactions pour les couches électriques et protocoles (voir Chapitre 7 et Chapitre 8). D'un point de vue de l'interconnexion, une interface bus USB similaire est fournie par l'appareil USB et l'hôte, comme illustré à titre d'exemple par le moteur d'interface série (SIE). Sur l'hôte, cependant, l'interface bus USB a des responsabilités supplémentaires en raison du rôle unique de l'hôte sur l'USB et est mise en œuvre comme contrôleur hôte. Le contrôleur hôte a un hub racine intégré fournissant des points de connexion avec le câble USB.

Le système USB utilise le contrôleur hôte pour gérer les transferts de données entre l'hôte et les appareils USB. L'interface entre le système USB et le contrôleur hôte dépend de la définition matérielle du contrôleur hôte. Le système USB, en relation avec le contrôleur hôte, effectue la traduction entre le point de vue du client des transferts de données et les transactions USB apparaissant sur l'interconnexion. Cela inclut l'ajout d'une prise en charge de la caractéristique USB comme les enveloppeurs de protocoles. Le système USB est également responsable de la gestion des ressources USB, comme la bande passante et la puissance de bus, de sorte que l'accès client à l'USB est possible.

Le système USB a trois composants de base:

- Pilote de contrôleur hôte
- Pilote USB
- Logiciel hôte

Le pilote du contrôleur hôte (HCD) existe pour mapper plus facilement les diverses mises en œuvre de contrôleur hôte dans le système USB, de sorte qu'un client peut interagir avec son appareil sans savoir à quel contrôleur hôte l'appareil est connecté. Le pilote USB (USBD) assure l'interface hôte de base (USBDI) pour les clients reliés aux appareils USB. L'interface entre le HCD et l'USBD est connue sous le nom d'interface de pilote de contrôleur hôte (HCDI). Cette interface n'est jamais directement disponible pour les clients et n'est ainsi pas définie par la spécification USB. Une HCDI particulière est cependant définie par chaque système d'exploitation qui prend en charge diverses mises en œuvre de contrôleur hôte.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

L'USBD assure les mécanismes de transfert de données sous la forme de paquets d'appel E/S (IRP), qui comprend un appel vers les données de transport via un canal spécifique. Outre la fourniture de mécanismes de transfert de données, l'USBD est responsable de la présentation à ses clients d'une abstraction d'un appareil USB qui peut être manipulé pour la configuration et la gestion d'états. Dans le cadre de cette abstraction, l'USBD possède le canal par défaut (voir Chapitre 5 et Chapitre 9) via lequel tous les appareils USB sont accessibles à des fins de contrôle USB normalisé. Ce canal par défaut représente une communication logique entre l'USBD et l'abstraction d'un appareil USB comme illustré dans la Figure 10-2.

Dans certains systèmes d'exploitation, un logiciel système non-USB supplémentaire est disponible et fournit les mécanismes de configuration et de chargement aux pilotes d'appareil. Dans certains systèmes d'exploitation, le pilote d'appareil doit utiliser les interfaces fournies au lieu d'accéder directement aux mécanismes USBDI.

La couche client décrit toutes les entités logicielles qui sont chargées d'interagir directement avec les appareils USB. Lorsque chaque appareil est relié au système, ces clients peuvent interagir directement avec le matériel périphérique. Les caractéristiques partagées de l'USB placent le logiciel système USB entre le client et son appareil; à savoir, un client ne peut pas accéder directement au matériel de l'appareil.

Partout, les couches hôtes proposent les fonctionnalités suivantes:

- Détection du raccordement et de la déconnexion des appareils USB
- Gestion du flux de contrôle normalisé USB entre l'hôte et les appareils USB
- Gestion du flux de données entre l'hôte et les appareils USB
- Collecte des statistiques d'état et d'activité
- Contrôle de l'interface électrique entre le contrôleur hôte et les appareils USB, notamment la fourniture d'une quantité limitée d'alimentation

Les sections suivantes décrivent ces responsabilités et les exigences vis-à-vis de l'USBDI en plus amples détails. Les interfaces actuelles utilisées pour une combinaison spécifique de plate-forme hôte et de système d'exploitation sont décrites dans le guide d'environnement du système d'exploitation approprié..

Tous les hubs (voir Chapitre 11) signalent les changements d'état interne et leur état de changement de port via le canal de changement d'état. Cela inclut une notification du moment où un appareil USB est branché ou déconnecté d'un de leurs ports. Un client USBD généralement connu comme pilote de hub reçoit ces notifications en tant que propriétaire du canal de changement d'état du hub. Pour les branchements d'appareils, le pilote de hub initie alors le processus de configuration d'appareils. Dans certains systèmes, ce pilote de hub fait partie du logiciel hôte fourni par le système d'exploitation pour gérer les appareils.

10.1.2 Mécanismes de contrôle

Les informations de contrôle peuvent être transmises entre l'hôte et un appareil USB à l'aide de la signalisation dans la bande et hors bande. La signalisation dans la bande mélange les informations de contrôle avec les données dans un canal en dehors de la perception de l'hôte. La signalisation hors bande place les informations de contrôle dans un canal séparé.

Il y a un canal de message appelé canal par défaut pour chaque appareil USB relié. Cette association logique entre un hôte et un appareil USB est utilisée pour le flux de contrôle normalisé USB comme l'énumération et la configuration de appareils. Le canal par défaut fournit une interface normalisée avec tous les appareils USB. Le canal par défaut peut également être utilisé pour les communications spécifiques aux appareils, comme transmises par l'USBD, qui possèdent les canaux par défaut de tous les appareils USB.

Un appareil USB particulier peut autoriser l'utilisation de canaux de message supplémentaires pour transférer les informations de contrôle spécifiques aux appareils. Ces canaux utilisent le

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

même protocole de communication que le canal par défaut, mais les informations transférées sont spécifiques à l'appareil USB et ne sont pas normalisées par la spécification USB.

L'USBD prend en charge le partage du canal par défaut, qu'il possède et utilise avec ses clients. Il fournit également l'accès aux autres canaux de contrôle associés à l'appareil.

10.1.3 Flux de données

Le contrôleur hôte est responsable du transfert des flux de données entre l'hôte et les appareils USB. Ces transferts de données sont traités sous forme de flux continu d'octets. L'USB prend en charge quatre types de base de transferts de données:

- Transferts de contrôle
- Transferts isochrones
- Transferts d'interruption
- Transferts en masse

Pour des informations supplémentaires sur les types de transfert, voir Chapitre 5.

Chaque appareil présente une ou plusieurs interfaces qu'un client peut utiliser pour communiquer avec l'appareil. Chaque interface est composée de zéro canal ou de plusieurs canaux qui transfèrent individuellement les données entre le client et un point d'extrémité particulier sur l'appareil. L'USBD établit les interfaces et canaux à la demande explicite du logiciel hôte. Le contrôleur hôte assure le service à partir des paramètres fournis par le logiciel hôte lorsque la demande de configuration est émise.

Un canal a plusieurs caractéristiques basées sur les exigences de livraison des données à transférer. Exemples de caractéristiques:

- Le débit auquel les données doivent être transférées
- Si les données sont fournies à un débit constant ou de manière sporadique
- Temps de retard des données avant livraison qui peut être appliqué
- Si la perte de données transférées est catastrophique

Un point d'extrémité d'appareil USB décrit les caractéristiques requises pour un canal spécifique. Les points d'extrémité sont décrits dans le cadre des informations de caractérisation de l'appareil USB. Pour des détails supplémentaires, voir Chapitre 9.

10.1.4 Collecte des statistiques d'état et d'activité

En tant que communicant commun pour tous les transferts de contrôle et de données entre l'hôte et les appareils USB, le système USB et le contrôleur hôte sont bien positionnés pour suivre les informations d'état et d'activité. Ces informations sont fournies sur demande au logiciel hôte, permettant au logiciel de gérer les informations relatives aux états et activités. Cette spécification n'identifie pas les informations spécifiques qu'il convient de suivre ou nécessite un format particulier pour rendre compte des informations relatives aux activités et états.

10.1.5 Considérations vis-à-vis de l'interface électrique

L'hôte alimente les appareils USB reliés au hub racine. La quantité de puissance fournie par un port est spécifiée au Chapitre 7.

10.2 Exigences vis-à-vis du contrôleur hôte

Dans toutes les mises en œuvre, les contrôleurs hôtes exécutent les mêmes tâches de base eu égard à l'USB et aux appareils reliés. Ces tâches de base sont décrites ci-après.

Le contrôleur hôte a des exigences de l'hôte et de l'USB. Voici une brève présentation de la fonctionnalité fournie. Chaque fonctionnalité est traitée en détail dans les sections suivantes.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Traitement des états	En tant que composant de l'hôte, le contrôleur hôte signale et gère ses états.
Sérialiseur/désérialiseur	Pour les données transmises par l'hôte, le contrôleur hôte convertit les informations sur les protocoles et données de son format natif en un train de bits transmis sur l'USB. Pour les données reçues dans l'hôte, l'opération inverse est effectuée.
Génération de (micro)trames	Le contrôleur hôte produit des jetons SOF à une période de 1 ms lorsqu'il fonctionne avec des appareils à vitesse intégrale, et à une période de 125 µs lorsqu'il fonctionne avec des appareils à grande vitesse.
Traitement des données	Le contrôleur hôte traite les demandes de transmission de données vers et de l'hôte.
Moteur de protocole	Le contrôleur hôte prend en charge le protocole spécifié par l'USB.
Traitement des erreurs de transmission	Tous les contrôleurs hôtes montrent le même comportement lors de la détection et de la réaction aux catégories d'erreurs définies.
Réveil à distance	Tous les contrôleurs hôtes doivent avoir la capacité de placer le bus dans l'état Suspendu et de répondre aux évènements de réveil de bus.
Hub racine	Le hub racine fournit la fonction de hub normalisé pour relier le contrôleur hôte à un ou plusieurs ports USB.
Interface du système hôte	Fournit un chemin de données à grande vitesse entre le contrôleur hôte et le système hôte.

Les sections suivantes présentent une discussion plus détaillée des fonctionnalités requises du contrôleur hôte.

10.2.1 Traitement des états

Le contrôleur hôte a une série d'états que le système USB gère. De plus, le contrôleur hôte fournit l'interface avec les deux domaines suivants d'état relevant de l'USB:

- Propagation du changement d'état
- Hub racine

Le hub racine présente au pilote de hub les mêmes états normalisés que les autres appareils USB. Le contrôleur hôte prend en charge ces états et leurs transitions pour le hub. Pour plus de détails sur les états USB, notamment leurs interrelations et transitions, voir Chapitre 9.

L'état général du contrôleur hôte est inextricablement lié à celui du hub racine et de l'USB général. Tout changement d'état du contrôleur hôte qui est visible pour les appareils reliés doit être reflété dans les informations de changement d'état de l'appareil correspondant de manière à ce que les états du contrôleur hôte et de l'appareil qui en résultent soient consistants.

Les appareils USB demandent un réveil via l'utilisation de la signalisation de reprise (voir Chapitre 7). Le contrôleur hôte doit informer le reste de l'hôte d'un évènement de reprise via un ou des mécanismes spécifiques à cette mise en œuvre du système. Le contrôleur hôte lui-même peut générer un évènement de reprise via la même méthode de signalisation.

10.2.2 Sérialiseur/désérialiseur

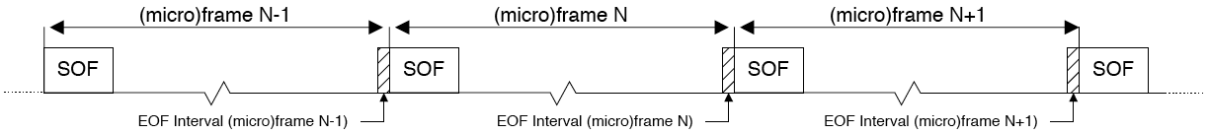
La transmission réelle des données dans l'USB physique a lieu sous forme de train de bits en série. Un moteur d'interface série (SIE), si mis en œuvre dans le cadre de l'hôte ou d'un appareil USB, traite la sérialisation et la désérialisation des transmissions USB. Sur l'hôte, ce SIE fait partie du contrôleur hôte.

10.2.3 Génération de trames et de microtrames

Il revient au contrôleur hôte de diviser le temps USB en quantités appelées "trames" lors du fonctionnement avec des appareils à vitesse intégrale, et "microtrames" lors du fonctionnement avec des appareils à grande vitesse. Les trames et microtrames sont créées par le contrôleur hôte via l'émission de jetons de début de trame (SOF) comme illustré dans la Figure 10-3. Le jeton SOF est la première transmission dans la période (micro)trame. Les contrôleurs hôtes fonctionnant avec des appareils à grande vitesse génèrent des jetons SOF à intervalles de 125 µs. Les contrôleurs hôtes fonctionnant avec des appareils à vitesse intégrale génèrent des jetons SOF à intervalles de 1,00 µs. Après l'émission d'un jeton SOF, le contrôleur hôte est libre de transmettre d'autres transactions pour le reste de la période de micro(trame). Lorsque le contrôleur hôte est dans son état de fonctionnement normal, les

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

jetons SOF doivent être générés en continu à un débit périodique approprié, sans tenir compte de l'activité d'un autre bus ou du manque de celle-ci. Si le contrôleur hôte entre dans un état où il ne fournit pas d'alimentation sur le bus, il ne doit pas générer de SOF. Lorsque le contrôleur hôte ne génère pas de SOF, il peut entrer dans un état à puissance réduite.



Légende

Anglais	Français
(micro)frame N-1	(micro)trame N-1
(micro)frame N	(micro)trame N
(micro)frame N+1	(micro)trame N+1
EOF Interval (micro)frame N-1	Intervalle EOF (micro)trame N-1
EOF Interval (micro)frame N	Intervalle EOF (micro)trame N
EOF Interval (micro)frame N+1	Intervalle EOF (micro)trame N+1
SOF	SOF

Figure 10-3. Création de trames et de microtrames

Le jeton SOF maintient l'accès avec la plus grande priorité au bus. Le circuit de babillage dans les hubs isole électriquement les transmetteurs actifs pendant l'intervalle Fin de microtrame ou Fin de trame (EOF), fournissant un bus ralenti pour la transmission SOF.

Le contrôleur hôte maintient le nombre de (micro)trames actuel pouvant être lues par le système hôte.

Ce qui suit s'applique au nombre de (micro)trames actuel maintenu par l'hôte:

- Utilisé pour identifier de manière unique les (micro)trames entre elles
- Incrémenté à la fin de chaque période de (micro)trame
- Valide via la (micro)trame suivante

Les contrôleurs hôtes fonctionnant avec des appareils à vitesse intégrale maintiennent un nombre de trames actuel (au moins 11 bits) qui incrémente à une période de 1 ms. L'hôte transmet les 11 bits inférieurs du numéro de trames actuel dans chaque transmission de jeton SOF.

Les contrôleurs hôtes fonctionnant avec des appareils à grande vitesse maintiennent un numéro de microtrames actuel (au moins 14 bits) qui incrémente à une période de 125 µs. L'hôte transmet les bits 3 à 13 du numéro de microtrame actuel dans chaque transmission de jeton SOF. Cela entraîne la transmission de la même valeur de paquet SOF pour huit microtrames consécutives avant l'incrémentation de la valeur de paquet SOF.

En cas de demande du contrôleur hôte, le numéro de (micro)trames actuel est le numéro de (micro)trames existant au moment où la demande a été remplie. Le numéro de (micro)trames actuel tel que retourné par l'hôte (contrôleur hôte ou HCD) est d'au moins 32 bits, bien que le contrôleur hôte lui-même ne soit pas tenu de maintenir plus de 11 bits lors du fonctionnement avec des appareils à vitesse intégrale ou de 14 bits lors du fonctionnement avec des appareils à grande vitesse.

Le contrôleur hôte doit cesser la transmission pendant l'intervalle EOF. Si l'intervalle EOF commence, toutes les transactions planifiées spécifiquement pour la (micro)trame qui vient de passer sont abandonnées. Si le contrôleur hôte exécute une transaction au moment où l'intervalle EOF est rencontré, le contrôleur hôte termine la transaction.

10.2.4 Traitement des données

Le contrôleur hôte est responsable de la réception des données du système USB et de leur envoi à l'USB et de la réception des données de l'USB et de leur envoi au système USB. Le

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

format particulier utilisé pour les communications de données entre le système USB et le contrôleur hôte est spécifique à la mise en œuvre, dans les règles du comportement de transfert décrites au Chapitre 5.

10.2.5 Moteur de protocole

Le contrôleur hôte gère l'interface au niveau du protocole USB. Il insère les informations appropriées concernant le protocole pour les transmissions sortantes. Il extrait et interprète, si approprié, les informations entrantes sur le protocole.

10.2.6 Traitement des erreurs de transmission

Le contrôleur hôte doit être capable de détecter les conditions d'erreur de transmission suivantes, qui sont définies du point de vue de l'hôte:

- Conditions de temporisation après un jeton ou paquet transmis par l'hôte. Ces erreurs surviennent lorsque le point d'extrémité adressé ne répond pas ou lorsque la structure de la transmission est tellement endommagée que le point d'extrémité cible ne la reconnaît pas.
- Erreurs de données résultant dans des transmissions manquantes ou invalides:
  - Le contrôleur hôte est incapable d'envoyer ou de recevoir complètement un paquet pour des raisons spécifiques à l'hôte, par exemple, une transmission s'étendant au-delà de l'EOF ou un manque de ressources disponibles pour le contrôleur hôte.
  - Un champ CRC invalide sur un paquet de données reçu.
- Erreurs de protocole:
  - Un PID d'établissement de liaison invalide, comme un établissement de liaison mal formé ou inapproprié
  - Un EOP faux
  - Une erreur de bourrage de bits

Pour chaque transaction en masse, de contrôle et d'interruption, l'hôte doit maintenir un compteur-enregistreur d'erreurs. Erreurs résultant des conditions décrites ci-dessus, et non d'un point d'extrémité recevant une demande. Cette valeur reflète le nombre de fois que la transaction a rencontré une erreur de transmission. Il convient que l'erreur ne puisse pas être incrémentée lorsqu'il y avait une erreur due à des raisons spécifiques à l'hôte (sous-utilisation ou sur-utilisation du tampon), et quand une transaction ne rencontre pas une erreur de transmission, le décompte des erreurs est remis à zéro.

Si le décompte d'erreurs pour une transaction donnée atteint trois, l'hôte retire le transfert. Lorsqu'un transfert est retiré pour des raisons excessives, le dernier type d'erreur doit être indiqué. Les transactions asynchrones sont essayées une seule fois, quel que soit le résultat, et cependant, aucun décompte d'erreurs n'est maintenu pour ce type.

10.2.7 Réveil à distance

Si le système USB souhaite placer le bus dans l'état Suspendu, il commande le contrôleur hôte d'arrêter tout le trafic de bus, notamment les SOF. Cela fait que tous les appareils USB entrent dans l'état Suspendu. Dans cet état, le système USB peut permettre au contrôleur hôte de répondre aux évènements de réveil de bus. Cela permet au contrôleur hôte de répondre à la signalisation de réveil de bus pour redémarrer le système hôte.

10.2.8 Hub racine

Le hub racine fournit la connexion entre le contrôleur hôte et un ou plusieurs ports USB. Le hub racine fournit la même fonctionnalité pour traiter la topologie USB que les autres hubs (voir Chapitre 11), sauf que l'interface matérielle et logicielle entre le hub racine et le contrôleur hôte est définie par la mise en œuvre matérielle spécifique.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.2.8.1 Réinitialisations de port

La Section 7.1.7.5 décrit les exigences d'un hub pour garantir que toutes les tentatives de reprise en amont sont suralimentées avec une longue réinitialisation en aval. Les hubs racines doivent fournir une période de réinitialisation d'agrégat d'au moins 50 ms. Si la durée de la réinitialisation est contrôlée dans le matériel et le minuteur du matériel est <50 ms, le système USB peut émettre plusieurs réinitialisations consécutives pour accumuler la durée de réinitialisation spécifiée comme décrit dans la Section 7.1.7.5.

10.2.9 Interface du système hôte

Le contrôleur hôte fournit une interface de maîtrise de bus à grande vitesse vers et de la mémoire système principale. Le transfert physique entre la mémoire et le câble USB est effectué automatiquement par le contrôleur hôte. Lorsque les tampons de données doivent être remplis ou vidés, le contrôleur hôte informe le système USB.

10.3 Présentation des mécanismes logiciels

Le HCD et l'USBD présentent les interfaces logicielles en fonction de différents niveaux d'abstraction. Mais, il est prévu qu'ils fonctionnent ensemble d'une manière spécifiée pour répondre aux exigences générales du système USB (voir Figure 10-2). Les exigences pour le système USB sont principalement exprimées sous forme d'exigences pour l'USBDI. La division des tâches entre l'USBD et le HCD n'est pas définie. Mais, la seule exigence du HCDI qui doit être satisfaite est de prendre en charge, dans le contexte du système d'exploitation spécifié, de multiples mises en œuvre de contrôleur hôte.

Le HCD fournit une abstraction du contrôleur hôte et une abstraction du point de vue du transfert de données du contrôleur hôte dans l'USB. L'USBD fournit une abstraction de l'appareil USB et des transferts de données entre le client de l'USBD et la fonction sur l'appareil USB. Globalement, le système USB agit comme un facilitateur pour la transmission de données entre le client et la fonction et comme point de contrôle pour les interfaces spécifiques à l'USB de l'appareil USB. Dans le cadre de la facilitation du transfert de données, le système USB fournit des fonctionnalités de gestion de tampons et permet la synchronisation de la transmission de données aux besoins du client et de la fonction.

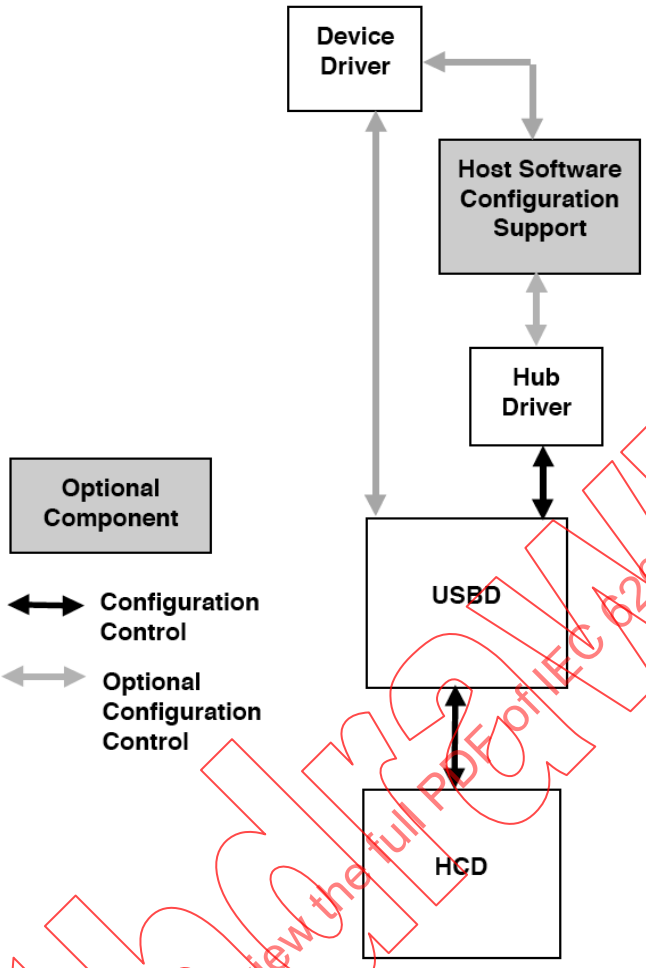
Les exigences spécifiques pour l'USBDI sont décrites ultérieurement dans ce chapitre. Les fonctions exactes qui remplissent ces exigences sont décrites dans le guide de l'environnement du système d'exploitation correspondant pour le HCDI et l'USBDI. Les procédures impliquées dans la réalisation de transferts de données via l'USBDI sont décrites dans les sections suivantes.

10.3.1 Configuration des appareils

Différents environnements de système d'exploitation effectuent la configuration des appareils à l'aide de différents composants logiciels et de différentes séquences d'évènements. Le système USB ne suppose pas une méthode de système d'exploitation spécifique. Mais, il existe certaines exigences de base à remplir par n'importe quelle mise en œuvre du système USB. Dans certains systèmes d'exploitation, le logiciel hôte existant fournit ces exigences. Dans d'autres, le système USB fournit les fonctionnalités.

Le système USB suppose un client spécialisé de l'USBD, appelé pilote de hub, qui agit comme centre d'échanges pour l'ajout et la suppression de appareils d'un hub particulier. Lorsque le pilote de hub reçoit ces notifications, il utilise un logiciel hôte supplémentaire et d'autres clients USBD, d'une manière spécifique au système d'exploitation, pour reconnaître et configurer l'appareil. Ce modèle, illustré dans la Figure 10-4, est la base de la discussion suivante.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Device Driver	Pilote d'appareil
Host Software Configuration Support	Prise en charge de la configuration du logiciel hôte
Hub driver	Pilote de hub
Optional Component	Composant en option
Configuration Control	Contrôle de configuration
Optional Configuration Control	Contrôle de configuration en option

Figure 10-4. Interactions des configurations

Lorsqu'un appareil est relié, le pilote de hub reçoit une notification du hub détectant le changement. Le pilote de hub, utilisant les informations fournies par le hub, demande un identifiant d'appareil à l'USBD. L'USBD, à son tour, configure le canal par défaut pour cet appareil et retourne un identifiant d'appareil au pilote de hub.

L'appareil est maintenant prêt à être configuré pour l'utilisation. Pour chaque appareil, trois configurations doivent être terminées avant que l'appareil ne soit prêt pour l'utilisation:

1. Configuration des appareils: Cela inclut la configuration de tous les paramètres USB de l'appareil et l'allocation de toutes les ressources hôtes USB qui sont visibles pour l'appareil. Cela est accompli en configurant la valeur de configuration sur l'appareil. Un ensemble limité de changements de configuration, comme les réglages alternatifs, est admis sans reconfigurer totalement l'appareil. Lorsque l'appareil est configuré, il est, de son point de vue, prêt pour l'utilisation.
2. Configuration de l'USB: Afin de créer réellement un canal USBD prêt pour utilisation par un client, des informations USB supplémentaires, non visibles pour l'appareil, doivent être spécifiées par le client. Ces informations, connues comme la politique du canal, décrivent

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

comment le client utilise le canal. Cela inclut des éléments tels que la quantité maximum de données que le client va transférer avec un IRP, l'intervalle de service maximum que le client va utiliser, l'identification de notification du client, etc.

3. Configuration des fonctions: Une fois que les types de configuration 1 et 2 ont été réalisés, le canal est complètement prêt pour utilisation d'un point de vue de l'USB. De plus, la configuration supplémentaire spécifique au fournisseur ou à la classe peut être requise avant que le client ne puisse réellement utiliser le canal. Cette configuration est une affaire privée entre l'appareil et le client et n'est pas normalisée par l'USBD.

Les alinéas suivants décrivent les exigences de configuration de l'appareil et de l'USB.

Le logiciel de configuration responsable effectue la configuration réelle des appareils. Selon la mise en œuvre particulière du système d'exploitation, le logiciel responsable de la configuration peut inclure ce qui suit:

- Le pilote de hub
- D'autres logiciels hôtes
- Un pilote d'appareil

Le logiciel de configuration lit d'abord le descripteur d'appareil, puis demande la description pour chaque configuration possible. Il peut utiliser les informations fournies pour charger un client particulier, comme un pilote d'appareil, qui interagit initialement avec l'appareil. Le logiciel de configuration, peut-être avec entrée de ce pilote d'appareil, choisit une configuration pour l'appareil. Le paramétrage de la configuration de l'appareil configure tous les points d'extrémité sur l'appareil et retourne une collection d'interfaces utilisées pour le transfert de données par les clients USBD. Chaque interface est une collection de canaux détenus par un seul client.

Cette configuration initiale utilise les réglages par défaut pour les interfaces et la bande passante par défaut pour chaque point d'extrémité. Une mise en œuvre USBD peut en outre autoriser le client à spécifier des interfaces alternatives lors de la sélection de la configuration initiale. Le système USB vérifie que les ressources requises pour la prise en charge du point d'extrémité sont disponibles et, si tel est le cas, alloue la bande passante requise. Voir Section 10.3.2 pour une discussion de la gestion des ressources.

L'appareil est maintenant configuré, mais les canaux créés ne sont pas prêts à être utilisés. La configuration USB est réalisée lorsque le client initialise chaque canal en définissant une politique pour spécifier la manière dont il interagit avec le canal. Parmi les informations spécifiées, on retrouver l'intervalle de service maximum du client et les informations de notification. Parmi les actions prises par le système USB, suite à la définition de la politique, on retrouve la détermination de la quantité d'espace de travail tampon requis en plus de l'espace de tampon de données fourni par le client. La taille des tampons requise est basée sur l'utilisation choisie par le client et les besoins de transfert du système USB.

Le client reçoit des notifications lorsque les IRP sont complets, avec succès ou en raison d'erreurs. Le client peut également se réveiller indépendamment de la notification USB pour vérifier l'état des IRP en cours.

Le client peut également choisir d'effectuer des modifications de configuration, comme permettre un réglage alternatif pour une interface ou changer la bande passante allouée à un canal particulier. Pour effectuer ces changements, l'interface ou le canal, respectivement, doit être inactif.

10.3.2 Gestion des ressources

Si un canal est configuré par l'USBD pour un point d'extrémité donné, le système USB doit déterminer s'il peut prendre en charge le canal. Le système USB effectue cette détermination en fonction des exigences définies dans le descripteur de point d'extrémité. Une des exigences de point d'extrémité, qui doit être prise en charge pour créer un canal pour un point d'extrémité, est la bande passante nécessaire pour les transferts de point d'extrémité. Il y a

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

deux étapes pour vérifier la bande passante disponible. Le temps d'exécution maximum pour une transaction est d'abord calculé. Le planning de (micro)trame est consulté pour déterminer si la transaction indiquée convient.

L'allocation de la bande passante garantie pour les canaux isochrones et d'interruption, et la détermination du caractère approprié ou non d'un contrôle particulier ou d'une transaction en masse dans une (micro)trame donnée, peuvent être déterminés par un logiciel heuristique dans le système USB. Si le temps d'exécution de la transaction réel dans le contrôleur hôte dépasse la valeur déterminée heuristiquement, le contrôleur hôte est chargé de garantir que l'intégrité des (micro)trames est maintenue (voir Section 10.2.3). La discussion suivante décrit les exigences pour le système USB heuristique.

Pour déterminer si la bande passante peut être allouée, ou si une transaction peut entrer dans une (micro)trame particulière, le temps d'exécution de la transaction maximum doit être calculé. Le calcul du temps d'exécution maximum nécessite que les informations suivantes soient fournies (noter qu'un agent autre que le client peut fournir certaines de ces informations):

- Nombre d'octets de données (*wMaxPacketSize*) à transmettre.
- Type de transfert.
- Profondeur dans la topologie. Si moins de précision est autorisée, la profondeur de topologie de maximum peut être supposée.

Ce calcul doit inclure le temps de transmission de bits, le retard de propagation du signal via la topologie, et les retards spécifiques à la mise en œuvre, comme le temps de préparation ou de récupération requis par le contrôleur hôte lui-même. Voir Chapitre 5 pour les exemples de formules qui peuvent être utilisées pour ces calculs.

10.3.3 Transferts de données

La base pour toute la communication client-fonction est l'interface: un faisceau de canaux associés avec un appareil USB particulier.

Exactement un client sur l'hôte gère une interface donnée. Le client initialise chaque canal d'une interface en définissant la politique pour ce canal. Cela inclut la quantité maximum de données à transmettre par IRP et l'intervalle de service maximum pour le canal. Un intervalle de service est exprimé en millisecondes et décrit l'intervalle pendant lequel les données de l'IRP sont transmises pour un canal isochrone. Il décrit l'intervalle de consultation pour un canal d'interruption. Le client est notifié lorsqu'une demande spécifiée est complétée. Le client gère la taille de chaque IRP de sorte que son cycle de service et ses contraintes de latence sont maintenus. Les informations sur la politique supplémentaires incluent les informations de notification pour le client.

Le client fournit l'espace tampon requis pour maintenir les données transmises. Le système USB utilise la politique pour déterminer l'espace de travail supplémentaire qu'il nécessite.

Le client consulte ses données sous forme d'un flux en série contigu, qu'il gère de manière similaire à ces flux fournis sur d'autres types de technologies de bus. En interne, le système USB peut, selon sa propre police et les contraintes de contrôleur hôte, diviser la demande du client en petites demandes à envoyer dans l'USB. Mais, deux exigences doivent être remplies lorsque le système USB choisit d'effectuer une telle division:

- La division du flux de données en petits fragments n'est pas visible pour le client.
- Les échantillons USB ne sont pas divisés dans les transactions de bus.

Lorsqu'un client souhaite transférer les données, il envoie un IRP à l'USBD. Selon la direction du transfert de données, un tampon de données plein ou vide est fourni. Lorsque la demande est terminée (avec succès ou en raison d'une erreur), l'IRP et son état sont retournés au client. Si pertinent, cet état est également fourni selon la transaction.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.3.4 Définitions de données communes

Pour permettre au client de recevoir les résultats de la demande le plus directement possible de son appareil, il est souhaitable de minimiser la quantité de traitement et de copie requise entre l'appareil et le client. Pour faciliter cela, certains aspects de contrôle de l'IRP sont normalisés de sorte que les couches différentes dans la pile peuvent utiliser directement les informations fournies par le client. Le format particulier pour ces données dépend de l'actualisation de l'USBDI dans le système d'exploitation. Certains éléments de données peuvent en fait ne pas être directement possibles pour le client, mais sont générés suite à la demande du client.

Ces éléments de données définissent les informations correspondantes pour une demande:

- Identification du canal associé à la demande. L'identification de ce canal décrit également les informations comme le type de transfert pour cette demande.
- Identification de la notification pour le client particulier.
- Emplacement et longueur du tampon de données qui doit être transmis ou reçu.
- Etat d'exécution pour la demande. L'état résumé et, si nécessaire, l'état détaillé par transaction doivent être fournis.
- Emplacement et longueur de l'espace de travail. Cela dépend de la mise en œuvre.

Les mécanismes actuels utilisés pour communiquer les demandes vers l'USBD fonctionnent selon le système. Mais, au-delà des exigences indiquées ci-dessus pour lesquelles les informations associées à la demande doivent être disponibles, il existe également des exigences vis-à-vis de la manière dont les demandes sont traitées. Les exigences de base sont décrites au Chapitre 5. De plus, l'USBD fournit un mécanisme pour désigner un groupe d'IRP isochrones pour lesquels la transmission de la première transaction de chaque IRP a lieu dans la même (micro)trame. L'USBD fournit également un mécanisme pour désigner un ensemble sans interruption de demandes spécifiques au fournisseur ou à la classe pour un canal par défaut. Aucune autre demande pour ce canal par défaut, y compris une demande normalisée, de classe ou de fournisseur ne peut être insérée dans le flux d'exécution pour un tel ensemble sans interruption. Si une demande dans cet ensemble échoue, l'ensemble entier est retiré.

10.4 Pilote de contrôleur hôte

Le pilote de contrôleur hôte (HCD) est une abstraction du matériel du contrôleur hôte et le point de vue de la transmission de données du contrôleur hôte sur l'USB. Le HCDI satisfait aux exigences suivantes:

- Fournit une abstraction du matériel du contrôleur hôte.
- Fournit une abstraction des transferts de données par le contrôleur hôte dans l'interconnexion USB.
- Fournit une abstraction pour l'allocation (et la dé-allocation) des ressources du contrôleur hôte pour prendre en charge le service garanti pour les appareils USB.
- Présente le hub racine et son comportement en fonction de la définition de la classe de hub. Cela inclut la prise en charge du hub racine de sorte que le pilote de hub interagit avec le hub racine exactement comme n'importe quel hub. En particulier, même si un hub racine peut être mis en œuvre dans une combinaison de matériels et logiciels, le hub racine répond initialement à l'adresse d'appareil par défaut (point de vue client), renvoi les informations de descripteur, prend en charge la possession de l'ensemble d'adresses de appareils, et prend en charge les autres demandes de classe de hub. Mais, les transactions de bus peuvent ou non devoir être générées pour accomplir ce comportement avec la meilleure intégration possible entre le contrôleur hôte et le hub racine.

Le HCD fournit une interface logicielle (HCDI) qui met en œuvre les abstractions requises. La fonction du HCD est de fournir une abstraction, qui masque les détails du matériel du

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

contrôleur hôte. Au-dessous du matériel du contrôleur hôte, on retrouver l'USB physique et tous les appareils USB reliés.

Le HCD est le niveau inférieur dans la pile logicielle USB. Le HCD a un seul client: le pilote de bus série universel (USBD). L'USBD mappe les demandes de nombreux clients vers le HCD approprié. Un HCD donné peut gérer de nombreux contrôleurs hôtes.

Le HCDI n'est pas directement accessible à partir d'un client. Par conséquent, les exigences d'interface spécifiques pour le HCDI ne sont pas traitées ici.

10.5 Pilote de bus série universel

L'USBD fournit une collection de mécanismes que les composants du système d'exploitation, généralement les pilotes d'appareil, utilisent pour accéder aux appareils USB. Le seul accès à un appareil USB est celui fourni par l'USBD. Les mises en œuvre USBD fonctionnent de manière spécifique au système. Les mécanismes fournis par l'USBD sont mis en œuvre, utilisation si appropriée et augmentation si nécessaire, les mécanismes fournis par l'environnement du système d'exploitation dans lequel l'USB fonctionne. La discussion suivante est centrée sur les fonctionnalités de base requises pour toutes les mises en œuvre USBD. Pour les spécificités du fonctionnement USBD dans un environnement spécifique, voir le guide de l'environnement du système d'exploitation correspondant pour l'USBD. Une seule instance de l'USBD dirige les accès vers un ou plusieurs HCD qui, à leur tour, relient à un ou plusieurs contrôleurs hôtes. Si autorisée, la manière dont est instancié l'USBD dépend de l'environnement du système d'exploitation. Mais, du point de vue du client, l'USBD avec lequel le client communique gère tous les appareils USB reliés.

10.5.1 Présentation de l'USBD

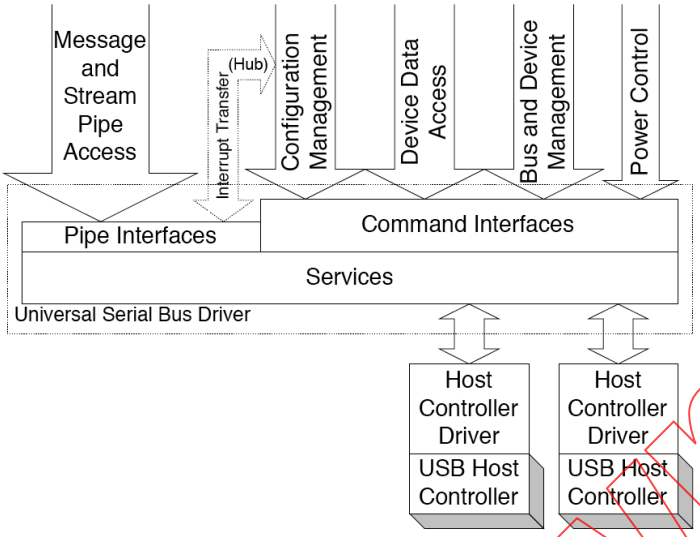
Les clients de l'USBD dirigent les commandes vers les appareils ou déplacent les flux de données vers ou des canaux. L'USBD présente deux groupes de mécanismes logiciels aux clients: mécanismes de commande et mécanismes de canaux.

Les mécanismes de commande permettent aux clients de configurer et de contrôler le fonctionnement USBD, de configurer et de contrôler génériquement un appareil USB. En particulier, ces mécanismes fournissent tous l'accès au canal par défaut de l'appareil.

Les mécanismes de canal permettent à un client USBD de gérer les données spécifiques aux appareils et de contrôler les transferts. Les mécanismes de canal ne permettent pas au client d'adresser directement le canal par défaut de l'appareil.

La Figure 10-5 présente une présentation de la structure USBD.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Message and Stream Pipe Access	Accès au canal de message et de flux
Interrupt Transfer (Hub)	Transfert d'interruption (hub)
Configuration Management	Gestion de la configuration
Device Data Access	Accès aux données d'appareil
Bus and Device Management	Gestion des bus et des appareils
Power Control	Commande de puissance
Pipe Interfaces	Interfaces de canaux
Command Interfaces	Interfaces de commandes
Services	Services
Universal Serial Bus Driver	Pilote de bus série universel
Host Controller Driver	Pilote de contrôleur hôte
USB Host Controller	Contrôleur d'hôte USB

Figure 10-5. Structure du pilote de bus série universel

10.5.1.1 Initialisation de l'USBD

L'initialisation de l'USBD spécifique fonctionne selon le système. Lorsqu'un USB particulier géré par l'USBD est initialisé, les informations de gestion pour cet USB sont également créées. La partie de ces informations de gestion est l'appareil d'adresse par défaut et son canal par défaut utilisé pour communiquer avec un appareil nouvellement réinitialisé.

Lorsqu'un appareil est relié à un USB, il répond à une adresse spéciale connue comme l'adresse par défaut (voir Chapitre 9) jusqu'à ce que son adresse unique soit assignée par l'énumérateur de bus. Pour que le système USB interagisse avec le nouvel appareil, l'adresse de l'appareil par défaut et le canal par défaut de l'appareil doivent être disponibles pour le pilote de hub lorsqu'un appareil est relié. Pendant l'initialisation de l'appareil, l'adresse par défaut est changée en une adresse unique.

10.5.1.2 Usage du canal USBD

Les canaux sont la méthode par laquelle un point d'extrémité d'appareil est associé à une entité de logiciel hôte. Les canaux sont possédés par exactement une entité sur l'hôte. Bien que le concept de base d'un canal soit le même, quel que soit le propriétaire, une certaine distinction des fonctionnalités fournies au client USBD a lieu entre deux groupes de canaux:

- les canaux par défaut, qui sont possédés par l'USBD
- tous les autres canaux, qui sont possédés et gérés par les clients de l'USBD

Les canaux par défaut ne sont jamais directement accessibles par les clients, bien qu'ils soient souvent utilisés pour remplir une partie des demandes client relayées via les mécanismes de commande.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

10.5.1.2.1 Canaux par défaut

L'USBD est responsable de l'allocation et de la gestion de la mise en tampon appropriée pour prendre en charge les transferts sur le canal par défaut qui ne sont pas directement visibles pour le client comme la définition d'une adresse d'appareil. Pour ces transferts directement visibles pour le client, comme l'envoi de commandes de fournisseur et de classe ou la lecture d'un descripteur d'appareil, le client doit fournir la mise en tampon requise.

10.5.1.2.2 Canaux client

Tout canal non possédé et géré par l'USBD peut être possédé et géré par un client USBD. Du point de vue de l'USBD, un seul client possède le canal. En fait, un groupe coopératif de clients peut gérer le canal, à condition qu'ils se comportent comme une seule entité coordonnée lors de l'utilisation du canal.

Le client est chargé de fournir la quantité de mise en tampon dont il a besoin pour entretenir le débit de transfert de données dans un intervalle de service atteignable par le client. Les exigences de mise en tampon supplémentaires pour l'espace de travail sont spécifiées par le système USB.

10.5.1.3 Fonctionnalités de service USBD

L'USBD fournit des services dans les catégories suivantes:

- Configuration via des mécanismes de commande
- Services de transfert via mécanismes de commande et de canal
- Notification d'évènements
- Compte-rendu d'état et reprise sur erreur

10.5.2 Exigences vis-à-vis des mécanismes de commande USBD

Les mécanismes de commande USBD permettent un accès générique à l'appareil USB. En général, ces commandes permettent d'établir des accès de lecture ou d'écriture à l'un des nombreux espaces potentiels de données et de contrôle d'appareil. Le client fournit le moins d'identifiants d'appareil et de données ou de pointeur de tampon vide.

Les transferts de commande USBD ne nécessitent pas que l'appareil USB soit configuré. Un grand nombre d'outils de configuration d'appareils fournis par l'USBD sont des transferts de commande.

Les exigences spécifiques des mécanismes de commande fournis sont indiquées ci-après.

10.5.2.1 Contrôle d'état d'interface

Les clients USBD doivent être capables de définir une interface spécifiée à n'importe quel état de canal configurable. La définition d'un état d'interface entraîne le transfert de tous les canaux dans cette interface dans cet état. De plus, tous les canaux dans une interface peuvent être réinitialisés ou abandonnés.

10.5.2.2 Contrôle d'état de canal

L'état du canal USBD a deux composants:

- Etat d'hôte
- Etat du point d'extrémité réfléchi

Si l'état du canal est signalé, la valeur pour les deux composants est identifiée. L'état du canal réfléchi par le point d'extrémité est le résultat du point d'extrémité situé dans un état particulier. Le client USBD gère l'état du canal comme signalé par l'USBD. Pour tout état de canal réfléchi par le point d'extrémité, le client doit également interagir avec le point d'extrémité pour changer l'état.

Un canal USBD est dans exactement l'un des états suivants:

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Actif: La règle du canal a été définie et le canal est capable de transmettre les données. Le client peut consulter pour savoir si les IRP sont en suspens pour un canal. Les canaux pour lesquels il n'y a pas d'IRP en suspens sont encore considérés comme étant dans l'état Actif tant qu'ils sont capables d'accepter de nouveaux IRP.
- Arrêté: Une erreur s'est produite sur le canal. Cet état peut également être le reflet du point d'extrémité Arrêté correspondant sur l'appareil.

Un canal et un point d'extrémité sont considérés comme actifs lorsque l'appareil est considéré et le canal et/ou le point d'extrémité n'est pas arrêté. Les clients peuvent manipuler l'état du canal des manières suivantes:

- Abandon d'un canal: Tous les IRP planifiés pour un canal sont retirés immédiatement et retournés au client avec un état indiquant qu'ils ont été abandonnés. Ni l'état de l'hôte ni l'état du point d'extrémité réfléchi du canal n'est affecté.
- Réinitialisation d'un canal: Les IRP du canal sont abandonnés. L'état de l'hôte est déplacé sur Actif. Si l'état du point d'extrémité réfléchi doit être changé, celui-ci doit être commandé explicitement par le client USBD.
- Effacement d'un canal Arrêté: L'état du canal est effacé de *Arrêté* à *Actif*.
- Arrêt d'un canal: L'état du canal est défini sur *Arrêté*.

10.5.2.3 Obtention de descripteurs

L'USBDI doit fournir un mécanisme pour récupérer l'appareil normalisé, la configuration, les descripteurs de chaîne, ainsi que les descripteurs spécifiques à la classe ou au fournisseur.

10.5.2.4 Obtention des réglages de configuration actuels

L'USBDI doit fournir un outil pour retourner, pour n'importe quel appareil spécifié, le descripteur de configuration actuel. Si l'appareil n'est pas configuré, aucun descripteur de configuration n'est retourné. Cette action équivaut à retourner le descripteur de configuration pour la configuration actuelle en demandant le descripteur de configuration spécifique. Elle ne nécessite cependant pas que le client connaisse l'identifiant pour la configuration actuelle. Cela retourne toutes les informations de configuration, notamment ce qui suit:

- Toutes les informations du descripteur de configuration comme enregistrées sur l'appareil, notamment tous les réglages alternatifs pour toutes les interfaces
- Indicateurs pour lesquels les réglages alternatifs pour les interfaces sont actifs
- Le canal traite les points d'extrémité dans les réglages alternatifs actifs pour interfaces
- Valeurs *w*MaxPacketSize réelles pour points d'extrémité dans les réglages alternatifs actifs pour interfaces

De plus, pour n'importe quel canal spécifié, l'USBDI doit fournir un outil pour retourner la valeur *wMaxPacketSize* qui est actuellement utilisée par le canal.

10.5.2.5 Ajout de appareils

L'USBDI doit fournir un mécanisme pour permettre au pilote de hub d'informer l'USBD de l'ajout d'un nouvel appareil dans un USB spécifié et de récupérer l'ID USBD du nouvel appareil USB. Les tâches USBD incluent l'assignation de l'adresse d'appareil et la préparation du canal par défaut de l'appareil pour utilisation.

10.5.2.6 Retrait de appareils

L'USBDI doit fournir un outil pour permettre au pilote de hub d'informer l'USBD qu'un appareil spécifique a été retiré.

10.5.2.7 Gestion d'états

L'USBDI doit fournir un mécanisme pour obtenir et effacer l'état basé sur l'appareil sur la base d'un appareil, d'une interface ou d'un canal.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.5.2.8 Envoi de commandes de classe

Ce mécanisme USBDI est utilisé par un client, généralement un pilote spécifique à la classe ou adaptatif, pour envoyer des commandes spécifiques à la classe à un appareil.

10.5.2.9 Envoi de commandes de fournisseur

Ce mécanisme USBDI est utilisé par un client pour envoyer une ou plusieurs commandes spécifiques au fournisseur à un appareil.

10.5.2.10 Etablissement de réglages alternatifs

L'USBDI doit fournir un mécanisme pour changer le réglage alternatif pour une interface spécifiée. Par conséquent, les descripteurs de canal pour le réglage précédent sont libérés et de nouveaux descripteurs de canal pour l'interface sont retournés. Pour que cette demande réussisse, l'interface doit être inactive; c'est-à-dire qu'aucun tampon de données ne peut être placé dans la file d'attente pour les canaux dans l'interface.

10.5.2.11 Etablissement d'une configuration

La configuration de logiciels nécessite qu'une configuration transfère un tampon contenant les informations de configuration à l'USBD. L'USBD demande des ressources pour les points d'extrémité dans la configuration, et si toutes les demandes de ressources réussissent, l'USBD définit la configuration de l'appareil et retourne les descripteurs d'interface avec les descripteurs de canal correspondants pour tous les points d'extrémité actifs. Les valeurs par défaut sont utilisées pour tous les réglages alternatifs pour interfaces.

Note: L'interface qui met en œuvre la configuration peut nécessiter l'identification d'autres réglages.

10.5.2.12 Définition de descripteurs

Pour les appareils prenant en charge ce comportement, l'USBDI permet de mettre à jour les descripteurs existants ou d'ajouter de nouveaux descripteurs.

10.5.3 Mécanismes du canal USBD

Cette partie de l'USBDI offre aux clients les services de transport de données supplémentaires les plus faibles et les plus rapides possible. Une performance accrue est obtenue en déléguant certaines responsabilités de gestion de canaux de l'USBD au client. Par conséquent, les mécanismes de canal sont mis en œuvre à un niveau plus primitif que les services de transfert de données fournis par les mécanismes de commande USBD. Les mécanismes de canal ne permettent pas au client d'accéder au canal par défaut d'un appareil.

Les transferts de canal USBD sont uniquement disponibles après avoir exécuté avec succès la configuration de l'appareil et de l'USB. Lors de la configuration de l'appareil, l'USBD demande les ressources requises pour prendre en charge tous les canaux d'appareil dans la configuration. Les clients peuvent modifier la configuration, selon si l'interface ou le canal spécifié(e) est inactif(ve) ou non.

Les clients fournissent des tampons pleins aux canaux sortants et récupèrent les informations sur l'état de transfert après l'exécution d'une demande. L'état de transfert retourné pour un canal sortant permet au client de déterminer le succès ou l'échec du transfert.

Les clients fournissent des tampons vides aux canaux entrants et récupèrent les tampons remplis et les informations sur l'état de transfert des canaux entrants après l'exécution d'une demande. L'état de transfert retourné pour un canal entrant permet au client de déterminer la quantité et la qualité des données reçues.

10.5.3.1 Types de canaux pris en charge

Les quatre types de canaux pris en charge, en fonction des quatre types de transfert, sont décrits dans les sections suivantes.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.5.3.1.1 Transferts de données isochrones

Chaque tampon en attente pour un canal isochrone doit être consultable comme flux d'échantillons. Comme avec tous les transferts de canaux, le client établit une politique pour utiliser ce canal isochrone, notamment l'intervalle de service correspondant pour ce client. Les octets perdus ou manquants, qui sont détectés à l'entrée, et les problèmes de transmission, qui sont notés à la sortie, sont indiqués au client.

Le client met en file d'attente un premier tampon, en commençant par le service de transmission continu de canal. Pour maintenir le modèle de transfert continu utilisé dans tous les transferts isochrones, le client met en file d'attente un tampon supplémentaire avant de retirer le tampon actuel.

L'USBD doit pouvoir fournir une vue du flux d'échantillons du flux de données du client. En d'autres mots, à l'aide de la méthode de synchronisation spécifiée par le client, la mise en paquets précise des données est cachée par le client. De plus, une transaction donnée est toujours contenue complètement dans certains tampons de données client.

Pour un canal de sortie, le client fournit un tampon de données. L'USBD alloue les données dans les (micro)trames pour la période de service à l'aide de la méthode de synchronisation choisie par le client.

Pour un canal d'entrée, le client doit fournir un tampon vide suffisamment grand pour contenir le nombre maximum d'octets délivrés par l'appareil du client dans la période de service. Si des octets manquants ou invalides sont indiqués, l'USBD peut laisser l'espace que les octets auraient occupé dans le tampon et identifier l'erreur. L'une des conséquences de la non-utilisation de méthode de synchronisation est que cet espace réservé est supposé être la taille de paquet maximum. La notification de tampon retiré a lieu lorsque l'IRP est terminé. Il faut noter que le tampon d'entrée peut ne pas être plein lorsqu'il est retourné au client.

L'USBD peut en option fournir des vues supplémentaires des flux de données isochrones. Il doit également pouvoir fournir une vue du flux de paquets du flux de données du client.

10.5.3.1.2 Transferts d'interruption

Le transfert de sortie d'interruption provient du client de l'USBD et est livré à l'appareil USB. Le transfert d'entrée d'interruption provient de l'appareil USB et est livré au client de l'USBD. Le système USB garantit que les transferts satisfont à la latence maximum spécifiée par le descripteur de point d'extrémité USB.

Le client met en file d'attente un tampon suffisamment grand pour tenir les données de transfert d'interruption (généralement une seule transaction USB). Lorsque toutes les données sont transférées, ou si le seuil d'erreur est dépassé, l'IRPS est retourné au client.

10.5.3.1.3 Transferts en masse

Les transferts en masse peuvent provenir de l'appareil ou du client. Aucune périodicité ou latence garantie n'est supposée. Lorsque toutes les données sont transférées, ou si le seuil d'erreur est dépassé, l'IRPS est retourné au client.

10.5.3.1.4 Transferts de contrôle

Tous les canaux de message transfèrent des données dans les deux directions. Dans tous les cas, le client génère un étage de configuration vers le point d'extrémité de l'appareil. L'étage de données en option peut être l'entrée ou la sortie et le statut final est toujours logiquement présenté à l'hôte. Pour plus de détails sur le protocole de message défini, voir Chapitre 8.

Le client prépare un tampon spécifiant la phase de commande et les données en option ou l'espace de tampon vide. Le client reçoit une notification de tampon retiré lorsque toutes les phases du transfert de contrôle sont terminées, ou une notification d'erreur si le transfert est abandonné en raison d'une erreur de transmission.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

10.5.3.2 Exigences vis-à-vis des mécanismes de canal USBD

Les mécanismes de canal suivants sont fournis.

10.5.3.2.1 Abandon des IRP

L'USBDI doit permettre l'abandon des IRP pour un canal particulier.

10.5.3.2.2 Gestion de la politique de canal

L'USBDI doit permettre à un client de définir et d'effacer la politique pour un canal individuel ou pour une interface entière. Les IRP effectués par le client avant de définir avec succès une politique sont rejetés par l'USBD.

10.5.3.2.3 Mise en file d'attente des IRP

L'USBDI doit permettre aux clients de mettre en file d'attente les IRP pour un canal donné. Lorsque les IRP sont retournés au client, l'état de la demande est également retourné. Un mécanisme est fourni par l'USBD pour identifier un groupe d'IRP isochrones dont les premières transactions ont toutes lieu dans la même (micro)trame.

10.5.4 Gestion de l'USB via les mécanismes USBD

A l'aide des mécanismes USBD fournis, les fonctionnalités générales suivantes sont prises en charge par n'importe quel système USB.

10.5.4.1 Services de configuration

Les services de configuration fonctionnent selon une base par appareil. Le logiciel de configuration indique à l'USBD quand effectuer la configuration de l'appareil. Un pilote de hub a un rôle spécial dans la gestion d'appareils et fournit au moins les fonctionnalités suivantes:

- Reconnaissance de branchement/déconnexion d'appareil, pilotée par un canal d'interruption possédé par le pilote de hub
- Réinitialisation des appareils, réalisée par le pilote de hub en redéfinissant le port de hub en amont de l'appareil
- Information de l'USBD pour effectuer l'assignation d'adresses d'appareils
- Commande de puissance

L'USBDI fournit également les outils de configuration suivants, qui peuvent être utilisés par le pilote de hub ou d'autres logiciels de configuration disponibles sur l'hôte:

- Identification des appareils et accès aux informations de configuration (via l'accès aux descripteurs sur l'appareil)
- Configuration des appareils via des mécanismes de commande

Lorsque le pilote de hub informe l'USBD d'un branchement d'appareil, l'USBD établit le canal par défaut pour le nouvel appareil.

10.5.4.1.1 Gestion des configurations

Les services de gestion des configurations sont principalement fournis sous forme d'ensemble de commandes d'interface spécifiques qui génèrent des transactions USB sur le canal par défaut. L'exception notoire est l'utilisation d'un canal d'interruption supplémentaire qui livre l'état du hub directement au pilote de hub.

Chaque hub initie un transfert d'interruption lorsqu'il y a un changement d'état d'un des ports de hub. Généralement, le changement d'état de port est la connexion ou le retrait d'un appareil USB en aval. (Voir Chapitre 11 pour plus de détails).

10.5.4.1.2 Configuration initiale des appareils

Le processus de configuration des appareils commence lorsqu'un hub signale, via son canal de changement d'état, la connexion d'un nouvel appareil USB.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Les services de gestion de configurations permettent de configurer les logiciels pour sélectionner une configuration d'appareil USB parmi l'ensemble de configurations listées dans l'appareil. L'USBD vérifie que la puissance adéquate est disponible et que les débits de transfert de données indiqués pour tous les points d'extrémité dans la configuration ne dépassent pas les fonctionnalités de l'USB avec la programmation actuelle avant de définir la configuration de l'appareil.

10.5.4.1.3 Modification d'une configuration d'appareil

Les services de gestion de configuration permettent de configurer les logiciels pour remplacer une configuration d'appareil USB par une autre parmi l'ensemble de configurations listées dans l'appareil. L'opération réussit si la puissance adéquate est disponible et si les débits de transfert de données indiqués pour tous les points d'extrémité dans la nouvelle configuration correspondent aux fonctionnalités de l'USB avec la programmation actuelle. Si la nouvelle configuration est rejetée, la configuration précédente demeure.

Les services de gestion de configurations permettent de configurer les logiciels pour retourner un appareil USB sur un état Non configuré.

10.5.4.1.4 Retrait des appareils

Le traitement de la récupération sur erreur et/ou du retrait des appareils commence lorsqu'un hub signale, via son canal de changement d'état, que l'appareil USB a été retiré.

10.5.4.2 Commande de puissance

Il y a deux niveaux de coopération de gestion de puissance pour l'USB: gestion au niveau bus et appareil. Cette spécification fournit les mécanismes permettant de gérer la puissance sur le bus USB. Les classes d'appareils peuvent définir des fonctionnalités de commande de puissance spécifiques à la classe.

Tous les appareils USB doivent prendre en charge l'état Suspendu (voir Chapitre 9). L'appareil est placé dans l'état Suspendu via la commande du port de hub auquel est relié l'appareil. Le fonctionnement de l'appareil normal cesse dans l'état Suspendu; mais, si l'appareil est capable de signaler le réveil et l'appareil est activé pour le réveil à distance, il peut générer la signalisation de reprise en réponse aux évènements externes.

Le système de gestion de puissance peut faire passer un appareil à l'état Suspendu ou mettre l'appareil hors tension pour contrôler et conserver la puissance. L'USB ne fournit ni exigences ni commandes pour l'état de l'appareil à enregistrer et restaurer dans ces transitions. Les classes d'appareils peuvent définir des fonctionnalités d'enregistrement et de restauration d'état d'appareil spécifiques aux classes.

L'USB coordonne l'action entre les états de puissance d'appareil et l'état Suspendu.

Il convient que lorsqu'un appareil n'est pas utilisé par le système (à savoir, aucune transaction n'est transmise vers ou de l'appareil outre les jetons SOF), l'appareil est suspendu dès que possible en suspendant de manière sélective le port auquel l'appareil est relié. Suspendre les appareils inactifs réduit les problèmes de fiabilité en raison des courants élevés passant dans un émetteur-récepteur fonctionnant en mode grande vitesse en présence des conditions de court-circuit décrites dans la Section 7.1.1. Certaines de ces conditions de court-circuit ne sont pas détectables en l'absence de transactions vers l'appareil. Suspendre l'appareil non utilisé place l'interface émetteur-récepteur en mode vitesse intégrale qui est plus fiable en présence de conditions de court-circuit.

10.5.4.3 Notification d'évènements

Les clients USBD reçoivent plusieurs notifications d'évènements via différentes sources:

- Exécution d'une action initiée par un client.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Les transferts d'interruption sur les canaux de flux peuvent livrer un avis d'évènements d'appareil directement aux clients USBD. Par exemple, les hubs utilisent un canal d'interruption pour livrer les évènements correspondant aux changements d'état de hub.
- Les données d'évènements peuvent être intégrées par les appareils dans les flux.
- Les commandes d'interface d'appareil normalisé, les commandes de classe d'appareil, les commandes spécifiques aux fournisseurs et même les transferts de commande génériques sur les canaux de message peuvent tous être utilisés pour analyser les appareils pour les conditions d'évènements.

10.5.4.4 Services de notification d'état et de récupération sur erreur

Les mécanismes de commande et de canal fournissent tous les deux la notification d'état sur les demandes individuelles lorsqu'elles sont invoquées et exécutées.

De plus, l'état d'appareil USB est disponible pour les clients USBD utilisant les mécanismes de commande.

L'USBD fournit aux clients des mécanismes de récupération sur erreur de canal en permettant aux canaux d'être réinitialisés ou abandonnés.

10.5.4.5 Gestion des appareils de réveil à distance

Le système USB peut minimiser la consommation de puissance de reprise d'une arborescence USB suspendue. Cela est accompli en activant explicitement les appareils capables de reprendre la signalisation et en contrôlant la propagation de la signalisation de reprise en sélectionnant de manière sélective et/ou en désactivant les ports de hub entre l'appareil et le hub de réveil autoalimenté le plus proche.

Dans certains scénarios de récupération sur erreur, le système USB doit ré-énumérer les sous-arborescences. La sous-arborescence peut être partiellement ou complètement suspendue. Pendant la récupération sur erreur, le système USB doit éviter toute contention entre un appareil émettant une signalisation de reprise et effectuant simultanément la réinitialisation du port. Cela est évité via la gestion de la caractéristique de réveil à distance des appareils et des fonctionnalités de port des hubs. Les règles sont les suivantes:

- Emettre une demande SetDeviceFeature(DEVICE_REMOTE_WAKEUP) à l'appareil feuille, juste avant de suspendre de manière sélective un port entre l'endroit où l'appareil est connecté et le port racine (via une demande SetPortFeature(PORT_SUSPEND)).
- Ne pas réinitialiser un port suspendu qui a eu un appareil activé pour le réveil à distance sans activer d'abord ce port.
- Vérifier qu'après un réveil à distance, les appareils dans la sous-arborescence affectée par le réveil à distance sont encore présents. Cela a généralement lieu en déterminant l'appareil de réveil à distance potentiel qui était la source du réveil. Il convient que cela ait lieu pour garantir qu'un appareil suspendu n'est pas déconnecté (et éventuellement reconnecté) ou réinitialisé (ex.: par le bruit) pendant un processus suspendre/reprendre.

10.5.5 Transfert d'une commande de pré-amorçage USB au système d'exploitation

Un seul pilote logiciel possède le contrôleur hôte. Si le système hôte met en œuvre les services USB avant le chargement du système d'exploitation, le contrôleur hôte doit fournir un mécanisme qui désactive l'accès par le logiciel de pré-amorçage et permet au système d'exploitation de gagner le contrôle. La configuration de l'USB de pré-amorçage n'est pas transférée au système d'exploitation. Lorsque celui-ci gagne le contrôle, il est chargé de configurer totalement le bus. Si le système d'exploitation fournit un mécanisme pour redonner le contrôle à l'environnement de pré-amorçage, le bus est dans un état inconnu. Il convient que le logiciel de pré-amorçage traite cet évènement comme une mise sous tension.

10.6 Guides d'environnement du système d'exploitation

Comme noté précédemment, les interfaces actuelles entre le système USB et le logiciel hôte sont spécifiques à la plate-forme hôte et au système d'exploitation. Il doit exister une

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

spécification de compagnie pour chaque combinaison de plate-forme et de système d'exploitation avec prise en charge USB. Ces spécifications décrivent les interfaces spécifiques utilisées pour intégrer l'USB dans l'hôte. Chaque fournisseur de système d'exploitation pour le système USB identifie une révision compatible de la spécification USB universelle.

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11 Chapitre 11
Spécification relative au hub

Ce chapitre décrit les exigences architecturales pour le hub USB. Il contient une description des trois sous-blocs principaux: le répéteur de hub, le contrôleur de hub et le traducteur de transactions. Le chapitre décrit également le fonctionnement du hub pour la récupération sur erreur, la réinitialisation et la suspension/reprise. La deuxième moitié du chapitre définit le comportement de demande de hub et les descripteurs de hub.

La spécification relative au hub fournit des informations supplémentaires suffisantes pour permettre à un exécutant de concevoir un hub qui est conforme à la spécification USB.

11.1 Présentation

Les hubs fournissent l'interface électrique entre les appareils USB et l'hôte. Les hubs sont directement responsables de la prise en charge de nombreux attributs qui rendent l'USB convivial et masquent sa complexité à l'utilisateur. Voici une liste des principaux aspects de la fonctionnalité USB que les hubs doivent prendre en charge:

- Comportement de connectivité
- Gestion de l'énergie
- Détection des connexions/déconnexions d'appareils
- Détection de défaut de bus et récupération
- Prise en charge des appareils à grande vitesse, vitesse intégrale et basse vitesse

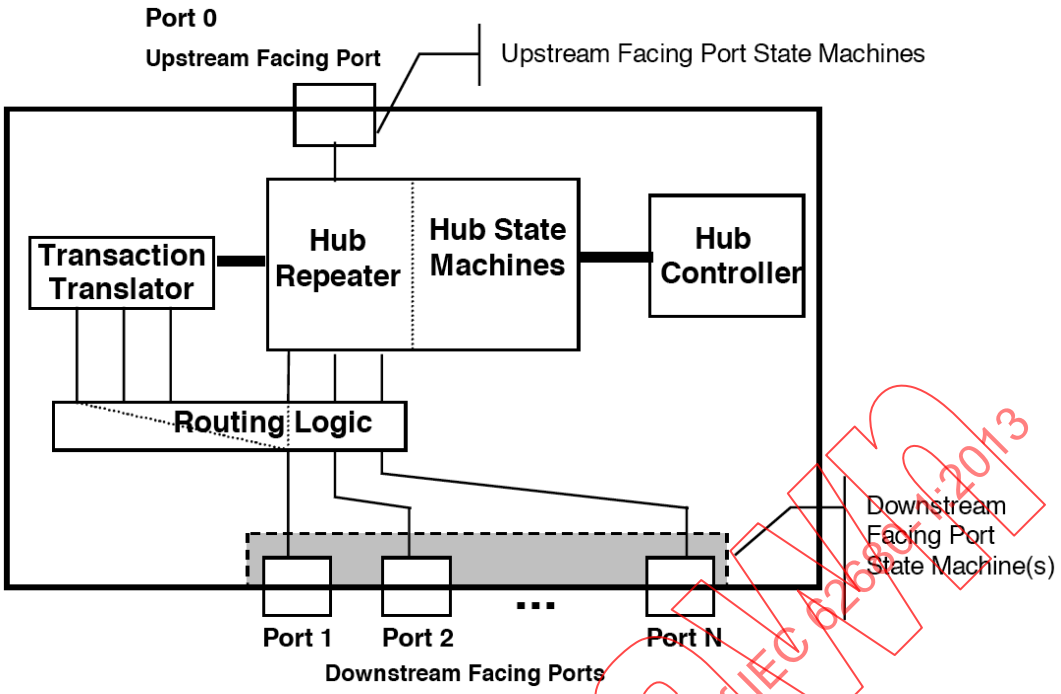
Un hub compte trois composants: le répéteur de hub, le contrôleur de hub et le traducteur de transactions. Le répéteur de hub est responsable de la configuration et de la désinstallation de la connectivité. Il assure également le traitement des exceptions, comme la détection des défauts de bus et la récupération ainsi que la détection des connexions/déconnexions. Le contrôleur de hub fournit le mécanisme pour la communication hôte-hub. L'état spécifique au hub et les commandes de contrôle permettent à l'hôte de configurer un hub et de surveiller et contrôler ses ports orientés en aval individuels. Le traducteur de transactions répond aux transactions fractionnées à grande vitesse et les traduit en transactions à vitesse intégrale/basse vitesse avec des appareils à vitesse intégrale/basse vitesse reliés aux ports orientés en aval.

11.1.1 Architecture d'un hub

La Figure 11-1 illustre un hub et les positions de ses ports orientés en amont et en aval. Un hub comprend une section Répéteur de hub, une section Contrôleur de hub et une section Traducteur de transactions. Le hub doit fonctionner à grande vitesse lorsque son port orienté en amont est relié à grande vitesse. Le hub doit fonctionner à vitesse intégrale lorsque son port orienté en amont est relié à vitesse intégrale.

Le répéteur de hub est chargé de gérer la connectivité entre les ports orientés en amont et en aval qui fonctionnent à la même vitesse. Le répéteur de hub prend en charge la connectivité à vitesse intégrale/basse vitesse et la connectivité à grande vitesse. Le contrôleur de hub fournit l'état et la commande et permet l'accès de l'hôte au hub. Le traducteur de transactions prend les transactions de division à grande vitesse et les traduit en transactions à vitesse intégrale/basse vitesse lorsque le hub fonctionne à grande vitesse et présente des appareils à vitesse intégrale/basse vitesse reliés. La vitesse de fonctionnement d'un appareil relié à un port orienté en aval détermine si la logique de routage connecte un port aux sections traducteur de transactions ou répéteur de hub.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Upstream Facing Port	Port orienté en amont
Upstream Facing Port State Machines	Diagrammes d'états de port orienté en amont
Transaction Translator	Traducteur de transactions
Hub Repeater	Répéteur de hub
Hub State Machines	Diagrammes d'état de hub
Hub Controller	Contrôleur de hub
Routing Logic	Logique de routage
Port 1	Port 1
Port 2	Port 2
Port N	Port N
Downstream Facing Port State Machine(s)	Diagramme(s) d'état de port orienté en aval
Downstream Facing Ports	Ports orientés en aval

Figure 11-1. Architecture d'un hub

Lorsqu'un port orienté en amont d'un hub est relié à un environnement électrique qui fonctionne à vitesse intégrale/basse vitesse, la fonctionnalité grande vitesse du hub est désactivée. Cela signifie que le hub fonctionne uniquement à vitesse intégrale/basse vitesse et le traducteur de transactions et le répéteur à grande vitesse ne fonctionnent pas. Dans cet environnement électrique, le répéteur de hub doit fonctionner comme un répéteur à vitesse intégrale/basse vitesse et la logique de routage connecte les ports au répéteur de hub.

Lorsque le port orienté en amont d'un hub est relié à un environnement électrique qui fonctionne à grande vitesse, le répéteur de hub à vitesse intégrale/basse vitesse n'est pas opérationnel. Dans cet environnement électrique, lorsqu'un appareil à grande vitesse est relié au port orienté en aval, la logique de routage connecte le port au répéteur de hub et le répéteur de hub doit fonctionner comme un répéteur à grande vitesse. Dans ce cas, lorsqu'un appareil à vitesse intégrale/basse vitesse est relié à un port orienté en aval, la logique de routage doit connecter le port au traducteur de transactions.

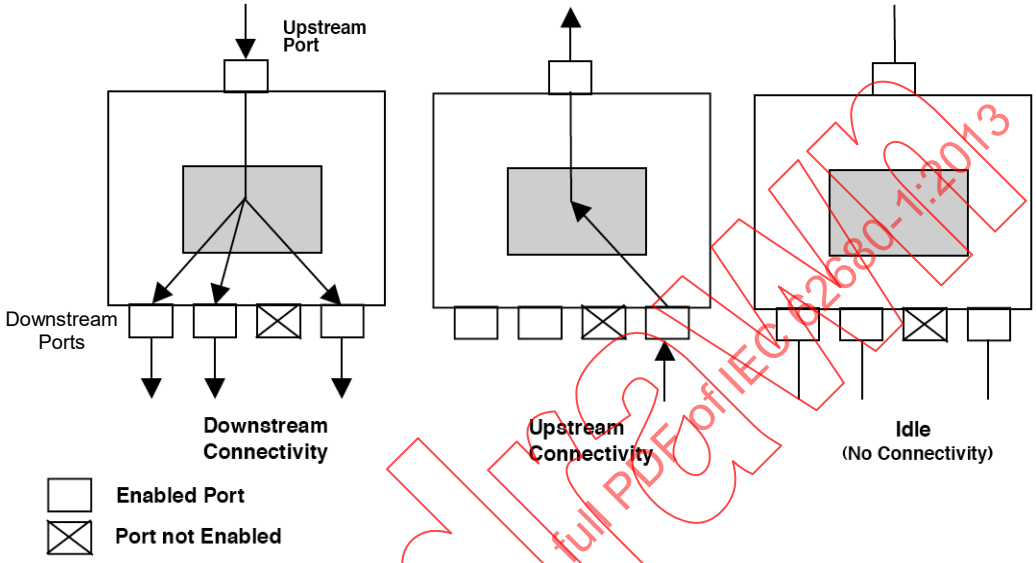
11.1.2 Connectivité du hub

Les hubs affichent un comportement de connectivité différent selon s'ils propagent le trafic de paquets ou reprennent la signalisation, ou s'ils sont dans l'état Inactif.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.1.2.1 Connectivité de la signalisation de paquets

Le répéteur de hub contient un port qui doit toujours connecter dans le sens amont (appelé port orienté en amont) et un ou plusieurs ports orientés en aval. La connectivité en amont est définie comme étant vers l'hôte, et la connectivité en aval est définie comme étant vers un appareil. La Figure 11-2 montre le comportement de connectivité de signalisation de paquets pour les hubs dans les directions en amont et en aval. Un hub a également un état Repos, pendant lequel le hub n'effectue aucune connectivité. Dans l'état Repos, tous les ports du hub sont en mode réception en attente du début du prochain paquet.



Légende

Anglais	Français
Upstream Port	Port en amont
Downstream Ports	Ports en aval
Downstream Connectivity	Connectivité en aval
Upstream Connectivity	Connectivité en amont
Idle (No Connectivity)	Inactivité (pas de connectivité)
Enabled Port	Port activé
Port not Enabled	Port non activé

Figure 11-2. Connectivité de la signalisation de hubs

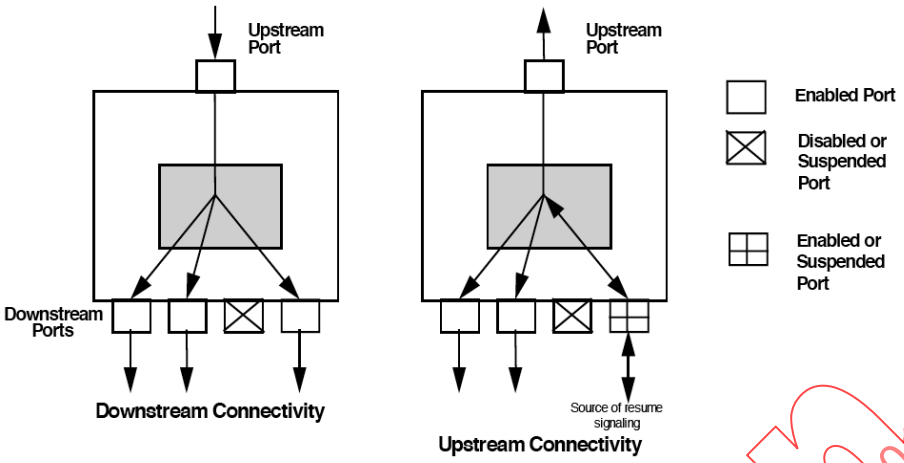
Si un port orienté en aval est activé (à savoir dans un état dans lequel il peut propager la signalisation dans le hub), et le hub détecte le début d'un paquet sur ce port, la connectivité est établie en amont vers le port orienté en amont de ce hub, mais pas vers les autres ports orientés en aval. Cela signifie que lorsqu'un appareil ou un hub transmet un paquet en amont, seuls les hubs en ligne entre l'appareil de transmission et l'hôte voient le paquet. Voir Section 11.8.3 pour le comportement en option lorsqu'un hub détecte la signalisation en amont simultanée sur plusieurs ports.

En aval, les hubs fonctionnent en mode diffusion. Lorsqu'un hub détecte le début d'un paquet sur son port orienté en amont, il établit la connectivité avec tous les ports orientés en aval. Si aucun port n'est activé, il ne propage pas la signalisation de paquets en aval.

11.1.2.2 Connectivité de reprise

Les hubs montrent différents comportements de connectivité pour la signalisation de reprise en amont et en aval. Un hub qui est suspendu reflète la signalisation de reprise de son port orienté en amont vers tous ses ports orientés en aval activés. La Figure 11-3 illustre la connectivité de reprise en amont et en aval du hub.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Upstream Port	Port en amont
Downstream Ports	Ports en aval
Downstream Connectivity	Connectivité en aval
Upstream Connectivity	Connectivité en amont
Source of resume signaling	Source de signalisation de reprise
Enabled Port	Port activé
Disabled or Suspended Port	Port désactivé ou suspendu
Enabled or Suspended Port	Port activé ou suspendu

Figure 11-3. Connectivité de reprise

Si un hub est suspendu et détecte la signalisation de reprise à partir d'un port orienté en aval suspendu de manière sélective ou activée, le hub reflète cette signalisation en amont et vers tous les ports orientés en aval activés, notamment le port qui a initié la séquence de reprise. La signalisation de reprise n'est pas reflétée vers les ports désactivés ou suspendus. La connectivité de reprise est davantage détaillée dans la Section 11.9.

11.1.2.3 Mécanismes d'élimination des défauts de hub

Les hubs représentent le composant USB essentiel pour établir la connectivité entre l'hôte et les autres appareils. Il est vital que les défauts de connectivité, notamment ceux qui peuvent donner lieu à une impasse, soient détectés et empêchés. Les hubs doivent traiter les défauts de connectivité uniquement lorsqu'ils sont en mode répéteur.

Les hubs doivent également être capables de détecter et de récupérer à partir des paquets perdus ou corrompus qui sont adressés au contrôleur de hub. Comme le contrôleur de hub est, en fait, un autre appareil USB, il doit observer les mêmes règles de temporisation que les autres appareils USB, comme décrit au Chapitre 8.

11.2 Minuteur de trame/microtrame de hub

Chaque hub a un minuteur de (micro)trame dont la synchronisation est dérivée de l'horloge locale du hub et est synchronisée avec la période de (micro)trame hôte par le début de (micro)trame généré par l'hôte (SOF). Le minuteur de (micro)trame fournit les références de synchronisation qui sont utilisées pour permettre au hub de détecter un appareil de babillage et d'empêcher le hub d'être désactivé par le hub en amont. Le minuteur de (micro)trame de hub doit suivre la période de (micro)trame hôte et être capable de rester synchronisé avec l'hôte même si deux jetons SOF consécutifs sont manqués par le hub.

Le minuteur de (micro)trame doit verrouiller la synchronisation de (micro)trame de l'hôte pour les pires précisions d'horloge et les écarts de synchronisation entre l'hôte et le hub. Il existe des exigences spécifiques pour les hubs lorsque leur port orienté en amont fonctionne à grande vitesse et à vitesse intégrale.

11.2.1 Plage de minuteur de microtrame à grande vitesse

La plage d'un minuteur de microtrame doit être entre 59 904 et 60 096 bits à grande vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

L'intervalle de microtrame nominal est 60 000 temps bits à grande vitesse. La plage de minuteur de microtrame de hub spécifiée ci-dessus est de 60 000 +/- 96 temps bits à grande vitesse pour permettre la précision de l'hôte, la précision du hub, la gigue du répéteur et la quantisation du hub. La variation du temps bit à vitesse intégrale +/-96 est calculée au Tableau 11-2.

Tableau 11-1. Contributions de la plage de minuteur de microtrame à grande vitesse

Source de variation	Variation (ppm)	Variation (bits) sur un intervalle de microtrame	Commentaire
Précision d'hôte	+/- 500	+/- 30	
Précision de hub	+/- 500	+/- 30	
Gigue de l'hôte		+/- 2	
Gigue de la chaîne de hub		+/- 20	Quatre hubs en série en amont du hub; 0 à 5 bits de gigue par hub
Quantisation		+/-14	Les bits doivent arrondir la variation totale en multiple de 16

11.2.2 Plage de minuteur de trame à vitesse intégrale

La plage de minuteur de trame doit être entre 11 958 et 12 042 bits à vitesse intégrale.

L'intervalle de trame nominal est 12 000 temps bits à vitesse intégrale. La plage de minuteur de trame de hub spécifiée ci-dessus est de 12 000 +/- 42 temps bits à vitesse intégrale pour permettre la précision de l'hôte et la précision du hub. La variation du temps bit à vitesse intégrale +/-42 est calculée dans le Tableau 11-2.

Tableau 11-2. Contributions de la plage de minuteur de trame à vitesse intégrale

Source de variation	Variation (ppm)	Variation (bits) sur un intervalle de trame	Commentaire
Précision d'hôte	+/- 500	+/- 6	
Précision de hub	+/- 3 000	+/- 36	+/-6 bits en raison de la précision de hub (500 ppm) +/-30 bits en raison de la précision de hub 1 x parent (2 500 ppm)

11.2.3 Synchronisation du minuteur de trame/microtrame

Un minuteur de (micro)trame de hub est rythmé par la source de l'horloge du hub et est synchronisé en paquets SOF qui sont dérivés du minuteur de (micro)trame de l'hôte. Après une réinitialisation ou une reprise, le minuteur de (micro)trame du hub n'est pas synchronisé. Lorsque le hub reçoit deux paquets SOF consécutifs, son minuteur de (micro)trame doit être synchronisé. Synchronisé est synonyme de rythmé. Voici un exemple de construction d'un minuteur qui synchronise correctement.

11.2.3.1 Exemple de méthode de synchronisation de minuteur de (micro)trame

Le hub maintient trois valeurs de minuteur: minuteur de (micro)trame (compteur descendant), (micro)trame actuelle (compteur ascendant) et (micro)trame suivante (registre). Après une réinitialisation ou reprise, un drapeau est placé pour indiquer que le minuteur de (micro)trame n'est pas synchronisé.

Lorsque le premier jeton SOF est détecté, le minuteur de (micro)trame actuel réinitialise et démarre le comptage une fois par temps bit de hub. Sur le SOF suivant, si le minuteur n'a pas basculé, la valeur dans le minuteur de (micro)trame actuel est chargée dans le registre de (micro)trame suivant et dans le minuteur de (micro)trame. Le minuteur de (micro)trame actuel est remis à zéro et continue de compter et le drapeau est placé pour indiquer que le minuteur de (micro)trame est verrouillé. Le minuteur de (micro)trame bascule lorsque le décompte dépasse 60 096 pour grande vitesse ou 12 042 pour vitesse intégrale (un essai à 65 535 pour grande vitesse ou 16 383 pour vitesse intégrale est adéquat). Si le minuteur de (micro)trame actuel a basculé, un SOF a alors été manqué et le minuteur de (micro)trame et les valeurs de (micro)trame suivantes ne sont pas chargés. Lorsqu'un SOF est raté, le drapeau indiquant que le minuteur n'est pas synchronisé reste en place.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Quand le minuteur de (micro)trame compte à rebours jusqu'à zéro, la valeur du registre de (micro)trame suivant est chargée dans le minuteur de (micro)trame. Quand un SOF est détecté et que le minuteur de (micro)trame actuel n'a pas basculé, la valeur de minuteur de (micro)trame est chargée dans le registre de (micro)trame et les registres de (micro)trame suivants. Le minuteur de (micro)trame actuel est alors défini sur zéro et continue à compter. Si le minuteur de (micro)trame actuel a basculé, la valeur dans le registre de (micro)trame suivant est alors chargée dans le minuteur de (micro)trame. Ce processus peut entraîner la mise à jour du minuteur de (micro)trame deux fois dans une seule (micro)trame: une fois lorsque le minuteur de (micro)trame atteint zéro et une fois lorsque le SOF est détecté.

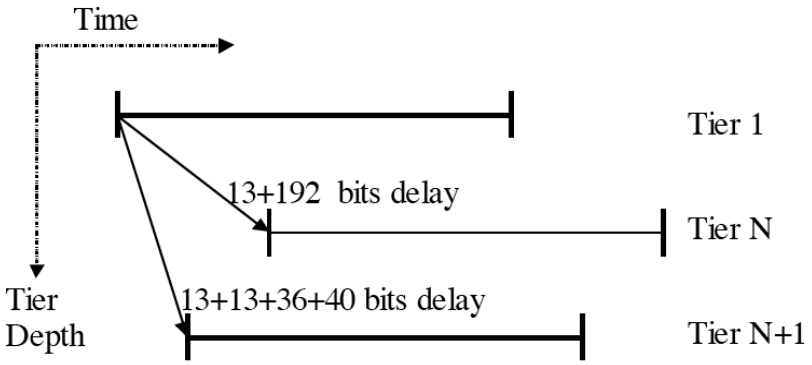
11.2.3.2 Avancement EOF

Le hub doit avancer ses points EOF en fonction de son délai de décodage pour garantir que dans la topologie multiniveaux, les hubs éloignés de l'hôte ont toujours des points EOF plus tard que les hubs plus proches de l'hôte. L'ampleur de l'avance dépend de la mise en œuvre; la plage possible d'avance est dérivée ci-après.

Le circuit de synchronisation décrit ci-dessus dépend du décodage réussi d'un identifiant de paquet SOF (PID). Cela signifie que le minuteur de (micro)trame est synchronisé selon un délai qui est postérieur au point de synchronisation dans le paquet SOF: postérieur d'au moins 40 temps bits pour grande vitesse ou 16 temps bits pour vitesse intégrale. Chaque mise en œuvre prend également du temps pour réagir au décodage SOF et définir les valeurs de minuteur/compteur appropriées. Ce délai de réaction dépend de la mise en œuvre, mais il est supposé être inférieur à 192 temps bits pour grande vitesse et quatre temps bits pour vitesse intégrale. Les sections suivantes décrivent les actions qui sont contrôlées par le minuteur de (micro)trame. Ces actions sont définies au niveau de EOF1, EOF2 et EOF. EOF1 et EOF2 sont définis dans les sections suivantes. Ces sections supposent que le minuteur de (micro)trame du hub est remis à zéro à la fin de la (micro)trame (EOF). L'ensemble de circuits décrit ci-dessus a un minuteur de (micro)trame qui est remis à zéro après 40 à 192 pour les temps bits à grande vitesse ou 16-20 temps bits à vitesse intégrale après le démarrage d'une (micro)trame (ou la fin de la (micro)trame précédente). Les synchronisations et décalages de bits dans les sections suivantes doivent être avancés pour tenir compte de ce retard (à savoir, ajouter 40-192 pour grande vitesse ou 16-20 temps bits pour vitesse intégrale aux points EOF1 et EOF2).

L'avancement des points EOF par le retard de traitement garantit que l'écart entre les EOF est seulement dû au retard de propagation. Par exemple, pour grande vitesse, l'écart maximum entre 2 points EOF n'importe où sur l'USB est inférieur à 216 bits (144 + 72). 144 temps bits sont dus à 36 temps bits de latence max dans 4 répéteurs. 72 temps bits sont dus à cinq retards maximum de câble et d'interconnexion de 30 ns chacun. Comme on peut le voir dans la Figure 11-4 sans avancement EOF, un hub avec un plus grand nombre de niveaux peut avoir un EOF qui se produit plus tôt qu'un hub avec un nombre de niveaux inférieur. Dans la Figure 11-5, l'avancement EOF garantit que dans la topologie multiniveaux, les hubs avec de plus grands nombres de niveaux ont toujours des points EOF ultérieurs par rapport aux hubs avec des nombres de niveaux inférieurs. Note: 13 temps bits dans les figures est un exemple de retard de câble maximum (environ 30 ns).

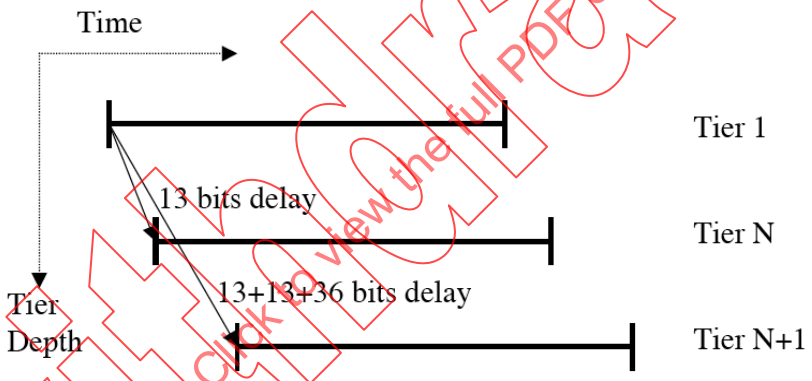
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Time	Temps
Tier Depth	Profondeur de niveau
Tier 1	Niveau 1
Tier N	Niveau N
Tier N+1	Niveau N+1
13+192 bits delay	Retard 13+192 bits
13+13+36+40 bits delay	Retard 13+13+36+40 bits

Figure 11-4. Exemple de décalages EOF à grande vitesse en raison du temps de propagation sans avancement EOF



Légende

Anglais	Français
Time	Temps
Tier Depth	Profondeur de niveau
Tier 1	Niveau 1
Tier N	Niveau N
Tier N+1	Niveau N+1
13 bits delay	Retard 13 bits
13+13+36 bits delay	Retard 13+13+36 bits

Figure 11-5. Exemple de décalages EOF à grande vitesse en raison du temps de propagation avec avancement EOF

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.2.3.3 Effet de synchronisation sur le comportement du répéteur

Le minuteur de (micro)trame fournit une indication au diagramme d'états du répéteur de hub que le minuteur de (micro)trame a synchronisé avec SOF et que le minuteur de (micro)trame est capable de générer les points de synchronisation EOF1 et EOF2. Ce signal est important après une reprise globale en raison de la possibilité qu'un appareil à vitesse intégrale/basse vitesse puisse avoir été débranché, et qu'un appareil à basse vitesse/vitesse intégrale branché tandis que l'hôte générait une longue reprise (plusieurs secondes) et la déconnexion ne puisse pas être détectée. Le nouvel appareil biaise D+ et D- pour apparaître comme un K sur le hub qui serait alors être traité comme un SOP et, sauf inhibé, ce SOP propagerait dans les hubs repris. Comme les hubs n'auraient pas vu les SOF à ce point, les hubs ne seraient pas synchronisés et ainsi capables de générer les points de synchronisation EOF1 et EOF2.

La seule reprise serait pour l'hôte de réinitialiser et de ré-énumérer la section du bus contenant l'appareil modifié. Ce scénario est empêché en empêchant tout port orienté en aval d'établir la connectivité jusqu'à ce que le hub soit verrouillé après une reprise.

11.2.4 Gigue de microtrame liée à la gigue de trame

La période entre les SOF du traducteur de transactions ne doit pas varier de plus de +/- 42 ns. Le compteur du minuteur de microtrame doit être utilisé par le traducteur de transactions pour générer des SOF sur les appareils à vitesse intégrale (et maintiens actifs sur les appareils à basse vitesse) raccordés à celui-ci.

Le SOF reçu sur le port orienté en amont du hub est répété avec une horloge locale. La fréquence de cette horloge peut être une version divisée du débit binaire. Cela peut entraîner une erreur de quantisation et une gigue entre microtrames. La gigue entre microtrames d'un répéteur de hub doit être comprise entre 0 et 5 temps bits. Cela signifie que la latence dans le répéteur de SOF consécutifs doit différer de moins de 5 bits. Un hub peut enregistrer le SOF à usage interne, ex.: synchronisation des microtrames. Cela nécessite la détection des PID SOF. L'ensemble de circuits utilisé pour l'enregistrement interne du SOF doit avoir une gigue qui est inférieure ou égale à 16 bits. Cela signifie que les valeurs du compteur de minuteur de microtrame entre SOF consécutifs équidistants doivent différer d'une valeur inférieure ou égale à 16 bits. La fréquence du contrôleur hôte peut dériver pendant la période d'une microtrame entraînant une gigue d'une période de microtrame. La gigue source du contrôleur hôte pour SOF doit être inférieure à 4 bits. Cela signifie que les périodes consécutives entre SOF doivent différer de moins de 4 bits. Ces exigences garantissent que la période de microtrame à la fin de cinq niveaux de hub a une gigue de moins de 40 bits (4 du contrôleur hôte + 4*5 des répéteurs de hub + 16 de l'enregistrement SOF interne). Cela signifie que les périodes consécutives entre SOF comme mesuré dans n'importe quel minuteur de microtrame diffèrent de moins de 40 bits (83,3 ns à 480 Mbs). Cela est inférieur à la variation +/- 42 ns autorisée.

11.2.5 Points de synchronisation EOF1 et EOF2

EOF1 et EOF2 sont des points de synchronisation qui sont dérivés du minuteur de (micro)trame du hub. Le Tableau 11-3 spécifie les points de synchronisation EOF de l'hôte et du hub requis pour le fonctionnement à grande vitesse et à vitesse intégrale.

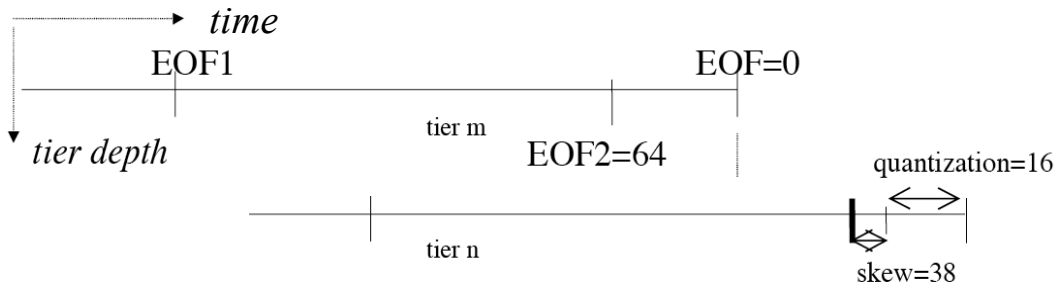
Tableau 11-3. Points de synchronisation EOF1/EOF2 du hub et de l'hôte

Libellé	Temps bits avant EOF pour grande vitesse	Temps bits avant EOF pour vitesse intégrale	Notes
EOF1	560	32	Point de fin de (micro)trame #1
EOF2	64	10	Point de fin de (micro)trame #2

Ces points de synchronisation sont utilisés pour garantir que les appareils et hubs n'interfèrent pas avec la transmission correcte du paquet SOF de l'hôte. *Ces points de synchronisation ont seulement du sens lorsque le minuteur de micro(trame) a été synchronisé avec le SOF.*

Les marqueurs de (micro)trame de l'hôte et du hub, tandis que tous les autres sont synchronisés avec le SOF de l'hôte, sont soumis à certains biais qui dictent le positionnement des points EOF. La Figure 11-6 illustre le point de synchronisation EOF2 pour le fonctionnement à grande vitesse. La Figure 11-7 illustre le point de synchronisation à grande vitesse EOF1. Les nombres dans les figures sont des temps bits à grande vitesse.

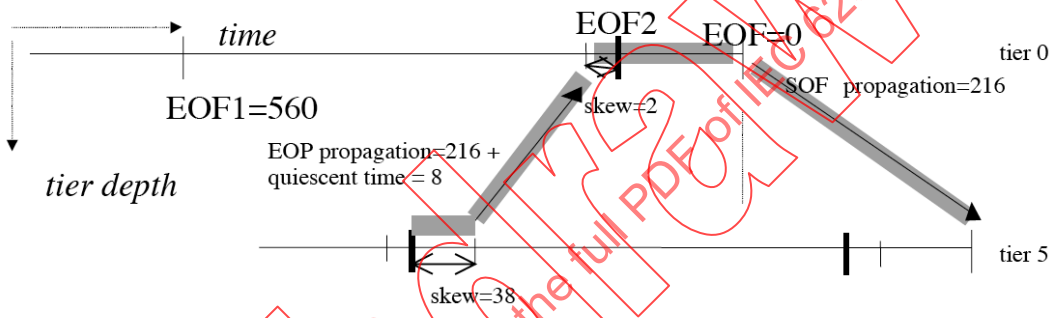
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Time	Temps
tier depth	profondeur de niveau
tier m	niveau m
tier n	niveau n
quantization	quantisation
skew	biais

Figure 11-6. Point de synchronisation EOF2 à grande vitesse



Légende

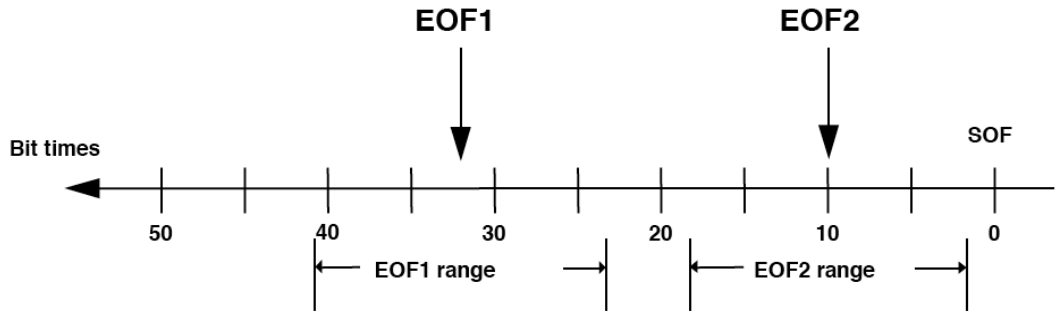
Anglais	Français
Time	Temps
tier depth	profondeur de niveau
EOP propagation=216 + quiescent time = 8	Propagation EOP = 216 + temps de repos = 8
skew=2	biais=2
EOF2	EOF2
EOF=0	EOF=0
SOF propagation=216	Propagation SOF=216
tier 0	niveau 0
tier 5	niveau 5
skew=38	biais=38

Figure 11-7. Point de synchronisation EOF1 à grande vitesse

Au point EOF2, n'importe quel port qui a une connectivité en amont, est désactivé comme un babilleur. Les hubs fonctionnant comme un répéteur à grande/basse vitesse empêchent de devenir désactivés en envoyant une fin de paquet au hub en amont avant que le hub n'atteigne son point EOF2 (à savoir, au niveau de EOF1).

La Figure 11-8 illustre les points de synchronisation EOF pour le fonctionnement du répéteur à grande/basse vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Bit times	Temps bit
EOF1	EOF1
EOF2	EOF2
EOF1 range	Plage EOF1
EOF2 range	Plage EOF2

Figure 11-8. Points de synchronisation EOF à vitesse intégrale

Le hub fonctionnant en tant que répéteur pleine/basse vitesse peut envoyer l'EOP si la connectivité en amont n'est pas établie au moment EOF1. Un répéteur pleine vitesse doit envoyer l'EOP si la connectivité est établie de n'importe quel port en aval du point EOF1.

Un répéteur à grande vitesse doit désinstaller la connectivité en amont au point EOF1.

Un répéteur à grande vitesse doit désinstaller la connectivité après que le bus soit retourné à l'état Repos et le tampon d'élasticité soit vidé (comme décrit dans la Section 11.7.2) et non lors du décodage d'un schéma EOP comme à vitesse intégrale/basse vitesse. Par conséquent, interrompre la fin de la signalisation (à savoir, sans EOP à grande vitesse) peut entraîner des paquets mal formés, et cela ne doit pas affecter le fonctionnement du répéteur. Le modèle de contrôleur hôte doit être capable de traiter correctement ces paquets.

11.2.5.1 Points de synchronisation EOF1 et EOF2 à grande vitesse

Le point EOF2 est 64 temps bits avant EOF comme illustré dans la Figure 11-6, et le point EOF1 est 560 temps bits avant EOF comme illustré dans la Figure 11-7.

Bien que le hub soit synchronisé avec le SOF, le biais de synchronisation peut s'accumuler entre l'hôte et un hub ou entre hubs. Ce biais de synchronisation représente la différence entre des minuteurs de microtrame sur différents hubs et l'hôte. Le biais accumulé total peut être égal à 38 temps bits. Celui-ci est composé de ±2 temps bits de gigue source de l'hôte de (micro)trame et de 0 à 36 temps bits de gigue de répéteur comme dérivé précédemment. Cette synchronisation de biais affecte le positionnement des points EOF1 et EOF2.

Note: La synchronisation du biais de hub suppose que l'intervalle de microtrame n'est pas modifié par l'hôte après la synchronisation des minuteurs de microtrame.

Le biais EOF peut être compris entre -2 et + 38 bits, de sorte que tous les EOF sont compris dans 256 bits (216 bits de temps de propagation EOF + 40 bits de biais EOF) chacun.

Note: Le point EOF2 est basé sur 16 temps bits pour la quantisation + 38 temps bits de biais; le point EOF2 doit donc être au moins 54 temps bits avant EOF. Le point EOF2 est défini sur 64 temps bits pour permettre la détection du babillage avec une version divisée par 16 de l'horloge binaire. Un paquet dirigé en amont terminant avant EOF1 doit atteindre chaque hub/hôte en amont avant qu'il n'accède à son point EOF2. Cela est possible si le point EOF1 est situé au moins 544 bits avant n'importe quelle EOF en amont (64 bits de décalage EOF2 + 216 bits de temps de propagation EOP + 8 bits de temps au repos + 216 bits de temps de propagation SOF + 38 bits de biais EOF1 + 2 bits de biais EOF2). Le point EOF1 est défini sur 560 temps bits pour l'utilisation de la version divisée (par 16) de l'horloge binaire.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.2.5.2 Points de synchronisation EOF1 et EOF2 à vitesse intégrale

Quand le hub fonctionne comme un répéteur à vitesse intégrale/basse vitesse, le point EOF1 est 10 temps bit avant EOF et EOF1 est 32 temps bit avant EOF; voir Figure 11-8.

Le point EOF2 est défini comme se produisant au moins un temps bit avant le premier bit de SYNC pour un SOP. La période autorisée pour un EOP est de quatre temps bit à vitesse intégrale (le port orienté en amont sur un hub est toujours à vitesse intégrale).

Bien que le hub soit synchronisé avec le SOF, le biais de synchronisation peut s'accumuler entre l'hôte et un hub ou entre hubs. Ce biais de synchronisation représente la différence entre différents minuteurs de trame sur différents hubs et l'hôte. Le biais accumulé total peut être égal à ±9 temps bit. Celui-ci est composé de ±1 temps bit par trame d'erreur de quantisation et de ±1 bit par trame de dérapage. L'erreur de quantisation se produit lorsque les temps de hub correspondent à l'intervalle entre SOF et arrive à une valeur qui est arrêtée par une fraction d'un temps bit, mais qui, en raison de la quantisation, est arrondie à un bit entier. Le dérapage de trame se produit lorsque le minuteur de trame de l'hôte est ajusté par le logiciel du système UBS de sorte que la valeur échantillonnée par le hub dans une trame précédente diffère de l'intervalle de trame utilisé par l'hôte. Note: Un tel ajustement a été autorisé dans la spécification USB 1.0 et 1.1 mais n'est plus permis. Ces valeurs s'accumulent sur de multiples trames, car les paquets SOF peuvent être perdus et le hub ne peut pas resynchroniser son minuteur de trame. Cette spécification permet la perte de SOF consécutifs. Pendant cet intervalle, l'erreur de quantisation accumule jusqu'à ±3 temps bit, et le dérapage s'accumule à ±1 ± 2 ± 3 = ±6 pour un total de ±9 temps bit de biais accumulé dans trois trames. Cette synchronisation de biais affecte le positionnement des points EOF1 et EOF2 comme suit.

Un hub doit atteindre son point EOF2 un temps bit avant la fin de la trame. Afin de garantir cela, une bande de garde de 9 temps bit doit être ajoutée de sorte que le point EOF2 est défini comme se produisant si le minuteur de trame locale du hub atteint 10. Un hub doit terminer son EOP avant que le hub auquel il est raccordé n'atteigne son point EOF2. Un hub peut atteindre son point EOF2 neuf temps bit avant le temps bit 10 (au temps bit 19 avant le SOF). Pour garantir que l'EOP est terminé par le temps bit 19, il doit démarrer avant le temps bit 23. Pour garantir que le hub démarre au temps bit 23 par rapport à un autre hub, un hub doit définir son point EOF1 neuf temps bit avant le temps bit 23 (au temps bit 32). Si un hub définit son minuteur pour générer un EOP au temps bit 32, cet EOP peut démarrer 9 temps bit plus tôt (au temps bit 41).

11.3 Comportement de l'hôte en fin de trame

Il incombe au contrôleur de l'hôte USB (l'hôte) de ne pas provoquer une réponse d'un appareil si la réponse entraîne l'envoi par l'appareil d'un paquet au point EOF2. De plus, comme un hub termine un paquet orienté en amont lorsque le hub atteint son point EOF1, il convient que l'hôte ne démarre pas une transaction si une réponse de l'appareil (données ou établissement de liaison) est en suspens ou en cours lorsqu'un hub atteint son point EOF1. Les implications de ces limitations sont décrites dans les sections suivantes.

Note: Les exigences ci-dessus peuvent être satisfaites si le contrôleur hôte garantit que la dernière transaction se termine par son EOF1. Le temps consommé par une transaction (et par conséquent la dernière heure de début de la transaction) peut être évalué en accumulant les divers composants de retard dans la transaction. Il convient que les longueurs de paquets incluent tous les champs et représentent le bourrage de bits maximum comme décrit au Chapitre 7 et au Chapitre 8. Les formules permettant de calculer les temps de transaction sont situées dans la Section 5.11.3.

A la définition des points de synchronisation ci-après, le dernier intervalle de bit dans une (micro)trame est appelé temps bit zéro. Les valeurs des temps bit dans une (micro)trame qui se produisent avant les derniers augmentent; ils proviennent du temps bit zéro (les temps bit plus tôt ont des nombres supérieurs). Ces désignations de temps bit ne sont utilisées que pour des raisons pratiques et ne sont pas destinées à impliquer une mise en œuvre

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

particulière. La seule exigence de mise en œuvre est que les relations temporelles relatives soient préservées.

Les contrôleurs hôtes émettant des transactions à grande vitesse sur un bus à grande vitesse doivent satisfaire aux exigences ci-dessus. Les contrôleurs hôtes émettant des transactions à vitesse intégrale/basse vitesse sur un bus à vitesse intégrale/basse vitesse peuvent également utiliser les trois comportements proches de l'EOF.

11.3.1 Dernier paquet hôte à vitesse intégrale/basse vitesse

Les hubs peuvent envoyer un EOP sur leurs ports orientés en amont au point EOF1 s'il n'y a pas de trafic orienté en aval en cours à ce moment-là. Pour empêcher toute contention potentielle, il n'est pas admis que l'hôte démarre un paquet si la connectivité n'est pas établie sur toutes les connexions avant qu'un hub n'atteigne son point EOF1. Cela signifie que l'hôte ne doit pas démarrer un paquet après le temps bit 42.

Note: Bien qu'il y ait un retard de six bits entre le moment où l'hôte démarre un paquet et où toutes les connexions sont établies, ce temps peut ne pas être ajouté à l'heure de début du paquet, car ce retard de phase existe également pour le paquet SOF, entraînant un retard de phase de tous les minuteurs de trame de hub par rapport à l'hôte par le temps de propagation. Il y a seulement un temps bit de retard de phase entre deux hubs adjacents et celui-ci a été inclus dans les calculs de biais.

11.3.2 Annulation du paquet à vitesse intégrale/basse vitesse

Si un appareil envoie un paquet (données ou établissement de liaison) lorsqu'un hub dans le chemin en amont de l'appareil atteint son point EOF1, le hub envoie un EOP à vitesse intégrale. Tout paquet qui est tronqué par un hub doit être éliminé.

Une mise en œuvre hôte peut éliminer un paquet qui est reçu au temps bit 41. Alternativement, une mise en œuvre hôte peut essayer de maximiser l'utilisation du bus en acceptant un paquet s'il est prévu que le paquet démarre au temps bit 41 ou avant.

11.3.3 Prédiction d'exécution de transaction à vitesse intégrale/basse vitesse

Un appareil peut envoyer deux types de paquets: données et établissement de liaison. Un paquet d'établissement de liaison mesure toujours exactement 16 temps bit de long (octet de synchronisation plus octet PID). Le temps entre la fin d'un paquet de l'hôte et où doit être vu sur l'hôte le premier bit de l'établissement de la liaison est de 17 temps bit. Cela donne une allocation totale de 35 temps bit à partir de la fin du paquet de données à partir de la racine (début de l'EOP) jusqu'à ce qu'il soit prédit que l'établissement de la liaison est terminé (début de l'EOP) à partir de l'appareil. Par conséquent, si l'hôte envoie un paquet de données pour lequel l'appareil peut retourner un établissement de liaison (toute autre chose qu'un paquet isochrone), alors si l'hôte termine le paquet de données et commence à envoyer l'EOP avant le temps bit 76, l'hôte peut alors prédire que l'appareil va terminer l'établissement de la liaison et démarrer l'EOP pour l'établissement de la liaison au moment du temps bit 41 ou avant. Pour un appareil à basse vitesse, les temps bits 36 du début de l'EOP de la racine au début de l'EOP de l'appareil sont des temps bit à basse vitesse, qui convertissent 1 à 8 en temps bit à vitesse intégrale. Par conséquent, si l'hôte complète le paquet de données à basse vitesse par le temps bit 329, on peut prévoir que l'appareil à basse vitesse termine alors l'établissement de la liaison avant le temps bit 41.

Note: Si l'hôte ne peut pas accepter un EOP à vitesse intégrale comme une fin valide d'un paquet à basse vitesse, l'EOP à basse vitesse a alors besoin de terminer avant le temps bit 41, qui va ajouter 13 temps bit à vitesse intégrale au temps d'établissement de la liaison à basse vitesse.

Comme l'hôte approche de la fin de la trame, il doit garantir qu'il n'est pas nécessaire qu'un appareil envoie un établissement de liaison si cet établissement de liaison ne peut pas être terminé avant le temps bit 41. L'hôte attend de recevoir un établissement de liaison après chaque paquet de données valide, non isochrone. Par conséquent, si l'hôte envoie un paquet de données non isochrone lorsqu'il atteint le temps bit 76 (329 pour basse vitesse), il convient

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

alors que l'hôte démarre une séquence de terminaison anormale pour garantir que l'appareil ne va pas essayer de répondre. Cette séquence de terminaison anormale comprend 7 bits consécutifs (non bourrés de bits) de 1, puis un EOP. La séquence de terminaison anormale est envoyée à la vitesse du paquet actuel. Note: Le but de cette séquence doit forcer une violation de bourrage de bits (et éventuellement d'autres erreurs) au niveau du récepteur.

Si l'hôte envisage d'envoyer un jeton IN, il ne peut pas envoyer le jeton si le paquet prévu par l'appareil n'est pas terminé au temps bit 41. La longueur valide maximum de la réponse de l'appareil est connue par l'hôte et il convient de l'utiliser dans le calcul de prédiction. Pour un paquet à vitesse intégrale, l'intervalle maximum entre le début du jeton IN et la fin d'un paquet de données est:

token_length + (packet_length + header + CRC) * 7/6 + 18

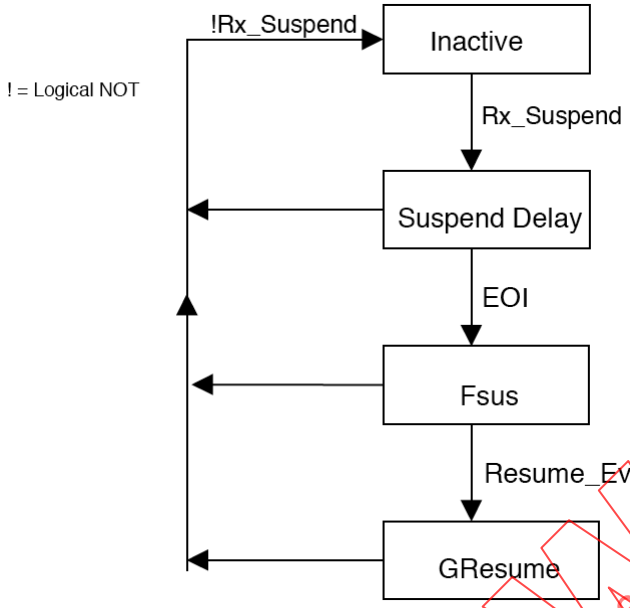
Où *token_length* correspond à 34 temps bit, *packet_length* correspond au nombre maximum de bits de données dans le paquet, *header* correspond à huit bits de synchronisation et huit bits de PID, et CRC correspond à 16 bits. Le multiplicateur 7/6 représente le pire bourrage de bits absolu sur le paquet, et les 18 bits supplémentaires permettent le pire temps de retournement. Pour un appareil à basse vitesse, le même calcul s'applique, mais le résultat doit être multiplié par 8 pour convertir en temps bit à vitesse intégrale, et 20 temps bit à vitesse intégrale supplémentaires doivent être ajoutés pour représenter le préfixe à basse vitesse. Cela donne le nombre maximum de temps bit entre le début du jeton IN et la fin du paquet de données, de sorte que le jeton ne peut pas être envoyé si ce nombre de temps bit n'existe pas avant le point EOF1 le plus tôt (temps bit 41). (Par exemple, prendre les résultats des calculs ci-dessus et ajouter 41. Si le nombre de bits laissés dans la trame est inférieur à cette valeur, le jeton peut ne pas être envoyé.)

L'hôte peut utiliser un algorithme plus prudent que celui ci-dessus pour décider si une transaction doit ou non être démarrée. Le calcul peut également inclure le temps nécessaire à l'hôte pour envoyer l'établissement de la liaison lorsque cela est requis, car il n'y a aucun avantage à démarrer un transfert si l'établissement de la liaison ne peut pas être terminé.

11.4 Port interne

Le port interne est la connexion entre le contrôleur de hub et le répéteur de hub. Outre le transport des données série vers/du contrôleur de hub, le port interne est la source de certains signaux de reprise. La Figure 11-9 illustre le diagramme d'états du port interne; le Tableau 11-4 définit les signaux et évènements du port interne.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
! = Logical NOT	! = NON logique
Inactive	Inactif
Suspend Delay	Retard suspension

Figure 11-9. Diagramme d'états du port interne

Tableau 11-4. Définitions des signaux/évènements du port interne

Nom de signal/d'évènement	Source d'évènement/de signal	Description
EOI	Interne	Fin de l'intervalle temporisé
Rx_Suspend	Récepteur	Le récepteur est dans l'état Suspendu
Resume_Event	Contrôleur de hub	Une condition de reprise existe dans le contrôleur de hub

11.4.1 Inactif

Cet état est accessible lorsque le récepteur n'est pas dans l'état Suspendu.

11.4.2 Retard Suspension

Cet état existe à partir de l'état Inactif lorsque le récepteur passe à l'état Suspendu.

Il s'agit d'un état temporisé avec un intervalle de 2 ms.

11.4.3 Suspension Totale (Fsus)

Cet état existe lorsque l'intervalle Retard Suspension expire.

11.4.4 Générer Reprise (GResume)

Cet état existe à partir de l'état Fsus lorsqu'une condition de reprise existe dans le contrôleur de hub. Une condition de reprise existe si le bit C_PORT_SUSPEND est défini dans n'importe quel port, ou si le hub est activé comme source de réveil et un bit est défini dans un champ Port Change ou dans le champ Hub Change (comme décrit dans les Figures 11-22 et 11-20, respectivement).

Dans cet état, le port interne génère la signalisation pour émuler un SOP_FD vers le répéteur de hub.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

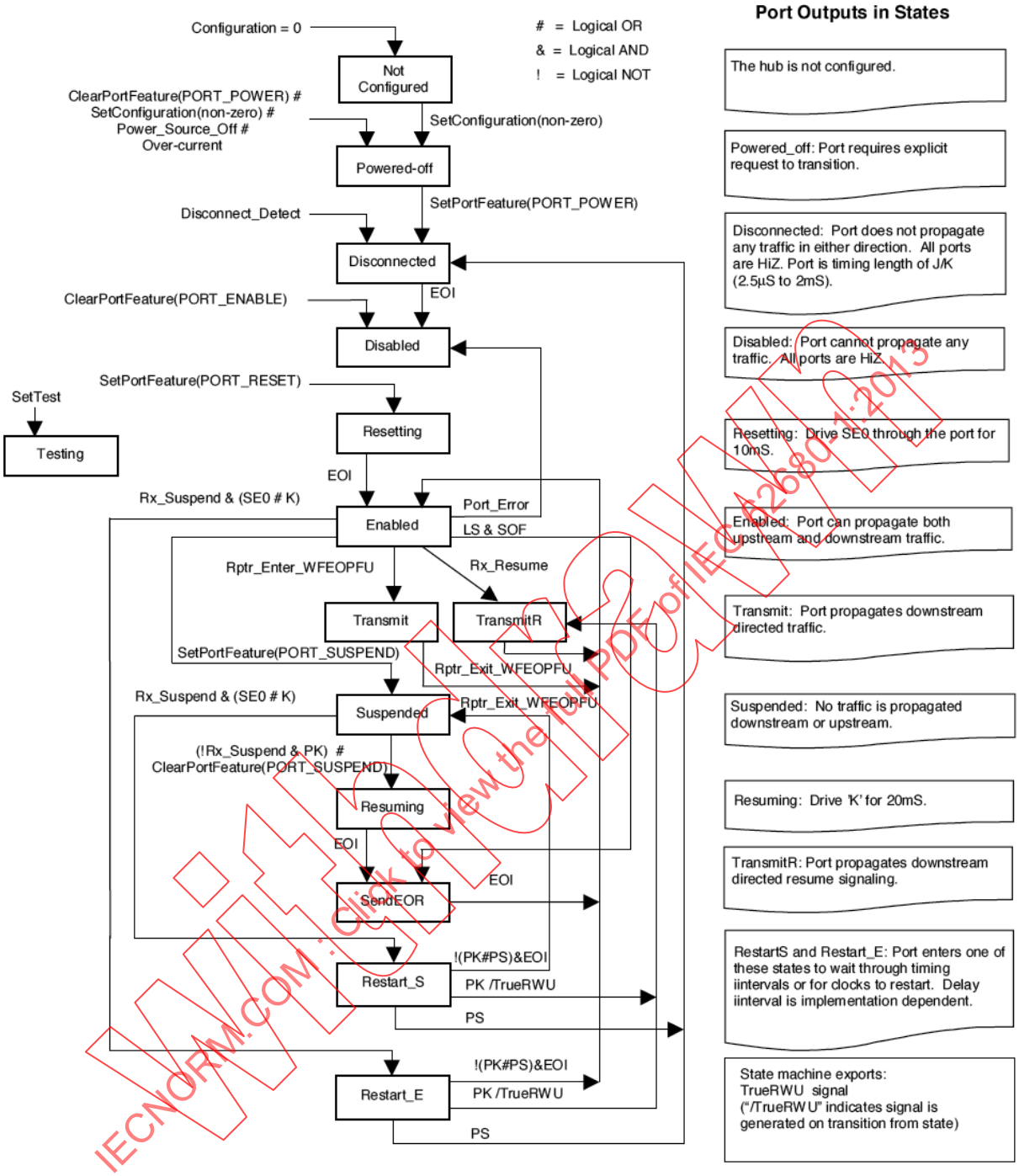
11.5 Ports orientés en aval

Les sections suivantes fournissent une description fonctionnelle d'un diagramme d'états qui présente le comportement correct pour un port orienté en aval.

La Figure 11-10 est une illustration du diagramme d'états du port orienté en aval. Les évènements et signaux sont définis dans le Tableau 11-5. Chacun des états est décrit dans la Section 11.5.1. Dans le schéma ci-après, certaines conditions d'entrée dans des états sont indiquées sans origine. Ces conditions ont de multiples états d'origine et les lignes de transitions individuelles ne sont pas illustrées afin que le schéma puisse être simplifié. La description de l'état actuel indique à partir de quels états s'applique la transition.

Note: Pour le hub racine, les signaux des diagrammes d'états du port orienté en amont dépendent de la mise en œuvre.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Not Configured	Non configuré
Powered-Off	Hors tension
Disconnected	Déconnecté
Disabled	Désactivé
Resetting	Réinitialisation
Enabled	Activé
Transmit	Transmission
Suspended	Suspendu
Resuming	Reprise
Testing	Essais
Port Outputs in States	Sorties de port dans les états
The hub is not configured.	Le hub n'est pas configuré.
Powered_off: Port requires explicit request to transition.	Hors tension: Le port nécessite une demande de transition explicite.

Anglais	Français
Disconnected: Port does not propagate any traffic in either direction. All ports are HiZ. Port is timing length of J/K (2.5µS to 2mS).	Déconnecté: Le port ne propage pas de trafic dans une des directions. Tous les ports sont HiZ. Le port synchronise la longueur de J/K (2,5µS à 2 mS).
Disabled: Port cannot propagate any traffic. All ports are HiZ.	Désactivé: Le port ne peut pas propager de trafic. Tous les ports sont HiZ.
Resetting: Drive SE0 through the port for 10mS.	Réinitialisation: Pilote SE0 dans le port pendant 10 mS.
Enabled: Port can propagate both upstream and downstream traffic.	Activé: Le port ne peut pas propager le trafic en amont et en aval.
Transmit: Port propagates downstream directed traffic.	Transmission: Le port propage le trafic orienté en aval.
Suspended: No traffic is propagated downstream or upstream.	Suspendu: Aucun trafic n'est propagé en aval ou en amont.
Resuming: Drive 'K' for 20mS.	Reprise: Pilote 'K' pendant 20 mS.
TransmitR: Port propagates downstream resume signaling.	TransmitR: Le port propage la signalisation de reprise en aval.
RestartS and Restart_E: Port enters one of theses states to wait through timing intervals or for clocks to restart. Delay interval is implementation dependant.	RestartS et Restart_E: Le port entre dans un de ces états pour attendre lors des intervalles de synchronisation ou pour le redémarrage des horloges. L'intervalle de retard dépend de la mise en œuvre.
State machine exports: TrueRWU signal ("/TrueRWU" indicates signal is generated on transition from state)	Exports de diagrammes d'états: signal TrueRWU ("/TrueRWU" indique que le signal est généré lors du passage de l'état)

Figure 11-10. Diagramme d'état du port du hub orienté en aval

Tableau 11-5. Définitions des signaux/évènements du port orienté en aval

Nom de signal/d'évènement	Source d'évènement/de signal	Description
Power_source_off	En fonction de la mise en œuvre	Alimentation du port indisponible en raison d'une surintensité ou de la coupure de l'alimentation source (ex.: alimentation externe coupée)
Over-current	Contrôleur de hub	Une condition de surintensité existe sur le hub ou le port
EOI	Interne	Fin d'un intervalle ou d'une séquence temporisé(e)
SE0	Interne	SE0 reçu sur le port
Disconnect_Detect	Interne	Déconnexion observée au niveau du port
LS	Contrôleur de hub	Appareil basse vitesse relié à ce port
SOF	Contrôleur de hub	Jeton SOF reçu
TrueRWU	Interne	K durant pendant au moins TDDIS (voir Tableau 7-13)
PK	Interne	K durant pendant au moins TDDIS
PS	Interne	SE0 durant pendant au moins TDDIS
K	Interne	"K" reçu au niveau du port
Rx_Resume	Récepteur	Récepteur en amont à l'état Reprise
Rx_Suspend	Récepteur	Récepteur en amont à l'état Suspension
Rptr_Exit_WFEOPFU	Répéteur de hub	Le répéteur de hub quitte l'état WFEOPFU
Rptr_Enter_WFEOPFU	Répéteur de hub	Le répéteur de hub entre dans l'état WFEOPFU
Port_Error	Interne	Condition d'erreur détectée (voir Section 11.8.1)
SetTest	Contrôleur de hub	OU logique de SetPortFeature(Test_SE0_NAK), SetPortFeature(Test_J), SetPortFeature(Test_K), SetPortFeature(Test_PRBS), SetPortFeature(Test_Force_Enable)
Configuration = 0	Contrôleur de hub	La valeur de configuration du contrôleur de hub est zéro

11.5.1 Descriptions d'état du port orienté en aval

11.5.1.1 Non configuré

Un port passe et reste dans cet état si la valeur de la configuration du hub est zéro. Tandis que le port est dans cet état, le hub conduit un SE0 sur le port (ce comportement est facultatif sur les hubs racines). Aucune autre signalisation active n'a lieu sur le port lorsqu'il est dans cet état.

11.5.1.2 Hors tension

Cet état est pris en charge pour tous les hubs.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Un port passe dans cet état dans une des situations suivantes:

- De n'importe quel état sauf Non configuré lorsque le hub reçoit une demande ClearPortFeature(port_power) pour ce port
- De n'importe quel état lorsque le hub reçoit une demande SetConfiguration() avec une valeur de configuration différente de zéro
- De n'importe quel état sauf Non configuré en cas de perte d'alimentation vers le port ou d'existence d'une condition de surintensité

Un port entre dans cet état à cause d'une condition de surintensité sur un autre port si cette condition peut avoir entraîné une chute de l'alimentation fournie vers ce port au-dessous des limites spécifiées pour l'alimentation du port (voir Section 7.2.1.2.1 et Section 7.2.4.1).

Si un hub a été configuré alors que le hub était autoalimenté, et si l'alimentation externe est alors perdue, le hub doit placer tous les ports dans l'état Hors tension. Si le hub est configuré alors que le bus est alimenté, le hub peut alors ne pas modifier le statut du port si le hub a commuté vers l'alimentation externe. Mais, si l'alimentation externe est ensuite perdue, le hub doit placer les ports dans l'état Hors tension.

Dans cet état, les émetteurs et récepteurs différentiels et à terminaison unique du port sont désactivés.

Le contrôle de l'alimentation du port est couvert dans la Section 11.11.

11.5.1.3 Déconnecté

Un port passe dans cet état dans une des situations suivantes:

- A partir de l'état Hors tension quand le hub reçoit une demande SetPortFeature(PORT_POWER)
- A partir de n'importe quel état sauf les états Non configuré et Hors tension si le minuteur de déconnexion du port temporise
- A partir de l'état Restart_S ou Restart_E à la fin de l'intervalle de redémarrage

Dans l'état Déconnecté, l'émetteur et le récepteur de différentiel du port sont désactivés et seule la détection de connexion est possible.

Il s'agit d'un état temporisé. Lorsqu'elle est dans cet état, le minuteur est réinitialisé tant que les lignes de signal du port sont dans l'état SE0 ou SE1. Si un autre état de signalisation est détecté, le minuteur démarre. Sauf si le hub est suspendu avec les horloges arrêtées, la durée de ce minuteur est de 2,5 µs à 2 ms.

Si le hub est suspendu avec sa caractéristique de réveil à distance activée, puis en transition vers un autre état que l'état SE0 ou SE1 sur un port Déconnecté, le hub démarre ses horloges et chronomètre cet évènement. Le hub doit pouvoir démarrer ses horloges et chronométrer cet évènement dans les 12 ms de la transition. Si un hub n'a pas sa caractéristique de réveil à distance activée, puis passe sur un port qui est dans l'état Déconnecté, il est ignoré jusqu'à ce que le hub ait repris.

11.5.1.4 Désactivé

Un port passe dans cet état dans une des situations suivantes:

- De l'état Déconnecté lorsque le minuteur expire, indiquant qu'une connexion est détectée sur le port
- De n'importe quel état sauf les états Hors tension, Déconnecté ou Non configuré à la réception d'une demande ClearPortFeature(PORT_ENABLE)
- De l'état Activé lorsqu'une condition d'erreur est détectée sur le port

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Un port dans l'état Désactivé ne propage pas la signalisation en amont ou en aval. Lorsqu'il est dans cet état, la durée de tout SE0 reçu sur le port est temporisée. Si le port utilise des terminaisons à grande vitesse lorsqu'il entre dans cet état, il commute sur les terminaisons à vitesse intégrale. Le port ne doit pas effectuer de détection de déconnexion normale pendant au moins 4 ms après être entré dans cet état.

11.5.1.5 Réinitialisation

A moins d'être dans l'état Hors tension ou Déconnecté, un port passe dans l'état Réinitialisation à la réception d'une demande SetPortFeature(PORT_RESET). Le hub pilote SE0 sur le port pendant cet intervalle temporisé. La durée de l'état Réinitialisation est nominalement comprise entre 10 ms et 20 ms (10 ms de préférence).

Un hub en mode grande vitesse utilise les terminaisons à grande vitesse du port lorsqu'il est dans cet état.

11.5.1.6 Activé

Un port passe dans cet état dans une des situations suivantes:

- A la fin de l'état Réinitialisation
- A partir de l'état Transmettre ou de l'état TransmitR lorsque le répéteur de hub quitte l'état WFEOPFU
- A partir de l'état Suspendu si le récepteur en amont est dans l'état Suspendre lorsqu'un "K" est détecté sur le port
- A la fin de l'état SendEOR
- A partir de l'état Restart_E lorsqu'un K persistant K ou SE0 persistant SE0 n'a pas été vu dans les 900 μs qui suivent l'entrée dans cet état

Dans cet état, la sortie du récepteur différentiel du port est disponible pour le répéteur de hub de sorte que les transitions de signalisation appropriées peuvent établir une connectivité en amont.

Un port qui utilise des terminaisons à grande vitesse dans cet état commute sur les terminaisons à vitesse intégrale sur Rx_Suspend (à savoir, lorsque le hub est suspendu). Le port ne doit pas effectuer de détection de déconnexion normale pendant au moins 1 ms après l'activation de Rx_Suspend.

11.5.1.7 Transmettre

Cet état existe à partir de l'état Activé lors de la transition du répéteur de hub vers l'état WFEOPFU. Dans cet état, le port transmet les données reçues sur le port orienté en amont.

Pour un port à basse vitesse, cet état existe à partir de l'état Activé si un PRE PID à vitesse intégrale est reçu sur le port orienté en amont. Dans cet état, le port retransmet les données qui sont reçues sur le port orienté en amont (après inversion correcte).

A grande vitesse, cet état est utilisé pour soumettre à essai la déconnexion au niveau du port. Le circuit de détection de déconnexion est activé après que 32 bits du même niveau de signalisation ("J" ou "K") aient été transmis vers le port.

Note: En raison du biais de synchronisation dans le chemin du répéteur vers les ports orientés en aval, tous les ports orientés en aval peuvent ne pas être activés pour la détection de la déconnexion au même moment.

11.5.1.8 TransmitR

Cet état existe dans une des situations suivantes:

- A partir de l'état Activé si le récepteur en amont est dans l'état Reprise
- A partir de l'état Restart_S ou Restart_E si un PK est détecté sur le port

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Dans cet état, le port répète le "K" de reprise sur le port orienté en amont vers le port orienté en aval. Selon la vitesse du port, deux comportements sont possibles sur la transition K->SE0 sur le port orienté en amont à la fin de la reprise.

- Port orienté en amont à grande vitesse et port orienté en aval à vitesse intégrale/basse vitesse: Après la transition K->SE0, le port pilote SE0 pour 16 à 18 temps bit à vitesse intégrale puis pilotage de J pendant au moins un temps bit à vitesse intégrale. Note: Le minuteur dans l'état Reprise du diagramme d'état du récepteur du port en amont qui génère EOITR peut être utilisé pour chronométrer cette exigence sur le ou les ports orientés en aval. La résistance de polarisation et la latence du traducteur de transactions (TT) entraînent le maintien dans cet état Repos pendant au moins un temps bit à basse vitesse, garantissant qu'un appareil voit la même fin de comportement de reprise au-dessous du TT que si c'était au-dessous d'un hub USB 1.x.
- Le port orienté en amont et le port orienté en aval ont la même vitesse: le port continue de répéter la signalisation qui suit la transition K->SE0.

Un port fonctionnant à grande vitesse retourne à ses terminaisons à grande vitesse dans un délai de 18 temps bit à vitesse intégrale après la transition K->SE0 comme décrit dans la Section 7.1.7.7.

11.5.1.9 Suspendu

Un port entre dans l'état Suspendu:

- A partir de l'état Activé lorsqu'il reçoit une demande SetPortFeature(port_suspend)
- A partir de l'état Restart_S lorsqu'un K persistant K ou SE0 persistant SE0 n'a pas été vu dans les 900 µs qui suivent l'entrée dans cet état.

Lorsqu'un port est dans l'état Suspendu, l'émetteur différentiel du port est désactivé. Un port à grande vitesse retourne des terminaisons à grande vitesse aux terminaisons à vitesse intégrale, mais son statut de vitesse continue à être à grande vitesse. Le port ne doit pas effectuer de détection de déconnexion normale pendant au moins 4 ms après être entré dans cet état.

Une mise en œuvre doit avoir un filtre de "bruit" K/SE0 pour un port qui est dans l'état suspendu. Ce filtre peut chronométrer la longueur de K/SE0 et, si la longueur de K/SE0 est plus courte que TDDIS, le port doit rester dans cet état. Si le hub est suspendu avec ses horloges arrêtées, une transition vers K/SE0 sur un port suspendu doit entraîner le passage immédiat du port dans l'état Restart_S.

11.5.1.10 REPRISE

Un port entre dans cet état à partir de l'état Suspendu dans une des situations suivantes:

- Si un "K" est détecté sur le port et persiste pendant au moins 2,5 µs et le récepteur n'est pas dans l'état Suspendu. La transition de l'état Suspendu doit avoir lieu dans les 900 µs qui suivent la transition J->K.
- Lorsqu'une demande ClearPortFeature(PORT_SUSPEND) est reçue.

Il s'agit d'un état temporisé avec une durée nominale de 20 ms (l'intervalle peut être plus long dans les conditions de la note ci-après). Dans cet état, le hub pilote un "K" sur le port.

Note: Un seul minuteur peut être utilisé pour chronométrer l'intervalle de réinitialisation et l'intervalle de reprise et ce minuteur peut être partagé entre de multiples ports. Lorsqu'il est partagé, le minuteur est réinitialisé lorsqu'un port entre dans l'état Reprise ou dans l'état Réinitialisation. S'il est partagé, il peut ne pas être partagé entre plus de dix ports, car le retard cumulé pourrait dépasser le temps maximum requis pour remplacer un appareil et une déconnexion pourrait être ratée.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.5.1.11 SendEOR

Cet état est accessible à partir de l'état Reprise si le minuteur 20 ms expire. Il est également accessible à partir de l'état Activé lorsqu'un SOF (ou autre jeton FS) est reçu et un appareil à basse vitesse est relié à ce port.

Il s'agit d'un état temporisé qui dure pendant trois temps bit à basse vitesse.

Dans cet état, si le port est à grande vitesse, il pilote le bus dans l'état Repos pendant trois temps bit à basse vitesse puis quitte cet état pour aller dans l'état Activé. Il doit également retourner à ses terminaisons à grande vitesse dans un délai de 18 temps bit à vitesse intégrale après la transition K->SE0 comme décrit dans la Section 7.1.7.7.

Si le port est à vitesse intégrale ou à basse vitesse, le port doit piloter deux temps bit à basse vitesse de SE0 suivis d'un temps bit à basse vitesse d'état Repos, puis quitter cet état pour aller dans l'état Activé.

Comme il convient que la période SE0 pilotée soit de longueur fixe, il convient que le minuteur SendEOR, si partagé, ne soit pas réinitialisé. Si la mise en œuvre du hub partage les circuits de synchronisation SendEOR entre les ports, puis pour un port avec un appareil à basse vitesse relié, il convient que l'état Reprise ne termine pas tant qu'un SOF (ou autre jeton FS) n'a pas été reçu (voir Section 11.8.4.1 pour les règles de génération de maintien actif).

11.5.1.12 Restart_S

Un port entre dans l'état Restart_S à partir de l'état Suspendu lorsqu'un SE0 ou "K" est vu au niveau du port et le récepteur est dans l'état Suspendu.

Dans cet état, le port surveille en continu l'état du bus. Si le bus est dans l'état "K" pendant au moins TDDIS, le port définit le bit C_PORT_SUSPEND, va vers TransmitR et génère un signal vers le répéteur appelé "TrueRWU". Si le bus est dans l'état "SE0" pendant au moins TDDIS, le port passe à l'état Déconnecté. Une de ces transitions doit se produire dans les 900 µs après l'entrée dans le port Restart_S; sinon, le port doit retourner dans l'état Suspendu.

11.5.1.13 Restart_E

Un port entre dans l'état Restart_E à partir de l'état Activé lorsqu'un SE0 ou "K" est vu au niveau du port et le récepteur est dans l'état Suspendu.

Dans cet état, le port surveille en continu l'état du bus. Si le bus est dans l'état "K" pendant au moins TDDIS, le port quitte l'état TransmitR et génère un signal vers le répéteur appelé "TrueRWU". Si le bus est dans l'état "SE0" pendant au moins TDDIS, le port passe à l'état Déconnecté. Une de ces transitions doit se produire dans les 900 µs après l'entrée dans le port Restart_E; sinon, le port doit retourner dans l'état Activé.

11.5.1.14 Essais

Un port entre dans cet état à partir de n'importe quel état lorsque le port voit SetTest.

Dans cet état, le port exécute la commande hôte comme décodée par le contrôleur de hub. Si la commande était SetPortFeature(PORT_TEST, Test_Force_Enable), le port prend en charge la connectivité de paquets en aval de manière identique à celle lorsque le port est dans l'état Activé.

11.5.2 Minuteur de détection de déconnexion

11.5.2.1 Détection de déconnexion à grande vitesse

La détection de déconnexion à grande vitesse est décrite dans la Section 7.1.7.3.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.5.2.2 Détection de déconnexion à vitesse intégrale/basse vitesse

Chaque port doit avoir un minuteur utilisé pour détecter la déconnexion lorsqu'un appareil à vitesse intégrale/basse vitesse est relié au port. Ce minuteur est utilisé pour surveiller en continu les récepteurs à terminaison unique du port et détecter un évènement de déconnexion. La raison d'une surveillance constante est qu'un évènement de bruit sur le bus peut entraîner la détection par l'appareil relié d'une condition de réinitialisation sur le bus après 2,5 µs de SE0 ou SE1 sur le bus. Si le hub ne met pas le port dans l'état déconnexion avant la réinitialisation de l'appareil, celui-ci peut être dans l'état Adresse par défaut avec le port activé, ce qui peut provoquer des erreurs système très difficiles à isoler et à corriger.

Ce minuteur doit être réinitialisé lorsque les lignes D+ et D- sur le port ne sont pas dans l'état SE0 ou SE1 ou lorsque le port n'est pas dans l'état Activé, Suspendu, Désactivé, Restart_E ou Restart_S. Ce minuteur doit être réinitialisé pendant 4 ms lors du passage dans les états Suspendu et Désactivé. Ce minuteur chronomètre un intervalle TDDIS. La plage de TDDIS est comprise entre 2,0 µs et 2,5 comme défini dans le Tableau 7-13. Lorsque ce minuteur expire, il génère le signal Disconnect_Detect au diagramme d'états du port.

Ce minuteur peut également être utilisé pour filtrer le signal K/SE0 dans les états Suspendu, Restart_E ou Restart_S comme décrit dans la Section 11.5.1.

11.5.3 Indicateur de port

Chaque port orienté en aval d'un hub peut prendre en charge un indicateur d'état facultatif. La présence d'indicateurs pour les ports orientés en aval est spécifiée par le bit 7 du champ *wHubCharacteristics* du descripteur de classe de hub. Chaque indicateur de port doit être situé dans une position qui associe évidemment l'indicateur au port. L'indicateur a deux couleurs: vert et orange. Il peut être mis en œuvre comme physiquement une LED deux couleurs ou deux LED séparées. Une combinaison de commande matérielle et logicielle est utilisée pour informer l'utilisateur d'état actuel du port ou de l'appareil relié au port et pour guider l'utilisateur lors de la résolution des problèmes. Les couleurs et le clignotement sont utilisés pour fournir des informations à l'utilisateur.

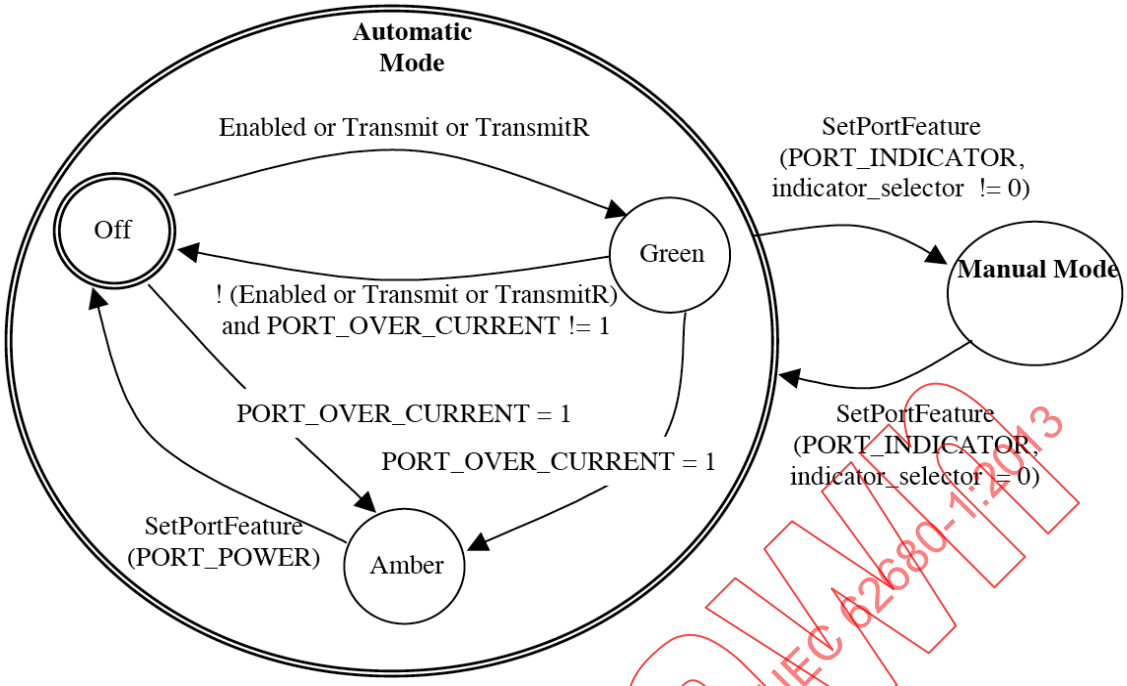
Un hub externe doit automatiquement commander la couleur de l'indicateur comme spécifié dans la Figure 11-11. La prise en charge de la configuration automatique de l'indicateur de port pour les hubs racines peut être mise en œuvre avec le matériel ou le logiciel. La valeur du sélecteur d'indicateur de port est zéro (indiquant la commande automatique) lorsque le hub passe à l'état d'appareil configuré. Lorsque le hub est suspendu ou non configuré, les indicateurs de port doivent être désactivés.

Le Tableau 11-6 identifie le mapping de la couleur sur l'état du port lorsque les indicateurs de port sont automatiquement commandés.

Tableau 11-6. Etat du port automatique sur le mapping de la couleur de l'indicateur de port

Mise sous tension	Etat du port du hub orienté en aval			
	Hors tension	Déconnecté, Désactivé, Non configuré, Réinitialisation, Essais	Activé, Transmit ou TransmitR	Suspendu, Reprise, SendEOR, Restart_E ou Restart_S
Avec	Eteint ou orange en raison d'une condition de surintensité	Eteint	Vert	Eteint
Sans	Eteint	Eteint ou orange en raison d'une condition de surintensité	Vert	Eteint

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Automatic Mode	Mode automatique
Enabled or Transmit or TransmitR	Activé ou Transmettre ou TransmitR
Off	Eteint
Green	Vert
Amber	Orange
Manual Mode	Mode manuel
! (Enabled or Transmit or TransmitR) and PORT_OVER_CURRENT ! =1	! (Activé ou Transmettre ou TransmitR) et PORT_OVER_CURRENT ! =1

Figure 11-11. Schéma d'états de l'indicateur de port

En **Mode Manuel**, la couleur d'un indicateur de port (Orange, Vert ou Eteint) est définie par une demande de classe de hub USB de logiciel système. En **Mode Automatique**, la couleur d'un indicateur de port est définie par les informations sur l'état du port.

Le Tableau 11-7 définit l'état du port tel que compris par l'utilisateur.

Tableau 11-7. Définitions des couleurs de l'indicateur de port

Couleur	Définition
Eteint	Non opérationnel
Orange	Condition d'erreur
Vert	Totalement opérationnel
Clignote Eteint/Vert	Attention logicielle
Clignote Eteint/Orange	Attention matérielle
Clignote Vert/Orange	Réservé

Il faut noter que les indicateurs reflètent l'état du port, pas nécessairement l'appareil relié à celui-ci. Le clignotement de l'indicateur permet d'attirer l'attention de l'utilisateur sur le port, quelle que soit sa couleur.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Les indicateurs de port permettent la commande par logiciel. Le logiciel hôte force l'état de l'indicateur à attirer l'attention sur le port ou à indiquer l'état actuel du port.

Voir Section 11.24.2.7.1.10 pour la spécification des demandes d'indicateur.

11.5.3.1 Etiquetage

Le logiciel du système USB utilise les numéros de port pour référencer un port individuel avec une demande ClearPortFeature ou SetPortFeature. Si un fournisseur fournit un étiquetage pour identifier les différents ports orientés en aval, chaque connecteur de port doit alors être identifié avec son numéro de port respectif.

11.6 Port orienté en amont

Le port orienté en amont a quatre composants: émetteur, diagramme d'états de l'émetteur, récepteur et diagramme d'états du récepteur. L'émetteur et son diagramme d'état sont l'émetteur tandis que le récepteur et son diagramme d'états sont le récepteur. L'émetteur et le récepteur fonctionnent à grande vitesse et à vitesse intégrale selon la configuration actuelle du hub.

11.6.1 Vitesse intégrale

L'émetteur et le récepteur ont des composants différentiels et à terminaison unique. L'émetteur et le récepteur différentiels peuvent envoyer/recevoir "J" ou "K" vers/du bus tandis que les composants à terminaison unique sont utilisés pour envoyer/recevoir la signalisation SE0, de suspension et de reprise. Les composants à terminaison unique sont également utilisés pour recevoir SE1. Dans cette section, lorsqu'il est nécessaire de différencier les signaux envoyés/reçus par le composant différentiel de l'émetteur/du récepteur de ceux des composants à terminaison unique, DJ et DK sont utilisés pour identifier le signal différentiel, tandis que SJ, SK, SE0 et SE1 sont utilisés pour les signaux à terminaison unique.

Lorsque le répéteur de hub a une connectivité en amont, l'émetteur ne doit pas envoyer ou propager la signalisation SE1. Le SE1 doit être propagé comme DJ.

11.6.2 Grande vitesse

L'émetteur et le récepteur ont des composants différentiels uniquement. Ces signaux sont appelés HJ et HK. L'état HS_Idle est l'état inactif du bus à grande vitesse.

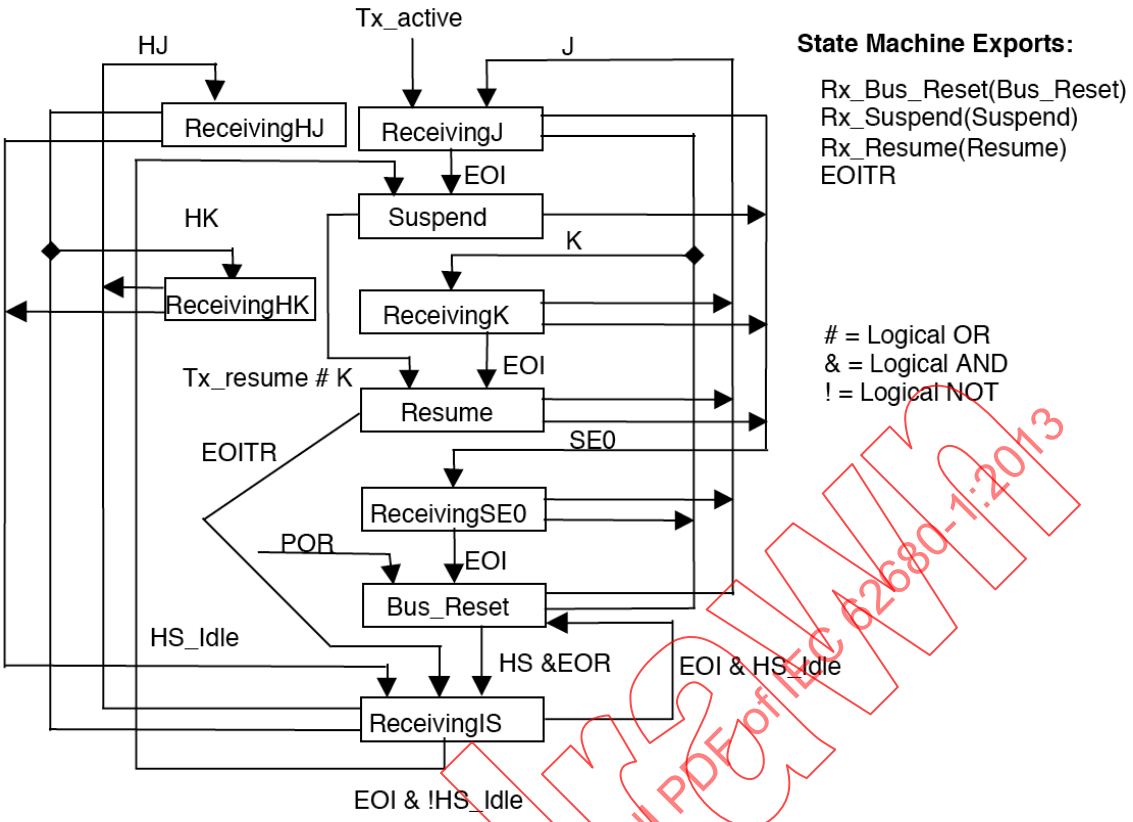
On suppose que l'émetteur et le récepteur différentiels sont éteints pendant la suspension pour minimiser la consommation électrique. Les composants à terminaison simple sont laissés sous tension à tout moment, car ils consomment peu.

11.6.3 Récepteur

Le diagramme d'états du récepteur est chargé de surveiller l'état de signalisation de la connexion en amont pour détecter les évènements de signalisation à long terme comme la réinitialisation, la reprise et la suspension de bus. Ce diagramme d'états détaille le fonctionnement du diagramme d'états de l'appareil illustré dans la Figure 9-1 dans l'état Par défaut, Adresse, Configuré et Suspendu. Les états Suspension, Reprise et ReceivingSE0 sont uniquement utilisés lorsque le port orienté en amont fonctionne en mode vitesse intégrale avec des terminaisons à vitesse intégrale. Les états ReceivingIS, ReceivingHJ et ReceivingHK sont uniquement utilisés lorsque le port orienté en amont fonctionne en mode vitesse intégrale avec des terminaisons à grande vitesse; de sorte que ces états sont classés comme des états HS (high-speed ou grande vitesse), et tous les autres états sont classés comme non-HS dans la description ci-après.

La Figure 11-12 illustre le diagramme de transition d'états.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
State Machine Exports:	Exports des diagrammes d'états:
# = Logical OR	# = OU logique
& = Logical AND	& = ET logique
! = Logical NOT	! = NON logique
Suspend	Suspension
Resume	Reprise

Figure 11-12. Diagramme d'états du récepteur du port orienté en amont

Le Tableau 11-8 définit les signaux et évènements référencés dans les figures.

Tableau 11-8. Définitions des signaux/évènements du récepteur du port orienté en amont

Nom de signal/d'évènement	Source d'évènement/de signal	Description
HS	Interne	Le port fonctionne à grande vitesse
Tx_active	Emetteur	Emetteur à l'état Actif
J	Interne	Réception d'un "J" (INACTIF) ou d'un "SE1" sur le port orienté en amont
HJ	Interne	Réception d'un HJ sur le port orienté en amont
EOI	Interne	Fin de l'intervalle temporisé
EOITR	Interne	24 temps bit à vitesse intégrale générés après la transition K->SE0 à la fin de la reprise
HK, K	Interne	Réception d'un HK, "K" sur le port orienté en amont
Tx_resume	Emetteur	L'émetteur est dans l'état Reprise
HS_Idle	Interne	Réception d'un état Repos sur le port orienté en amont à grande vitesse

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Nom de signal/d'évènement	Source d'évènement/de signal	Description
SE0	Interne	Réception d'un SE0 sur le port orienté en amont à vitesse intégrale
EOR	Interne	Signalisation fin de reprise de l'amont
POR	En fonction de la mise en œuvre	Power_On_Reset

11.6.3.1 ReceivingIS

Cet état est activé

- De l'état ReceivingHJ ou ReceivingHK lorsqu'un SE0 est vu au niveau du port et le port est en mode grande vitesse
- De l'état Reprise lorsqu'un EOITR est vu au niveau du port et le port est en mode grande vitesse
- De l'état Réinitialisation du bus au niveau de la signalisation Fin de réinitialisation de l'amont lorsque le port est en mode grande vitesse

Il s'agit d'un état temporisé avec un intervalle de 3 ms. Le minuteur est réinitialisé chaque fois que cet état est activé.

11.6.3.2 ReceivingHJ

Cet état est accessible à partir d'un état HS lorsqu'un HJ est vu sur le bus.

11.6.3.3 ReceivingJ

Cet état est accessible à partir d'un état non HS sauf l'état Suspension si le récepteur détecte une condition SJ (ou Repos) ou SE1 sur le bus ou tandis que l'émetteur est dans l'état Actif.

Il s'agit d'un état temporisé avec un intervalle de 3 ms. Le minuteur est réinitialisé chaque fois que cet état est activé.

Le minuteur avance uniquement si l'émetteur est dans l'état Repos.

11.6.3.4 Suspension

Cet état est accessible lorsque:

- Le minuteur 3 ms expire dans ReceivingJ
- Le minuteur 3 ms expire dans l'état ReceivingIS, le port a retiré ses terminaisons à grande vitesse et a connecté sa résistance de polarisation D+, l'état de bus n'est pas SE0.

Lorsque le récepteur entre dans cet état, le contrôleur de hub démarre un minuteur 2 ms. Si ce minuteur expire alors que le récepteur est encore dans cet état, le contrôleur de hub est alors suspendu. Lorsque le contrôleur de hub est suspendu, il peut générer la signalisation de reprise.

11.6.3.5 ReceivingHK

Cet état est accessible à partir d'un état HS lorsqu'un HK est vu sur le bus.

11.6.3.6 ReceivingK

Cet état est accessible à partir de l'état non HS sauf l'état Reprise lorsque le récepteur détecte une condition SK sur le bus et le répéteur de hub est dans l'état WFSOP ou WFSOPFU.

Il s'agit d'un état temporisé avec une durée de 2,5 μs à 100 μs. Le minuteur est réinitialisé chaque fois que cet état démarre.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.6.3.7 Reprise

Cet état est activé:

- De l'état ReceivingK lorsque le minuteur expire
- De l'état Suspension alors que l'émetteur est dans l'état Sresume ou s'il y a une transition vers l'état K sur le port orienté en amont

Si le hub entre dans cet état lorsque sa référence de synchronisation n'est pas disponible, le hub peut rester dans cet état jusqu'à ce que la référence de synchronisation du hub devienne stable (les références de synchronisation doivent se stabiliser en moins de 10 ms). Si cet état est maintenu en cours de stabilisation de l'horloge du hub, le récepteur doit fournir un K au répéteur pour la propagation vers les ports orientés en aval. Lorsque les horloges sont stables, le récepteur doit répéter les signaux d'entrée.

Note: Les références de synchronisation de hub sont stables en moins de 10 ms, car les exigences de réinitialisation spécifient déjà qu'elles sont stables en moins de 10 ms et qu'un hub doit prendre en charge la réinitialisation à partir de la suspension.

11.6.3.8 ReceivingSE0

Cet état est accessible à partir de l'état non HS sauf Bus_Reset lorsque le récepteur détecte une condition SE0 et le répéteur de hub est dans l'état WFSOP ou WFSOPFU.

Il s'agit d'un état temporisé. L'intervalle minimum pour cet état est de 2,5 µs. Le maximum dépend du hub, mais cet intervalle doit expirer suffisamment tôt de sorte que si la largeur de SE0 sur le port orienté en amont est seulement de 10 ms, le récepteur entre dans l'état Bus_Reset avec suffisamment de temps restant dans l'intervalle de 10 ms pour que le hub termine son traitement de réinitialisation. De plus, si le hub est suspendu lorsque le récepteur entre dans cet état, le hub doit être capable de démarrer ses horloges, de chronométrer cet intervalle et de terminer son protocole de réinitialisation (chirp) et le traitement dans l'état Bus_Reset dans un délai de 10 ms. Il est recommandé que cet intervalle soit le plus long possible étant donné les contraintes énumérées ici. Cela fournit une immunité au bruit maximum sur le port orienté en amont et réduit la probabilité que l'appareil soit réinitialisé en présence de bruit avant que le hub en amont ne désactive le port.

Le minuteur est réinitialisé chaque fois que cet état démarre.

11.6.3.9 Bus_Reset

Cet état est activé:

- De l'état RéceptionSE0 lorsque le minuteur expire. Tant que le port continue à recevoir SE0, le récepteur reste dans cet état.
- Cet état est également activé lorsque la mise sous tension-réinitialisation (POR) est générée par l'ensemble de circuits local. Le diagramme d'états ne peut pas quitter cet état alors que le POR est actif.
- Le minuteur 3 ms expire dans l'état ReceivingIS et le port a retiré ses terminaisons à grande vitesse et a connecté sa résistance de polarisation D+ et l'état de bus résultant est encore SE0.

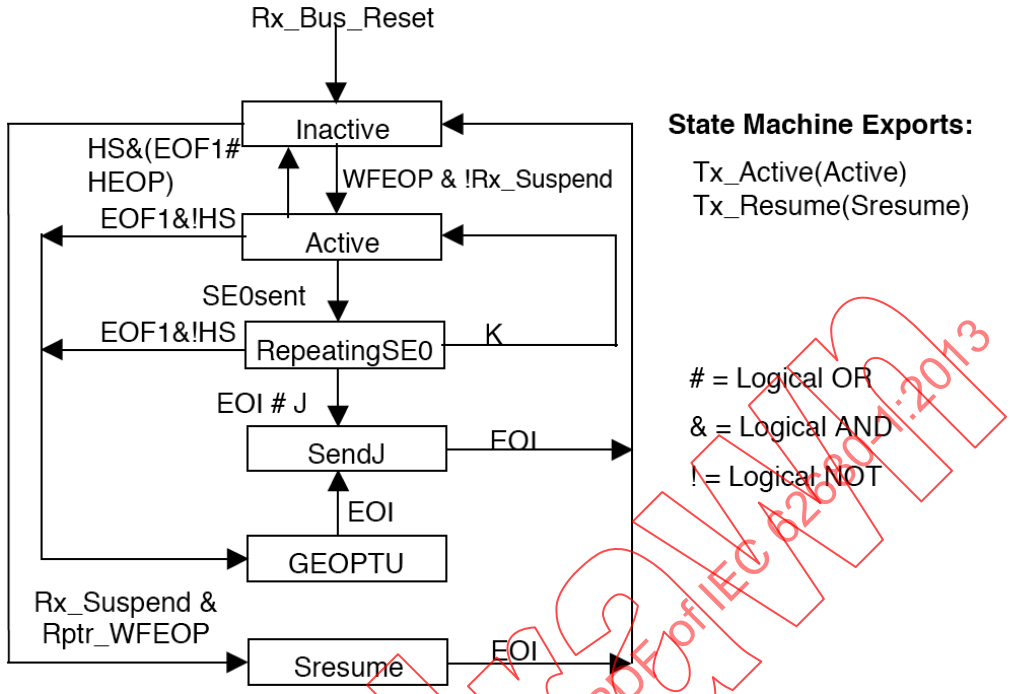
Dans cet état, un port compatible à grande vitesse va appliquer la signalisation chirp, l'établissement de liaison et le protocole de synchronisation comme décrit dans la Section 7.1.7.5.

11.6.4 Emetteur

Ce diagramme d'états est utilisé pour surveiller le port orienté en amont alors que le répéteur de hub a une connectivité en amont. L'objet de cette activité de surveillance est d'empêcher la propagation d'indications erronées en amont. En particulier, cette machine empêche les évènements de babillage et de déconnexion sur les ports orientés en aval de ce hub de se propager et de faire en sorte que ce hub soit désactivé ou déconnecté par le hub auquel il est

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

relié. La Figure 11-13 illustre le diagramme de transition d'état de l'émetteur. Le Tableau 11-9 définit les signaux et évènements référencés dans la Figure 11-13.



Légende

Anglais	Français
State Machine Exports:	Exports des diagrammes d'états:
# = Logical OR	# = OU logique
& = Logical AND	& = ET logique
! = Logical NOT	! = NON logique
Tx_Active(Active)	Tx_Active(Actif)
Tx_Resume(Sresume)	Tx_Resume(Sresume)
Inactive	Inactif
Active	Actif

Figure 11-13. Diagramme d'états de l'émetteur du port orienté en amont

Tableau 11-9. Définitions des signaux/évènements de l'émetteur du port orienté en amont

Nom de signal/d'évènement	Source d'évènement/de signal	Description
Rx_Bus_Reset	Récepteur	Le récepteur est dans l'état Bus_Reset
EOF1	Minuteur de (micro)trames	Le temps de (micro)trame du hub a atteint le point EOF1 ou est entre EOF1 et la fin de la (micro)trame
J	Interne	L'émetteur passe à l'envoi d'un "J" et transmet un "J"
Rptr_WFEOP	Répéteur de hub	Le répéteur de hub est dans l'état WFOEP
K	Interne	L'émetteur transmet un "K"
SE0sent	Interne	Au moins un temps bit de SE0 a été envoyé via l'émetteur
Rx_Suspend	Récepteur	Le récepteur est dans l'état Suspension
HEOP	Répéteur	Exécution de la transmission de paquets en amont
HS	Interne	Le port orienté en amont fonctionne comme un port à grande vitesse
EOI	Interne	Fin de l'intervalle temporisé

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.6.4.1 Inactif

Cet état est accessible à la fin de l'état SendJ ou lorsque le récepteur est dans l'état Bus_Reset. Cet état est également accessible à la fin de l'état Sresume. Tandis que l'émetteur est dans cet état, les circuits de transmission différentielle et à terminaison unique sont désactivés et placés dans leur état à impédance élevée.

Lorsque le port fonctionne comme un port à grande vitesse, cet état est accessible à partir de l'état Actif au niveau de EOF1 ou après un HEOP de l'aval.

11.6.4.2 Actif

Cet état est accessible à partir de l'état Inactif lorsque le répéteur de hub passe à l'état WFEOP. Cet état est accessible à partir de l'état RepeatingSE0 si la première transition après le SE0 n'est pas vers l'état J. Dans cet état, les données d'un port orienté en aval sont répétées et transmises sur le port orienté en amont.

11.6.4.3 RepeatingSE0

Le port entre dans cet état à partir de l'état Actif lorsqu'un temps bit de SE0 a été envoyé sur le port orienté en amont. Lorsqu'il est dans cet état, l'émetteur est encore actif et la signalisation en aval est répétée sur le port. Il s'agit d'un état temporisé avec une durée de 23 temps bit à vitesse intégrale.

11.6.4.4 SendJ

Le port entre dans cet état à partir de l'état RepeatingSE0 si le minuteur de bits atteint 23 ou si la signalisation répétée passe de SE0 à "J" ou "SE1". Cet état est également accessible à la fin de l'état GEOPTU. Cet état dure pendant un temps bit à vitesse intégrale. Pendant cet état, le hub pilote un SJ sur le port.

11.6.4.5 Génération de fin de paquet vers le port amont (GEOPTU)

Le port entre dans cet état à partir de l'état Actif ou RepeatingSEO si le minuteur de trame atteint le point EOF1.

Dans cet état, le port transmet SE0 pour deux temps bit à vitesse intégrale.

11.6.4.6 Reprise Envoi (Sresume)

Le port entre dans cet état à partir de l'état Inactif si le récepteur est dans l'état Suspension et le répéteur de hub passe à l'état WFEOP. Cela indique qu'un appareil en aval (ou le port vers le contrôleur de hub) a généré une signalisation de reprise entraînant l'établissement de la connectivité en amont.

Lors de l'entrée dans cet état, le hub redémarre les horloges si elles ont été éteintes pendant l'état Suspend. Dans cet état, l'émetteur pilote un "K" sur le port orienté en amont. Lorsque l'émetteur est dans cet état, le récepteur est maintenu dans l'état Reprise. Lorsque le récepteur est dans l'état Reprise, tous les ports orientés en aval qui sont dans l'état Activé sont placés dans l'état TransmitR et la reprise sur ce port est transmise à ces ports orientés en aval.

Le port reste dans cet état pendant au moins 1 ms, mais pas plus de 15 ms.

11.7 Répéteur de hub

Le répéteur de hub fournit les fonctions suivantes:

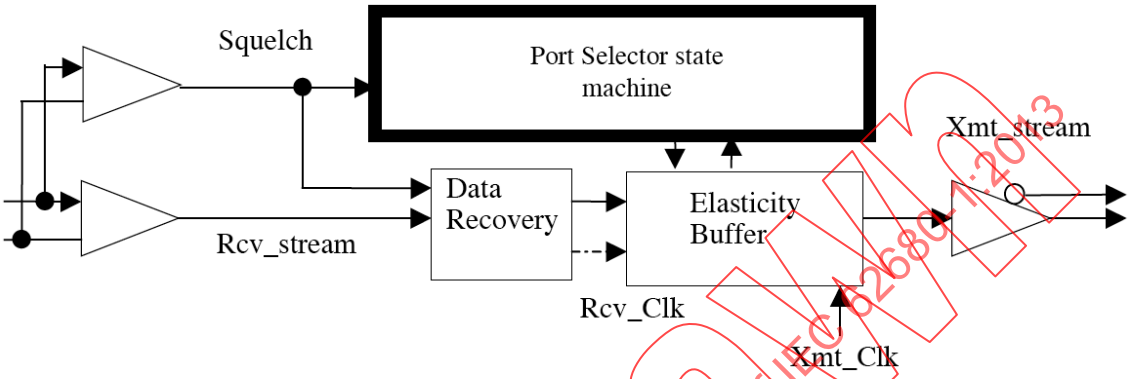
- Configure et désinstalle la connectivité sur les limites de paquet
- Garantit l'entrée correcte dans et en dehors de l'état Suspend, notamment le traitement correct des réveils à distance

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.7.1 Connectivité de paquet à grande vitesse

Les répéteurs de paquet à grande vitesse doivent recadencer les paquets dans les deux directions. Le recadençage signifie que le répéteur extrait les données du flux reçu et retransmet le flux à l'aide de sa propre horloge locale. Cela est nécessaire pour maintenir la gigue observée sur un récepteur dans les limites acceptables (voir Chapitre 7 pour la définition et les limites de la gigue).

Le recadençage génère plusieurs exigences qui peuvent être mieux comprises avec l'exemple de chemin du signal du répéteur illustré dans la Figure 11-14.



Légende

Anglais	Français
Squelch	Appareil de silence
Port Selector state machine	Machine d'état du sélecteur de ports
Data Recovery	Récupération des données
Elasticity Buffer	Tampon d'élasticité

Figure 11-14. Exemple d'organisation du répéteur de hub

11.7.1.1 Circuit de l'appareil de silence

En raison de la détection de l'appareil de silence, les bits initiaux du champ SYNC peuvent ne pas être vus dans le reste du répéteur. Au maximum 4 bits du champ SYNC peuvent être sacrifiés dans l'ensemble du chemin du répéteur.

Le circuit de l'appareil de silence peut prendre au maximum 4 temps bit pour désactiver le répéteur après que le bus soit retourné à l'état Repos. Cela entraîne l'ajout de bits après la fin du paquet. Cela est également connu sous le nom de débordement d'EOP et jusqu'à 4 bits aléatoires peuvent être ajoutés après le paquet par l'ensemble du chemin du répéteur.

11.7.1.2 Unité de récupération des données

L'unité de récupération des données extrait l'horloge de réception et reçoit les données de ce flux. Il faut noter qu'il s'agit d'un mode conceptuel uniquement; les mises en œuvre réelles (ex., DLL) peuvent atteindre le recadençage avec l'horloge locale sans séparation de l'horloge de réception et des données.

11.7.1.3 Tampon d'élasticité

La demi-profondeur du tampon d'élasticité dans le répéteur doit être d'au moins 12 bits.

La latence totale d'un paquet dans un répéteur doit être inférieure à 36 temps bit. Cela inclut la latence dans le tampon d'élasticité.

Le tampon d'élasticité est utilisé pour traiter la différence de fréquence entre l'horloge de réception et l'horloge locale et fonctionne comme suit. Le tampon d'élasticité est amorcé (rempli d'au moins 12 bits) par l'horloge de réception avant que les données ne soient pointées en sortie par l'horloge de transmission. Si l'horloge de transmission est plus rapide que l'horloge de réception, le tampon est vidé plus rapidement qu'il ne se remplit. Si l'horloge de transmission est plus lente, le tampon est vidé plus lentement qu'il ne se remplit. Si la

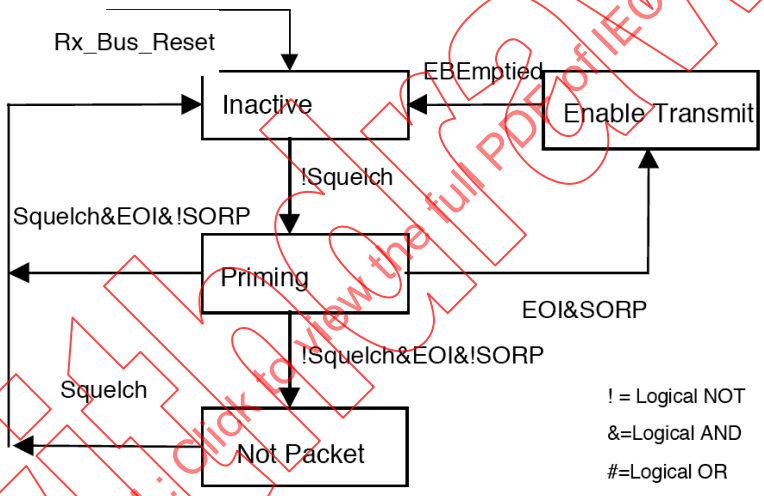
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

demi-profondeur du tampon est choisie comme étant égale à la différence maximum du débit d'horloge sur la longueur d'un paquet, les bits ne sont pas perdus ou ajoutés au paquet. La demi-profondeur est calculée comme suit.

La tolérance d'horloge autorisée est de 500 ppm. Cela tient compte de l'effet de la tension, de la température, du vieillissement, etc. L'horloge reçue et l'horloge locale peuvent ainsi être différentes de 1 000 ppm. Le paquet le plus long a une charge utile de données de 1 Koctet. La longueur maximum d'un paquet est calculée en ajoutant la longueur de tous les champs et en supposant le bourrage de bits maximum. Cette longueur maximum est de 9 644 bits (9 624 bits de paquet + 20 bits de débordement d'EOP). Cela signifie que lorsque le répéteur pointe en sortie un paquet avec son horloge locale, il peut être en avance ou en retard de l'horloge de réception de 9 644 bits (1 000 ppm*9 644). Ce calcul obtient 10 bits. La demi-profondeur du tampon d'élasticité dans le répéteur doit être d'au moins 12 bits pour fournir la marge de synchronisation du système.

11.7.1.4 Diagramme d'états du sélecteur de port à grande vitesse

Ce diagramme d'états est utilisé pour établir la connectivité sur un paquet valide et pour empêcher le répéteur d'établir la connectivité à partir d'un port qui voit du bruit. Ce diagramme d'états doit appliquer le comportement illustré dans la Figure 11-15. (Note: Ce diagramme d'états peut être appliqué par port ou par hub).



Légende

Anglais	Français
Inactive	Inactif
Priming	Amorçage
Not Packet	Non paquet
Squelch	Appareil de silence
Enable Transmit	Activer Transmission
# = Logical OR	# = OU logique
& = Logical AND	& = ET logique
! = Logical NOT	! = NON logique

Figure 11-15. Diagramme d'états du sélecteur de port à grande vitesse

Tableau 11-10. Définitions des signaux/évènements du sélecteur de port à grande vitesse

Nom de signal/d'évènement	Source d'évènement/de signal	Description
Rx_Bus_Reset	Interne	Le récepteur est dans l'état Bus_Reset.
EBEmptied	Interne	Tous les bits accumulés dans le tampon d'élasticité ont été transmis.
EOI	Interne	Fin d'intervalle de temps nécessaire pour l'amorçage du

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Nom de signal/d'évènement	Source d'évènement/de signal	Description
		tampon d'élasticité
Appareil de silence	Interne	Le bus est dans l'état appareil de silence
SORP	Interne	Début du modèle de répétition; un modèle "JKJK" ou "KJKJ" a été vu dans les données dans le tampon d'élasticité.

11.7.1.4.1 Inactif

Cet état est activé

- A partir de l'état Activer transmission lorsque tous les bits accumulés dans le tampon d'élasticité ont été transmis
- A partir de l'état Amorçage si l'appareil de silence est vu et le tampon d'élasticité est amorcé sans qu'un SORP ne soit vu
- A partir de l'état Non paquet lorsque le circuit de l'appareil de silence indique un état de l'appareil du silence sur le port
- A partir de n'importe quel état sur Rx_Bus_Reset

11.7.1.4.2 Amorçage

Cet état est accessible à partir de l'état Inactif lorsque le circuit de l'appareil de silence indique que les niveaux de signal valides ont été observés sur le port. Il s'agit d'un état temporisé et l'intervalle d'amorçage est le temps nécessaire pour la mise en œuvre pour remplir le tampon d'élasticité avec au moins 12 bits.

11.7.1.4.3 Transmission activée

Cet état est accessible à partir de l'état Amorçage lorsque l'intervalle d'amorçage du tampon d'élasticité s'est écoulé et les bits dans le tampon d'élasticité incluent le modèle SORP.

Dans cet état, le diagramme d'états génère un signal "début de paquet à grande vitesse" (SOHP) au diagramme d'états du répéteur qui permet au répéteur d'établir la connectivité à partir de ce port vers le port orienté en amont (ou les ports orientés en aval).

11.7.1.4.4 Non paquet

Cet état est accessible à partir de l'état Amorçage lorsque l'intervalle d'amorçage du tampon d'élasticité s'est écoulé et les bits dans le tampon d'élasticité n'incluent pas le modèle SORP et le signal de l'appareil de silence n'est pas actif.

11.7.2 Diagramme d'états du répéteur de hub

Le diagramme d'états du répéteur de hub dans la Figure 11-16 montre le statut et les transitions nécessaires pour appliquer le répéteur de hub. Le Tableau 11-11 définit les signaux et évènements du répéteur de hub. Les sections suivantes décrivent les états et les transitions.

11.7.2.1 Fonctionnement du répéteur à grande vitesse

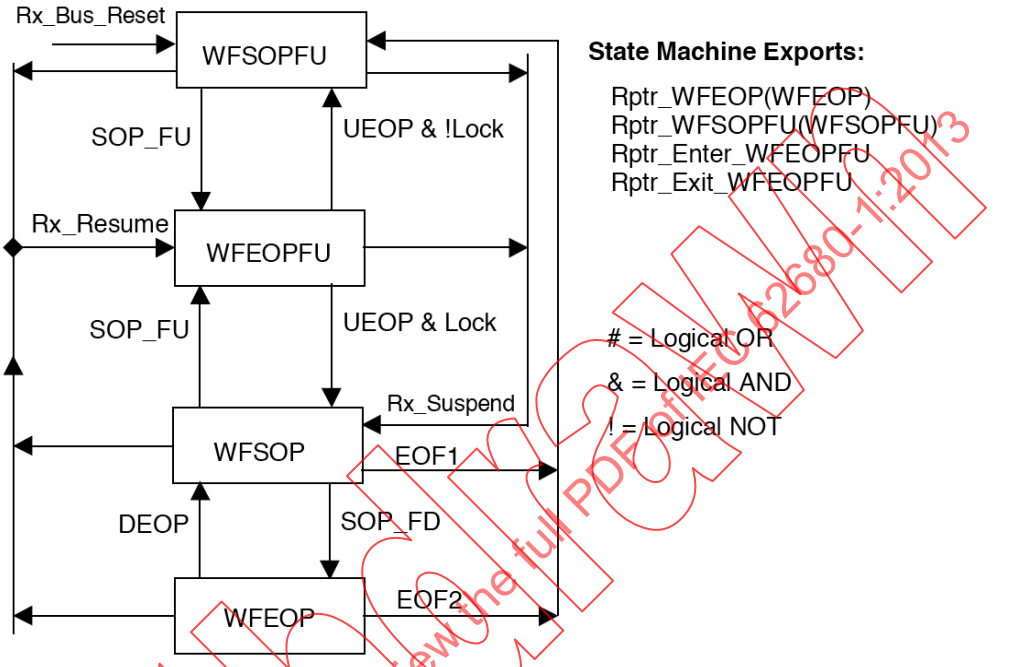
La connectivité est configurée sur SOHP et désinstallée sur HEOP. (HEOP est le signal EBemptied du diagramme d'états du sélecteur de port "OU" le signal EOI qui entraîne le passage de l'état SendEOR dans le diagramme d'états du port orienté en aval.) Plusieurs transitions d'état ci-après ont lieu lorsque le HEOP est vu. Lorsqu'une telle transition est indiquée, la transition n'a pas lieu tant que le hub n'a pas répété le dernier bit dans le tampon d'élasticité. Certaines transitions sont déclenchées par un SOHP. Les transitions de ce type ont lieu dès que le hub détecte le SOHP à partir du diagramme d'états du sélecteur de port, garantissant qu'un démarrage de paquet valide a été vu.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.7.2.2 Fonctionnement du répéteur à vitesse intégrale/basse vitesse

La connectivité est configurée sur SOP et désinstallée sur EOP. Plusieurs transitions d'état ci-après ont lieu lorsque l'EOP est vu. Lorsqu'une telle transition est indiquée, la transition ne se produit pas tant que le hub n'a pas répété la transition SE0 vers "J" et a piloté "J" pendant au moins un temps bit (le temps bit est déterminé par la vitesse du port.) Certaines transitions sont déclenchées par un SOP. Les transitions de ce type ont lieu dès que le hub détecte la transition "J" vers "K", garantissant que le front initial du champ SYNC est préservé.

11.7.2.3 Diagramme d'états du répéteur



Légende

Anglais	Français
State Machine Exports:	Exports des diagrammes d'états:
# = Logical OR	# = OU logique
& = Logical AND	& = ET logique
! = Logical NOT	! = NON logique

Figure 11-16. Diagramme d'états du répéteur de hub

Tableau 11-11. Définitions des signaux/évènements du répéteur de hub

Nom de signal/d'évènement	Source d'évènement/de signal	Description
Rx_Bus_Reset	Récepteur	Le récepteur est dans l'état Bus_Reset
HEOP	Interne (sélecteur de port, port en aval, récepteur de port en amont)	Trois sources de HEOP: Signal EBEmptied du diagramme d'états du sélecteur de port OU transition au niveau de EOI à partir de l'état SendEOR dans le diagramme d'états du port orienté en aval OU EOITR à partir du diagramme d'états du récepteur de port orienté en amont
UEOP	Interne	(HEOP)EOP reçu à partir du port orienté en amont
DEOP	Interne	Généré lorsque l'émetteur entre dans l'état SendJ (Inactif)
EOF1	Minuteur de (micro)trame	Le minuteur de (micro)trame est au point EOF1 ou entre EOF1 et la fin de (micro)trame
EOF2	Minuteur de (micro)trame	Le minuteur de (micro)trame est au point EOF2 ou entre EOF2 et la fin de (micro)trame

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Nom de signal/d'évènement	Source d'évènement/de signal	Description
Lock	Minuteur de (micro)trame	Le minuteur de micro(trame) est verrouillé
Rx_Suspend	Récepteur	Le récepteur est dans l'état Suspendu
Rx_Resume	Récepteur	Le récepteur est dans l'état Reprise
SOP_FD	Interne	(SOHP)SOP reçu à partir du port orienté en aval ou du contrôleur de hub Généré (après SOHP identifié) sur la transition de l'état Repos à l'état K sur un port
SOP_FU	Interne	(SOHP)SOP reçu à partir du port orienté en amont Généré (après SOHP identifié) sur la transition de l'état Repos à l'état K sur le port orienté en amont

11.7.3 Attente de début de paquet du port en amont (WFSOPFU)

Cet état est accessible dans une des situations suivantes:

- A partir de n'importe quel état lorsque le récepteur en amont est dans l'état Bus_Reset
- A partir de l'état WFSOP si le minuteur de (micro)trame est au point EOF1 ou l'a dépassé
- A partir de l'état WFEOP au point EOF2
- A partir de WFEOPFU si le minuteur de (micro)trame n'est pas synchronisé (verrouillée) lorsqu'un (HEOP)EOP est reçu sur le port orienté en amont

Dans cet état, le hub attend un (SOHP)SOP sur le port orienté en amont, et les transitions sur les ports orientés en aval sont ignorées par le répéteur de hub. Lorsque le répéteur de hub est dans cet état, la connectivité n'est pas établie.

Cet état est utilisé pendant la fin de la (micro)trame (après le point EOF1) pour garantir que le hub est capable de recevoir le SOF lorsqu'il est envoyé par l'hôte.

11.7.4 Attente de fin de paquet du port en amont (WFEOPFU)

Le hub entre dans cet état si le hub est dans l'état WFSOP ou WFSOPFU et un (SOHP)SOP est détecté sur le port orienté en amont. Le hub entre également dans cet état à partir des états WFSOP, WFSOPFU ou WFEOP lorsque le récepteur entre dans l'état Reprise.

Dans cet état, la connectivité est établie à partir du port orienté en amont vers tous les ports orientés en aval activés. Les ports orientés en aval qui sont dans l'état Activé sont placés dans l'état Transmission lors de la transition vers cet état.

11.7.5 Attente de début de paquet (WFSOP)

Cet état est accessible dans une des situations suivantes:

- A partir de l'état WFEOP lorsqu'un (HEOP)EOP est détecté à partir du port orienté en aval
- A partir de l'état WFEOPFU si le minuteur de (micro)trame est synchronisé (verrouillée) lorsqu'un (HEOP)EOP est reçu de l'amont
- A partir des états WFSOPFU ou WFEOPFU lorsque le récepteur en amont passe à l'état Suspension

Un hub dans cet état attend un (SOHP)SOP sur le port orienté en amont ou sur n'importe quel port orienté en aval qui est dans l'état Activé. Lorsque le répéteur de hub est dans cet état, la connectivité n'est pas établie.

11.7.6 Attente de fin de paquet (WFEOP)

Cet état est accessible à partir de l'état WFSOP lorsqu'un (SOHP)SOP est reçu à partir d'un port orienté en aval dans l'état Activé.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Dans cet état, le hub a une connectivité établie en amont et la signalisation reçue sur un port orienté en aval activé est répétée et pilotée sur le port orienté en amont. L'émetteur en amont est placé dans l'état Actif lors de la transition dans cet état.

Si le répéteur de hub est dans cet état lorsque le point EOF2 est atteint, le port orienté en aval pour lequel la connectivité est établie est désactivé comme port de babillage.

Note: L'émetteur à vitesse intégrale envoie un EOP au niveau d'EOF1, mais le répéteur reste dans cet état jusqu'à ce que l'appareil envoie un (HEOP)EOP ou que le point EOF2 soit atteint.

11.8 Evaluation d'état de bus

Un hub doit évaluer l'état de la connexion sur un port pour effectuer les transitions d'état de port appropriées. Cette section décrit les temps et moyens appropriés pour plusieurs de ces évaluations.

11.8.1 Erreur de port

Une erreur de port peut se produire sur un port orienté en aval qui est dans l'état Activé. Une condition Erreur de port existe lorsque:

- Le hub est dans l'état WFEOP avec connectivité établie en amont du port lorsque le minuteur de (micro)trame atteint le point EOF2.
- Au point EOF2, le répéteur de hub est dans l'état WFSOPFU et il y a un autre état que l'état Repos sur le port.

Si la connectivité orientée en amont est établie lorsque le minuteur de (micro)trame atteint le point EOF1, l'émetteur en amont génère (retourne à l'état Inactif) un EOP à vitesse intégrale pour empêcher le hub d'être désactivé par le hub en amont. Le port connecté est alors désactivé s'il n'a pas terminé le paquet et n'est pas retourné à l'état Repos avant que le minuteur de (micro)trame n'atteigne le point EOF2.

11.8.2 Détection de vitesse

A la fin de la réinitialisation, le bus est dans l'état Repos pour la vitesse enregistrée dans le registre d'état de port. La détection de vitesse est décrite dans la Section 7.1.7.5.

Si l'appareil connecté au port orienté aval est à grande vitesse, le répéteur (non le traducteur de transactions) est utilisé pour la signalisation entre ce port et le port orienté amont.

En raison des transitoires de connexion et de démarrage, le hub peut ne pas déterminer de manière fiable la vitesse de l'appareil tant que les transitoires ne sont pas terminés. Le logiciel du système USB doit "assurer l'antirebond" de la connexion et fournir un délai entre le moment où une connexion est détectée et l'appareil est utilisé (voir Section 7.1.7.3). A la fin de l'intervalle d'antirebond, l'appareil devrait avoir placé son port orienté en amont dans l'état Repos et être capable de réagir à la signalisation de réinitialisation. Le logiciel du système USB doit envoyer une demande SetPortFeature(PORT_RESET) au port pour activer le port et rendre l'appareil relié prêt à l'emploi.

Le port orienté en aval surveille l'état des lignes D+ et D- pour déterminer si l'appareil connecté est à basse vitesse. Si tel est le cas, le bit d'état PORT_LOW_SPEED est défini sur un pour indiquer un appareil à basse vitesse. Si tel n'est pas le cas, le bit d'état PORT_LOW_SPEED est défini sur zéro pour indiquer un appareil à vitesse intégrale/élevée. Lors de la sortie du processus de réinitialisation, le hub doit définir le bit d'état PORT_HIGH_SPEED selon la vitesse détectée. Le port orienté en aval effectue le traitement de réinitialisation requis comme défini dans la section 7.1.7.5. A la fin de l'état Réinitialisation, le hub retourne le bus à l'état Repos qui est approprié pour la vitesse de l'appareil relié et passe à l'état Activé.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.8.3 Collision

Si le répéteur de hub est dans l'état WFEOP et un (SOHP)SOP est détecté sur un autre port activé, une condition de collision existe. Il y a deux comportements autorisés pour le hub dans cette instance. Dans l'un des cas, la désinstallation de la connectivité au niveau d'EOF1 et la détection du babillage au niveau d'EOF2 sont requises.

Le premier comportement, préféré, consiste à "altérer" le message de manière à ce que l'hôte puisse détecter le problème. Le hub altère le message en transmettant un ("J" ou) "K" sur le port orienté en amont. Il convient que ce ("J" ou) "K" persiste jusqu'à la fin du trafic de paquets en provenance de tous les ports orientés en aval. Il convient que le hub utilise le dernier ("J" ou "K") EOP pour terminer le paquet altéré. La détection du babillage est activée pendant ce message altéré.

Un second comportement consiste à bloquer le second paquet et, lorsque le premier message termine, à retourner le hub dans l'état WFSOPFU ou WFSOP si approprié. Si le second flux est encore actif, le hub peut rétablir la connectivité en amont. Cette méthode n'est pas la méthode préférée, car elle ne transfère pas le problème vers l'hôte. De plus, si le second flux fait que le hub ré-établit la connectivité en amont comme l'hôte essaie d'établir la connectivité en aval, des paquets supplémentaires peuvent être perdus et l'hôte ne peut pas associer correctement le problème.

Note: dans les répéteurs à grande vitesse, l'utilisation du SOHP pour détecter les collisions nécessiterait la réplication du chemin de données illustré dans la Figure 11-14 sur tous les ports. Le signal d'absence d'appareil de silence sur chaque port peut être utilisé à la place de SOHP pour détecter les collisions; dans ce cas, le second comportement (blocage) décrit ci-dessus doit être utilisé.

11.8.4 Comportement de port à basse vitesse

Lorsqu'un hub est configuré pour le fonctionnement à vitesse intégrale/basse vitesse, les données basse vitesse sont envoyées ou reçues par le port orienté en amont du hub au niveau de la signalisation à vitesse intégrale même si les temps bit sont à basse vitesse.

La signalisation à vitesse intégrale ne doit pas être transmise aux ports à basse vitesse.

Si un port est détecté comme étant relié à un appareil à basse vitesse, les tampons de sortie du port du hub sont configurés pour fonctionner à une vitesse de balayage lente (75-300 ns), et le port ne propage pas les paquets orientés en aval tant qu'ils ne sont pas préfacés avec un PID PRE. Lorsqu'un PID PRE est reçu, l'état "J" doit être piloté sur les ports à basse vitesse activés dans les quatre temps bit qui suivent la réception du dernier bit du PID PRE.

Les données à basse vitesse suivent le PID et sont propagées vers les deux appareils à basse vitesse/vitesse intégrale. Les hubs continuent de propager la signalisation en aval vers tous les ports activés jusqu'à ce qu'un EOP en aval soit détecté, moment où tous les émetteurs de sortie sont hors tension.

Les appareils à vitesse intégrale ne vont pas mal interpréter le trafic à basse vitesse, car aucun modèle de données à basse vitesse ne peut générer un PID à vitesse intégrale valide.

Lorsqu'un appareil à basse vitesse transmet, il ne préface pas son paquet de données avec un PID PRE. Les hubs propagent les paquets orientés en amont à vitesse intégrale/basse vitesse à l'aide de la polarité de signalisation à vitesse intégrale et des taux de front.

Pour les données à basse vitesse en amont et en aval, le hub est responsable de l'inversion de la polarité des données avant la transmission vers/d'un port à basse vitesse.

Bien qu'un appareil à basse vitesse envoie un EOP à basse vitesse pour terminer correctement un paquet, un hub peut tronquer un paquet à basse vitesse au point EOF1 avec un EOP à vitesse intégrale. Les hubs doivent ainsi toujours être capables de désinstaller la connectivité en réponse à un EOP à vitesse intégrale du débit de données du paquet.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

En raison des transitions lentes sur les ports à basse vitesse, lorsque les lignes de signaux D+ et D- commutent entre 'J' et 'K', elles peuvent toutes les deux être inférieures à 2,0 V pour une période de temps qui est supérieure à un temps bit à vitesse intégrale. Un hub doit garantir que ces transitions lentes n'entraînent pas un arrêt de la connectivité et ne doivent pas entraîner l'envoi d'un SE0 en amont.

11.8.4.1 Maintien actif à basse vitesse

Tous les ports de hub auxquels les appareils à basse vitesse sont connectés doivent générer un maintien actif à basse vitesse, généré au début de la trame, qui comprend un EOP basse vitesse valide (décrit dans la Section 7.1.13.2). Le maintien actif doit être généré au moins une fois dans chaque trame dans laquelle un SOF est reçu. Ce maintien actif est utilisé pour empêcher la suspension des appareils à basse vitesse s'il n'y a pas d'autre trafic à basse vitesse sur le bus. Le hub peut générer le maintien actif sur n'importe quel paquet de jeton valide à vitesse intégrale. Les règles suivantes de génération d'un maintien actif à basse vitesse doivent être observées:

- Un maintien actif doit au minimum être dérivé de chaque SOF. Il convient qu'un maintien actif soit généré sur un jeton à vitesse intégrale valide.
- Le maintien actif doit démarrer au huitième bit après le PID du jeton à vitesse intégrale.

11.9 Suspension et Reprise

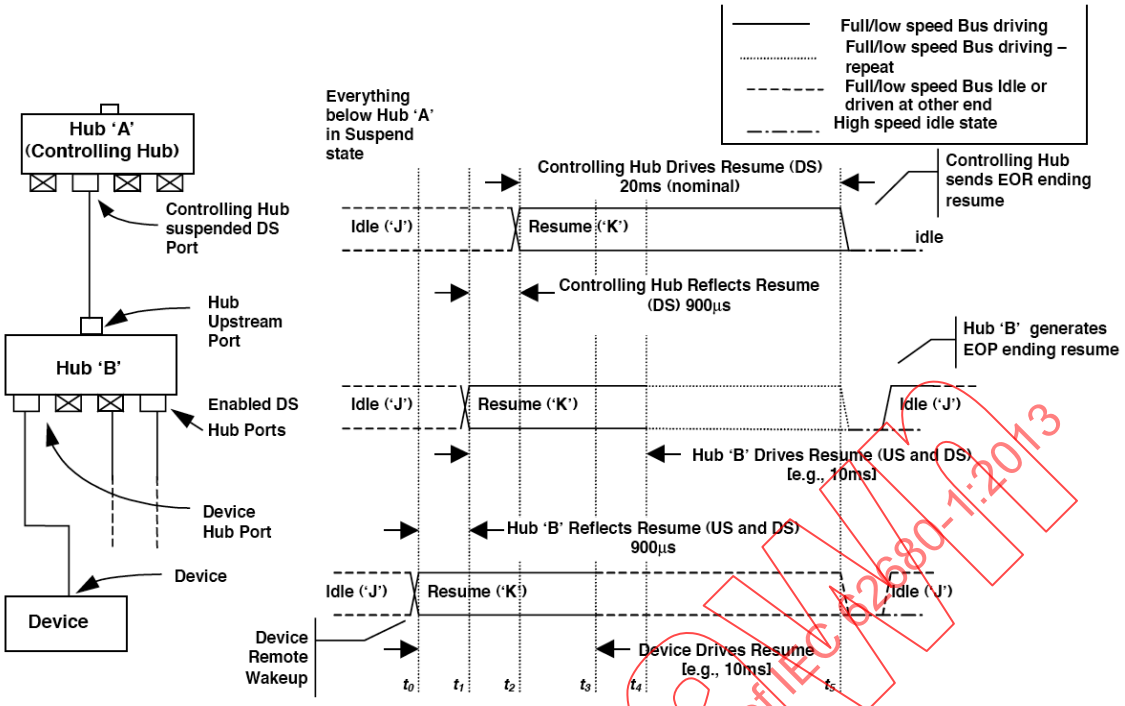
Les hubs doivent prendre en charge la suspension et la reprise en tant qu'appareil USB et en termes de propagation de la signalisation de suspension et de reprise. Les hubs prennent en charge la suspension et la reprise globales et sélectives. Les suspensions globale et sélective sont définies dans la Section 7.1.7.6. La suspension/reprise globale fait référence à l'ensemble du bus suspendu ou repris sans affecter des états de ports orientés en aval du hub; la suspension/reprise sélective fait référence à un port orienté en aval d'un hub suspendu ou repris sans altérer l'état du hub. La suspension/reprise globale est appliquée dans le ou les ports racines au niveau de l'hôte. La suspension/reprise sélective est appliquée via les demandes à un hub. La reprise initiée par l'appareil est appelée réveil à distance (voir Section 7.1.7.7).

Si le port orienté en amont du hub est à vitesse intégrale (vitesse élevée), le comportement requis est le même que pour une fonction avec port orienté en amont à vitesse intégrale (vitesse élevée) et est décrit au Chapitre 7.

Lorsqu'un port orienté en aval et fonctionnant à vitesse élevée passe dans l'état Suspension, il commute sur les terminaisons à vitesse intégrale, mais continue d'avoir un état de port à vitesse élevée. En réponse à un réveil à distance ou une reprise sélective, ce port pilote le "K" sélectif dans son état Reprise. Les exigences et synchronisations sont les mêmes que pour les ports à vitesse intégrale et sont décrites ci-après. A la fin de cette signalisation, le bus est retourné à l'état Repos à vitesse élevée (à l'aide de l'état SendEOR). Puis, le port retourne à l'état Activé. Le statut à vitesse élevée du port est maintenu pendant le cycle suspension-reprise.

La Figure 11-17 et la Figure 11-18 illustrent les relations de synchronisation pour un exemple de séquence de réveil à distance. Cet exemple illustre un appareil initiant une signalisation de reprise dans un hub suspendu ('B') vers un hub de réveil ('A'). Le hub "A" dans cet exemple cadence et exécute la séquence de reprise et est le "hub de contrôle". Les synchronisations et évènements sont définis dans la Section 7.1.7.7.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

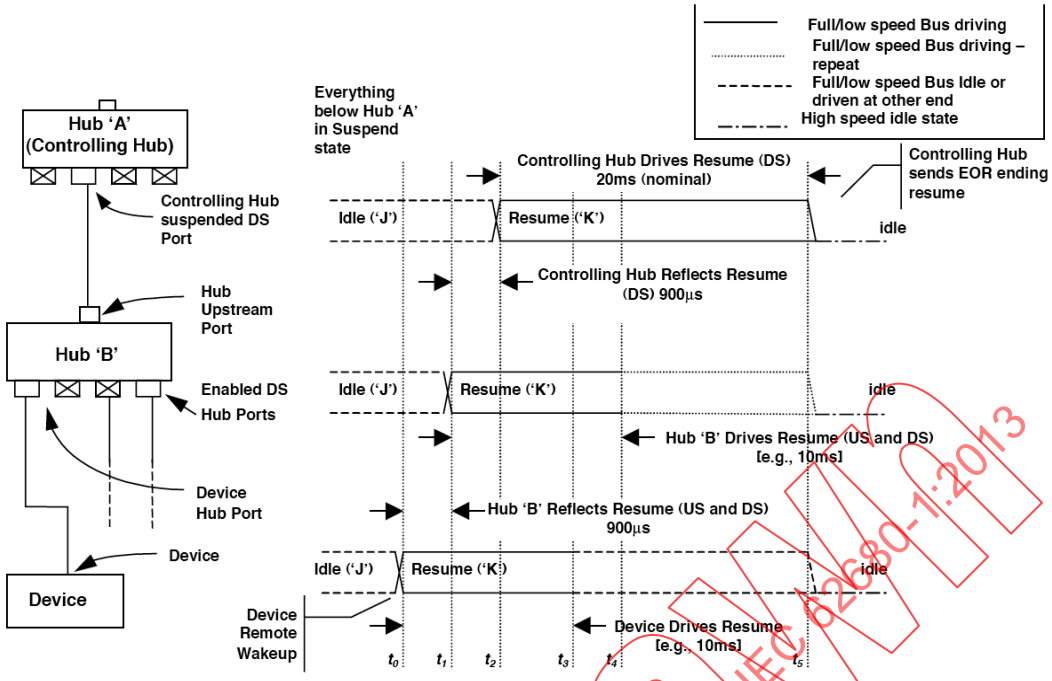


Légende

Anglais	Français
Hub 'A' (Controlling Hub)	Hub 'A' (Hub de contrôle)
Controlling Hub suspended DS Port	Port DS suspendu du hub de contrôle
Hub 'B '	Hub 'B '
Hub Upstream Port	Port en amont du hub
Enabled DS Hub Ports	Ports de hub DS activés
Device Hub Port	Port de hub d'appareil
Device	Appareil
Everything below Hub 'A' in Suspend state	Tout au-dessous du hub 'A' à l'état Suspension
Idle ('J')	Ralenti ('J')
Device Remote Wakeup	Réveil à distance de l'appareil
Controlling Hub Drives Resume (DS) 20ms (nominal)	Le hub de contrôle pilote la reprise (DS) 20 ms (nominal)
Resume ('K')	Reprise ('K')
Controlling Hub Reflects Resume (DS) 900µs	Le hub de contrôle reflète la reprise (DS) 900 µs
Hub 'B' Drives Resume (US and DS) [e.g., 10ms]	Le hub 'B' pilote la reprise (US et DS) [ex.: 10 ms]
Hub 'B' Reflects Resume (US and DS) 900µs	Le hub 'B' reflète la reprise (US et DS) 900 µs
Device Drives Resume [e.g., 10ms]	L'appareil pilote la reprise [ex.: 10 ms]
Idle	Ralenti
Controlling Hub sends EOR ending resume	Le hub de contrôle envoie une reprise de fin d'EOR
Hub 'B' generates EOP ending resume	Le hub 'B' génère une reprise de fin d'EOP
Full/low speed Bus driving Full/low speed Bus driving - Repeat Full/low speed Bus Idle or driven at other end High speed idle state	Pilotage de bus à vitesse intégrale/basse vitesse Pilotage de bus à vitesse intégrale/basse vitesse - Répétition Bus à vitesse intégrale/basse vitesse au ralenti ou piloté à l'autre extrémité Grande vitesse état ralenti

Figure 11-17. Exemple de signalisation de reprise de réveil à distance avec appareil à vitesse intégrale/basse vitesse

IECNORM.COM : Click to view the PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Hub 'A' (Controlling Hub)	Hub 'A' (Hub de contrôle)
Controlling Hub suspended DS Port	Port DS suspendu du hub de contrôle
Hub 'B '	Hub 'B '
Hub Upstream Port	Port en amont du hub
Enabled DS Hub Ports	Ports de hub DS activés
Device Hub Port	Port de hub d'appareil
Device	Appareil
Everything below Hub 'A' in Suspend state	Tout au-dessous du hub 'A' à l'état Suspension
Idle ('J')	Ralenti ('J')
Device Remote Wakeup	Réveil à distance de l'appareil
Controlling Hub Drives Resume (DS) 20ms (nominal)	Le hub de contrôle pilote la reprise (DS) 20 ms (nominal)
Resume ('K')	Reprise ('K')
Controlling Hub Reflects Resume (DS) 900µs	Le hub de contrôle reflète la reprise (DS) 900 µs
Hub 'B' Drives Resume (US and DS) [e.g., 10ms]	Le hub 'B' pilote la reprise (US et DS) [ex.: 10 ms]
Hub 'B' Reflects Resume (US and DS) 900µs	Le hub 'B' reflète la reprise (US et DS) 900 µs
Device Drives Resume	L'appareil pilote la reprise

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
[e.g., 10ms]	[ex.: 10 ms]
idle	ralenti
Controlling Hub sends EOR ending resume	Le hub de contrôle envoie une reprise de fin d'EOR
Full/low speed Bus driving Full/low speed Bus driving - repeat Full/low speed Bus Idle or driven at other end High speed idle state	Pilotage de bus à vitesse intégrale/basse vitesse Pilotage de bus à vitesse intégrale/basse vitesse - Répétition Bus à vitesse intégrale/basse vitesse au ralenti ou piloté à l'autre extrémité Grande vitesse état ralenti

Figure 11-18. Exemple de signalisation de reprise de réveil à distance avec appareil à vitesse élevée

Voici une explication de ce qui se produit à chaque t_n:

t0 L'appareil suspendu initie le réveil à distance en pilotant un "K" sur les lignes de données.

t1 Le hub suspendu "B" détecte le "K" sur son port orienté en aval et se réveille suffisamment dans les 900 µs pour filtrer et refléter la reprise en amont et dans tous les ports activés.

t2 Le hub "A" n'est pas suspendu (l'implication est que le port auquel "B" est relié est suspendu de manière sélective), détecte le "K" sur le port suspendu de manière sélective auquel "B" est relié, filtre et reflète le signal de reprise vers "B" dans un délai de 900 µs.

t3 L'appareil cesse de piloter "K" en amont.

t4 Le hub "B" cesse de piloter "K" en amont et dans tous les ports activés et commence à répéter la signalisation en amont vers tous les ports orientés en aval activés.

t5 Le hub "A" termine la séquence de reprise, après l'intervalle de synchronisation approprié, en pilotant une fin de reprise en aval adaptée à la vitesse. (La fin de reprise est un état Repos pour un appareil à grande vitesse ou un EOP à basse vitesse pour un appareil à vitesse intégrale/basse vitesse.)

Le temps de réflexion du hub est beaucoup plus petit que la durée minimum pendant laquelle un appareil USB pilote la reprise en amont. Cette relation garantit que la reprise est propagée en amont et en aval sans écarts.

11.10 Comportement de réinitialisation du hub

La signalisation de la réinitialisation à un hub est définie uniquement en aval, qui est au niveau du port orienté en amont du hub. La signalisation de la réinitialisation requise du hub est décrite dans la Section 7.1.7.5.

Un hub suspendu doit interpréter le début de la réinitialisation sous forme d'évènement de réveil; il doit être réveillé et avoir terminé sa séquence de réinitialisation à la fin de la signalisation de la réinitialisation.

A la fin de la séquence de réinitialisation, un hub est dans l'état suivant:

- L'adresse par défaut du contrôleur de hub est 0.
- Les bits de changement d'état de hub sont définis sur zéro.
- Le répéteur de hub est dans l'état WFSOPFU.
- L'émetteur est dans l'état Inactif.
- Les ports orientés en aval sont dans l'état Non configuré et SE0 est piloté sur tous les ports orientés en aval.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.11 Contrôle d'alimentation de port de hub

Les hubs autoalimentés peuvent avoir des interrupteurs d'alimentation qui contrôlent l'alimentation électrique des ports orientés en aval, mais cela n'est pas requis. Les hubs alimentés par bus doivent avoir des interrupteurs d'alimentation. Un hub avec interrupteurs d'alimentation peut mettre sous tension tous les ports sous forme de groupe/association, chaque port individuellement ou avoir un nombre arbitraire d'associations d'un ou plusieurs ports.

Un hub indique s'il prend ou non en charge la commutation de l'alimentation avec le réglage du champ Logical Power Switching Mode dans *wHubCharacteristics*. Si un hub prend en charge la commutation de l'alimentation par port, l'alimentation d'un port est alors établie lorsqu'une demande SetPortFeature(PORT_POWER) est reçue pour le port. L'alimentation du port est coupée lorsque le port est dans les états Hors tension ou Non configuré. Si un hub prend en charge la commutation de l'alimentation en associations, l'alimentation de tous les ports dans une association est alors établie lorsqu'un port dans une association reçoit une demande SetPortFeature(PORT_POWER). L'alimentation d'une association n'est pas coupée si tous les ports dans une association ne sont pas dans les états Hors tension ou Non configuré. Il faut noter que l'alimentation d'un port n'est pas établie par une demande SetPortFeature(PORT_POWER) si C_HUB_LOCAL_POWER et Local Power Status (dans *wHubStatus*) sont définis sur 1B au moment où la demande est exécutée et la fonctionnalité PORT_POWER est activée.

Bien qu'un hub autoalimenté ne soit pas requis pour effectuer la commutation de l'alimentation, le hub doit prendre en charge l'état Hors tension pour tous les ports. De plus, le hub doit appliquer *PortPwrCtrlMask* (tous les bits définis sur 1B) même si le hub n'a pas d'interrupteurs d'alimentation qui peuvent être commandés par le logiciel du système USB.

Note: Afin de garantir la compatibilité avec les versions précédentes du logiciel USB, les hubs doivent appliquer le champ Logical Power Switching Mode dans *wHubCharacteristics*. Cela vient du fait que certaines versions du logiciel n'utilisent pas la demande SetPortFeature() si le hub indique dans *wHubCharacteristics* que le port ne prend pas en charge la commutation de l'alimentation des ports. Le cas échéant, le champ Logical Power Switching Mode dans *wHubCharacteristics* deviendrait redondant par rapport à cette version de la spécification.

Il convient que le paramétrage de Logical Power Switching Mode pour les hubs sans interrupteurs d'alimentation reflète la manière selon laquelle la surintensité est signalée. Par exemple, si le hub signale des conditions de surintensité sur une base par port, il convient alors que le Logical Power Switching Mode soit défini pour indiquer que la commutation de l'alimentation est commandée sur une base par port.

Pour un hub sans interrupteurs d'alimentation, *bPwrOn2PwrGood* doit être défini sur zéro.

11.11.1 Associations multiples

Un hub peut appliquer un certain nombre d'associations d'alimentation et/ou de surintensité. Un hub qui applique plusieurs associations de surintensité et/ou de commutation d'alimentation, doit définir les champs Logical Power Switching Mode et le Over-current Reporting Mode pour indiquer que l'indication de la commutation de l'alimentation et de la surintensité sont sur une base par port (ces champs sont dans *wHubCharacteristics*). Tous les bits dans *PortPwrCtrlMask* doivent également être définis sur 1B.

Lorsqu'une condition de surintensité se produit sur un appareil de protection contre les surintensités, la surintensité est signalée sur tous les ports qui sont protégés par cet appareil. Lorsque la surintensité est signalée, tous les ports dans le groupe sont placés dans l'état Hors tension et le champ C_PORT_OVER-CURRENT est défini sur 1B sur tous les ports. Lorsque l'état du port est lu à partir de n'importe quel port dans le groupe, le champ PORT_OVER-CURRENT est défini sur 1B tant que la condition de surintensité existe. Le champ C_PORT_OVER-CURRENT doit être effacé dans chaque port individuellement.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

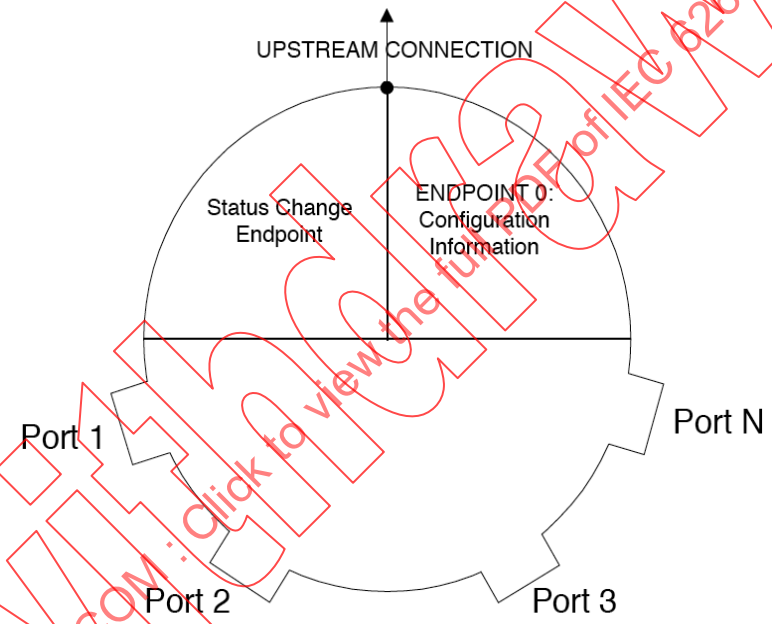
Lorsque de multiples ports partagent un interrupteur d'alimentation, la configuration de PORT_POWER sur n'importe quel port dans le groupe entraîne la mise sous tension de tous les ports. Mais cela n'entraîne pas que les autres ports dans ce groupe quittent l'état Hors tension. Lorsque tous les ports dans un groupe sont dans l'état Hors tension ou lorsque le hub n'est pas configuré, l'alimentation des ports est coupée.

Si un hub applique la commutation de l'alimentation et la surintensité, il n'est pas nécessaire que les groupes de surintensité soient identiques aux groupes de commutation d'alimentation.

Si une condition de surintensité se produit et des interrupteurs d'alimentation sont présents, tous les interrupteurs d'alimentation associés à un circuit de protection contre les surintensités doivent alors être mis hors tension. Si de multiples appareils de protection contre les surintensités sont associés à un seul interrupteur d'alimentation, cet interrupteur est alors éteint lorsqu'un des circuits de protection contre les surintensités indique une condition de surintensité.

11.12 Contrôleur de hub

Le contrôleur de hub est organisé logiquement comme illustré dans la Figure 11-19.



Légende

Anglais	Français
UPSTREAM CONNECTION	CONNEXION EN AMONT
Status Change Endpoint	Point d'extrémité de changement d'état
ENDPOINT 0: Configuration Information	POINT D'EXTREMITE 0: Informations relatives à la configuration

Figure 11-19. Exemple d'organisation du contrôleur de hub

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

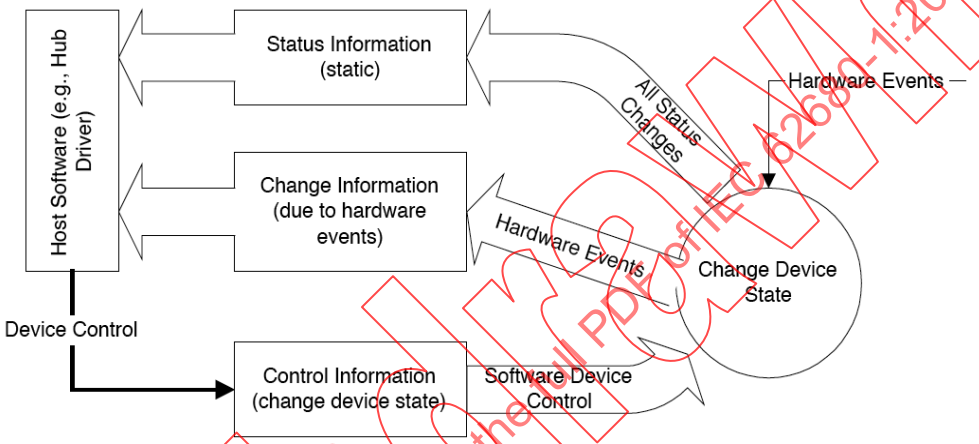
11.12.1 Organisation des points d'extrémité

La classe de hub définit un point d'extrémité supplémentaire en plus du canal de contrôle par défaut, qui est requis pour tous les hubs: le point d'extrémité Changement d'état. Le système hôte reçoit des notifications de changement d'état du port et du hub via le point d'extrémité Changement d'état. Le point d'extrémité Changement d'état est un point d'extrémité d'interruption. Si aucun bit de changement d'état de hub ou de port n'est défini, le hub retourne alors un NAK lorsque le point d'extrémité Changement d'état est analysé. Si un bit de changement d'état est défini, le hub répond avec des données, comme illustré dans la Section 11.12.4, indiquant l'entité (hub ou port) avec un bit de changement défini. Le logiciel du système USB peut utiliser ces données pour déterminer les registres d'état auxquels il faut accéder pour déterminer la cause exacte de l'interruption de changement d'état.

11.12.2 Architecture et utilisation des informations sur le hub

La Figure 11-20 montre comment l'état, le changement d'état et les informations de contrôle font référence aux états d'appareil. Les descripteurs de hub, l'état du hub/port et le contrôle sont accessibles via le canal de contrôle par défaut. Les descripteurs de hub peuvent être lus à tout moment. Lorsqu'un hub détecte un changement sur un port ou lorsque le hub change son propre état, le point d'extrémité Changement d'état transfère les données vers l'hôte sous la forme spécifiée dans la Section 11.12.4.

Les bits de changement d'état du hub ou du port peuvent être définis en raison des évènements matériels ou logiciels. Lorsqu'ils sont définis, ces bits restent définis jusqu'à ce qu'ils soient clairement effacés par le logiciel du système USB via une demande ClearPortFeature() ou une réinitialisation du hub. Tandis qu'un bit de changement est défini, le hub continue à signaler un changement d'état lorsqu'il est analysé jusqu'à ce que tous les bits de changement aient été effacés par le logiciel du système USB.



Légende

Anglais	Français
Host Software (e.g., Hub Driver)	Logiciel hôte (ex.: pilote de hub)
Status Information (static)	Informations relatives à l'état (statiques)
Change Information (due to hardware events)	Informations relatives au changement (en raison d'évènements matériels)
Device Control	Commande d'appareil
Control Information (change device state)	Informations relatives à la commande (changement d'état d'appareil)
All Status Changes	Tous les changements d'état
Hardware Events	Evènements matériels
Software Device Control	Commande d'appareil logicielle
Change Device State	Changement d'état d'appareil

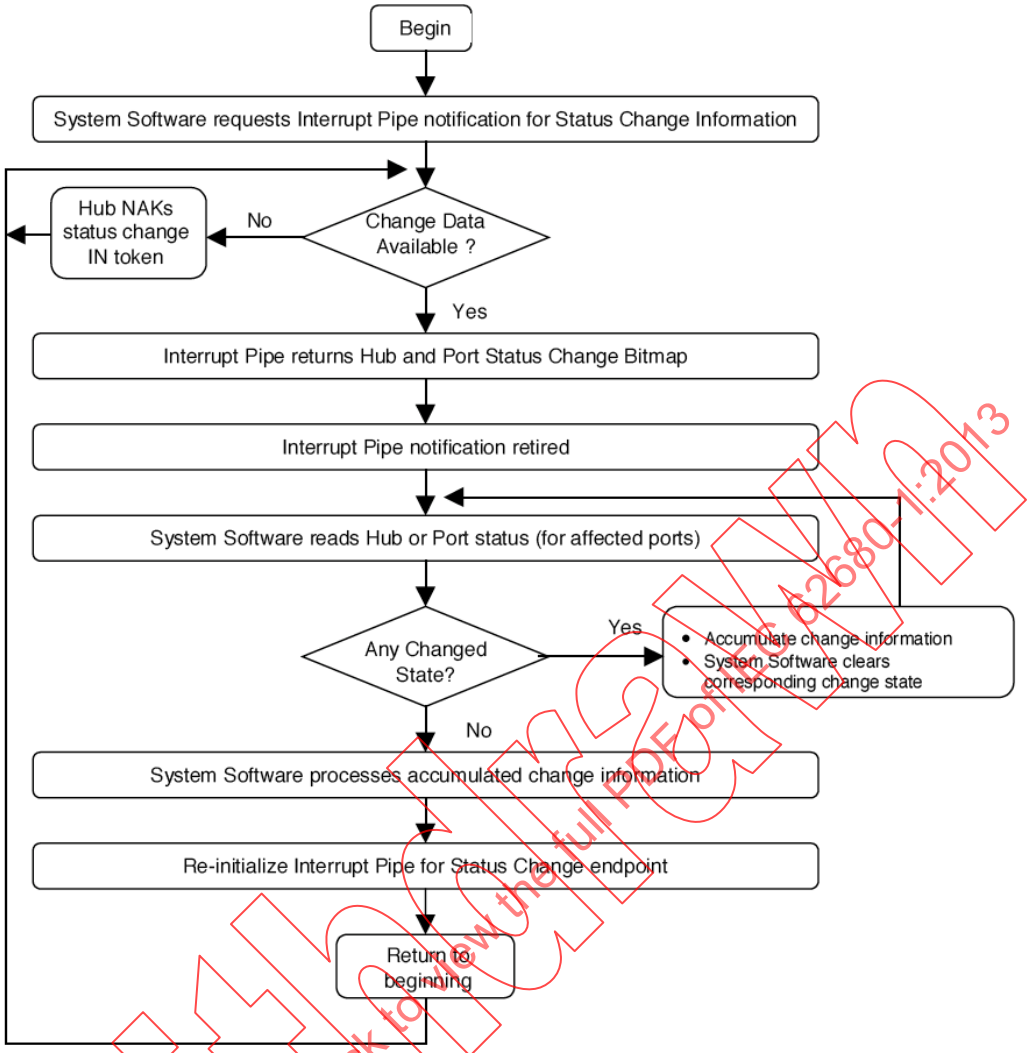
Figure 11-20. Relations entre Etat, Changement d'état et Informations de contrôle et les états d'appareil

Le logiciel du système USB utilise le canal d'interruption associé au point d'extrémité Changement d'état pour détecter les changements d'état de hub et de port.

11.12.3 Traitement d'informations de changement de port

Le hub signale l'état d'un port via les commandes de port pour chaque port. Le logiciel du système UBS acquitte un changement de port en effaçant l'état de changement correspondant au changement d'état signalé par le hub. L'acquittement efface l'état de changement pour ce port de sorte que les futurs transferts de données vers le point d'extrémité Changement d'état ne signalent pas l'évènement précédent. Cela permet de répéter le processus pour d'autres changements (voir Figure 11-21).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Begin	Début
System Software requests Interrupt Pipe notification for Status Change Information	Le logiciel système demande une notification de canal d'interruption pour les informations de changement d'état
Hub NAKs status change IN token	Jeton IN de changement d'état NAK de hub
No	Non
Change Data available?	Données de changement disponibles?
Yes	Oui
Interrupt Pipe returns Hub and Port Status Change Bitmap	Le canal d'interruption retourne le topogramme binaire de changement d'état de hub et de port
Interrupt Pipe notification retired	Notification de canal d'interruption retirée
System Software reads Hub or Port status (for affected ports)	Le logiciel système lit l'état du hub ou du port (pour les ports concernés)
Any Changed State?	Un état modifié?
- Accumulate change information - System Software clears corresponding change state	- Accumuler les informations relatives au changement - Le logiciel système efface l'état de changement correspondant
System Software processes accumulated change information	Le logiciel système traite les informations relatives au changement accumulées
Re-initialize Interrupt Pipe for Status Change endpoint	Réinitialiser le canal d'interruption pour le point d'extrémité Changement d'état
Return to beginning	Retourner au début

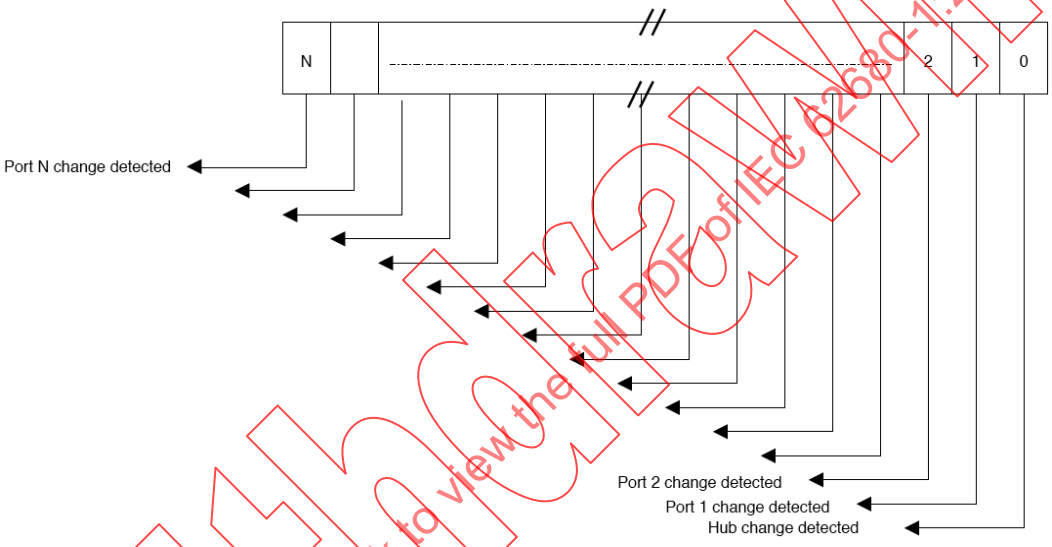
Figure 11-21. Méthode de traitement d'état de port

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.12.4 Topogramme binaire de changement de hub et de port

Le Topogramme binaire de changement de hub et de port, illustré dans la Figure 11-22, indique si le hub ou un port a rencontré un changement d'état. Ce topogramme binaire indique également quel port a eu un changement d'état. Le hub retourne cette valeur sur le point d'extrémité Changement d'état. Les hubs signalent cette valeur par incréments d'octets. Ainsi, si un hub a six ports, il retourne une quantité d'octets et signale un zéro dans les emplacements de champ de numéro de port invalides. Le logiciel du système USB est informé du nombre de ports sur un hub (cela est indiqué dans le descripteur de hub) et décode le Topogramme binaire de changement de hub et de port en conséquence. Le hub signale tout changement d'état de hub dans le bit zéro du Topogramme binaire de changement de hub et de port.

La taille du Topogramme binaire de changement de hub et de port varie d'une taille minimum d'un octet. Les hubs signalent uniquement autant de bits qu'il y a de ports sur le hub, selon l'exigence de granularité d'octet (à savoir, arrondi à l'octet le plus proche).



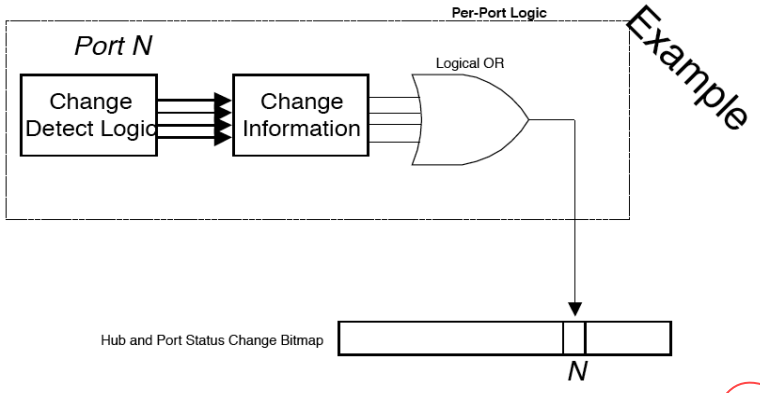
Légende

Anglais	Français
Port N change detected	Changement de port N détecté
Port 2 change detected	Changement de port 2 détecté
Port 1 change detected	Changement de port 1 détecté
Hub change detected	Changement de hub détecté

Figure 11-22. Topogramme binaire de changement de hub et de port

Chaque fois que le point d'extrémité Changement d'état est analysé par le contrôleur hôte et un des bits Etat modifié est différent de zéro, le Topogramme binaire de changement de hub et de port est retourné. La Figure 11-23 montre un exemple de mécanisme de création pour les bits de changement de hub et de port.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Port N	Port N
Per-Port Logic	Logique par port
Example	Exemple
Change Detect Logic	Logique de détection de changement
Change Information	Informations relatives au changement
Logical OR	OU logique
Hub and Port Status Change Bitmap	Topogramme binaire de changement d'état de hub et de port

Figure 11-23. Exemple d'échantillonnage de bit de changement de hub et de port

11.12.5 Reporting et rétablissement de surintensité

Les appareils USB doivent être conçus pour satisfaire aux normes de sécurité applicables. Généralement, cela signifie qu'un hub autoalimenté applique une limitation de courant à ses ports orientés en aval. Si une condition de surintensité se produit, cela entraîne un changement de statut et d'état dans un ou plusieurs ports. Ce changement est signalé au logiciel du système USB de sorte qu'il peut prendre une action corrective.

Un hub peut être conçu pour signaler une surintensité comme un évènement de port ou de hub. Le champ du descripteur de hub *wHubCharacteristics* est utilisé pour indiquer les fonctionnalités de reporting d'un hub particulier (voir Section 11.23.2). Le bit d'état de surintensité dans le champ d'état de hub ou de port indique l'état de la détection de surintensité lorsque l'état est retourné. Le bit de changement d'état de surintensité dans le champ Hub ou Port indique si l'état de surintensité a changé.

Lorsqu'un hub rencontre une condition de surintensité, il doit placer tous les ports affectés dans l'état Hors tension. Si un hub a une commutation d'alimentation par port et une limitation de courant par port, une surintensité sur un port peut encore entraîner une chute de l'alimentation au-dessous des minima spécifiés sur un autre port. Dans ce cas, le port affecté est placé dans l'état Hors tension et C_PORT_OVER_CURRENT est défini pour le port, mais PORT_OVER_CURRENT n'est pas défini. Si le hub a une détection de surintensité sur une base de hub, une condition de surintensité sur le hub entraîne le passage de tous les ports dans l'état Hors tension. Mais, dans ce cas, ni C_PORT_OVER_CURRENT ni PORT_OVER_CURRENT n'est défini pour les ports affectés.

Il convient que les actions de récupération d'hôte pour un évènement de surintensité incluent ce qui suit:

1. L'hôte obtient la notification de changement du hub avec évènement de surintensité.
2. L'hôte extrait les informations de changement de hub ou de port appropriées (selon les informations dans le topogramme binaire de changement).
3. L'hôte attend que le bit d'état de surintensité soit effacé à 0.
4. L'hôte met sous/hors tension tous les ports nécessaires (ex.: émet une demande SetPortFeature(PORT_POWER) pour chaque port).
5. L'hôte ré-énumère tous les ports affectés.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.12.6 Traitement des énumérations

Les commandes de classe d'appareil de hub sont utilisées pour manipuler son état de port orienté en aval. Lorsqu'un appareil est relié, l'évènement de raccordement d'appareil est détecté par le hub et signalé sur l'interruption de changement d'état. L'hôte accepte le report de changement d'état et demande un SetPortFeature(PORT_RESET) sur le port. Dans le cadre de la séquence de réinitialisation de bus, une détection de vitesse est effectuée par le matériel de port du hub.

La demande Get_Status(PORT) invoquée par l'hôte retourne une indication "pas PORT_LOW_SPEED et PORT_HIGH_SPEED" pour un port orienté en aval fonctionnant à grande vitesse. Le Get_Status(PORT) signale "PORT_LOW_SPEED" pour un port orienté en aval fonctionnant à basse vitesse. Le Get_Status(PORT) signale "pas PORT_LOW_SPEED et pas PORT_HIGH_SPEED" pour un port orienté en aval à vitesse intégrale.

Quand l'appareil est déconnecté du port, le port signale le changement d'état via le point d'extrémité de changement d'état et le port est reconnecté au répéteur grande vitesse. Le processus est alors prêt à être répété à la prochaine détection de raccordement d'appareil.

11.13 Configuration de hub

Les hubs sont configurés via les commandes de configuration d'appareil USB normalisées. Un hub qui n'est pas configuré se comporte comme un autre appareil qui n'est pas configuré eu égard aux exigences d'alimentation et à l'adressage. Si un hub applique la commutation d'alimentation, aucune alimentation n'est fournie sur les ports orientés en aval alors que le hub n'est pas configuré. La configuration d'un hub active le point d'extrémité Changement d'état. Le logiciel du système USB peut alors émettre des commandes au hub pour mettre sous/hors tension l'alimentation du port aux moments appropriés.

Le logiciel du système USB examine les informations du descripteur de hub pour déterminer les caractéristiques du hub. En examinant les caractéristiques du hub, le logiciel du système USB garantit que les topologies d'alimentation illégales ne peuvent pas alimenter les ports du hub si cela enfreint la topologie d'alimentation USB. Les informations d'état d'appareil et de configuration peuvent être utilisées pour déterminer s'il convient d'utiliser le hub comme un bus ou un appareil autoalimenté. Le Tableau 11-12 résume les informations et comment il est possible de les utiliser pour déterminer les exigences d'alimentation actuelles du hub.

Tableau 11-12. Résumé du mode de fonctionnement d'alimentation du hub

Descripteur de configuration		Etat de l'appareil de hub (autoalimentation)	Explication
MaxPower	**bmAttributes (autoalimentés)**		
0	0	N/A	N/A Il s'agit d'un ensemble illégal d'informations.
0	1	0	N/A Appareil autoalimenté uniquement, mais sans alimentation locale, ne peut pas se connecter au bus et communiquer.
0	1	1	Hub autoalimenté uniquement et l'alimentation locale est bonne. L'état du hub indique également une alimentation locale bonne, voir Section 11.16.2.5. La fonctionnalité du hub est valide si aucune restriction de profondeur n'est violée.
> 0	0	N/A	Hub alimenté par bus uniquement. Les ports orientés en aval peuvent ne pas être alimentés sauf si la topologie actuelle l'autorise. L'état de l'appareil du hub Autoalimenté est sans intérêt en combinaison avec un *bmAttributes.Self-Powered* à zéro.
> 0	1	0	Ce hub est capable des modes autoalimenté et alimenté par bus. Il est actuellement disponible uniquement comme hub alimenté par bus.
> 0	1	1	Ce hub est capable des modes autoalimenté et alimenté par bus. Il est actuellement disponible comme hub autoalimenté.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Un hub autoalimenté a une alimentation électrique locale, mais peut en option tirer une charge unitaire de sa connexion en amont. Cela permet à l'interface de fonctionner lorsque l'alimentation locale n'est pas disponible (voir Section 7.2.1.2). Lorsque l'alimentation locale est coupée (une condition de surintensité dans tout un hub ou une alimentation locale est coupée), un hub de ce type reste dans l'état Configuré mais passe tous les ports (que ce soit amovible ou non amovible) à l'état Hors tension. Lorsque l'alimentation locale est coupée, toutes les informations de port et de changement indiquent zéro et toutes les demandes SetPortFeature() sont ignorées (la demande est traitée comme non opérationnelle). Le hub utilise le point d'extrémité Changement d'état pour informer le logiciel du système USB de l'évènement de hub (voir Section 11.24.2.6 pour plus de détails sur l'état du hub).

Le champ *MaxPower* dans le descripteur de configuration est utilisé pour signaler au système la puissance maximum que le hub tire de VBUS lorsque la configuration est sélectionnée. Pour les hubs alimentés par bus, la valeur indiquée ne doit pas inclure la puissance pour l'un des ports orientés en aval externes. Les appareils externes reliés au hub indiquent leurs exigences d'alimentation individuelles.

Un appareil composite peut alimenter l'électronique du hub et les appareils raccordés en permanence du VBUS. Toute la charge peut être signalée dans le descripteur de configuration du hub avec les appareils raccordés en permanence indiquant chacun autoalimenté, avec *MaxPower* zéro dans leurs descripteurs de configuration respectifs.

11.14 Traducteur de transactions

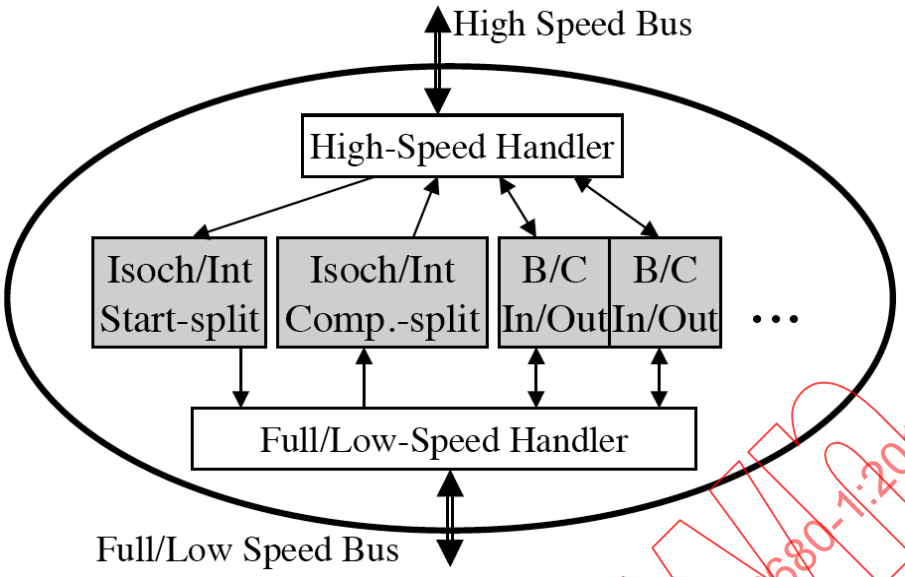
Un hub a une responsabilité spéciale lorsqu'il fonctionne à grande vitesse et a des appareils à vitesse intégrale/basse vitesse raccordés aux ports orientés en aval. Dans ce cas, le hub doit isoler l'environnement de signalisation à grande vitesse de l'environnement de signalisation à vitesse intégrale/basse vitesse. Cette fonction est exécutée par la portion Traducteur de transactions (TT) du hub.

Cette section définit le comportement requis du traducteur de transactions.

11.14.1 Présentation

La Figure 11-24 présente un aperçu du traducteur de transactions. Le TT est chargé de participer aux transactions fractionnées à grande vitesse sur le bus à grande vitesse via son port orienté en amont et d'émettre les transactions à vitesse intégrale/basse vitesse correspondantes sur ses ports orientés en aval qui fonctionnent à vitesse intégrale/basse vitesse. Le TT agit comme une fonction à grande vitesse sur le bus à grande vitesse et remplit le rôle d'un contrôle hôte pour ses ports orientés en aval qui fonctionnent à vitesse intégrale/basse vitesse. Le TT inclut un gestionnaire à grande vitesse pour traiter les transactions à grande vitesse. Le TT inclut également un gestionnaire à vitesse intégrale/basse vitesse qui remplit le rôle d'un contrôle hôte sur les ports orientés en aval qui fonctionnent à vitesse intégrale/basse vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Anglais	Français
High Speed Bus	Bus à grande vitesse
High-Speed Handler	Gestionnaire à grande vitesse
B/C In/Out	In/Out BC
Full/Low-Speed Handler	Gestionnaire à vitesse intégrale/basse vitesse
Full/Low Speed Bus	Bus à vitesse intégrale/basse vitesse

Figure 11-24. Présentation du traducteur de transactions

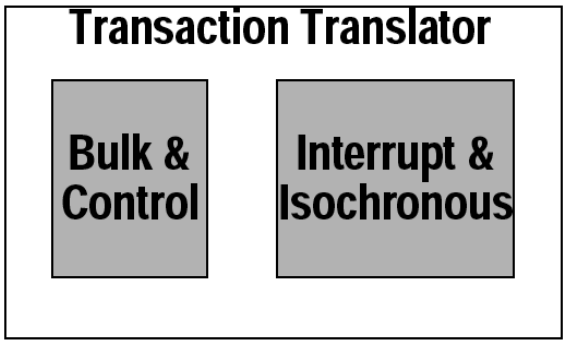
Le TT a des tampons (illustrés en gris dans la figure) pour contenir les transactions qui sont en cours et suit l'état de chaque transaction mise en tampon lorsqu'elle est traitée par le TT. Les tampons fournissent la connexion entre les systèmes de traitement à grande vitesse et à vitesse intégrale/basse vitesse. L'état suivant ce que fait le TT pour chaque transaction dépend du type de transfert USB spécifique de la transaction (à savoir, en masse, contrôle, interruption et isochrone). Le gestionnaire à grande vitesse accepte les transactions start-split à grande vitesse ou répond aux transactions complete-split à grande vitesse. Le gestionnaire à grande vitesse place les transactions start-split dans des tampons locaux pour l'utilisation du gestionnaire à vitesse intégrale/basse vitesse.

Les transactions start-split mises en tampon fournissent au gestionnaire à vitesse intégrale/basse vitesse les informations dont il a besoin pour émettre les transactions à vitesse intégrale/basse vitesse correspondantes aux appareils à vitesse intégrale/basse vitesse raccordés sur les ports orientés en aval. Le gestionnaire à vitesse intégrale/basse vitesse met en tampon les résultats de ces transactions à vitesse intégrale/basse vitesse de manière à ce qu'elles puissent être retournées avec une transaction complete-split correspondante sur le bus à grande vitesse.

La conversion générale entre les transactions à vitesse intégrale/basse vitesse et le protocole de transaction fractionnée à grande vitesse correspondant est décrite dans la Section 8.4.2. Plus de détails sur les types de transfert spécifiques pour les transactions fractionnées figurent ultérieurement dans ce chapitre.

Le gestionnaire à grande vitesse du TT fonctionne indépendamment du gestionnaire à vitesse intégrale/basse vitesse. Les deux systèmes de traitement utilisent les tampons de transactions locaux pour échanger des informations si nécessaire.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Anglais	Français
Transaction Translator	Traducteur de transactions
Bulk & Control	Masse & Contrôle
Interrupt & Isochronous	Interruption & Isochrone

Figure 11-25. Sections de tampon périodique et non périodique du TT

Le TT a deux sections de tampon et de suivi d'état (en gris dans la Figure 11-24 et dans la Figure 11-25): périodique (pour les transactions isochrone/interruption à vitesse intégrale/basse vitesse) et non périodique (pour les transactions à vitesse intégrale/basse vitesse en masse/de contrôle). Les exigences vis-à-vis du TT pour ces deux sections de tampon et de suivi d'état sont différentes. Chaque exigence est décrite une à une ultérieurement dans ce chapitre.

11.14.1.1 Traitement des données entre grande vitesse et vitesse intégrale/basse vitesse

L'hôte convertit les demandes de transfert impliquant un appareil à vitesse intégrale/basse vitesse en transactions fractionnées à grande vitesse correspondantes vers le TT auquel est raccordé l'appareil.

Les paquets de préambule à basse vitesse (PRE) ne sont jamais utilisés sur le bus à grande vitesse pour indiquer une transaction à basse vitesse. Une transaction à basse vitesse est codée dans le jeton de transaction fractionnée.

L'hôte peut avoir un seul planning des transactions qui doivent être émises vers les appareils. L'unique planning peut être utilisé pour contenir les transactions à grande vitesse et les transactions fractionnées à grande vitesse utilisées pour communiquer avec les appareils à vitesse intégrale/basse vitesse.

11.14.1.2 Transactions fractionnées de contrôleur hôte et de TT

Le contrôleur hôte utilise le protocole de transaction fractionnée pour initier des transactions à vitesse intégrale/basse vitesse via le TT, puis pour déterminer l'état d'exécution de la transaction à vitesse intégrale/basse vitesse. Cette approche permet au contrôleur hôte de démarrer une transaction à vitesse intégrale/basse vitesse, puis de continuer avec d'autres transactions à grande vitesse tout en évitant d'avoir à attendre le traitement/l'exécution de la transaction plus lente à sa vitesse. Une transaction fractionnée à grande vitesse a deux parties: un start-split et un complete-split. Les transactions fractionnées sont uniquement utilisées entre le contrôleur hôte et un hub. Aucun autre appareil à grande vitesse/vitesse intégrale/basse vitesse ne participe aux transactions fractionnées.

Quand le contrôleur hôte envoie une transaction fractionnée à grande vitesse, la transaction fractionnée est adressée au TT pour cet appareil. Ce TT accepte la transaction et la met en tampon localement. Le gestionnaire à grande vitesse répond avec un établissement de liaison approprié pour informer le contrôleur hôte que la transaction est acceptée. Toutes les transactions fractionnées n'ont pas de phase d'établissement de liaison vers le start-split. Les transactions start-split sont maintenues temporairement dans un tampon de transaction TT.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Le gestionnaire à vitesse intégrale/basse vitesse traite les transactions périodiques start-split enregistrées dans le tampon de transaction périodique (dans l'ordre) comme le bus à vitesse intégrale/basse vitesse en aval à vitesse intégrale/basse vitesse est prêt pour la transaction "suivante". Le gestionnaire à vitesse intégrale/basse vitesse accepte toute information résultante du bus en aval (en réponse à la transaction à vitesse intégrale/basse vitesse) et les accumule dans un tampon local pour une transmission ultérieure vers le contrôleur hôte.

A un moment ultérieur approprié, le contrôleur hôte envoie une transaction complete-split à grande vitesse pour récupérer l'état/les données/le résultat pour des transactions à vitesse intégrale/basse vitesse appropriées. Le gestionnaire à grande vitesse vérifie cette transaction complete-split à grande vitesse avec la réponse en tête du tampon de transaction local approprié et répond en conséquence. Les séquences de transaction fractionnée spécifiques sont définies pour chaque type de transfert USB dans les sections ultérieures.

11.14.1.3 Traducteurs de transactions multiples

Un hub a deux choix pour organiser les traducteurs de transactions (TT). Un hub peut avoir un TT pour tous les ports orientés en aval qui ont des appareils à vitesse intégrale/basse vitesse raccordés ou le hub peut avoir un TT pour chaque port orienté en aval. Le hub doit signaler son organisation dans le descripteur de classe de hub.

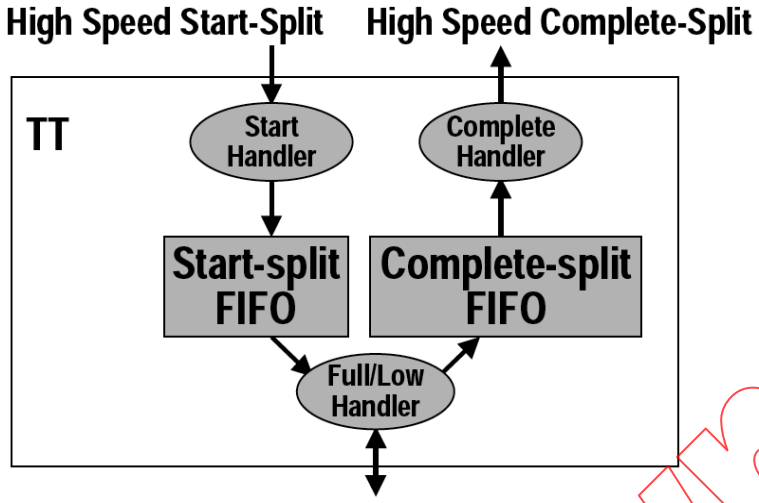
11.14.2 Programmation du traducteur de transactions

Comme le gestionnaire à grande vitesse accepte les start-splits, les informations sur les transactions à vitesse intégrale/basse vitesse et les données pour les SORTIES ou les informations sur les transactions pour les ENTREES s'accumulent dans des tampons attendant leur service sur le bus en aval. L'hôte gère les transactions TT périodiques différemment des tampons de transactions non périodiques.

11.14.2.1 Tampon de transaction (périodique) isochrone/interruption TT

Les transactions périodiques ont des exigences de synchronisation strictes à satisfaire sur un bus à vitesse intégrale/basse vitesse (comme défini par le point d'extrémité spécifique et le type de transfert). Les transactions doivent donc se déplacer dans le bus à grande vitesse, via le TT, dans le bus à vitesse intégrale/basse vitesse, puis de nouveau dans le TT, et sur le bus à grande vitesse de manière synchronisée. Un aperçu du pipeline de microtrame de mise en tampon dans le TT est illustré à la Figure 11-26. Une transaction commence comme un start-split sur le bus à grande vitesse, est acceptée par le gestionnaire à grande vitesse et est enregistrée dans le tampon de transaction start-split. Le gestionnaire à vitesse intégrale/basse vitesse utilise la transaction start-split suivante en tête du tampon de transaction start-split quand il est temps d'émettre la transaction à vitesse intégrale/basse vitesse périodique suivante sur le bus en aval. Les résultats de la transaction sont accumulés dans le tampon de transaction complete-split. Le TT répond à un complete-split de l'hôte et extrait la bonne réponse du tampon de transaction complete-split. Cela exécute le flux pour une transaction périodique dans le TT, ce qu'on appelle le pipeline de transaction périodique.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Anglais	Français
High Speed Start-Split	Start-split à grande vitesse
High Speed Complete-Split	Complete-Split à grande vitesse
Start Handler	Gestionnaire Start
Complete Handler	Gestionnaire Complete
Start-split FIFO	FIFO Start-split
Complete-Split FIFO	FIFO Complete-split
Full/Low Handler	Gestionnaire à vitesse intégrale/basse vitesse

Figure 11-26. Pipeline de microtrame TT pour transactions fractionnées périodiques

Le TT applique un pipeline traditionnel de transactions avec ses tampons de transactions périodiques. Il y a un espace de tampon séparé pour les start-splits et les complete-splits. L'hôte est responsable du remplissage du tampon de transaction start-split et du vidage du tampon de transaction complete-split. Le logiciel hôte gère le contrôleur hôte pour générer les transactions fractionnées à grande vitesse aux moments corrects pour éviter les sur/sous-utilisations dans les tampons de transactions périodiques du TT. Le contrôleur hôte envoie les données "juste à temps" pour les SORTIES à vitesse intégrale/basse vitesse et récupère les données de réponse des ENTREES à vitesse intégrale/basse vitesse pour garantir que l'espace de tampon de transaction périodique requis dans le TT est le minimum possible. Voir Section 11.18 pour plus d'informations.

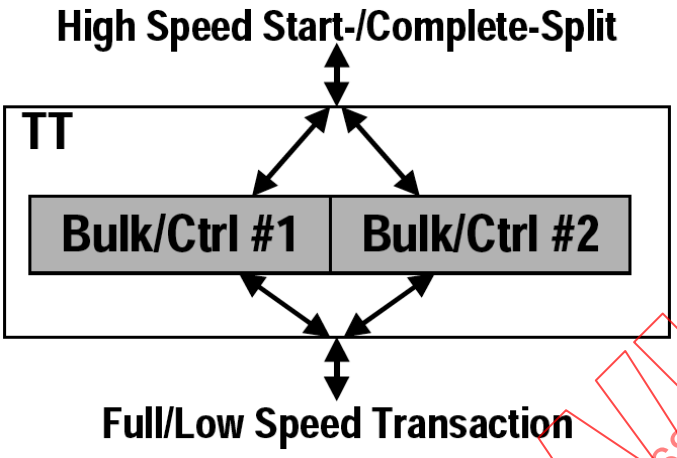
L'USB définit strictement les exigences de synchronisation des transactions périodiques et les fonctionnalités de transport isochrones des bus à grande vitesse et à vitesse intégrale/basse vitesse. Cela permet à l'hôte de prédire avec précision quand les données des transactions périodiques doivent être déplacées sur les bus à vitesse intégrale/basse vitesse et grande vitesse, lorsqu'un client demande un transfert de données avec un point d'extrémité périodique à vitesse intégrale/basse vitesse. Par conséquent, l'hôte peut "canaliser" les données vers/du TT de manière à ce qu'elles se déplacent de manière synchronisée avec son point d'extrémité cible. Lorsque la configuration d'un appareil à vitesse intégrale/basse vitesse avec des points d'extrémité périodiques est défini, l'hôte fait circuler les données vers/du TT pour maintenir le fonctionnement normal des points d'extrémité de l'appareil.

11.14.2.2 Mise en mémoire tampon des transactions (non périodiques) en masse/de contrôle du TT

Les transactions non périodiques n'ont pas d'exigences de synchronisation, mais le TT prend en charge le débit à vitesse intégrale/basse vitesse maximum autorisé. Un TT fournit quelques tampons de transaction pour les transactions à vitesse intégrale/basse vitesse en masse/de contrôle. L'hôte et le TT utilisent des mécanismes simples de contrôle de débit (NAK) pour gérer les tampons de transactions non périodiques en masse/de contrôle. L'hôte émet une transaction start-split, et si un espace tampon est disponible, le TT accepte la transaction. Le gestionnaire à vitesse intégrale/basse vitesse utilise les informations mises en tampon pour émettre la transaction à vitesse intégrale/basse vitesse en aval, puis utilise le même tampon pour conserver les résultats (ex.: établissement de liaison ou données ou

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

expiration). Le tampon est alors vidé avec un complete-split à grande vitesse correspondant et le processus continue. La Figure 11-27 montre un exemple d'aperçu d'un TT qui a deux tampons en masse/de contrôle.



Anglais	Français
High Speed Start-/Complete-Split	Start-/Complete-Split à grande vitesse
Bulk/Ctrl # 1	Masse/Ctrl # 1
Bulk/Ctrl # 2	Masse/Ctrl # 2
Full/Low Speed Transaction	Transaction à vitesse intégrale/basse vitesse

Figure 11-27. Mise en tampon non périodique du TT

11.14.2.3 Programmation des transactions du gestionnaire à vitesse intégrale/basse vitesse

Le gestionnaire à vitesse intégrale/basse vitesse utilise un schéma de priorité programmé simple pour entretenir les transactions en cours sur le bus en aval. Lorsque le gestionnaire à vitesse intégrale/basse vitesse finit une transaction sur le bus en aval, il prend la transaction start-split suivante dans le tampon de transaction périodique start-split (si disponible). S'il n'y a pas de transactions périodiques start-split disponibles dans le tampon, le gestionnaire à vitesse intégrale/basse vitesse peut essayer une transaction en masse/de contrôle. S'il y a des transactions start-split en cours dans les tampons en masse/de contrôle et s'il reste suffisant de temps dans la trame 1 ms à vitesse intégrale/basse vitesse pour compléter la transaction, le gestionnaire à vitesse intégrale/basse vitesse émet une des transactions en masse/de contrôle (à la ronde). La Figure 11-28 montre un pseudocode pour l'algorithme de programmation de transactions start-split du gestionnaire à vitesse intégrale/basse vitesse.

Le TT séquence également le pipeline de transactions basé sur le minuteur de microtrame à grande vitesse pour garantir qu'il ne démarre pas les transactions périodiques à vitesse intégrale/basse vitesse trop tôt ou trop tard. La procédure "Advance_pipeline" dans le pseudocode est utilisée pour aider le TT à maintenir l'avancement du "pipeline" de microtrame. Cette procédure est décrite en détail dans la Figure 11-67.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

```
While (1) loop
   While (not end of microframe) loop
      -- process next start-split transaction
      If available periodic start-split transaction then
         Process next full-/low-speed periodic transaction
      Else if (available bulk/control transaction) and
              (fits in full-/low-speed 1 ms frame) then
         Process one transaction
      End if
   End loop

   Advance_Pipeline();  -- see description in Figure 11-67(below)
End loop
```

Anglais	Français
While (1) loop	Pendant (1) boucle
While (not end of microframe) loop	Pendant (non fin de microtrame) boucle
Process next start-split transaction	Procéder à la transaction start-split suivante
If available periodic start-split transaction then	Puis, si disponible, transaction start-split périodique
Process next full-/low-speed periodic transaction	Procéder à la transaction périodique à vitesse intégrale/basse vitesse suivante
Else if (available bulk/control transaction) and	Sauf si (transaction de masse/de contrôle disponible) et
(fits in full-low speed 1 ms frame the)	(trame 1 ms à vitesse intégrale/basse vitesse) puis
Process one transaction	Procéder à une transaction
End if	Fin si
End loop	Fin boucle
See description in Figure 11-67 (below)	Voir description à la Figure 11-67 (ci-dessous)

Figure 11-28. Exemple de programmation du gestionnaire à vitesse intégrale/basse vitesse pour start-splits

Comme décrit précédemment dans ce chapitre, le TT dérive le minuteur SOF 1 ms du bus en aval de la microtrame 125 µs à grande vitesse. Cela signifie que l'hôte et le TT ont le même temps de trame 1 ms pour tous les TT. Etant donné la relation stricte entre les trames et la microtrame zéro, il n'est pas nécessaire de transporter des informations de synchronisation explicites dans les transactions split périodiques envoyées au TT. Voir Section 11.18 pour plus d'informations.

11.15 Informations sur la notation des transactions fractionnées

Les sections suivantes décrivent les détails des phases de transactions et séquences de flux des transactions fractionnées pour les différents types de transfert USB: masse/contrôle, interruption et isochrone. Chaque description affiche également un exemple détaillé de diagrammes d'états d'hôte et de TT pour obtenir les définitions de transactions requises. Il convient que les schémas ne soient pas pris comme une mise en œuvre requise, mais pour spécifier le comportement requis. L'Annexe A inclut un exemple de séquences de transactions à grande vitesse et à vitesse intégrale avec différents résultats pour clarifier les relations entre le contrôleur hôte, le TT et un point d'extrémité à vitesse intégrale.

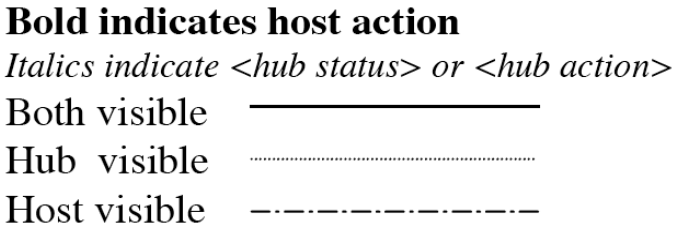
La basse vitesse n'est pas traitée en détail, car outre la gestion du paquet PRE (qui est défini au Chapitre 8), il n'y a pas de différences de séquençage de paquet entre la basse vitesse et la vitesse intégrale.

Pour chaque direction de transfert de données, les figures de référence montrent également les séquences de flux possibles pour la portion start-split et complete-split de chaque type de transfert de transaction split.

Les transitions dans les figures des séquences de flux ont des libellés qui correspondent aux transitions dans les diagrammes d'états de l'hôte et du TT. Ces libellés sont également inclus dans les exemples en Annexe A. Les libellés à trois caractères ont la forme: < S | C >< T | D | H | E ><nombre>. S indique qu'il s'agit d'un libellé start-split. S indique qu'il s'agit d'un libellé complete-split. T indique une phase de jeton; D indique une phase de données; H indique une phase d'établissement de données; E indique un cas d'erreur. Le numéro distingue simplement les différents libellés de même cas/phase dans la même partie de transaction fractionnée.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Les figures de séquences de flux identifient en outre la visibilité des transitions selon la légende dans la Figure 11-29. Les séquences de flux incluent également certaines indications d'état requises dans l'hôte ou le TT ou les actions prises. La légende illustrée dans la Figure 11-29 indique comme elles sont identifiées.



Anglais	Français
Bold indicates host action	Gras indique l'action de l'hôte
Italics indicate < hub status > or < hub action >	Italique indique < état du hub > ou < action du hub >
Both visible	Les deux visibles
Hub visible	Hub visible
Host visible	Hôte visible

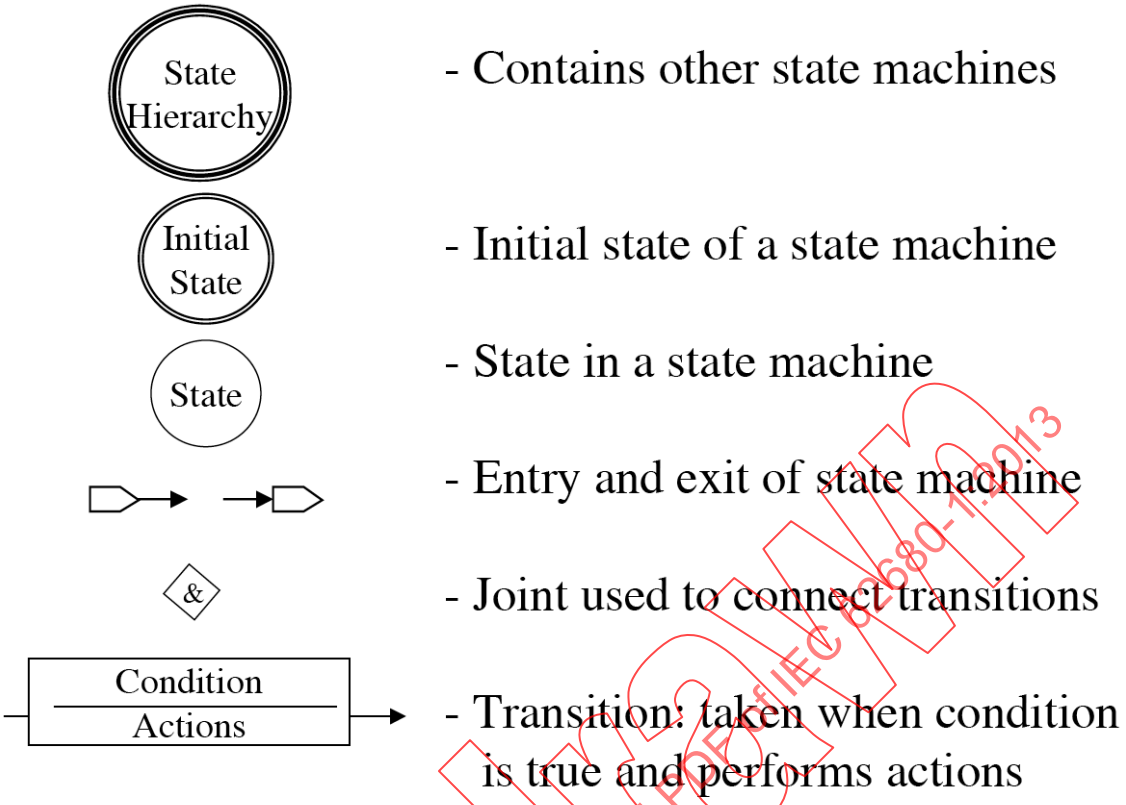
Figure 11-29. Légende de la séquence de flux

La Figure 11-30 montre la légende des diagrammes d'états. Un cercle avec une bordure à trois lignes indique une référence à un autre diagramme d'états (hiérarchique). Un cercle avec une bordure à deux lignes indique un état initial. Un cercle avec une bordure à une ligne indique un état simple.

Un diamant (joint) est utilisé pour assembler plusieurs transitions en un point commun. Un joint permet une transition d'entrée simple avec de multiples transitions de sortie ou de multiples transitions d'entrée et une seule transition de sortie. Toutes les conditions sur les transitions d'un chemin impliquant un joint doivent être vraies pour le chemin à prendre. Un chemin est simplement une séquence de transitions impliquant un ou plusieurs joints.

Une transition est identifiée avec un bloc avec une ligne au milieu séparant la condition (partie supérieure) et les actions (partie inférieure). La condition doit être vraie pour effectuer la transition. Les actions sont exécutées si la transition est effectuée. La syntaxe des actions et conditions est VHDL. Un cercle inclut un nom en gras et en option une ou plusieurs actions qui sont exécutées à l'entrée dans l'état.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
State Hierarchy	Hiérarchie des états
Contains other state machines	Contient d'autres diagrammes d'états
Initial State	Etat initial
Initial state of a state machine	Etat initial d'un diagramme d'états
State	Etat
State in a state machine	Etat dans un diagramme d'états
Entry and exit of state machine	Entrée et sortie du diagramme d'états
Joint used to connect transitions	Assemblage utilisé pour connecter les transitions
Condition Actions	Condition Actions
Transition: taken when condition is true and perform actions	Transition: prise lorsque la condition est vraie et exécuter les actions

Figure 11-30. Légende des diagrammes d'états

Les descriptions des transactions fractionnées pour les quatre types de transfert font référence au statut de la transaction à vitesse intégrale/basse vitesse sur le bus en aval du TT. Ce statut est utilisé par le gestionnaire à grande vitesse pour déterminer sa réponse à une transaction complete-split. Le statut est uniquement visible dans une mise en œuvre TT et est utilisé dans la spécification pour faciliter l'explication uniquement. Les valeurs de statut définies sont les suivantes:

- Prêt - La transaction est terminée sur le bus à vitesse intégrale/basse vitesse orienté en aval avec le résultat suivant:
  - Prêt/NAK - Un établissement de liaison NAK a été reçu.
  - Prêt /trans_err - La transaction à vitesse intégrale/basse vitesse a rencontré une erreur dans la transaction. Les erreurs possibles sont les suivantes: Défaillance de contrôle PID to PID_invert bits, défaillance de contrôle CRC5, PID incorrect, temporisation, défaillance de contrôle CRC16, longueur de paquet incorrecte, erreur de bourrage de bits, EOP erroné.
  - Prêt /ACK - Un établissement de liaison ACK a été reçu.
  - Prêt/Arrêt - Un établissement de liaison STALL a été reçu.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

  - Prêt /Données - Un paquet de données a été reçu et le contrôle CRC a réussi. (ENTREE en masse/de contrôle).
  - Prêt /dernières données - Un paquet de données a été reçu fini. (ENTREE isochrone/interruption).
  - Prêt /plus de données - Un paquet de données a été reçu lorsque le minuteur de microtrame s'est produit (IN isochrone/interruption).

- Ancien - Un complete-split a été reçu par le gestionnaire à grande vitesse pour une transaction qui avait précédemment un statut "prêt". Les résultats de statut possibles sont les mêmes que pour le statut Prêt. Il s'agit de l'état initial pour un tampon avant qu'il n'ait été utilisé pour une transaction.
- En cours - La transaction attend d'être complétée sur le bus à vitesse intégrale/basse vitesse orienté en aval.

Les figures utilisent "ancien/x" et "prêt/x" pour indiquer un des statuts ancien ou prêt respectivement.

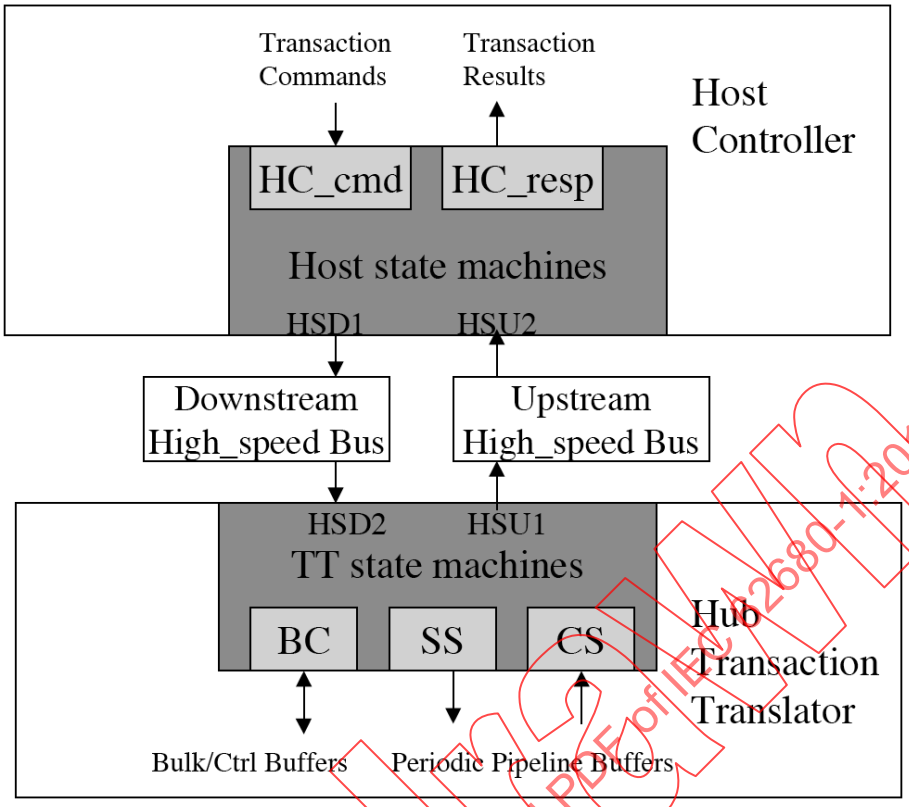
Les diagrammes d'états des transactions fractionnées dans le reste de ce chapitre sont présentés dans le cadre de la Figure 11-31. Les diagrammes d'états du contrôleur hôte sont situés dans le contrôleur hôte. Le contrôleur hôte entraîne l'émission des paquets en aval (étiquetés comme HSD1) et il reçoit les paquets en amont (étiquetés comme HSU2).

Les diagrammes d'états du traducteur de transactions sont situés dans le TT. Le TT entraîne l'émission des paquets en amont (étiquetés comme HSU1) et il reçoit les paquets en aval (étiquetés comme HSD2).

Le contrôleur hôte a des commandes qui lui indiquent quelle transaction fractionnée suivante à émettre pour un point d'extrémité. Le contrôle hôte suit les transactions pour plusieurs points d'extrémité. Le TT a un état dans les tampons qui suit les transactions pour plusieurs points d'extrémité.

L'Annexe B inclut quelques déclarations qui ont été utilisées pour établir les diagrammes d'états et peuvent être utiles pour comprendre les détails supplémentaires des diagrammes d'états. Il y a plusieurs procédures et fonctions de pseudocode pour les conditions et actions. Des descriptions simples figurent également en Annexe B.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Transaction Commands	Commandes de transactions
Transaction Results	Résultats de transactions
Host Controller	Contrôleur hôte
Host state machines	Diagrammes d'états d'hôte
Downstream High-speed Bus	Bus à grande vitesse en aval
Upstream High-speed Bus	Bus à grande vitesse en amont
TT state machines	Diagrammes d'états de TT
Hub Transaction Translator	Traducteur de transactions de hub
Bulk/Ctrl Buffers	Tampons en masse/de contrôle
Periodic Pipeline Buffers	Tampons de pipeline périodique

Figure 11-31. Présentation du contexte du diagramme d'états

11.16 Diagrammes d'états communs aux transactions fractionnées

Il y a plusieurs diagrammes d'états communs à tous les types de transaction fractionnée spécifiques. Ces diagrammes d'états sont utilisés dans le contrôleur hôte et le traducteur de transactions pour déterminer le type de transaction fractionnée spécifique (ex.: start-split OUT d'interruption contre complete-split IN en masse). Un aperçu de la hiérarchie du diagramme d'états du contrôleur hôte est illustré à la Figure 11-32. L'aperçu de la hiérarchie du diagramme d'états du traducteur de transactions est illustré à la Figure 11-33. Chacun des encadrés étiquetés dans les figures montre un diagramme d'états individuel. Les encadrés dans un autre encadré indiquent un diagramme d'états dans un autre diagramme d'états. Tous les diagrammes d'états à l'exception de ceux de niveau inférieur sont illustrés dans les autres figures dans cette section. Les diagrammes d'états de niveau inférieur illustrés dans les sections suivantes décrivent le type de transaction fractionnée spécifique.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

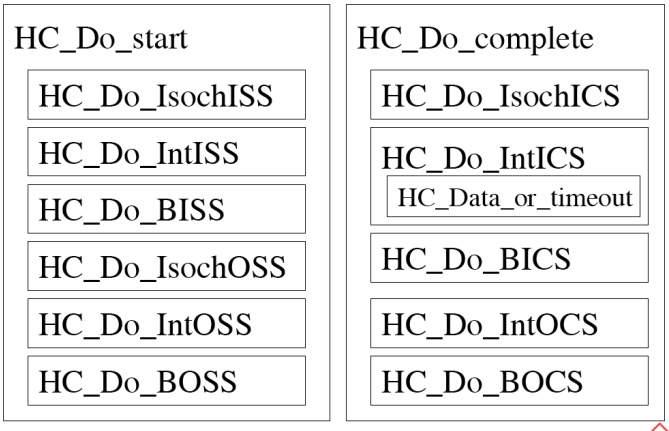


Figure 11-32. Présentation de la hiérarchie du diagramme d'états de transaction fractionnée du contrôleur hôte

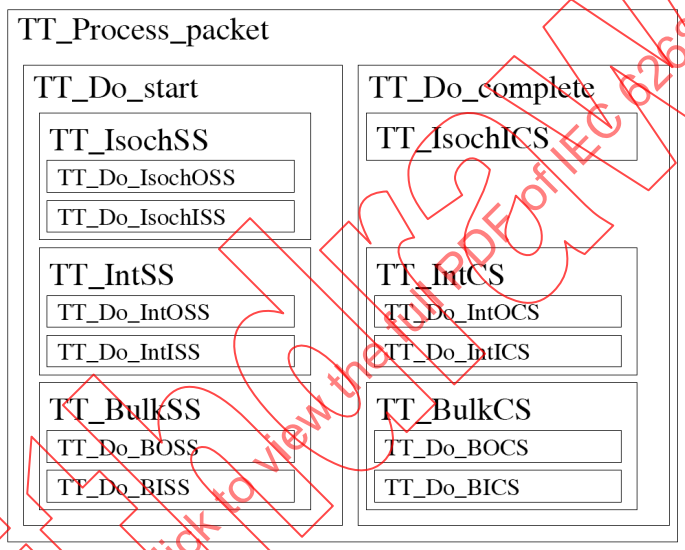
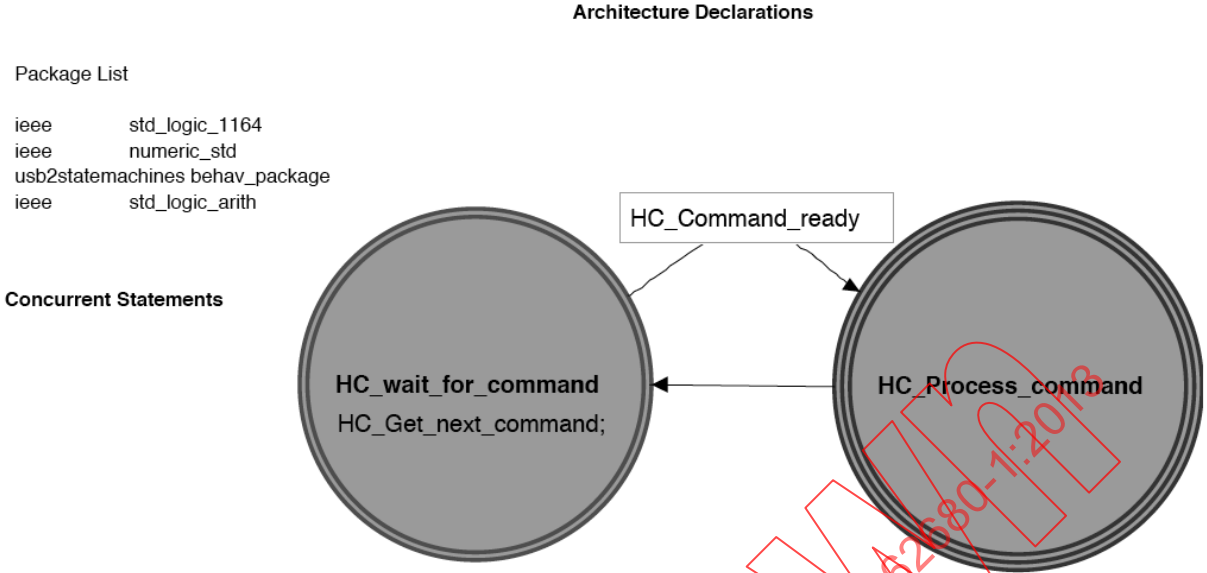


Figure 11-33. Présentation de la hiérarchie du diagramme d'états du traducteur de transactions

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.1 Diagramme d'états du contrôleur hôte



Légende

Anglais	Français
Architecture Declarations	Déclarations d'architecture
Package List	Liste de packages
Concurrent Statements	Instructions concurrentes

Figure 11-34. Contrôleur hôte

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.1.1 Diagramme d'états HC_Process_command

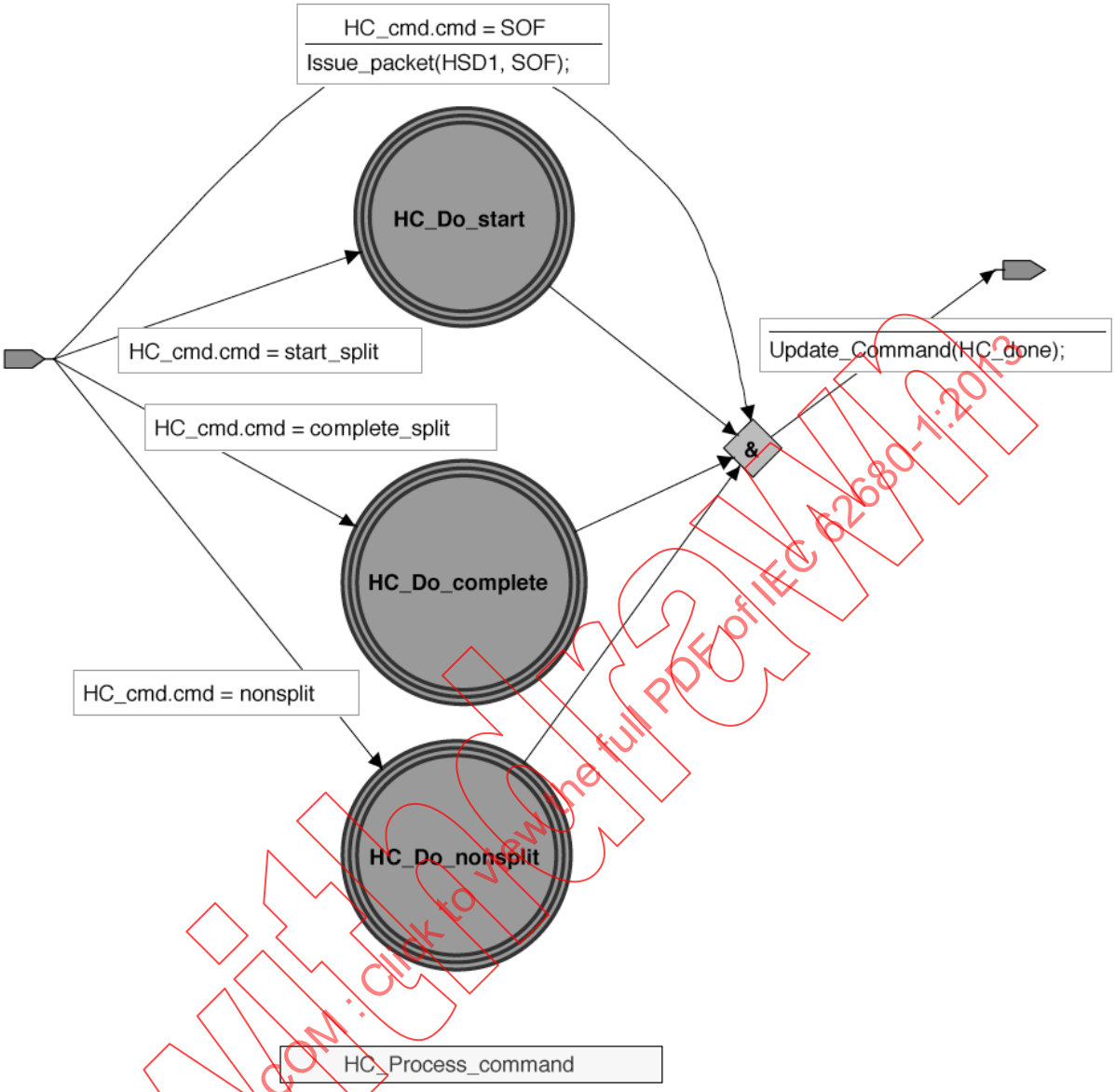
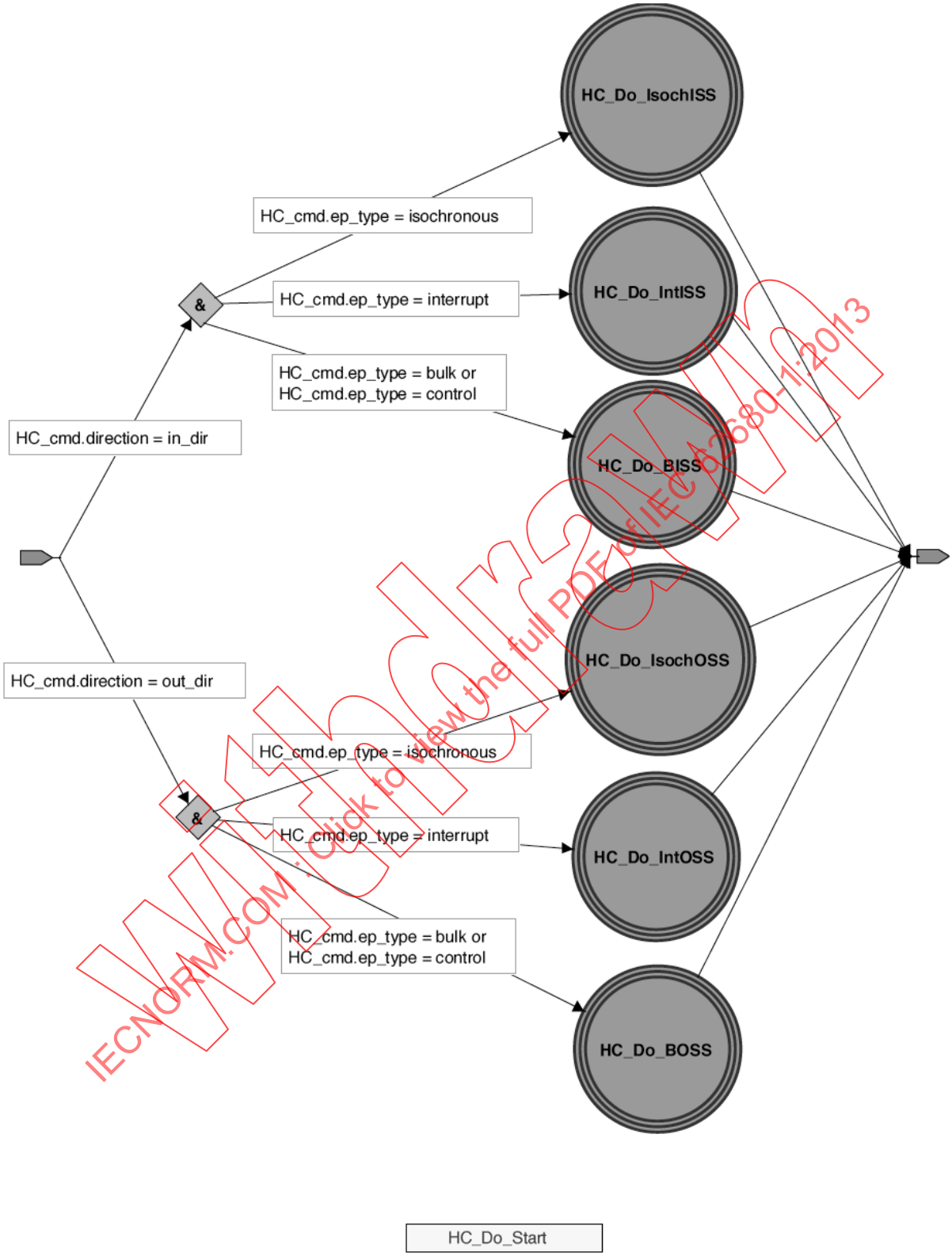


Figure 11-35. HC_Process_Command

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.1.1.1 Diagramme d'états HC_Do_start



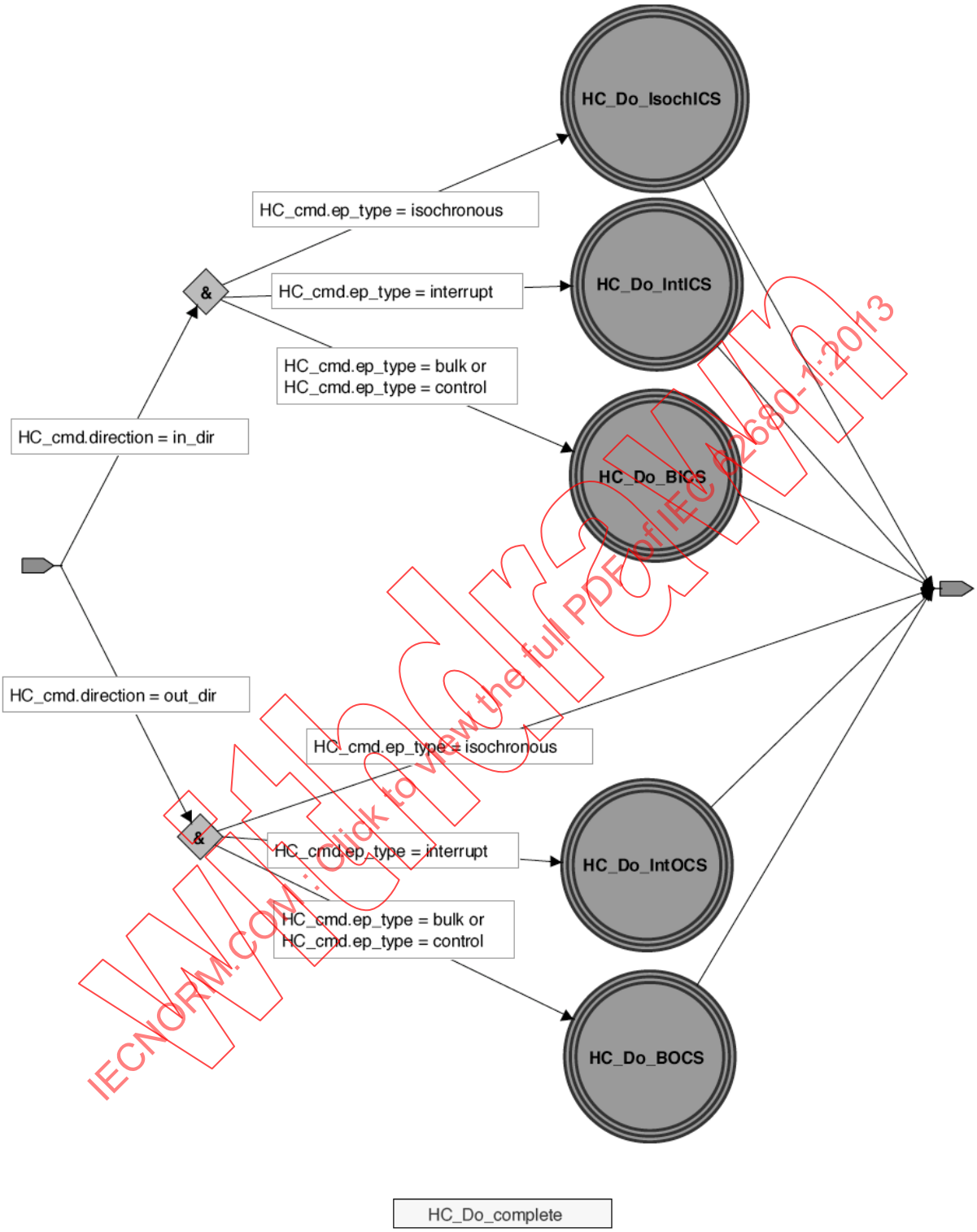
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
HC_cmd.ep_type = isochronous	HC_cmd.ep_type = isochrones
HC_cmd.ep_type = interrupt	HC_cmd.ep_type = interruption
HC_cmd.ep_type = bulk or HC_cmd.ep_type = control	HC_cmd.ep_type = masse ou HC_cmd.ep_type = contrôle

Figure 11-36. HC_Do_Start

11.16.1.1.2 Diagramme d'états HC_Do_complete



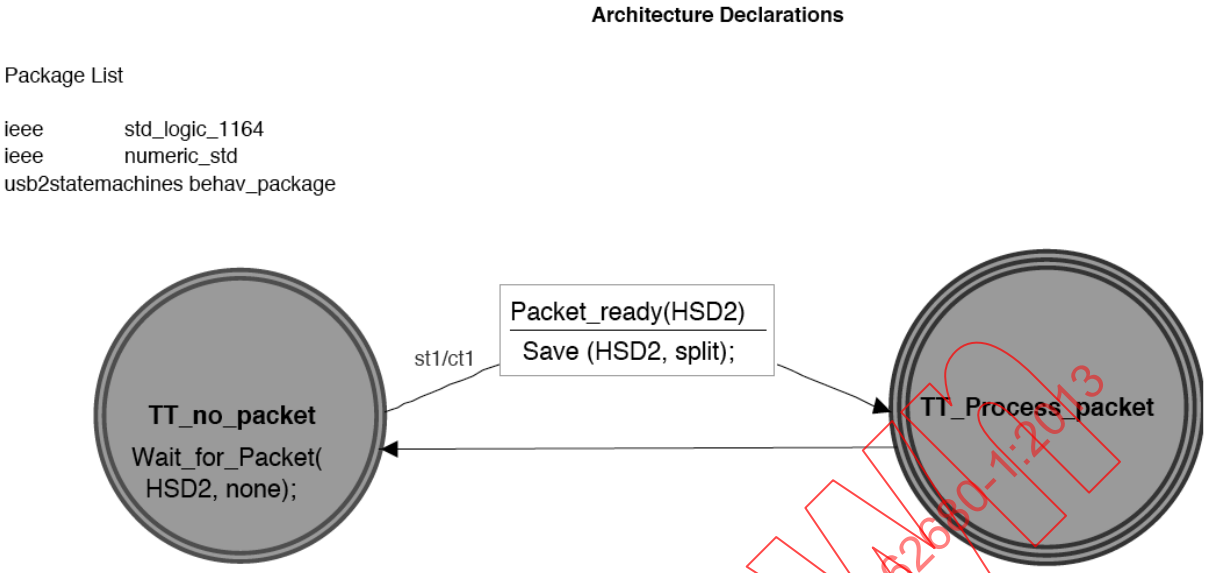
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
HC_cmd.ep_type = isochronous	HC_cmd.ep_type = isochrones
HC_cmd.ep_type = interrupt	HC_cmd.ep_type = interruption
HC_cmd.ep_type = bulk or HC_cmd.ep_type = control	HC_cmd.ep_type = masse ou HC_cmd.ep_type = contrôle

Figure 11-37. HC_Do_Complete

11.16.2 Diagramme d'états du traducteur de transactions



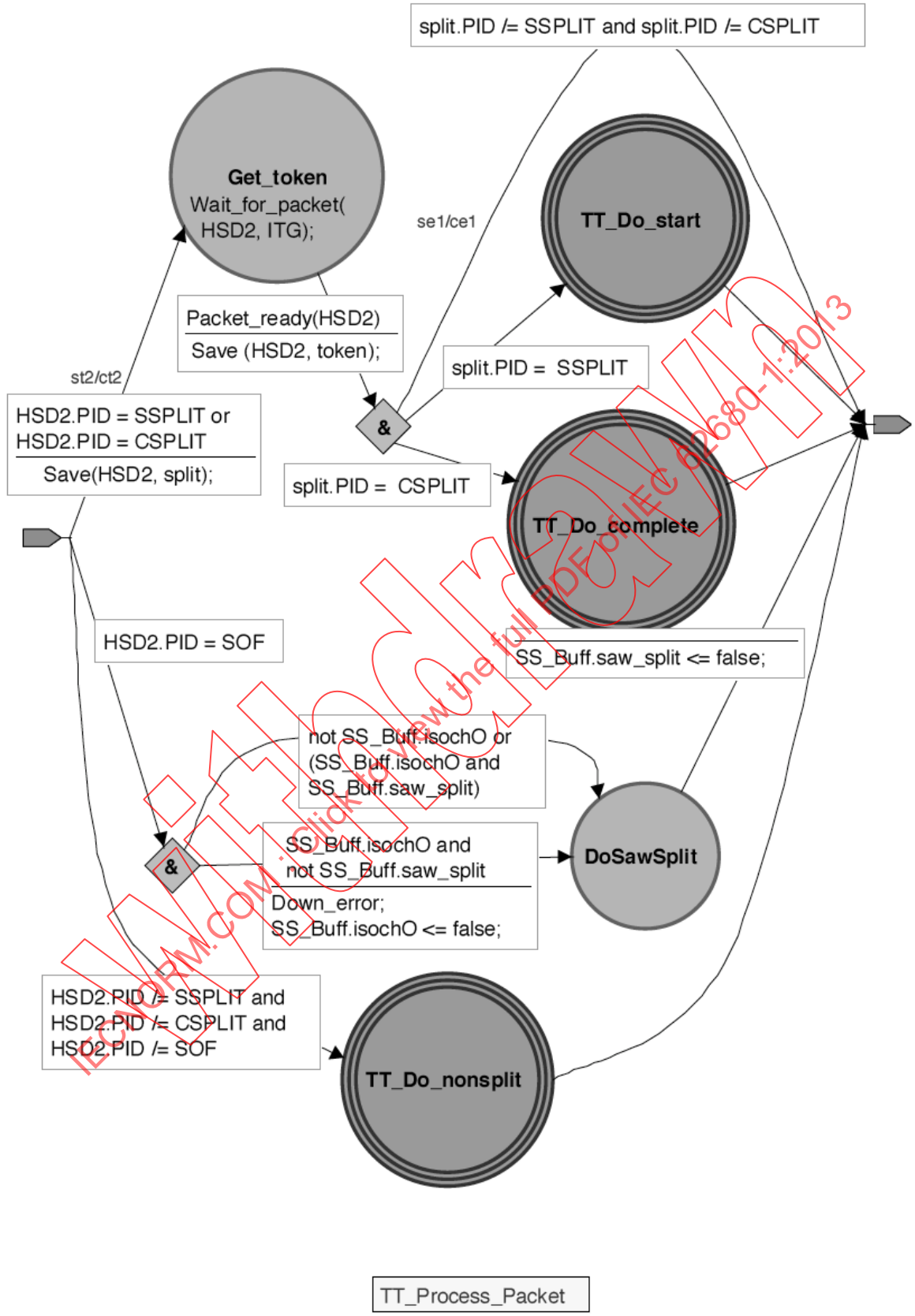
Légende

Anglais	Français
Architecture Declarations	Déclarations d'architecture
Package List	Liste de packages

Figure 11-38. Traducteur de transactions

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.2.1 Diagramme d'états TT_Process_packet



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

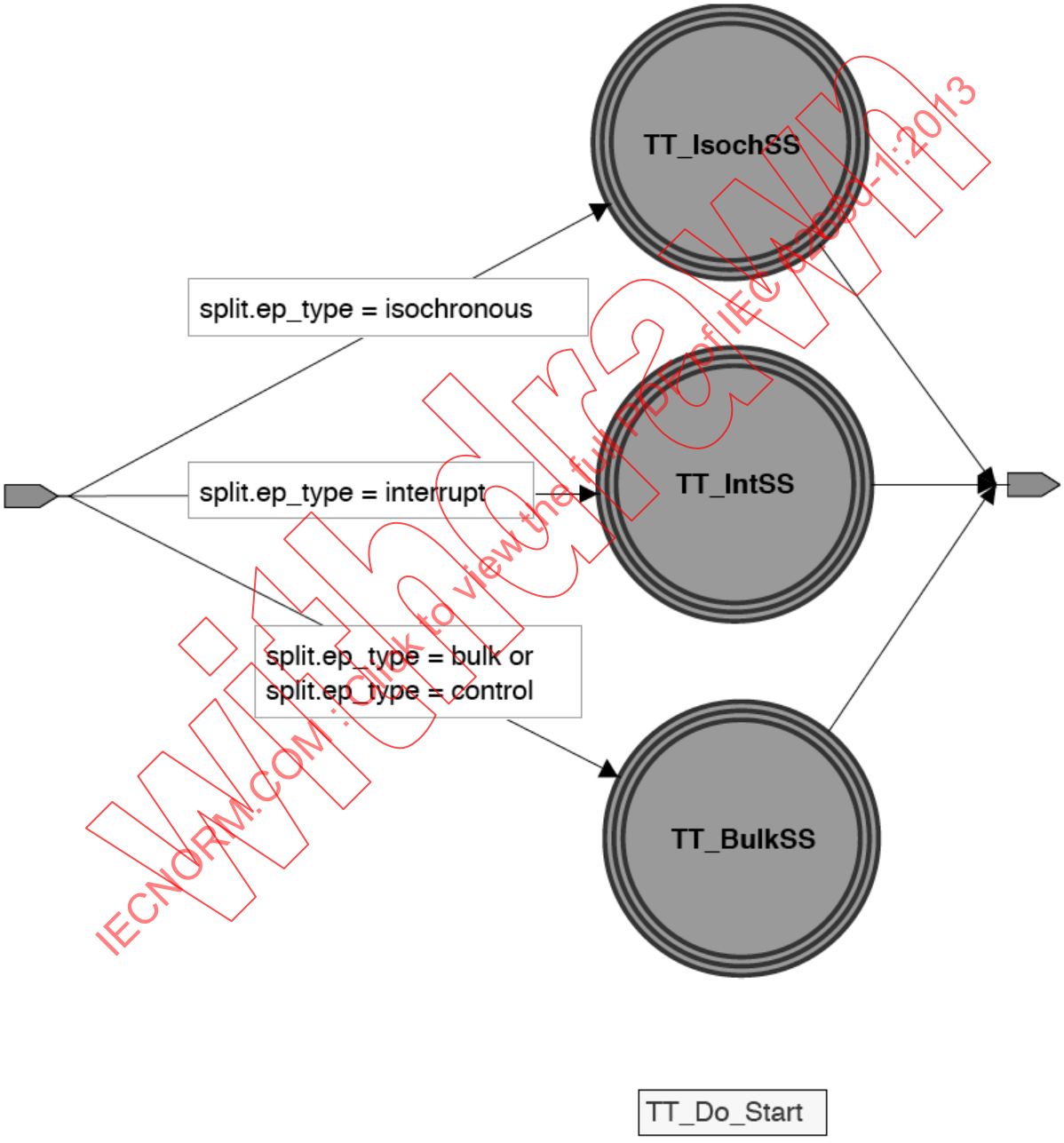
Légende

Anglais	Français
split.PID/= SSPLIT and split.PID /= CSPLIT	split.PID/= SSPLIT et split.PID /= CSPLIT
Save (HSD2, token)	Enregistrer (HSD2, jeton)

Save (HSD2, split)	Enregistrer (HSD2, split)
not SS_Buff.isochO or (SS_Buff.isochO and SS_Buff.saw_split)	not SS_Buff.isochO ou (SS_Buff.isochO et SS_Buff.saw_split)
SS_Buff.isochO and not SS_Buff.saw_split	SS_Buff.isochO et non SS_Buff.saw_split
SS_Buff.isoO <= false;	SS_Buff.isoO <= faux;
HSD2.PID /= SSPLIT and HSD2.PID /= CSPLIT and HSD2.PID /= SOF	HSD2.PID /= SSPLIT et HSD2.PID /= CSPLIT et HSD2.PID /= SOF

Figure 11-39. TT_Process_Packet

11.16.2.1.1 Diagramme d'états TT_Do_Start



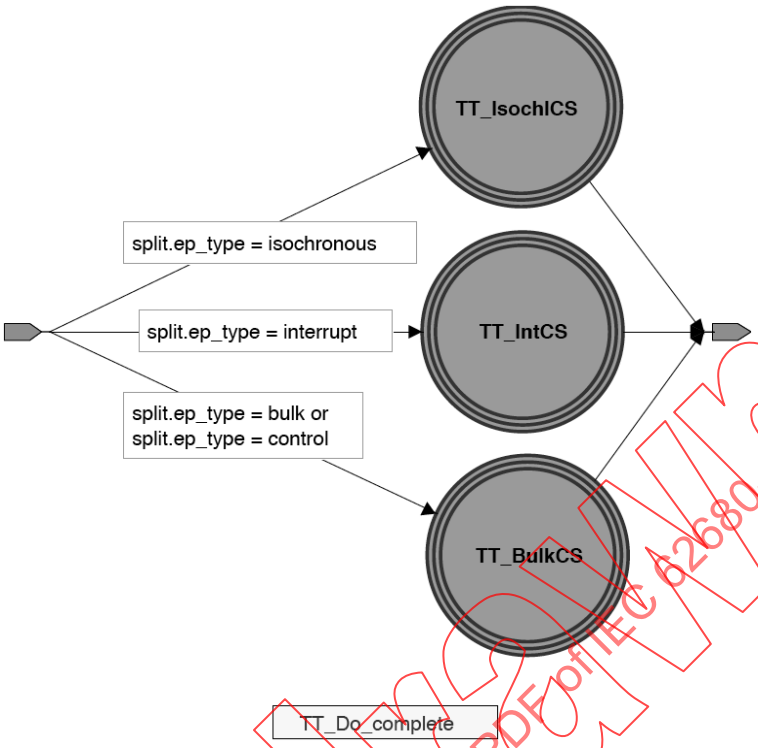
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
split.ep_type = isochronous	split.ep_type = isochrones
split.ep_type = interrupt	split.ep_type = interruption
split.ep_type = bulk or split.ep_type = control	split.ep_type = masse ou split.ep_type = contrôle

Figure 11-40. TT_Do_Start

11.16.2.1.2 Diagramme d'états TT_Do_Complete



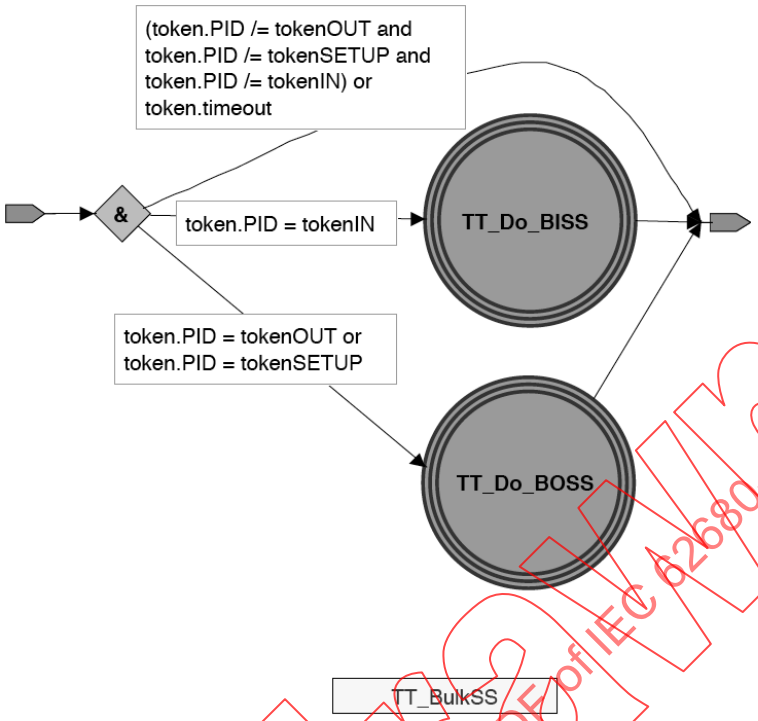
Légende

Anglais	Français
split.ep_type = isochronous	split.ep_type = isochrones
split.ep_type = interrupt	split.ep_type = interruption
split.ep_type = bulk or split.ep_type = control	split.ep_type = masse ou split.ep_type = contrôle

Figure 11-41. TT_Do_Complete

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.16.2.1.3 Diagramme d'états TT_BulkSS



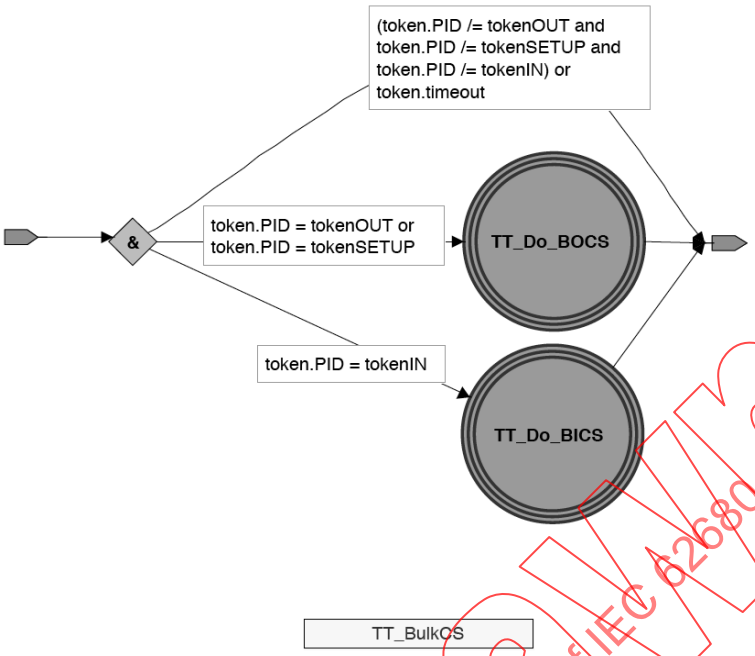
Légende

Anglais	Français
(token.PID /= tokenOUT and token.PID /= tokenSETUP and token.PID /= tokenIN) or token.timeout	(token.PID /= tokenOUT et token.PID /= tokenSETUP et token.PID /= tokenIN) ou token.timeout
token.PID /= tokenIN	token.PID /= tokenIN
token.PID /= token OUT or token.PID /= tokenSETUP	token.PID /= token OUT ou token.PID /= tokenSETUP

Figure 11-42. TT_BulkSS

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.16.2.1.4 Diagramme d'états TT_BulkCS



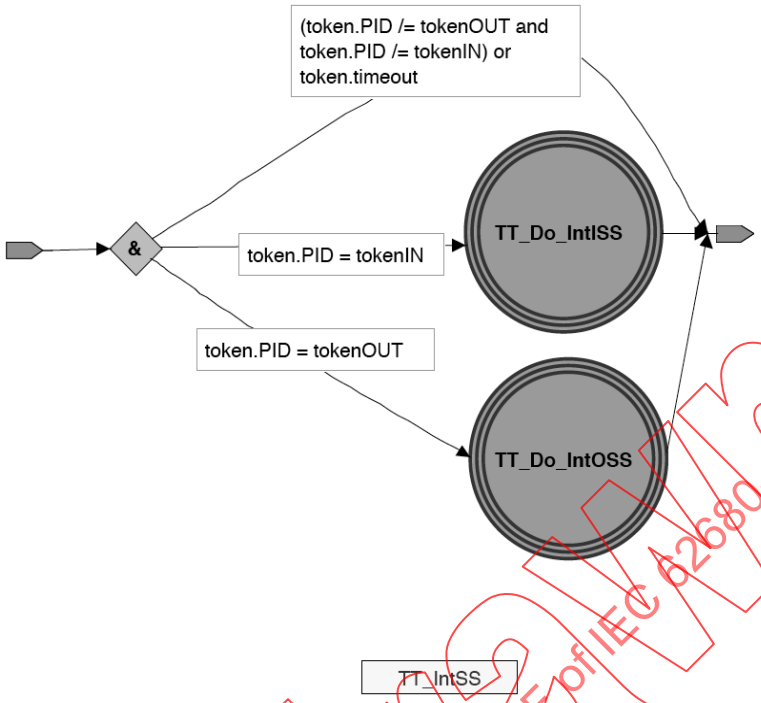
Légende

Anglais	Français
(token.PID /= tokenOUT and token.PID /= tokenSETUP and token.PID /= tokenIN) or token.timeout	(token.PID /= tokenOUT et token.PID /= tokenSETUP et token.PID /= tokenIN) ou token.timeout
token.PID /= tokenOUT or token.PID /= tokenSETUP	token.PID /= token OUT ou token.PID /= tokenSETUP

Figure 11-43. TT_BulkCS

IECNORM.COM : Click to view the PDF of IEC 62680-1:2013
Withdrawn

11.16.2.1.5 Diagramme d'états TT_IntSS



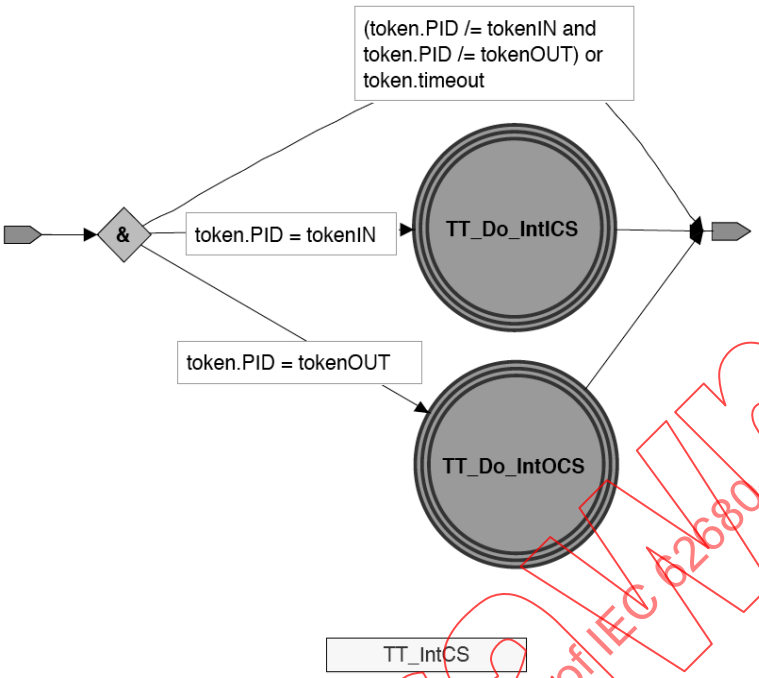
Légende

Anglais	Français
(token.PID /= tokenOUT and token.PID /= tokenIN) or token.timeout	(token.PID /= tokenOUT et token.PID /= tokenIN) ou token.timeout
token.PID /= tokenOUT	token.PID /= tokenOUT

Figure 11-44. TT_IntSS

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.2.1.6 Diagramme d'états TT_IntCS



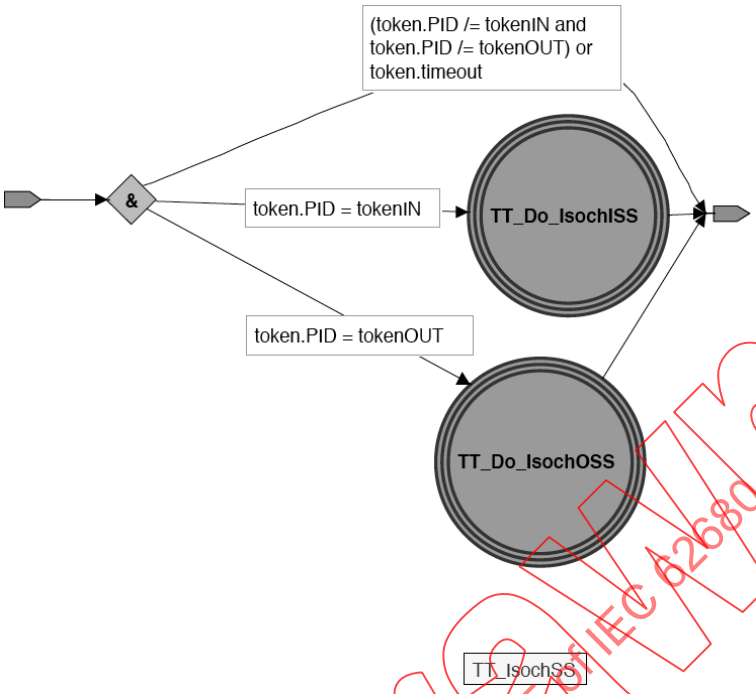
Légende

Anglais	Français
(token.PID /= token IN and token.PID /= tokenOUT) or token.timeout	(token.PID /= tokenIN et token.PID /= tokenOUT) ou token.timeout

Figure 11-45. TT_IntCS

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.16.2.1.7 Diagramme d'états TT_IsochSS



Légende

Anglais	Français
(token.PID /= token IN and token.PID /= tokenOUT) or token.timeout	(token.PID /= tokenIN et token.PID /= tokenOUT) ou token.timeout

Figure 11-46. TT_IsochSS

11.17 Présentation de la traduction de transactions en masse/de contrôle

Chaque TT doit avoir au moins deux tampons de transactions en masse/de contrôle. Chaque tampon conserve les informations pour une transaction start- ou complete-split et représente une seule transaction à vitesse intégrale/basse vitesse qui est en attente (ou a terminé le) de transfert sur le bus en aval. Le tampon est utilisé pour contenir les informations de transaction du start-split (et les données pour OUT) et l'établissement de la liaison/le résultat de la transaction à vitesse intégrale/basse vitesse (et les données pour IN). Ce tampon est rempli et vidé par les transactions fractionnées du bus à grande vitesse via le gestionnaire à grande vitesse. Le tampon est également mis à jour par le gestionnaire à vitesse intégrale/basse vitesse tandis que la transaction est en cours sur le bus en aval.

Le gestionnaire à grande vitesse doit accepter une transaction start-split du contrôleur hôte pour un point d'extrémité en masse/de contrôle lorsque le gestionnaire à grande vitesse a un espace approprié dans un tampon en masse/de contrôle.

Le contrôleur hôte essaie une transaction start-split selon son programme de transaction à grande vitesse en masse/de contrôle. Dès que le gestionnaire à grande vitesse répond à une transaction complete-split avec les résultats du tampon correspondant, le prochain start-split pour certains points d'extrémité à vitesse intégrale/basse vitesse (éventuellement d'autres) peut être enregistré dans le tampon.

Il n'y a pas de méthode pour contrôler la transaction start-split acceptée ensuite par le gestionnaire à grande vitesse. Le séquençage des transactions start-split est simplement déterminé par l'espace tampon du TT disponible et l'état actuel du programme du contrôle hôte (ex.: quelle transaction start-split est la suivante que le contrôleur hôte essaie comme une partie normale du traitement des transactions à grande vitesse).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Il n'est pas nécessaire pour le e contrôleur hôte de séparer les transactions fractionnées en masse (ou de contrôle) des transactions en masse (de contrôle) à grande vitesse lors de l'établissement de son programme. Le contrôleur hôte doit suivre si une transaction est une transaction à grande vitesse normale ou une transaction fractionnée à grande vitesse.

Les sections suivantes décrivent les détails des phases de transaction, des séquences de flux et des diagrammes d'états pour les transactions fractionnées utilisées pour prendre en charge les transactions OUT et IN en masse et de contrôle à vitesse intégrale/basse vitesse. Il existe seulement des différences mineures entre les transactions fractionnées en masse et de contrôle. Dans les figures, certaines zones sont ombrées pour indiquer qu'elles ne s'appliquent pas pour les transactions de contrôle.

11.17.1 Séquences de transactions fractionnées en masse/de contrôle

Les figures des diagrammes d'états montrent les transitions requises pour les transactions fractionnées à grande vitesse pour les types de transfert en masse/de contrôle à vitesse intégrale/basse vitesse pour un seul point d'extrémité. Ces figures ne doivent pas être interprétées comme illustrant une synchronisation spécifique particulière. Elles définissent le comportement de séquençage requis des différents paquets d'une transaction fractionnée en masse/de contrôle. En particulier, d'autres transactions à grande vitesse ou fractionnées pour les autres points d'extrémité se produisent avant ou après ces séquences de transactions fractionnées.

La Figure 11-47 montre un exemple d'algorithme de code qui décrit le comportement des transitions étiquetées avec *Is_new_SS*, *Is_old_SS* et *Is_no_space* illustrées dans les figures pour les transactions start-split IN et OUT en masse/de contrôle mises en tampon dans le TT pour n'importe quel point d'extrémité. Cet algorithme garantit que le TT met uniquement en tampon une seule transaction fractionnée en masse/de contrôle pour n'importe quel point d'extrémité. La définition du protocole complete-split exige qu'un point d'extrémité ait un seul résultat mis en tampon dans le TT à tout moment. Il faut noter que l'essai "correspondance des tampons" est différent pour les points d'extrémité en masse et de contrôle. Une correspondance des tampons pour une transaction en masse doit inclure la direction de la transaction dans l'essai, car les points d'extrémité en masse sont unidirectionnels. Une transaction de contrôle ne doit pas utiliser la direction dans le cadre de l'essai de correspondance.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

```
procedure Compare_buffs IS
      variable match:boolean:=FALSE;
begin
--
--  Is_new_SS is true when BC_buff.status == NEW_SS
--  Is_old_SS is true when BC_buff.status == OLD_SS
--  Is_no_space is true when BC_buff.status == NO_SPACE
--
       -- Assume nospace and intialize index to 0.
       BC_buff.status := NO_SPACE;
       BC_buff.index  := 0;

       FOR i IN 0 to num_buffs-1 LOOP
           IF NOT match THEN
               -- Re-use buffer with same Device Address/End point.
               IF (token.endpt = cam(i).store.endpt AND
                   token.dev_addr = cam(i).store.dev_addr AND
                   ((token.direction = cam(i).store.direction AND
                     split.ep_type /= CONTROL) OR
                    split.ep_type = CONTROL)) THEN

                 -- If The buffer is already pending/ready this must be a retry.
                 IF (cam(i).match.state = READY OR cam(i).match.state = PENDING) THEN
                     BC_buff.status := OLD_SS;
                 ELSE
                     BC_buff.status := NEW_SS;
                 END IF;
                 BC_buff.index := i;
                 match := TRUE;

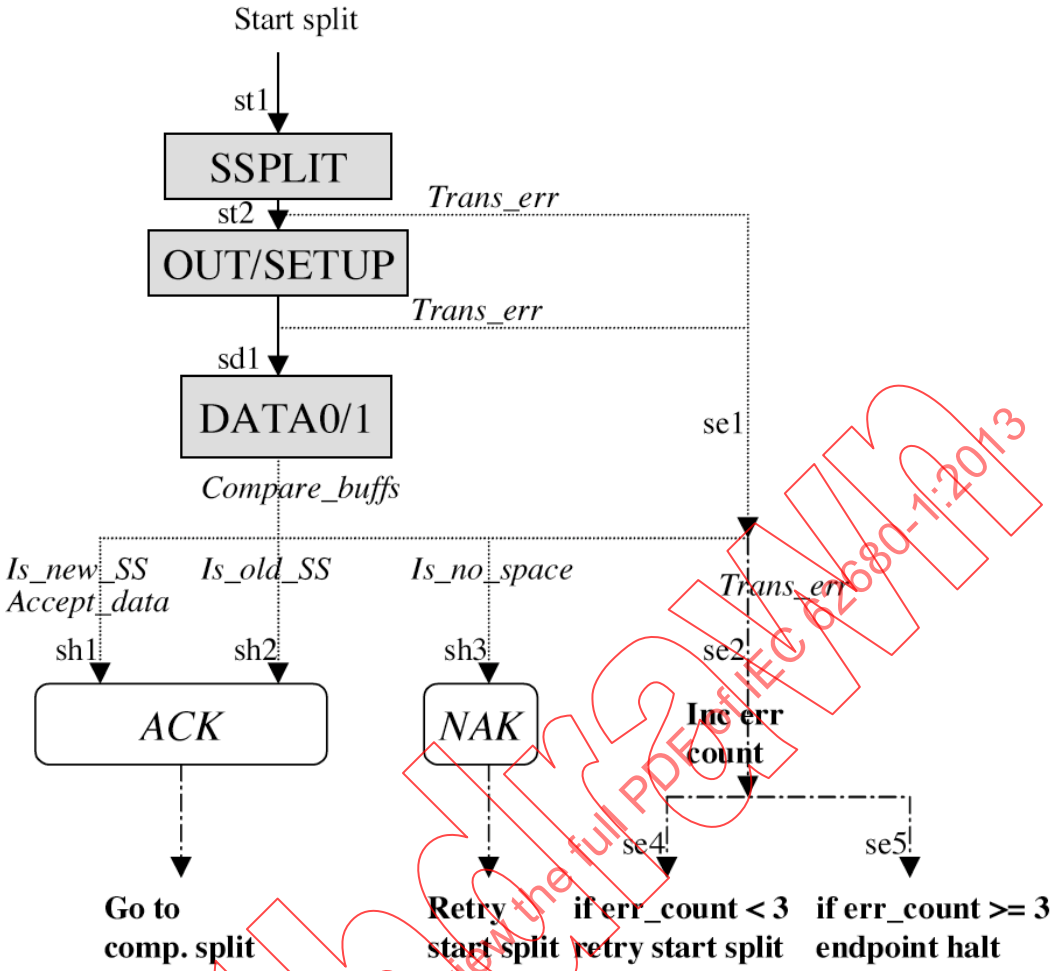
               -- Otherwise use the buffer if it's old.
               ELSIF (cam(i).match.state = OLD) THEN
                 BC_buff.status := NEW_SS;
                 BC_buff.index := i;
               END IF;
           END IF;
       END LOOP;

    end Compare_buffs;
```

Figure 11-47. Exemple d'algorithme pour Compare_buffs

La Figure 11-48 montre la séquence de paquets pour une transaction start-split pour le type de transfert OUT en masse à vitesse intégrale/basse vitesse. Le bloc étiqueté SSPLIT représente un paquet de jeton de transaction fractionnée comme décrit au Chapitre 8. Il est suivi d'un paquet de jeton OUT (ou paquet de jeton SETUP pour une transaction de configuration de contrôle). Si le gestionnaire à grande vitesse expire après les paquets de jetons SSPLIT ou OUT, et ne reçoit pas les paquets OUT/SETUP ou DATA0/1 suivants, il ne répond pas avec un établissement de liaison comme indiqué par les transitions en pointillés étiquetées "se1" ou "se2". L'hôte voit alors une erreur de transaction (temporisation) (étiquetée "se2" et indiquée avec une ligne de tirets). Si le gestionnaire à grande vitesse reçoit le paquet DATA0/1 et celui-ci échoue au contrôle CRC, il prend la transition "se2" qui pousse l'hôte à expirer et à suivre la transition "se2".

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Inc err count	Inc décompte err
Go to comp. split	Aller à comp. split
Retry start split	Réessayer start split
if err_count < 3 retry start split	si err_count < 3 réessayer start split
if err_count >= 3 endpoint halt	si err_count >= 3 arrêt point d'extrémité

Figure 11-48. Séquence de transaction start-split OUT en masse/de contrôle

L'hôte doit réessayer le start-split pour ce point d'extrémité jusqu'à ce que le err_count atteigne trois pour ce point d'extrémité avant de continuer sur un autre start-split pour ce point d'extrémité. Mais, l'hôte peut émettre d'autres start-splits pour d'autres points d'extrémité avant de réessayer le start-split pour ce point d'extrémité. Le err_count est utilisé pour compter combien d'erreurs ont été rencontrées pendant les tentatives d'émission d'une transaction particulière pour un point d'extrémité particulier.

S'il n'y a pas d'espace dans les tampons de transaction pour contenir le start-split, le gestionnaire à grande vitesse répond avec un NAK via la transition "sh3". Cela pousse l'hôte à réessayer ce start-split ultérieurement en fonction de son programme normal. L'hôte n'augmente pas son err_count pour une réponse d'établissement de liaison NAK. Lorsque l'hôte a reçu une réponse NAK à un start-split, il peut ignorer les autres start-splits pour ce TT pour les points d'extrémité en masse/de contrôle jusqu'à ce qu'il termine un complete-split en masse/de contrôle.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

S'il y a un espace tampon pour le start-split, le gestionnaire à grande vitesse prend la transition "sh1" et répond avec un ACK. Cela indique à l'hôte qu'il doit essayer un complete-split la prochaine fois qu'il essaie de traiter une transaction pour ce point d'extrémité à vitesse intégrale/basse vitesse. Après avoir reçu un établissement de liaison ACK, l'hôte ne doit pas émettre un autre start-split pour ce point d'extrémité jusqu'à ce que le complete-split correspondant ait été terminé.

Si le gestionnaire à grande vitesse a déjà un start-split pour ce point d'extrémité à vitesse intégrale/basse vitesse en cours ou prêt, il suit la transition "sh2" et répond également à un ACK, mais ignore les données. Celui-ci traite le cas où un établissement de liaison ACK a été tiré et raté par le contrôleur hôte et le contrôleur hôte réessaie maintenant le start-split; ex.: transition d'un gestionnaire à grande vitesse de "sh1" mais transition de l'hôte de "se2".

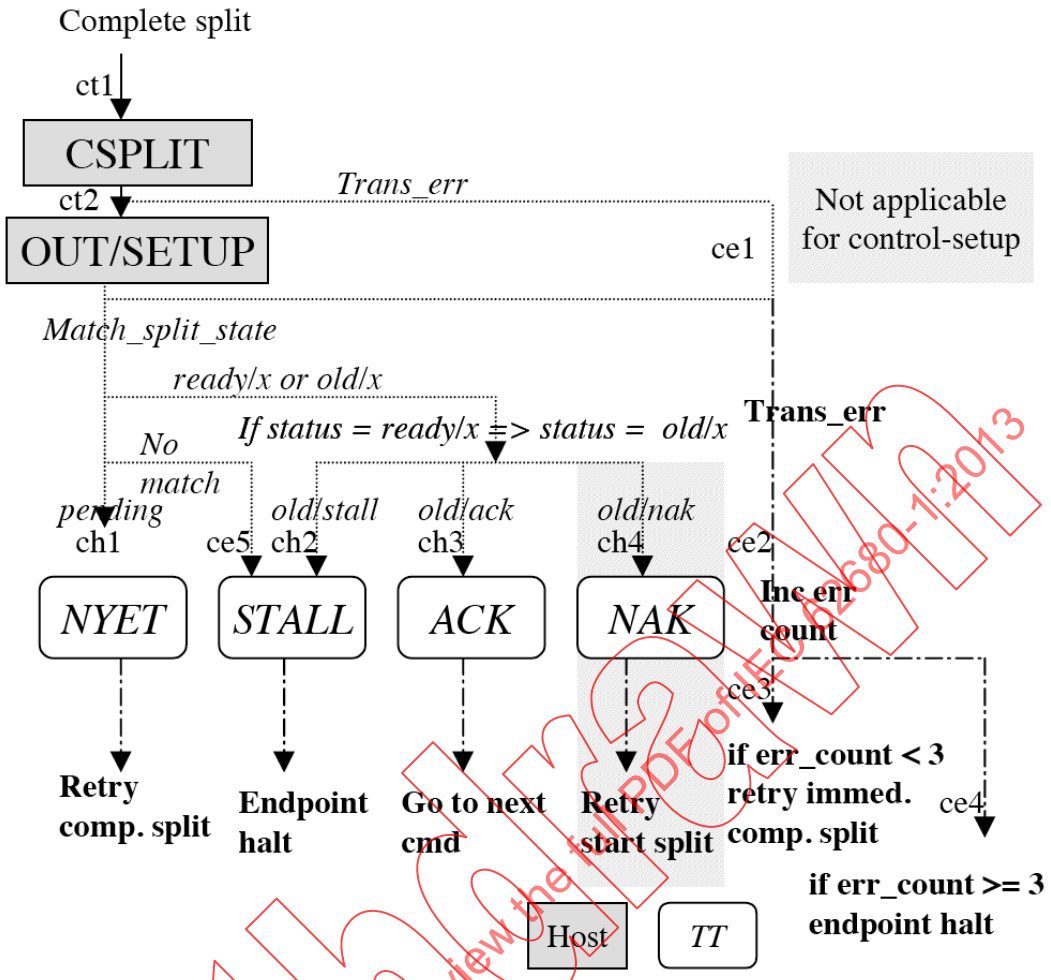
Dans les cas d'erreur du contrôleur hôte, le contrôleur hôte applique le mécanisme "des trois essais et vous êtes éliminé". Il incrémente ainsi un décompte des erreurs (err_count) et, si le décompte est inférieur à trois (transition "se4"), il réessaie la transaction. Si le err_count est supérieur ou égal à trois (transition "se5"), le contrôleur hôte effectue un traitement d'arrêt du point d'extrémité et ne réessaie pas la transaction. Si pour une raison ou une autre, une mémoire d'hôte ou un retard de bus non-USB (ex.: une mémoire système "suspendue") se produit et fait en sorte que la transaction n'est pas terminée normalement, le err_count ne doit pas être incrémenté. Lorsqu'une transaction se termine normalement, le err_count est remis à zéro.

Le gestionnaire à grande vitesse dans le TT n'a pas de connaissance immédiate de ce que l'hôte voit, les transitions "se2", "se4" et "se5" affichent uniquement la visibilité de l'hôte.

Cette séquence de flux de paquets illustrant les interactions entre l'hôte et le hub est également représentée par les diagrammes d'états du gestionnaire à grande vitesse dans la section suivante. Ces diagrammes d'états utilisent les mêmes libellés pour corréler les transitions entre les deux représentations des règles de transactions fractionnées.

La Figure 11-49 illustre la séquence de flux correspondante pour la transaction complete-split pour le type de transfert OUT en masse/de contrôle à vitesse intégrale/basse vitesse. La notation "prêt/x" ou "ancien/x" indique que le statut de la transaction fractionnée est dans un des états prêt ou ancien. Après avoir exécuté une transaction à vitesse intégrale/basse vitesse sur le bus en aval, le statut de la transaction est mis à jour pour refléter le résultat de la transaction. Le statut de résultat possible est: nak, stall, ack. Le "x" désigne un des résultats de statut de transaction à vitesse intégrale/basse vitesse NAK, ACK, STALL. Chaque résultat de statut reflète la réponse de l'établissement de liaison de la transaction à vitesse intégrale/basse vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



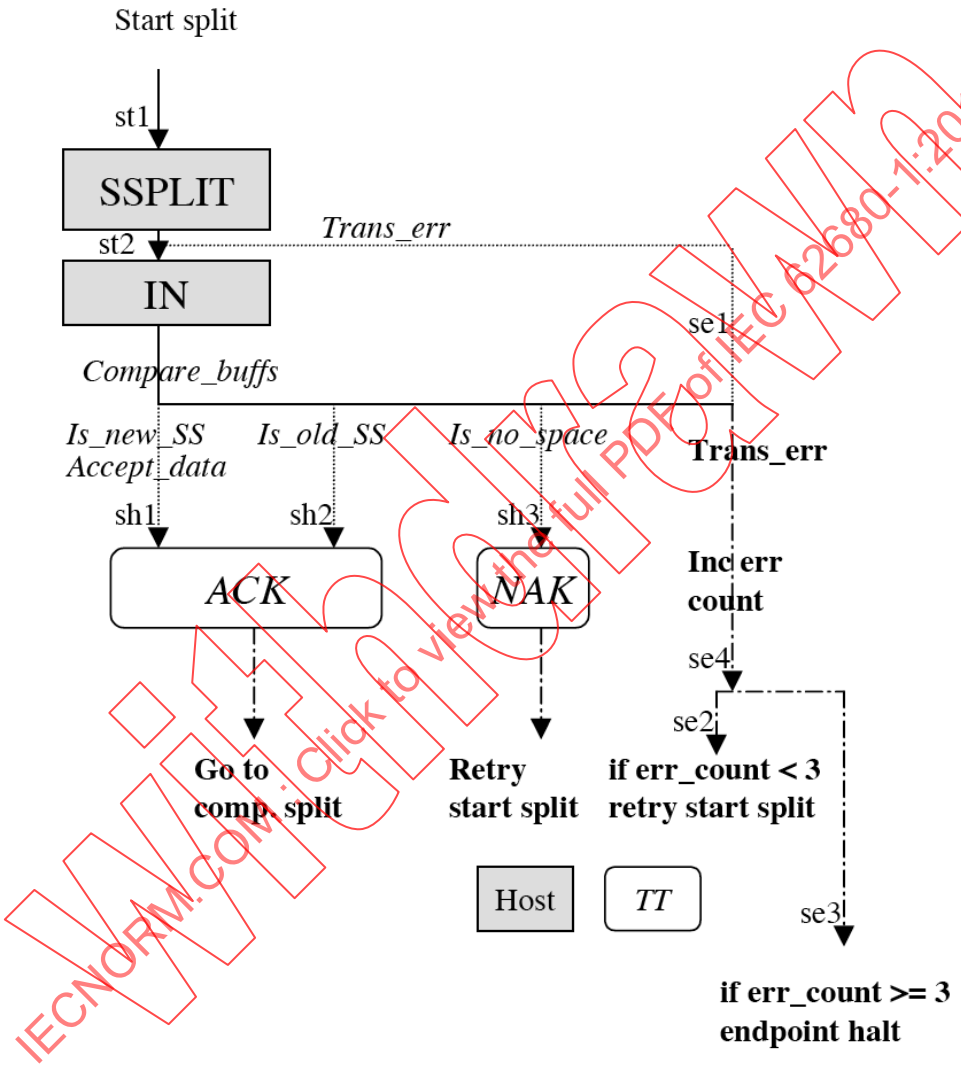
Légende

Anglais	Français
Complete split	Complete split
Not applicable for control-setup	Non applicable pour configuration de commande
No match	Pas de correspondance
ready/x or old/x	prêt/x ou ancien/x
If status = ready/x = > status = old/x	Si état = prêt/x = > état = ancien/x
pending	en cours
old/stall	ancien/arrêt
old/ack	ancien/ack
old/nak	ancien/nak
Inc err count	Inc décompte err
Retry comp. split	Réessayer comp. split
Endpoint halt	Arrêt point d'extrémité
Go to next cmd	Aller à cmd suivante
Retry start split	Réessayer start split
if err_count < 3 retry immed. comp. split	si err_count < 3 réessayer imméd. comp. split
if err_count >= 3 endpoint halt	si err_count >= 3 arrêt point d'extrémité
Host	Hôte

Figure 11-49. Séquence de transaction complete-split OUT en masse/de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Il n'y a pas de statut de réponse de temporisation pour une transaction, car le gestionnaire à vitesse intégrale/basse vitesse doit effectuer une nouvelle tentative locale d'une transaction en masse ou de contrôle à vitesse intégrale/basse vitesse qui rencontre une erreur de transaction. Il applique localement le mécanisme "des trois essais et vous êtes éliminé". Cela signifie que la transaction à vitesse intégrale/basse vitesse résout un des résultats d'établissement de liaison NAK, STALL ou ACK. Si la transaction rencontre une erreur de transaction trois fois, le gestionnaire à vitesse intégrale/basse vitesse le reflète comme un résultat de statut d'arrêt. Le gestionnaire à vitesse intégrale/basse vitesse ne doit pas effectuer une nouvelle tentative locale de la transaction en réponse à un établissement de liaison ACK, NAK ou STALL.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Go to comp. split	Aller à comp. split
Retry start split	Réessayer start split
if err_count < 3 retry start split	si err_count < 3 réessayer start split
if err_count >= 3 endpoint halt	si err_count >= 3 arrêt point d'extrémité
Inc err count	Inc décompte err
Start split	Start split
Host	Hôte

Figure 11-50. Séquence de transaction start-split IN en masse/de contrôle

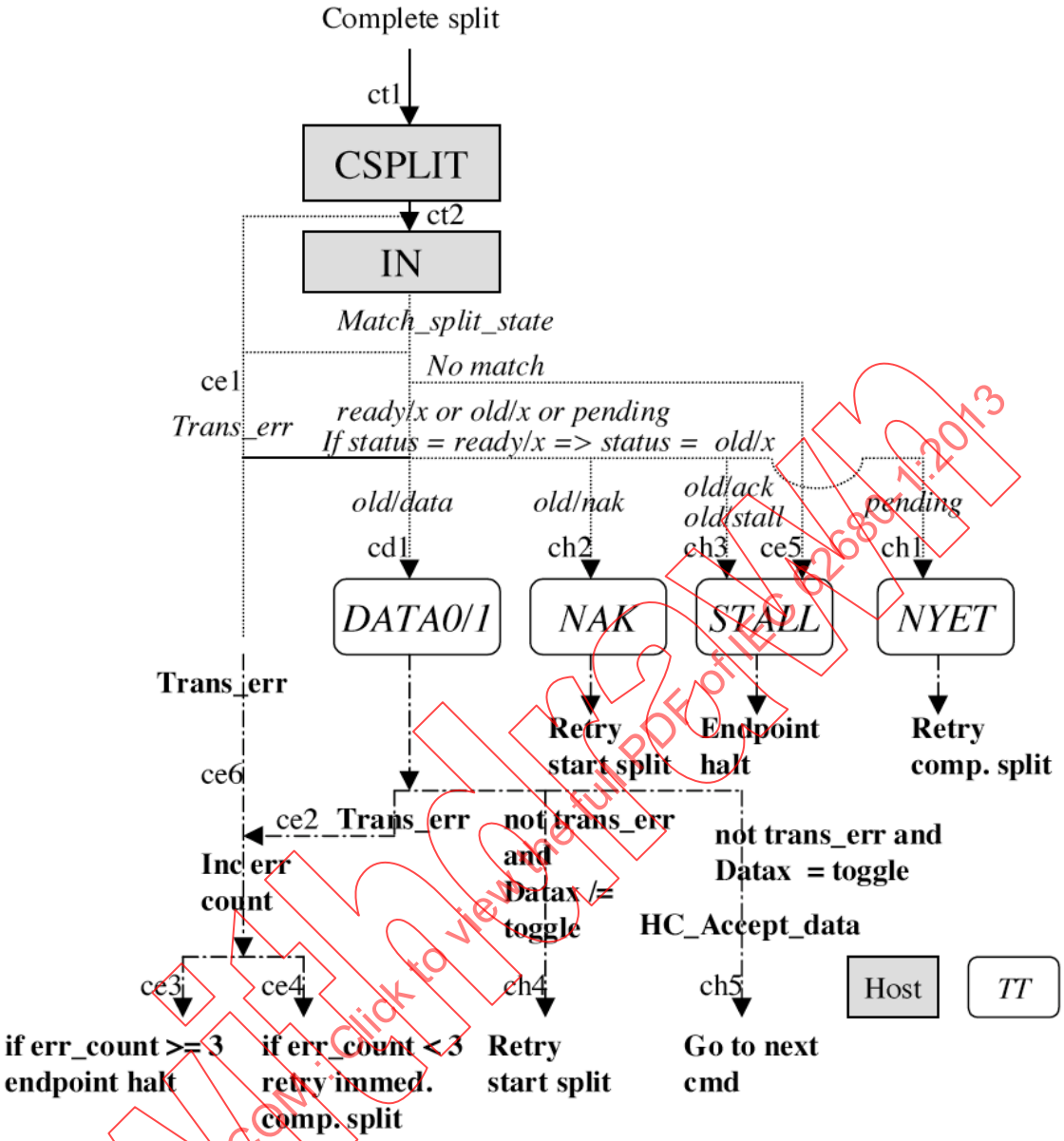
Si le gestionnaire à grande vitesse reçoit le paquet de jeton complete-split (et le paquet de jeton) alors que la transaction à vitesse intégrale/basse vitesse n'est pas terminée (ex.: le statut de transaction est "en cours"), le gestionnaire à grande vitesse répond avec un établissement de liaison NYET. L'hôte réessaie alors le complete-split pour ce point d'extrémité un peu plus tard.

Si le gestionnaire à grande vitesse reçoit un paquet de jeton complete-split (et le paquet de jeton) et ne trouve pas de tampon local avec une transaction correspondante, le TT répond avec un STALL pour indiquer une violation de protocole.

Lorsque le gestionnaire à vitesse intégrale/basse vitesse a terminé une transaction à vitesse intégrale/basse vitesse, il change le statut de transaction de en cours à prêt et enregistre le résultat de la transaction. Cela permet au gestionnaire à grande vitesse de répondre à la transaction complete-split avec quelque chose à côté de NYET. Lorsque le gestionnaire à grande vitesse a vu un complete-split, il change le statut de transaction de prêt/x à ancien/x. Cela permet au gestionnaire à grande vitesse de réutiliser son tampon local pour d'autres transactions en masse/de contrôle une fois que ce complete-split est terminé.

Si l'hôte fait expirer la transaction ou ne reçoit pas un établissement de liaison valide, il réessaie immédiatement le complete-split avant de passer aux autres transactions en masse/de contrôle pour ce TT. Le mécanisme des "trois essais" s'applique ici également à l'hôte; à savoir, le err-count est incrémenté. Si pour une raison ou une autre, une mémoire d'hôte ou un retard de bus non-USB (ex.: une mémoire système "suspendue") se produit et fait en sorte que la transaction n'est pas terminée normalement, le err_count ne doit pas être incrémenté.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
Go to next cmd	Aller à cmd suivante
Retry start split	Réessayer start split
if err_count < 3 retry immed. comp. split	si err_count < 3 réessayer imméd. comp. split
if err_count >= 3 endpoint halt	si err_count >= 3 arrêt point d'extrémité
Inc err count	Inc décompte err
Endpoint halt	Arrêt point d'extrémité
Retry comp. split	Réessayer comp. split
No match	Pas de correspondance
ready/x or old/x or pending	prêt/x ou ancien/x ou en cours
If status = ready/x => status = old/x	Si état = prêt/x => état = ancien/x
pending	en cours
old/data	ancien/données
old/nak	ancien/nak
old/ack old/stall	ancien/ack ancien/arrêt
Host	Hôte
Complete split	Complete split

Figure 11-51. Séquence de transaction complet_split IN en masse/de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Si l'hôte reçoit un établissement de liaison STALL, il effectue le traitement d'arrêt du point d'extrémité et n'émet plus de transactions fractionnées pour ce point d'extrémité à vitesse intégrale/basse vitesse jusqu'à ce que la condition d'arrêt soit éliminée.

Si l'hôte reçoit un ACK, il enregistre les résultats de la transaction à vitesse intégrale/basse vitesse et passe à la transaction fractionnée suivante pour ce point d'extrémité. La transaction suivante est émise un peu plus tard selon les règles de programmation normales.

Si l'hôte reçoit un NAK, il réessaie la transaction start-split pour ce point d'extrémité un peu plus tard selon les règles de programmation normales. L'hôte ne doit pas incrémenter le err_count dans ce cas.

L'hôte doit continuer à réessayer le start-split actuel jusqu'à ce que le err_count atteigne trois pour ce point d'extrémité avant de passer à la transaction fractionnée suivante pour ce point d'extrémité. Mais, l'hôte peut émettre d'autres start-splits pour d'autres points d'extrémité avant de réessayer le start-split pour ce point d'extrémité.

Après que l'hôte ait reçu un établissement de liaison NAK, ACK ou STALL en réponse à une transaction complete-split, il peut ensuite émettre une transaction start-split pour le même point d'extrémité. L'hôte peut choisir alors d'émettre une transaction start-split pour un point d'extrémité différent qui n'est pas en attente d'une réponse complete-split.

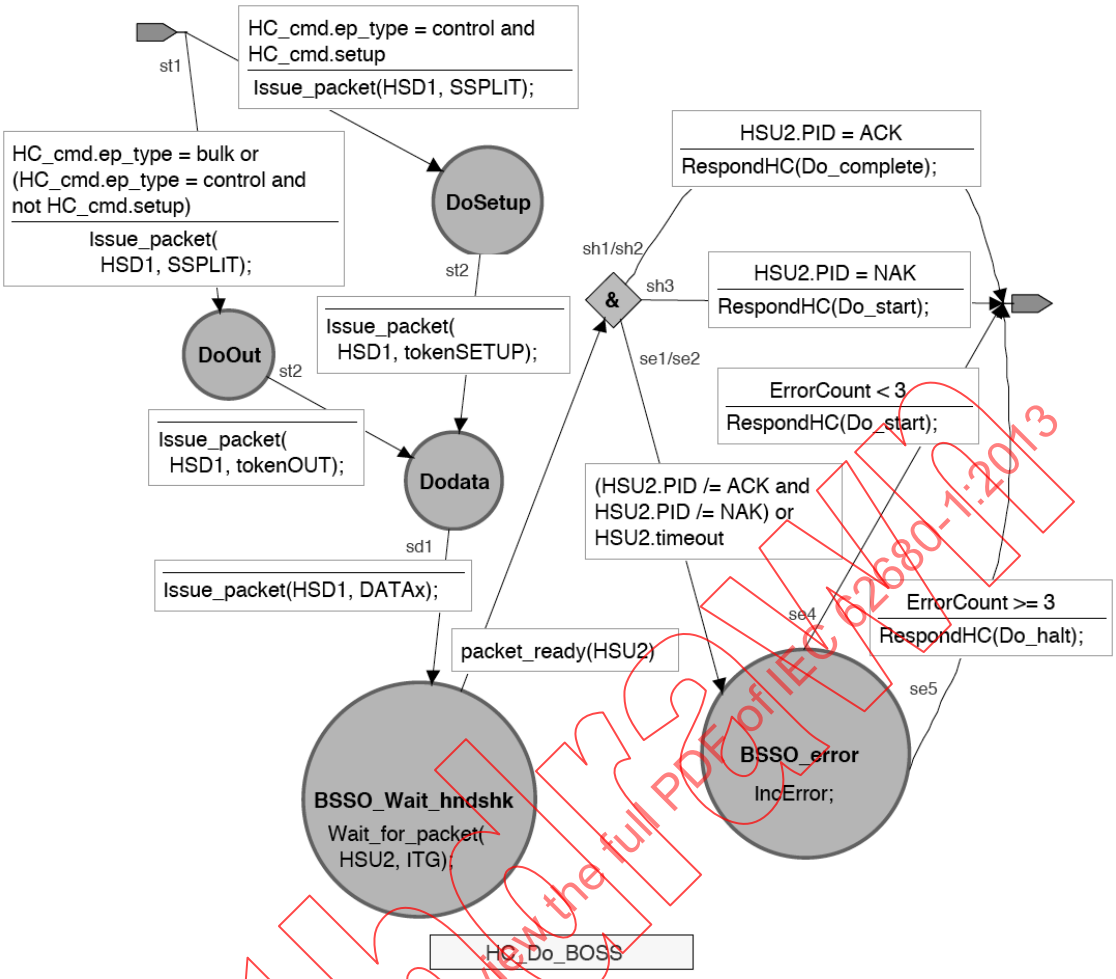
Le cas ombré illustré dans la figure indique qu'il convient qu'une transaction de configuration de contrôle ne rencontre jamais une réponse NAK car cela n'est pas admis pour les transactions à vitesse intégrale/basse vitesse.

La Figure 11-50 et la Figure 11-51 montrent les séquences de flux correspondantes pour les transactions fractionnées IN en masse/de contrôle.

11.17.2 Diagrammes d'état de transactions fractionnées en masse/de contrôle

Les diagrammes d'états de l'hôte et du TT pour les transactions fractionnées IN et OUT en masse/de contrôle sont illustrés dans les figures suivantes. Les transitions pour ces diagrammes d'états sont identifiées de la même manière que dans les figures de séquences de flux.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

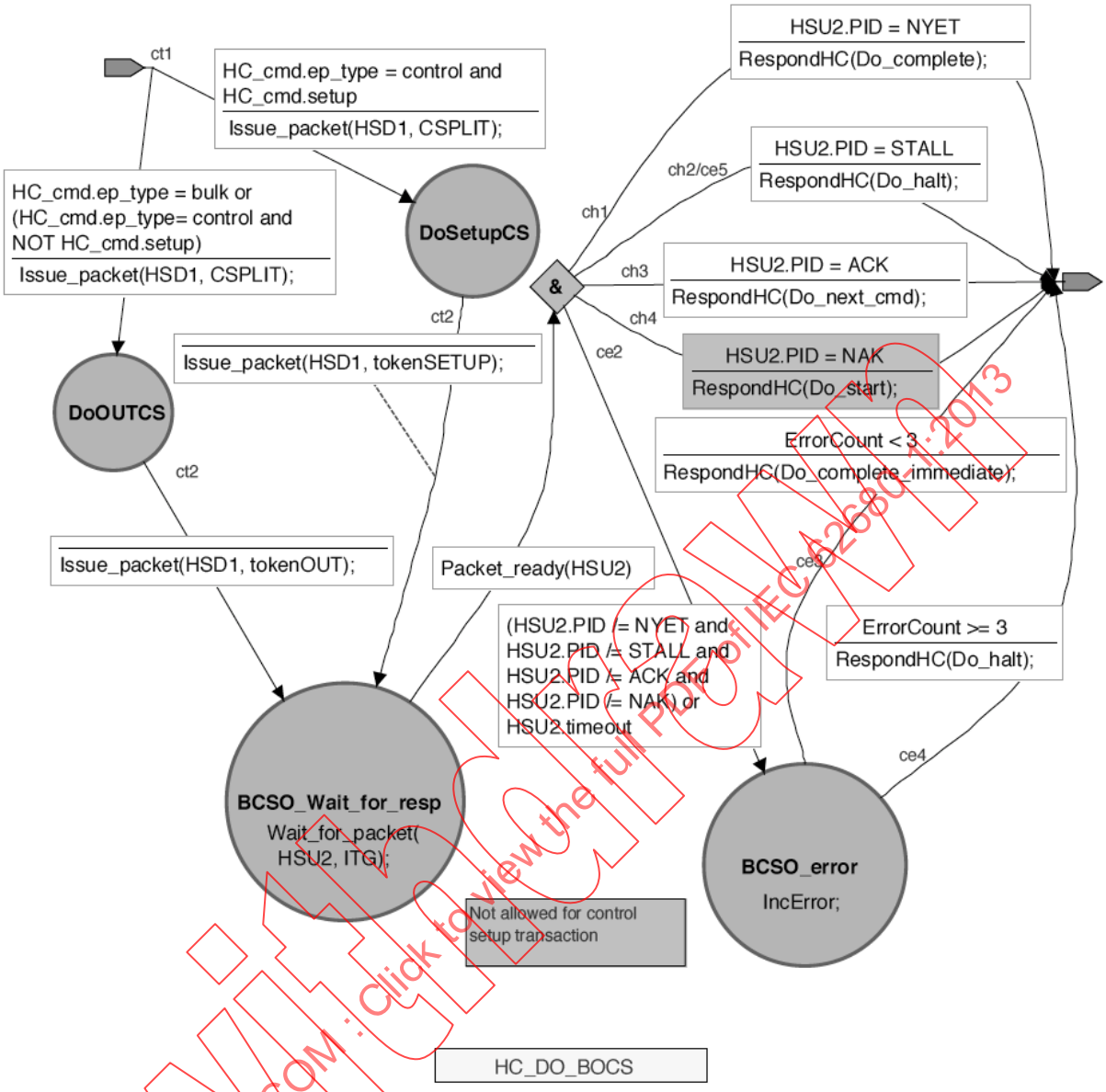


Légende

Anglais	Français
HC_cmd.ep_type = control and HC_cmd.setup	HC_cmd.ep_type = contrôle et HC_cmd.setup
HC_cmd.ep_type = bulk or (HC_cmd.ep_type = control and not HC_cmd.setup)	HC_cmd.ep_type = en masse ou (HC_cmd.ep_type = contrôle et non HC_cmd.setup)
(HSU2.PID / = ACK and HSU2.PID / = NAK) or HSU2.timeout	(HSU2.PID / = ACK et HSU2.PID / = NAK) ou HSU2.timeout

Figure 11-52. Diagramme d'états de l'hôte de transaction start-split OUT en masse/de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

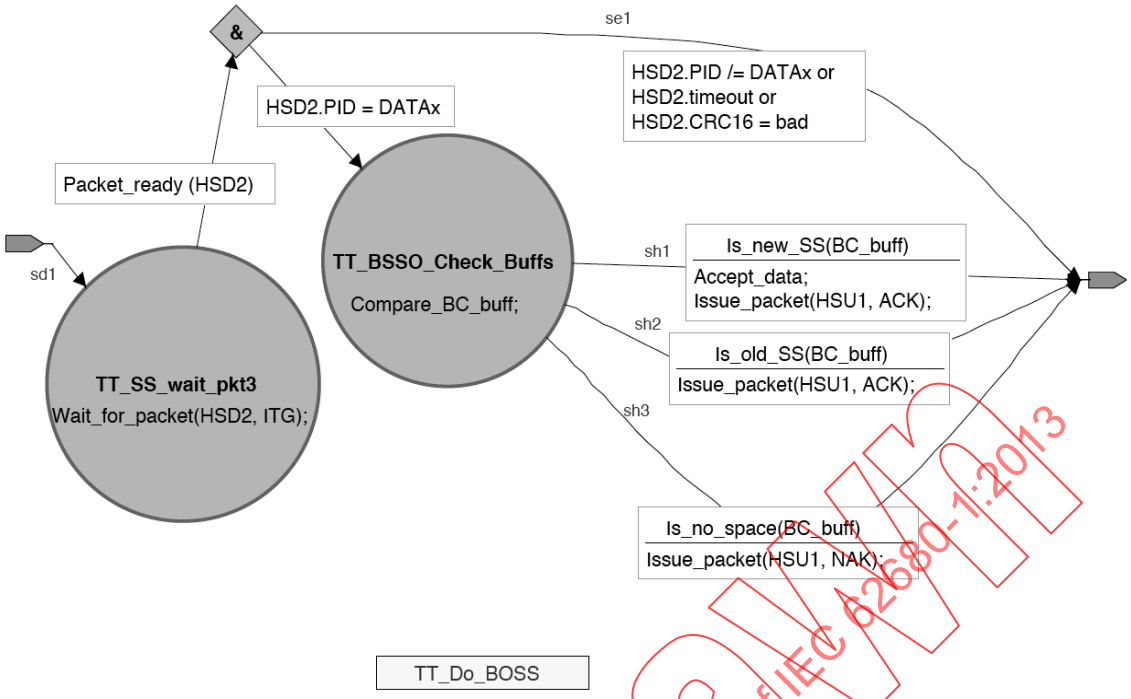


Légende

Anglais	Français
HC_cmd.ep_type = control and HC_cmd.setup	HC_cmd.ep_type = contrôle et HC_cmd.setup
HC_cmd.ep_type = bulk or (HC_cmd.ep_type = control and NOT HC_cmd.setup)	HC_cmd.ep_type = en masse ou (HC_cmd.ep_type = contrôle et NON HC_cmd.setup)
(HSU2.PID / = NYET and HSU2.PID / = STALL and HSU2.PID / = ACK and HSU2.PID / = NAK) or HSU2.timeout	(HSU2.PID / = NYET et HSU2.PID / = STALL et HSU2.PID / = ACK et HSU2.PID / = NAK) ou HSU2.timeout
Not allowed for control setup transaction	Pas autorisé pour transaction de configuration de commande

Figure 11-53. Diagramme d'états de l'hôte de transaction complete-split OUT en masse/de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

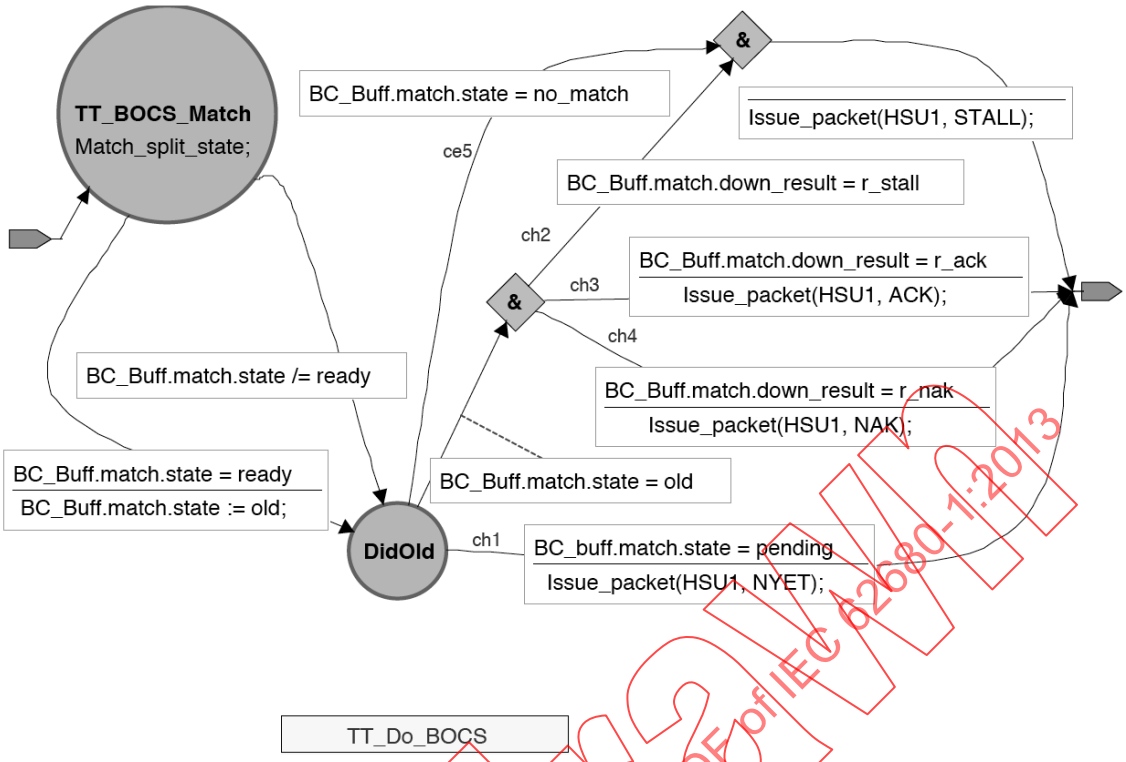


Légende

Anglais	Français
HSD2.PID / = DATAx or HSD2.timeout or HSD2.CRC16 = bad	HSD2.PID / = DATAx ou HSD2.timeout ou HSD2.CRC16 = bad

Figure 11-54. Diagramme d'états du TT de transaction start-split OUT en masse/de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

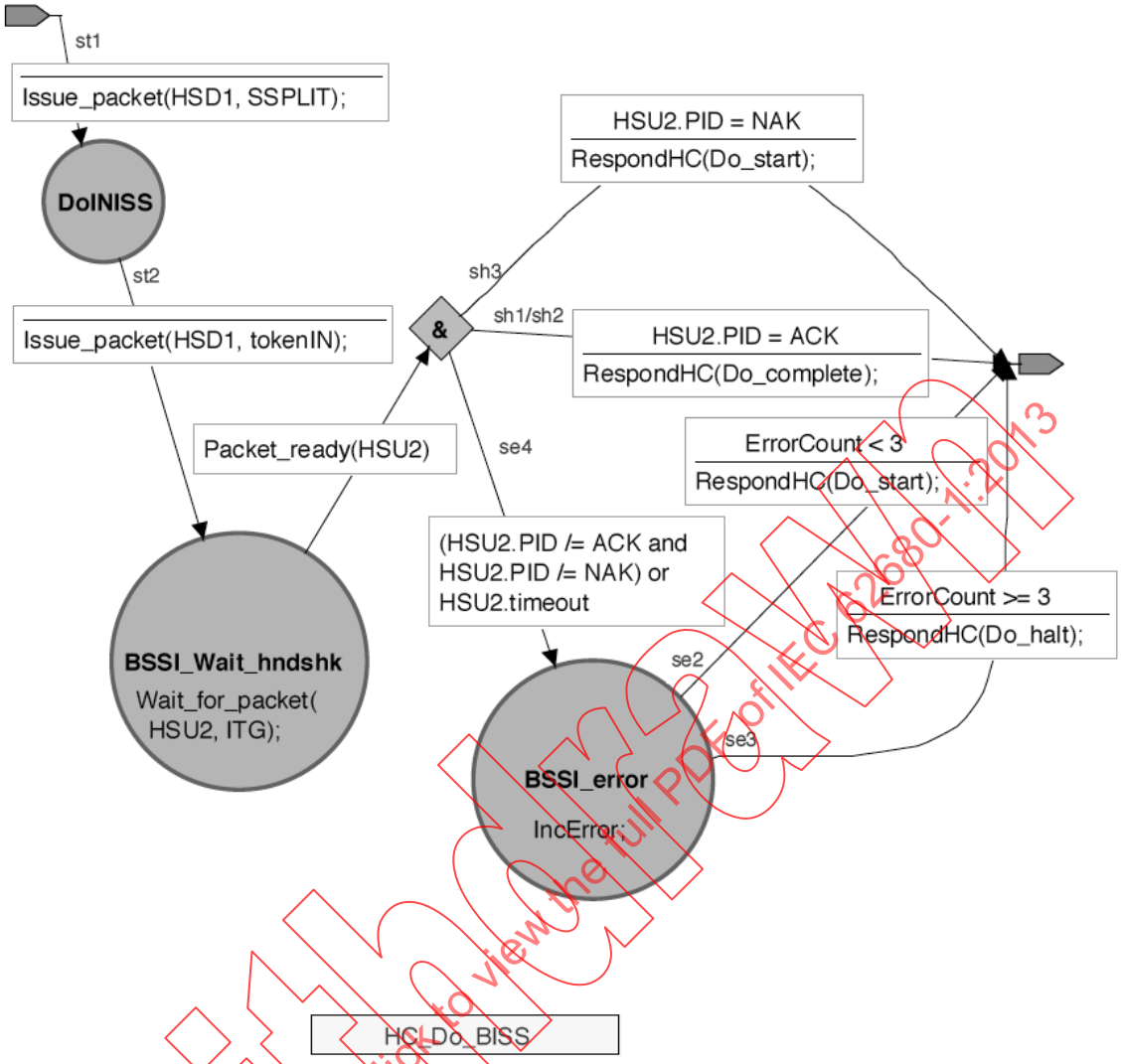


Légende

Anglais	Français
BC_Buff.match.state /= ready	BC_Buff.match.state /= prêt
BC_Buff.match.state = ready	BC_Buff.match.state = prêt
BC_Buff.match.state := old	BC_Buff.match.state := ancien
BC_Buff.match.state = old	BC_Buff.match.state = ancien
BC_Buff.match.state = pending	BC_Buff.match.state = en cours

Figure 11-55. Diagramme d'états du TT de transaction complete-split OUT en masse/de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

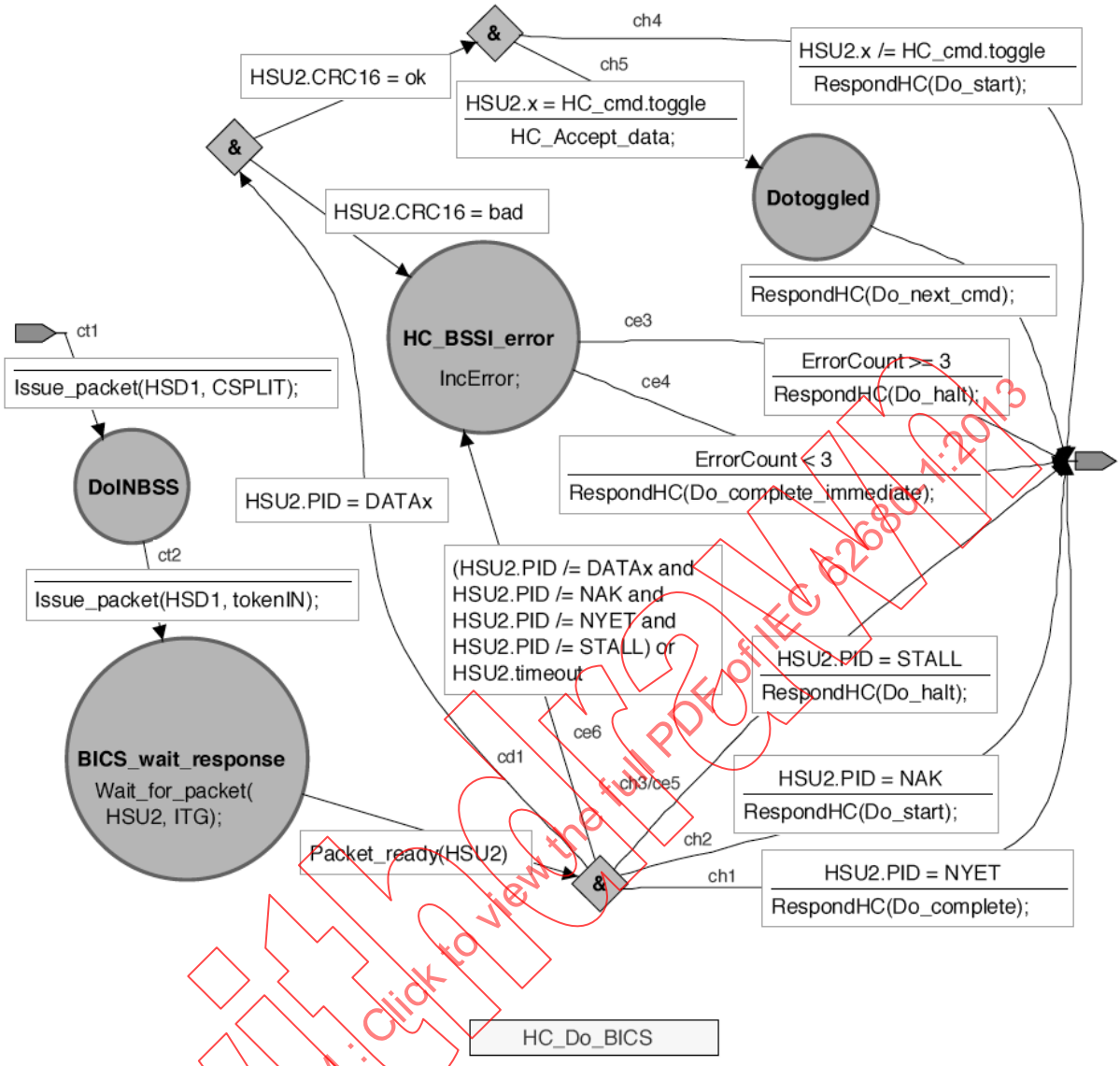


Légende

Anglais	Français
(HSU2.PID /= ACK and HSU2.PID /= NAK) or HSU2.timeout	(HSU2.PID /= ACK et HSU2.PID /= NAK) ou HSU2.timeout

Figure 11-56. Diagramme d'états de l'hôte de transaction start-split IN en masse/de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
(HSU2.PID / = DATAx and HSU2.PID / = NAK and HSU2.PID / = NYET and HSU2.PID / = STALL) or HSU2.timeout	(HSU2.PID / = DATAx et HSU2.PID / = NAK et HSU2.PID / = NYET et HSU2.PID / = STALL) ou HSU2.timeout

Figure 11-57. Diagramme d'états de l'hôte de transaction complete-split IN en masse/de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

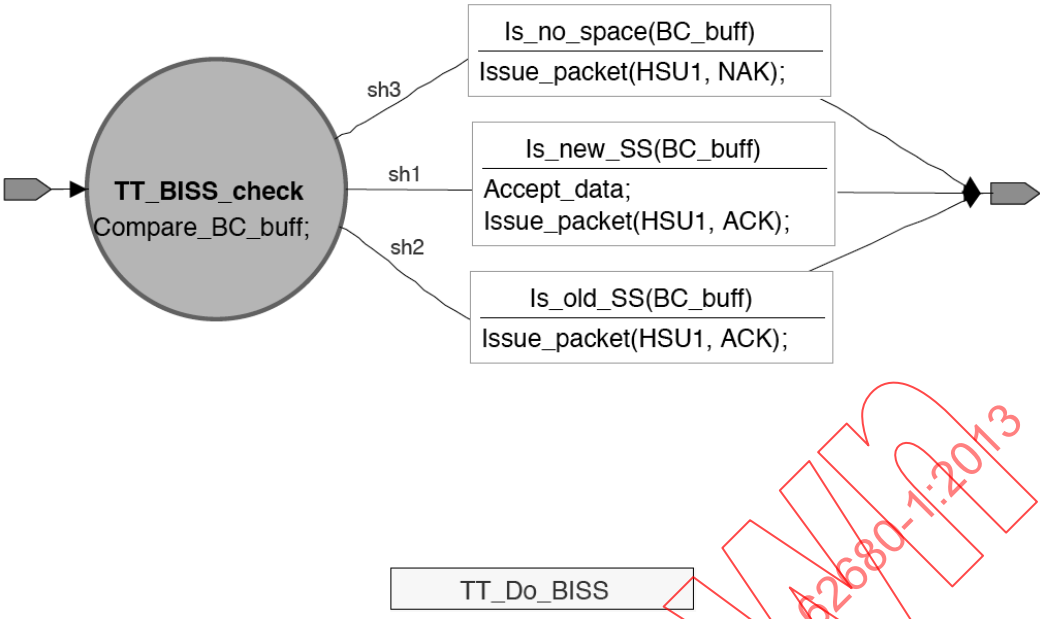
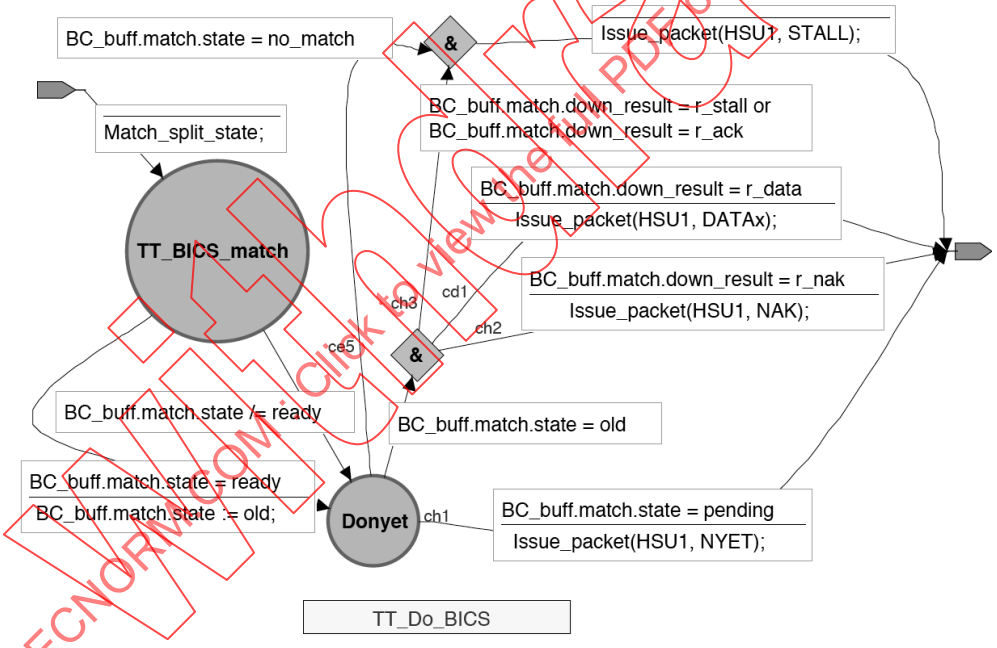


Figure 11-58. Diagramme d'états du TT de transaction start-split IN en masse/de contrôle



Légende

Anglais	Français
BC_buff.match.down_result = r_stall or BC_buff.match.down_result = r_ack	BC_buff.match.down_result = r_stall ou BC_buff.match.down_result = r_ack
BC_buff.match.state /= ready	BC_buff.match.state /= prêt
BC_buff.match.state = ready	BC_buff.match.state = prêt
BC_buff.match.state := old	BC_buff.match.state := ancien
BC_buff.match.state = old	BC_buff.match.state = ancien
BC_buff.match.state = pending	BC_buff.match.state = en cours

Figure 11-59. Diagramme d'états du TT de transaction complete-split IN en masse/de contrôle

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.17.3 Séquençage en masse/de contrôle

Lorsque le gestionnaire à grande vitesse a reçu un start-split pour un point d'extrémité et l'a enregistré dans un tampon local, il répond avec un établissement de liaison de transaction fractionnée ACK. Cela indique au contrôleur hôte d'effectuer une transaction complete-split la prochaine fois que ce point d'extrémité est analysé.

Dès que possible (soumis aux règles de programmation décrites précédemment), le gestionnaire à vitesse intégrale/basse vitesse émet la transaction à vitesse intégrale/basse vitesse et enregistre le statut d'établissement de liaison (pour OUT) ou le statut données/établissement de liaison (pour IN) dans le même tampon.

Un peu plus tard (selon le programme du contrôleur hôte), ce point d'extrémité est analysé pour la transaction complete-split. Le gestionnaire à grande vitesse répond au complete-split de retourner le statut du point d'extrémité à vitesse intégrale/basse vitesse pour cette transaction (comme enregistré dans le tampon). Si le contrôleur hôte analyse la transaction complete-split pour ce point d'extrémité avant que le gestionnaire à vitesse intégrale/basse vitesse ait terminé le traitement de cette transaction sur le bus en aval, le gestionnaire à grande vitesse répond avec un établissement de liaison NYET. Cela indique au contrôleur hôte que la transaction n'est pas encore terminée. Dans ce cas, le contrôleur hôte réessaie le complete-split un peu plus tard.

Lorsque le gestionnaire à vitesse intégrale/basse vitesse termine la transaction à vitesse intégrale/basse vitesse, il enregistre les données/le statut dans le tampon comme prêt pour la transaction complete-split suivante du contrôleur hôte pour ce point d'extrémité. Lorsque l'hôte envoie le complete-split, le gestionnaire à grande vitesse répond aux données/au statut indiqué comme enregistré dans le tampon. Le statut de transaction du tampon est mis à jour de prêt à ancien de sorte que le gestionnaire à grande vitesse est prêt pour une nouvelle tentative ou une nouvelle transaction start-split pour ce (ou un autre) point d'extrémité à vitesse intégrale/basse vitesse.

En cas d'erreur sur la transaction complete-split, le contrôleur hôte réessaie la transaction complete-split pour ce point d'extrémité en masse/de contrôle "immédiatement" avant de traiter d'autres transactions fractionnées en masse/de contrôle. Le contrôleur hôte peut émettre d'autres transactions fractionnées périodiques ou d'autres transactions non périodiques avant d'effectuer cette nouvelle tentative de transaction complete-split.

En cas de transaction en masse/de contrôle en cours sur le bus orienté en aval lorsque le temps EOF se produit, le TT doit observer la définition dans la Section 11.3 pour son comportement sur le bus orienté en aval. Cela entraîne une augmentation du décompte d'erreurs pour cette transaction. Les règles de nouvelle tentative normales déterminent si la transaction est réessayée ou non sur le bus orienté en aval.

11.17.4 Exigences vis-à-vis de la mise en tampon en masse/de contrôle

Le TT doit fournir au moins deux transactions de mise en tampon non périodique pour permettre au TT de fournir le débit à vitesse intégrale/basse vitesse maximum sur un bus en aval lorsque le bus à grande vitesse est inactif.

Comme le bus à grande vitesse devient plus occupé, le débit possible sur les bus à vitesse intégrale/basse vitesse en aval diminue.

Un TT peut fournir plus de deux transactions de mise en tampon non périodique et cela peut améliorer le débit pour les bus en aval pour des combinaisons spécifiques de configurations des appareils.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.17.5 Autres détails en masse/de contrôle

Lorsqu'une transaction fractionnée en masse/de contrôle échoue, elle peut quitter le tampon de transaction du TT associé dans un état occupé (prêt/x). Cet état de tampon ne permet pas de réutiliser le tampon pour les autres transactions fractionnées en masse/de contrôle. Par conséquent, dans le cadre du traitement d'arrêt du point d'extrémité pour les points d'extrémité à vitesse intégrale/basse vitesse raccordés via un TT, le logiciel hôte doit utiliser la demande Clear_TT_Buffer vers le TT pour garantir que le tampon n'est pas dans l'état occupé.

L'Annexe A montre des exemples de séquences de paquets pour les transactions en masse/de contrôle à vitesse intégrale/basse vitesse et leurs relations avec les start-splits et complete-splits dans diverses conditions normales et d'erreur.

11.18 Mise en pipeline des transactions fractionnées périodiques et gestion des tampons

Il existe des exigences concernant le comportement de l'hôte et du TT pour garantir que le pipeline de microtrame séquence correctement les transactions isochrones/interruption à vitesse intégrale/basse vitesse sur les bus à vitesse intégrale/basse vitesse orientés en aval. L'hôte doit déterminer les microtrames dans lesquelles une transaction start-split et complete-split doit être émise à grande vitesse pour séquencer correctement une transaction à vitesse intégrale/basse vitesse correspondante sur le bus orienté en aval. C'est ce qu'on appelle la "programmation" des transactions fractionnées.

Dans les descriptions suivantes, les 8 microtrames dans chaque trame à vitesse intégrale (1 ms) sont appelées microtrame Y_0, Y_1, Y_2, …, Y_7. Cette notation signifie que la première microtrame de chaque trame à vitesse intégrale est identifiée Y_0. La seconde microtrame est identifiée Y_1, etc. La dernière microtrame de chaque trame à vitesse intégrale est identifiée Y_7. Les libellés se répètent pour chaque trame à vitesse intégrale.

Cette section décrit les détails du pipeline de microtrame qui affectent les transactions isochrones à vitesse intégrale et les transactions d'interruption à vitesse intégrale/basse vitesse. Les règles de transactions fractionnées pour interruption et isochrone sont ensuite décrites.

Les transactions en masse/de contrôle ne sont pas programmées avec ce mécanisme. Elles sont traitées comme décrit dans la section précédente.

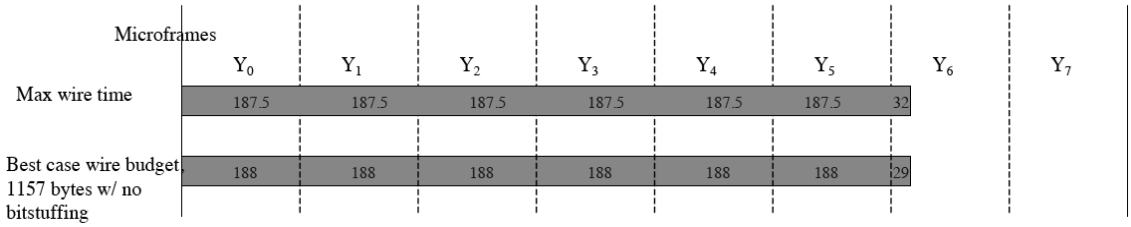
11.18.1 Meilleur budget à vitesse intégrale

Une microtrame de temps permet au maximum 187,5 octets bruts de signalisation sur un bus à vitesse intégrale. Pour estimer quand les transactions à vitesse intégrale/basse vitesse apparaissent sur un bus en aval, l'hôte doit calculer le meilleur budget à vitesse intégrale. Ce budget suit dans quelles microtrames apparaît une transaction à vitesse intégrale/basse vitesse. Le meilleur budget à vitesse intégrale suppose que les 188 octets à vitesse intégrale se produisent dans chaque microtrame. La Figure 11-60 montre comment une trame de 1 ms est sous-divisée en microtrames de temps de budget. Cette estimation suppose qu'aucun bourrage de bits n'a lieu pour allonger le temps requis pour déplacer les transactions sur le bus.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Le nombre maximum d'octets dans une trame de 1 ms est calculé comme suit:

1 157 maximum_periodic_bytes_per_frame = 12 Mb/s * 1 ms / 8 bits_per_byte * 6 data_bits / 7 bit-stuffed_data_bits * 90% maximum_periodic_data_per_frame



Légende

Anglais	Français
Microframes	Microtrames
Max wire time	Temps de câblage max
Best case wire budget 1157 bytes w/no bitstuffing	Meilleur budget de câblage 1 157 octets avec ou sans bourrage de bits

Figure 11-60. Meilleur temps de câblage à vitesse intégrale budgété sans bourrage de bits

11.18.2 Pipeline de microtrame TT

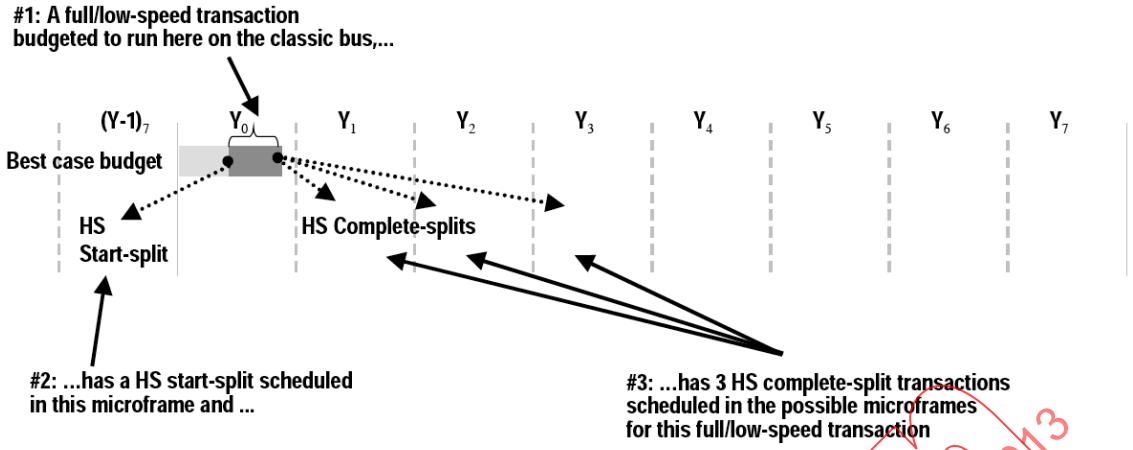
Le TT applique un pipeline de microtrame des transactions fractionnées pour assister un bus à vitesse intégrale/basse vitesse. Les transactions start-split sont programmées une microtrame avant le temps le plus tôt où leur transaction à vitesse intégrale/basse vitesse correspondante devrait démarrer. Les transactions complete-split sont programmées dans des microtrames que la transaction à vitesse intégrale/basse vitesse peut finir.

Lorsqu'un appareil à vitesse intégrale/basse vitesse est raccordé au bus et configuré, l'hôte assigne un peu de temps sur le bus à vitesse intégrale-/basse vitesse à un certain temps budgété, en fonction des exigences de point d'extrémité de l'appareil configuré.

Les effets de bourrage de bits peuvent différer lorsque la transaction à vitesse intégrale/basse vitesse est réellement exécutée. Les résultats des autres transactions à vitesse intégrale/basse vitesse précédentes peuvent entraîner l'exécution de la transaction plus tôt ou plus tard sur le bus à vitesse intégrale/basse vitesse.

L'hôte utilise toujours la taille de charge utile de données maximum pour un point d'extrémité à vitesse intégrale/basse vitesse lors de sa budgétisation. Il n'essaie pas de programmer les charges utiles de données réelles qui peuvent être utilisées dans des transactions spécifiques sur les points d'extrémité à vitesse intégrale/basse vitesse. L'hôte doit inclure l'écart interpaquet de durée maximum, les délais de rotation de bus et "TT think time". Le TT exige un peu de temps pour passer à la transaction à vitesse intégrale/basse vitesse suivante. Ce temps est appelé "TT think time" et est spécifié dans le champ du descripteur de hub *wHubCharacteristics* bits 5 et 6.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
#1: A full-low-speed transaction budgeted to run here on the classic bus, …	#1: Une transaction à vitesse intégrale/basse vitesse budgétisée pour fonctionner ici sur le bus classique, …
Best case budget	Meilleur budget
HS Start-split	Start-split HS
HS Complete-splits	Complete-splits HS
#2:.. has a HS start-split scheduled in the microframe and …	#2:.. a un start-split HS planifié dans la microtrame et …
#3:…has 3 HS complete-split transactions scheduled in the possible microframes for this full/low-speed transaction	#3:…a 3 transactions complete-split HS planifiées dans les microtrames possibles pour cette transaction à vitesse intégrale/basse vitesse

Figure 11-61. Programmation de pipeline de microtrame TT

La Figure 11-61 montre un exemple de nouveau point d'extrémité qui est assigné à une portion d'une trame à vitesse intégrale/basse vitesse et où ses start et complete-splits sont généralement programmés. L'acte d'assignation d'une portion de la trame à vitesse intégrale/basse vitesse à une transaction particulière est appelé détermination du budget de la transaction. Des règles plus précises de programmation et de budgétisation sont présentées ultérieurement. Le start-split pour cet exemple de transaction est programmé dans la microtrame $Y\text{-}1_7$, la transaction est budgétisée pour être exécutée dans la microtrame Y_0, et les complete-splits sont programmés pour les microtrames Y_1, Y_2 et Y_3. La Section 11.18.4 décrit les règles de programmation plus en détail.

L'hôte doit déterminer précisément quand les start- et complete- splits sont programmés pour éviter les sur- et sous-utilisations dans les tampons de transactions périodiques fournis par le TT.

11.18.3 Génération de trames à vitesse intégrale

Le TT doit générer des SOF sur le bus à vitesse intégrale pour établir l'horloge de trame 1 ms dans les tolérances de gigue définies pour les appareils à vitesse intégrale. Le TT a sa propre horloge de trame qui est synchronisée sur les SOF de microtrame sur le bus à grande vitesse. Le SOF qui reflète un changement de numéro de trame qu'il transporte est identifié comme SOF de microtrame zéro. Le SOF de microtrame zéro à grande vitesse correspond au SOF à vitesse intégrale sur le bus orienté en aval du TT. Le TT doit adhérer à toutes les exigences de synchronisation/gigue d'un contrôleur hôte associé aux trames comme défini dans les autres parties de cette spécification.

Le TT doit arrêter l'émission des SOF à vitesse intégrale après qu'il détecte 250 µs d'inactivité à grande vitesse. Cela est nécessaire pour garantir que le bus orienté en aval à vitesse intégrale/basse vitesse entre dans l'état suspension pas plus de 250 µs après l'entrée du bus à grande vitesse dans l'état suspension.

Le TT doit générer un SOF à vitesse intégrale sur le bus orienté en aval en fonction de son minuteur de trame. La génération du SOF à vitesse intégrale doit se produire dans +/-3 temps

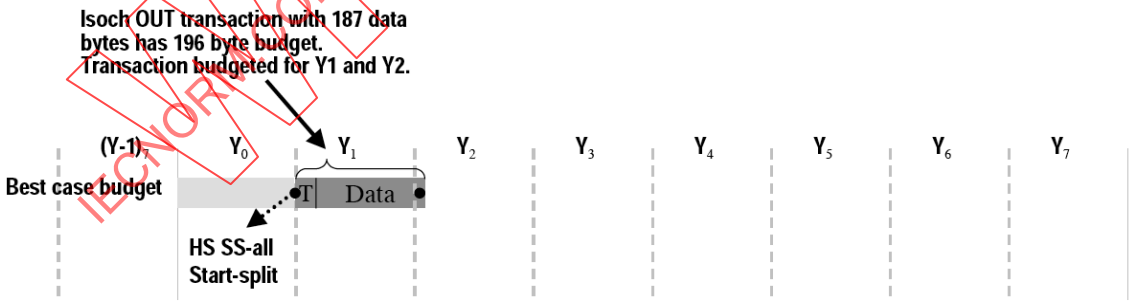
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

bit à vitesse intégrale à partir de l'occurrence du SOF à grande vitesse zéro. Voir Section 11.22.1 pour plus d'informations sur la génération de SOF TT.

11.18.4 Exigences vis-à-vis de la programmation de transactions programmées de l'hôte

La programmation de transactions fractionnées est effectuée par l'hôte (généralement dans le logiciel) en fonction de la meilleure estimation de la manière dont les transactions à vitesse intégrale/basse vitesse peuvent être exécutées sur le bus orienté en aval. Cette meilleure estimation est appelée meilleur budget. L'hôte est libre d'émettre les transactions fractionnées n'importe quand dans la microtrame programmée, mais chaque transaction fractionnée doit être émise parfois dans la microtrame programmée. Cette description des exigences de programmation s'applique aux transactions fractionnées pour une seule transaction à vitesse intégrale/basse vitesse à la fois.

1. L'hôte ne doit jamais programmer un start-split dans la microtrame Y_6. Certaines conditions d'erreur peuvent entraîner l'émission erronée par le contrôleur hôte d'un start-split dans cette microtrame. La réponse du TT à ce start-split est indéfinie.
2. L'hôte doit calculer le programme start-split en déterminant le meilleur budget pour la transaction et:
   a. Pour les transactions à vitesse intégrale OUT isochrones, pour chaque microtrame dans laquelle la transaction est budgétisée, l'hôte doit programmer une transaction start-split de 188 (ou la taille de données restante) d'octets de données. La transaction start-split doit être programmée dans la microtrame avant que les données ne soient budgétisées pour commencer sur le bus à vitesse intégrale. Les transactions start-split doivent utiliser les codages de jeton de transaction fractionnée début/milieu/fin/tout correspondant à l'élément de données à vitesse intégrale qui est envoyé sur le bus à grande vitesse. Par exemple, si un seul start-split est requis, un codage "tout" est utilisé. Si de multiples start-splits sont requis, un codage "début" est utilisé pour le premier start-split et un codage "fin" est utilisé pour le dernier start-split. Si plus de deux start-splits sont requis, les start-splits supplémentaires qui ne sont pas le premier ou le dernier utilisent un codage "milieu". Une charge utile de données à vitesse intégrale de longueur zéro doit uniquement être programmée avec un start-split "all". Une transaction start-split pour un start-split de début, milieu ou fin doit toujours avoir une charge utile de données de longueur différente de zéro. La Figure 11-62 montre un exemple d'un OUT isochrone qui semblerait avoir budgétisé une charge utile de données de longueur zéro dans un start-split (fin). Cet exemple doit être programmé avec une transaction start-split (tout).



Schedule SS-all with 187 data bytes, not SS-begin(187 data) and SS-end (0 data).

An Isoch OUT only ever has zero length data in SS-all.

Légende

Anglais	Français
Isoch OUT transaction with 187 data bytes has 196 byte budget. Transaction budgeted for Y1 and Y2.	La transaction OUT isoch avec 187 octets de données a un budget de 196 octets. Transaction budgétisée pour Y1 et Y2.
Best case budget	Meilleur budget

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Data	Données
HS SS-all Start-split	Start-split SS-tout HS
Schedule SS-all with 187 data bytes, not SS-begin (187 data) and SS-end (0 data). An Isoch OUT only ever has zero length data in SS-all.	Planifier SS-tout avec 187 octets de données, pas SS-début (187 données) et SS-fin (0 données). Un OUT isoch a seulement des données de longueur zéro dans SS-tout.

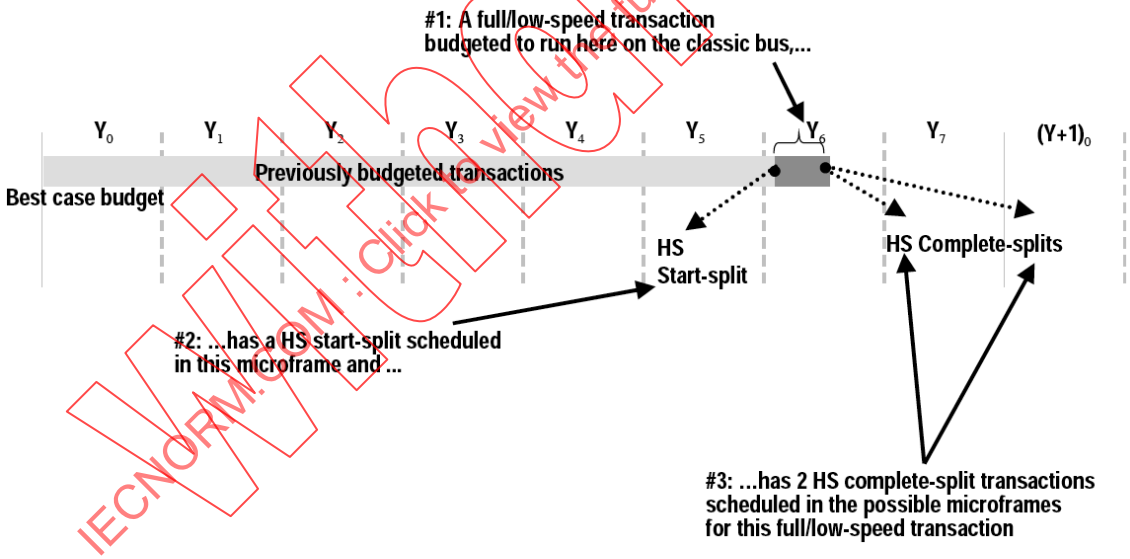
Figure 11-62. Exemple de OUT isochrone qui évite une fin start-split avec des données égales à zéro

b. Pour les transactions IN isochrones et les transactions à vitesse intégrale/basse vitesse IN/OUT d'interruption, un seul start-split doit être programmé dans la microtrame avant que la transaction ne soit budgétisée pour démarrer sur le bus à vitesse intégrale/basse vitesse.

3. L'hôte ne programme jamais plusieurs complete-split dans une microtrame pour la même transaction à vitesse intégrale/basse vitesse.

   a. Pour les transactions à vitesse intégrale OUT isochrones, l'hôte ne doit jamais programmer un complete-split. La réponse du TT à un complete-split pour un OUT isochrone est indéfinie.

   b. Pour les transactions à vitesse intégrale/basse vitesse IN/OUT d'interruption, l'hôte doit programmer une transaction complete-split dans chacune des deux microtrames suivant la première microtrame dans laquelle la transaction à vitesse intégrale/basse vitesse est budgétisée. Un complete-split supplémentaire doit également être programmé dans la troisième microtrame suivante à moins que la transaction à vitesse intégrale/basse vitesse n'ait été budgétisée pour démarrer dans la microtrame Y_6. La Figure 11-63 montre un exemple avec seulement deux complete-splits.



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

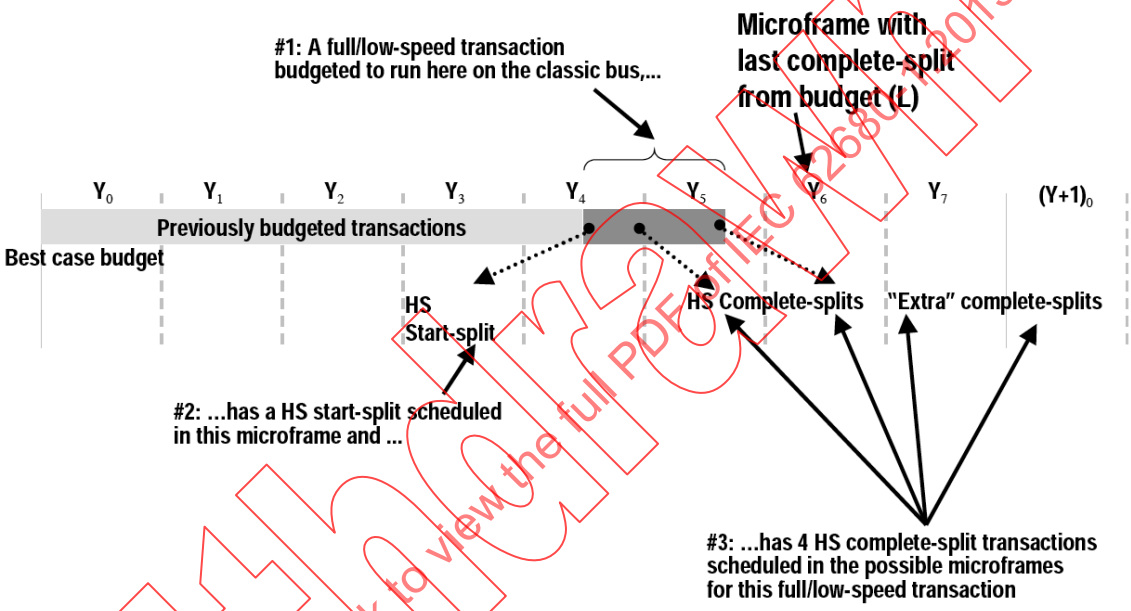
Anglais	Français
#1: A full-low-speed transaction budgeted to run here on the classic bus, …	#1: Une transaction à vitesse intégrale/basse vitesse budgétisée pour fonctionner ici sur le bus classique, …
Best case budget	Meilleur budget
#2:.. has a HS start-split scheduled in this microframe and …	#2:.. a un start-split HS planifié dans cette microtrame et …
#3:…has 2 HS complete-split transactions scheduled in the possible microframes for this full/low-speed transaction	#3:…a 2 transactions complete-split HS planifiées dans les microtrames possibles pour cette transaction à vitesse intégrale/basse vitesse
Previously budgeted transactions	Transactions précédemment budgétisées

Figure 11-63. Exemple de programmation de pipeline TT Fin de trame

c. Pour les transactions à vitesse intégrale IN isochrones, pour chaque microtrame dans laquelle la transaction à vitesse intégrale est budgétisée, un complete-split doit être programmé pour chaque microtrame suivante. Déterminer aussi la dernière microtrame dans laquelle un complete-split est programmé, l'appeler L. Si L est inférieur à Y_6, planifier des complete-splits supplémentaires dans la microtrame L+1 et L+2.

Si L est égal à Y_6, programmer un complete-split dans la microtrame Y_7. Programmer également un complete-split dans la microtrame Y_0 de la trame suivante, à moins que la transaction à vitesse intégrale n'ait été budgétisée pour démarrer dans la microtrame Y_0.

Si L est égal à Y_7, programmer un complete-split dans la microtrame Y_0 de la trame suivante, à moins que la transaction à vitesse intégrale n'ait été budgétisée pour démarrer dans la microtrame Y_0. La Figure 11-64 et la Figure 11-65 montrent des exemples des cas pour L= Y_6 et L=Y_7.

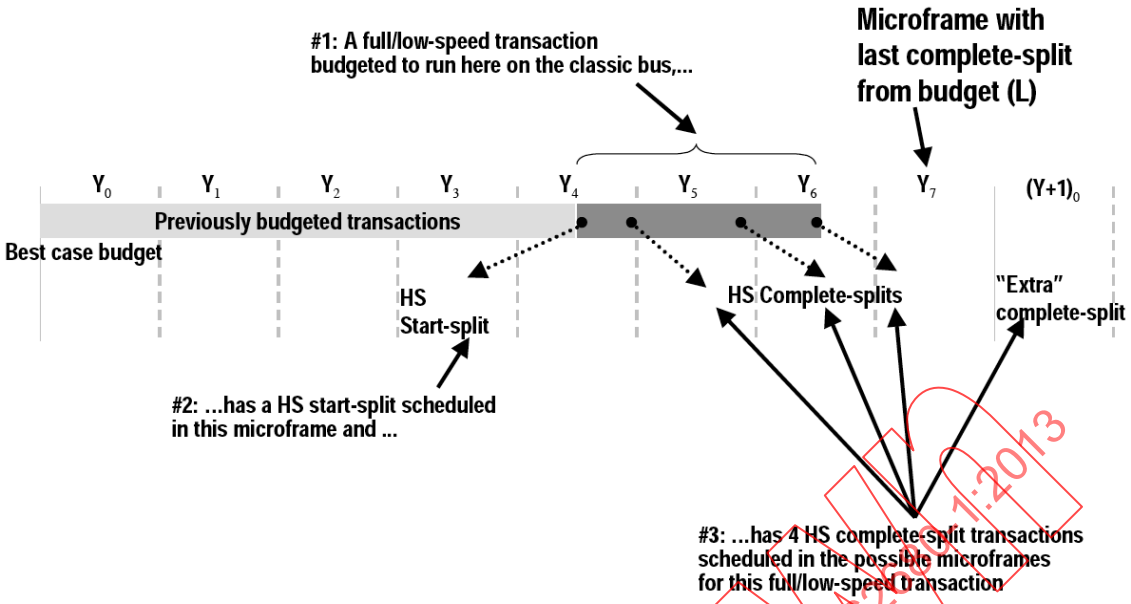


Légende

Anglais	Français
#1: A full-low-speed transaction budgeted to run here on the classic bus, ...	#1: Une transaction à vitesse intégrale/basse vitesse budgétisée pour fonctionner ici sur le bus classique, ...
Best case budget	Meilleur budget
#2:.. has a HS start-split scheduled in this microframe and ...	#2:.. a un start-split HS planifié dans cette microtrame et ...
#3:...has 4 HS complete-split transactions scheduled in the possible microframes for this full/low-speed transaction	#3:...a 4 transactions complete-split HS planifiées dans les microtrames possibles pour cette transaction à vitesse intégrale/basse vitesse
Previously budgeted transactions	Transactions précédemment budgétisées
Microframe with last complete-split from budget (L)	Microtrame avec dernier complete-split du budget (L)
"Extra" complete-splits	Complete-splits "supplémentaires"

Figure 11-64. Exemple de programme complete-split IN isochrone avec L=Y_6

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Anglais	Français
#1: A full-low-speed transaction budgeted to run here on the classic bus, …	#1: Une transaction à vitesse intégrale/basse vitesse budgétisée pour fonctionner ici sur le bus classique, …
Best case budget	Meilleur budget
HS Start-split	Start-split HS
HS Complete-splits	Complete-splits HS
#2:.. has a HS start-split scheduled in this microframe and …	#2:.. a un start-split HS planifié dans cette microtrame et …
#3:...has 4 HS complete-split transactions scheduled in the possible microframes for this full/low-speed transaction	#3:...a 4 transactions complete-split HS planifiées dans les microtrames possibles pour cette transaction à vitesse intégrale/basse vitesse
Previously budgeted transactions	Transactions précédemment budgétisées
Microframe with last complete-split from budget (L)	Microtrame avec dernier complete-split du budget (L)
"Extra" complete-split	Complete-split "supplémentaire"

Figure 11-65. Exemple de programme complete-split IN isochrone avec L=Y_7

4. L'hôte ne doit jamais émettre plus de 16 start-splits dans une microtrame à grande vitesse pour n'importe quel TT.
5. L'hôte doit uniquement émettre une transaction fractionnée dans la microtrame dans laquelle elle a été programmée.
6. Comme précisément identifié dans la séquence de flux et les figures des diagrammes d'états, le contrôleur hôte doit immédiatement réessayer un complete-split après une erreur de transaction à grande vitesse ('trans_err").

Le "modèle" des transactions fractionnées programmées pour une transaction à vitesse intégrale/basse vitesse peut être calculé une fois que chaque point d'extrémité est configuré. Le modèle ne change alors pas à moins que certains changements ne se produisent concernant la collection des points d'extrémité à vitesse intégrale/basse vitesse actuellement configurés et raccordés via un TT.

Finalement, pour tous les points d'extrémité périodiques qui ont des transactions fractionnées programmées dans une microtrame particulière, l'hôte doit émettre des transactions complete-split dans le même ordre dans lequel les transactions start-split correspondantes ont été émises.

11.18.5 Génération de réponse TT

L'approche utilisée pour les IN isochrones à vitesse intégrale et les IN/OUT d'interruption garantit qu'il y a toujours une opportunité pour le TT de retourner les données/résultats lorsqu'il a quelque chose à retourner de la transaction à vitesse intégrale/basse vitesse. Alors, lorsque le gestionnaire à vitesse intégrale/basse vitesse démarre la transaction à vitesse intégrale/basse vitesse, il accumule simplement les résultats dans chaque microtrame, puis la

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

retourne en réponse à un complete-split de l'hôte. Le TT agit de manière similaire à un appareil isochrone dans la mesure où il utilise la limite de microtrame pour "morceler" les données à vitesse intégrale/basse vitesse à retourner à l'hôte. Le TT n'effectue pas de calcul sur la quantité de données à retourner et à quel moment. En réponse au complete-split à grande vitesse "suivant", le TT retourne simplement les données de point d'extrémité qu'il a reçues du bus à vitesse intégrale/basse vitesse dans une microtrame.

Lorsque le TT doit retourner des données en réponse à un complete-split pour une transaction à vitesse intégrale/basse vitesse d'interruption ou une transaction à vitesse intégrale isochrone, il utilise DATA0/1 ou MDATA pour le PID de paquet de données.

Si le gestionnaire à vitesse intégrale/basse vitesse termine la transaction IN isochrone/d'interruption à vitesse intégrale/basse vitesse pendant une microtrame avec un CRC16 valide, il utilise le PID DATA0/1 pour le paquet de données de la transaction complete-split. Cela indique qu'il s'agit des dernières données de la transaction à vitesse intégrale/basse vitesse. Un PID DATA0 est toujours utilisé pour les transactions isochrones. Pour les transactions d'interruption, un PID DATA0/1 correspondant au PID du paquet de données à vitesse intégrale/basse vitesse reçu est utilisé.

Si le gestionnaire à vitesse intégrale/basse vitesse termine la transaction IN isochrone/d'interruption à vitesse intégrale/basse vitesse pendant une microtrame avec un CRC16 incorrect, il utilise la réponse ERR à la transaction complete-split et ne retourne pas les données reçues de l'appareil à vitesse intégrale/basse vitesse.

Si le TT continue de recevoir les données sur le bus orienté en aval à la limite de microtrame, le TT répond avec un PID MDATA ou un NYET pour le complete-split correspondant. Si le TT a reçu plus de deux octets du champ de données du paquet de données à vitesse intégrale/basse vitesse, il répond avec un PID MDATA. De plus, le paquet de données qui est retourné dans le complete-split doit contenir les données reçues de l'appareil à vitesse intégrale/basse vitesse moins les deux derniers octets. Les deux derniers octets ne doivent pas être inclus, car ils peuvent être le champ CRC16, mais le TT ne va pas le savoir avant la microtrame suivante. Le champ CRC16 reçu de l'appareil à vitesse intégrale/basse vitesse n'est jamais retourné dans un paquet de données complete-split pour les transactions isochrones/d'interruption. Si moins de trois octets de données du paquet de données à vitesse intégrale/basse vitesse ont été reçus à la fin d'une microtrame, le TT doit répondre avec un NYET au complete-split à grande vitesse correspondant. Ces deux réponses indiquent à l'hôte que davantage de données sont reçues et qu'une autre transaction complete-split est requise.

Lorsque le contrôleur hôte reçoit un PID DATA0/1 pour les complete-splits IN d'interruption ou isochrones (et ACK, NAK, STALL, ERR pour les complete-splits IN/OUT d'interruption), il arrête d'émettre le reste de complete-splits qui pourraient être programmés pour ce point d'extrémité pour cette transaction à vitesse intégrale/basse vitesse.

Si le TT n'a pas démarré la transaction à vitesse intégrale/basse vitesse lorsqu'il reçoit un complete-split, le TT ne trouve pas d'entrée dans l'étage de pipeline complete-split. Lorsque cela se produit, les diagrammes d'états de protocole montrent que le TT répond avec un NYET (ex.: le cas "pas de correspondance"). Cette réponse NYET indique à l'hôte qu'il n'y a pas de résultats disponibles actuellement, mais qu'il convient que l'hôte continue avec d'autres transactions fractionnées programmées pour ce point d'extrémité dans les microtrames suivantes.

En général, il y a deux (ou plus) transactions complete-split programmées pour un point d'extrémité périodique. Mais, pour les points d'extrémité d'interruption, la taille maximum de la transaction à vitesse intégrale/basse vitesse garantit qu'elle ne peut pas nécessiter plus de deux transactions complete-split. Deux transactions complete-split sont uniquement requises lorsque la transaction s'étend sur une limite de microtrame. Si la transaction à vitesse intégrale/basse vitesse démarre réellement et se termine dans la même microtrame, un seul complete-split retourne les données; les autres complete-splits avancés ont une réponse NYET.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Pour les transactions IN isochrones, davantage de transactions complete-split peuvent être programmées en fonction de la longueur de la transaction à vitesse intégrale. Une transaction IN isochrone à vitesse intégrale peut atteindre 1 023 octets de données, ce qui peut nécessiter des portions de maximum 8 microtrames de temps sur le bus orienté en aval (avec le pire alignement dans la trame et le pire bourrage de bits). Une telle transaction à vitesse intégrale de dimension maximum peut nécessiter 8 transactions complete-split. Si l'appareil génère moins de données, l'hôte arrête d'émettre des complete-splits après celui qui retourne les données finales de l'appareil pour une trame.

11.18.6 Exigences vis-à-vis du traitement des transactions périodiques TT

Le TT a deux méthodes qu'il doit utiliser pour réagir aux évènements associés à la synchronisation qui affectent le pipeline de microtrame: abandon de la transaction actuelle et autorisation des start-splits en cours. Ces méthodes doivent être utilisées pour gérer le pipeline de microtrame.

Le TT doit également réagir (comme décrit dans la Section 11.22.1) lorsque son minuteur de microtrame ou de trame perd la synchronisation avec le bus à grande vitesse.

Le TT ne doit pas émettre trop de transactions à vitesse intégrale/basse vitesse dans une microtrame.

Chacune de ces exigences est décrite ci-après.

11.18.6.1 Abandon de la transaction actuelle

Lorsqu'une transaction actuelle est en cours sur le bus orienté en aval et n'est plus appropriée pour que le TT continue la transaction, la transaction est "abandonnée".

Le gestionnaire à vitesse intégrale/basse vitesse TT doit abandonner la transaction à vitesse intégrale/basse vitesse actuelle:

1. Pour tous les types de transactions périodiques, si le temps d'EOF de trame à vitesse intégrale se produit
2. Si la transaction est une transaction d'interruption et le start-split de la transaction a été reçu dans certaines microtrames (l'appeler X) et le minuteur de microtrame TT indique la microtrame X+4

Il faut noter qu'aucun traitement d'abandon supplémentaire n'est requis pour les transactions isochrones outre le traitement IN/OUT générique décrit ci-après. L'abandon a différentes exigences de traitement eu égard au bus orienté en aval pour les transactions IN et OUT. Pour n'importe quel type de transaction, le TT ne doit pas générer de réponse complete-split pour une transaction abandonnée; ex.: aucune entrée n'est effectuée dans l'étage de pipeline complete-split pour une transaction abandonnée.

1. Au moment où le TT décide d'abandonner une transaction IN, le TT ne doit pas émettre le paquet d'établissement de liaison pour la transaction si l'établissement de liaison n'a pas déjà été démarré sur le bus orienté en aval. Le TT peut choisir de ne pas émettre le paquet de jeton IN, si possible. Si la transaction est dans la phase de données (ex.: dans le milieu du paquet DATA généré par l'appareil cible), le TT attend simplement l'exécution de ce paquet et ignore les données reçues et ne doit pas répondre avec un établissement de liaison à vitesse intégrale/basse vitesse. Le TT ne doit pas effectuer d'entrée dans l'étage de pipeline complete-split. Ce traitement entraîne une réponse NYET au complete-split correspondant sur le bus à grande vitesse.
2. Au moment où le TT décide d'annuler une transaction OUT, le TT peut choisir de ne pas émettre les paquets TOKEN ou DATA, si possible. Si le TT est au milieu du paquet DATA, il doit arrêter d'émettre les octets de données dès que possible et forcer une erreur de bourrage de bits sur le bus orienté en aval. Dans tous les cas, le TT ne doit pas effectuer d'entrée dans l'étage de pipeline complete-split. Ce traitement entraîne une réponse NYET au complete-split correspondant sur le bus à grande vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.18.6.2 Libération des start-splits en cours

Un start-split peut être mis en tampon dans l'étage de pipeline start-split qui n'est plus approprié pour entraîner une transaction à vitesse intégrale/basse vitesse sur le bus orienté en aval. Une telle transaction start-split doit être "libérée" de l'étape de pipeline start-split. Cela signifie que le start-split est simplement ignoré par le TT et le TT doit répondre à un complete-split correspondant avec un NYET. Par exemple, aucune entrée n'est effectuée dans l'étage de pipeline complete-split pour le start-split libéré.

Un start-split dans le pipeline start-split doit être libéré:

1. Si le temps d'EOF de trame à vitesse intégrale se produit, sauf pour les start-splits reçus dans $(Y\text{-}1)_7$
2. Si la transaction start-split a été reçue dans certaines microtrames (l'appeler X) et le minuteur de microtrame TT indique la microtrame X+4.

Si le TT reçoit une transaction start-split périodique dans la microtrame Y_6, son comportement est indéfini. Il s'agit d'une erreur de programmation d'hôte.

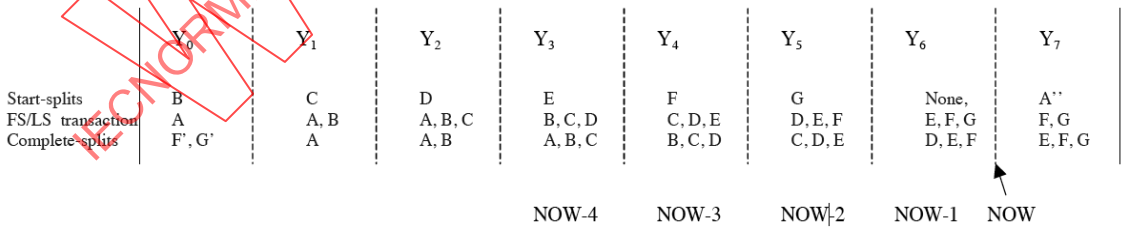
11.18.6.3 Transactions à vitesse intégrale/basse vitesse maximum par microtrame

Le TT ne doit pas démarrer une transaction à vitesse intégrale/basse vitesse s'il n'y a pas assez d'espace disponible dans l'étage de pipeline complete-split pour contenir les résultats de la transaction. S'il n'y a pas assez d'espace, le TT doit attendre pour émettre la transaction jusqu'à ce qu'il y ait suffisamment d'espace. Le nombre maximum de transactions à vitesse intégrale fonctionnant normalement qui peuvent être complétées dans une microtrame est 16.

11.18.7 Suivi des transactions TT

La Figure 11-66 montre le pipeline de microtrame TT des transactions. Les 8 microtrames à grande vitesse qui composent une trame à vitesse intégrale/basse vitesse, sont identifiées avec Y_0 via Y_7 en supposant que le minuteur de microtrame s'est produit au moment indiqué par la flèche (ex.: temps "NOW").

Comme illustré dans la figure, une transaction à grande vitesse start-split que le gestionnaire à grande vitesse reçoit dans la microtrame Y_0 (ex.: un start-split "B") peut être exécutée sur le bus à vitesse intégrale/basse vitesse pendant les temps de microtrame Y_1 ou Y_2 ou Y_3. Cette variation de démarrage sur le bus à vitesse intégrale/basse vitesse est due au bourrage de bits et à la réclamation en masse/de contrôle qui peut se produire sur le bus à vitesse intégrale/basse vitesse. Lorsque la transaction à vitesse intégrale/basse vitesse est terminée, ses transactions complete-split (si elles sont requises) sont exécutées sur le bus à grande vitesse pendant les microtrames Y_2, Y_3 ou Y_4.



Légende

Anglais	Français
Start-splits FS/LS transaction Complete-splits	Start-splits Transaction à vitesse intégrale/basse vitesse Complete-splits
NOW-4	MAINTENANT-4
NOW-3	MAINTENANT-3
NOW-2	MAINTENANT-2

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
NOW-1	MAINTENANT-1
NOW	MAINTENANT

Figure 11-66. Pipeline de microtrame

Lorsque le minuteur de microtrame indique une nouvelle microtrame, le gestionnaire à grande vitesse doit marquer les start-splits dans l'étage de pipeline start-split qu'il a reçus dans la microtrame précédente comme "en cours" de sorte qu'ils peuvent être traités sur le bus à vitesse intégrale/basse vitesse si approprié. Cela empêche les transactions à vitesse intégrale/basse vitesse d'être exécutées sur le bus en aval trop tôt.

Au début de chaque microtrame (l'appeler "NOW"), le gestionnaire à grande vitesse doit libérer (comme défini dans la Section 11.18.6.2) les transactions start-split de l'étage de pipeline start-split qui sont en cours de la microtrame NOW-4 (ou plus tôt) et les ignorent. Si la transaction est en cours sur le bus orienté en aval, la transaction doit être abandonnée (avec méthodes à vitesse intégrale/basse vitesse comme défini au Chapitre 8). Cela est décrit plus en détail dans les sections précédentes. Cela garantit que même si le bus à vitesse intégrale/basse vitesse a rencontré une condition de babillage sur le bus (ou une autre condition de retard), le TT laisse son pipeline de transaction périodique fonctionner (ex.: les transactions ne sont pas exécutées trop tard). Cela garantit également que lorsque la dernière transaction complete-split programmée est reçue par le TT, la transaction à vitesse intégrale/basse vitesse est terminée (avec succès ou abandonnée).

Finalement, au début de chaque microtrame, le gestionnaire à grande vitesse doit changer les réponses de transaction complete-split dans l'étage de pipeline complete-split de la microtrame NOW-2 sur l'état libre de sorte que leur espace peut être réutilisé pour les réponses dans cette microtrame.

Cet algorithme est illustré dans le pseudocode dans la Figure 11-67. Ce pseudocode correspond à la procédure Advance_pipeline identifiée précédemment.

```
-- Clean up start-split state in case full-/low-speed bus fell behind
while start-splits in pending state received by TT before microframe-4 loop
   Free start-split entry
End loop

-- Clean up complete-split pipeline in case no complete-splits were received
While complete-split transaction states from (microframe-2) loop
   Free complete-split response transaction entry
End loop

-- Enable full-/low-speed transactions received in previous microframe
While start-split transactions from (previous_microframe) loop
   Set start-split entry to pending status
End loop
```

Figure 11-67. Pseudocode Advance_Pipeline

11.18.8 Recherche d'état de transaction complete-split TT

Un hôte doit émettre les transactions complete-split dans une microtrame pour un ensemble de points d'extrémité à vitesse intégrale/basse vitesse dans le même ordre dans lequel les start-splits ont été émis dans une microtrame pour ce TT. Mais, les erreurs sur les start- ou complete-splits peuvent faire en sorte que le gestionnaire à grande vitesse reçoive une transaction complete-split qui ne "correspond" pas à la transaction suivante prévue conformément au pipeline de transaction du TT.

Le TT a un pipeline d'état de transaction complete-split qui attend d'être utilisé pour répondre aux transactions complete-split. Normalement, l'hôte émet le complete-split que le gestionnaire à grande vitesse attend et le complete-split correspond à l'entrée à l'avant du pipeline complete-split.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Mais, lorsque des erreurs se produisent, la transaction complete-split que le gestionnaire à grande vitesse reçoit peut ne pas correspondre à l'entrée à l'avant du pipeline complete-split. Cela peut se produire par exemple, lorsqu'un start-split est endommagé sur le bus à grande vitesse et le gestionnaire à grande vitesse ne le reçoit pas avec succès. Ou le gestionnaire à grande vitesse peut avoir une correspondance, mais l'entrée correspondante est située après l'état pour les autres complete-splits prévus que le gestionnaire à grande vitesse n'a pas reçus (en raison des erreurs complete-split sur le bus à grande vitesse).

Le gestionnaire à grande vitesse doit répondre à une transaction complete-split avec les résultats d'une transaction à vitesse intégrale/basse vitesse qu'il a terminée. Cela signifie que le gestionnaire à grande vitesse doit chercher à trouver l'entrée de suivi d'état correcte parmi plusieurs entrées de réponse complete-split possibles. Cette recherche prend du temps. Le gestionnaire à grande vitesse doit seulement rechercher les réponses complete-split accumulées pendant la microtrame précédente. Il doit seulement y avoir au maximum 1 microtrame d'entrées de réponse complete-split; la microtrame des réponses qui se sont déjà accumulées et attendent d'être retournées via les complete-splits à grande vitesse.

Le protocole de transaction fractionnée est défini pour permettre au gestionnaire à grande vitesse de temporiser la première transaction complete-split à grande vitesse tandis qu'il recherche la réponse correcte. Cela laisse du temps au gestionnaire à grande vitesse pour terminer sa recherche et pour répondre correctement au complete-split suivant (réessayé).

Les figures de séquence de flux d'interruption et isochrone suivantes montrent ces cas avec les transitions identifiées "Recherche pas terminée à temps" et "Aucune réponse fractionnée trouvée".

Le gestionnaire à grande vitesse associe la transaction complete-split à l'entrée correcte dans l'étage de pipeline complete-split et avance le pipeline de manière appropriée. Il existe cinq cas que le TT doit traiter correctement:

1. Avec le jeton complete-split à grande vitesse et la première entrée de la correspondance de pipeline complete-split, le gestionnaire à grande vitesse répond avec les données/le statut indiqué(es). Ce cas se produit la première fois que le TT reçoit un complete-split.
2. Identique à ci-dessus, mais il s'agit d'une nouvelle tentative d'un complete-split que le TT a déjà reçu comme le contrôleur hôte ne recevait pas les informations de réponse (précédentes).
3. Si la transaction complete-split correspond à d'autres entrées dans le pipeline complete-split en plus de la première, le gestionnaire à grande vitesse avance le pipeline complete-split (ex.: libère les informations de réponse pour les entrées complete-split précédentes) et répond avec les informations pour l'entrée correspondante. Ce cas peut se produire en raison de transactions complete-split précédentes normales ou ratées. Un exemple de cas anormal pourrait être que le contrôleur hôte n'a pas réussi à émettre une transaction complete-split vers le gestionnaire à grande vitesse et a effectué un traitement d'arrêt de point d'extrémité pour ce point d'extrémité. Cela signifie que le complete-split suivant ne correspond pas à la première entrée de l'étage de pipeline complete-split.
4. Le gestionnaire à grande vitesse peut également recevoir un complete-split avant qu'il ait démarré une transaction à vitesse intégrale/basse vitesse. S'il n'y a pas d'entrée dans le pipeline complete-split, le gestionnaire à grande vitesse répond avec un établissement de liaison NYET pour informer l'hôte qu'il n'a pas d'informations de statut. Lorsque l'hôte émet le dernier complete-split programmé pour ce point d'extrémité pour cette trame, il doit interpréter le NYET comme une condition d'erreur. Cela stimule le traitement des erreurs normal "trois essais". S'il y a eu plus que trois erreurs, l'hôte arrête ce point d'extrémité. S'il y a eu moins de trois erreurs, l'hôte continue de traiter les transactions programmées de ce point d'extrémité (ex.: un start-split est émis comme la transaction suivante pour ce point d'extrémité au prochain moment programmé pour ce point d'extrémité). Il faut noter qu'une réponse NYET est possible dans ce cas en raison d'une erreur de transaction sur le start-split ou d'une erreur de programmation de l'hôte (ou du TT).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

5. Le gestionnaire à grande vitesse peut temporiser sa première transaction complete-split à grande vitesse alors qu'il cherche l'étage de pipeline complete-split pour une entrée correspondante. Mais, le gestionnaire à grande vitesse doit répondre correctement à la transaction complete-split suivante. Si le gestionnaire à grande vitesse ne répond pas correctement pour un IN d'interruption après avoir accepté la transaction à vitesse intégrale/basse vitesse, le logiciel de point d'extrémité et l'appareil peuvent perdre la synchronisation des données et des erreurs plus catastrophiques peuvent se produire.

Le contrôleur hôte doit émettre les transactions complete-split dans le même ordre que les transactions start-split correspondantes.

11.19 Espace tampon TT approximatif requis

Un traducteur de transactions nécessite un espace tampon et de suivi d'état pour ses portions périodiques et non périodiques.

Le pipeline de microtrame TT nécessite moins de:

- 752 octets de données pour l'étage start-split
- 2x 188 octets de données pour l'étage complete-split
- 16x 4x statut de transaction (<4 octets pour chaque transaction) pour l'étage start-split
- 16x 2x statut de transaction (<4 octets pour chaque transaction) pour l'étage complete-split

Il y a, au maximum, 4 microtrames de mise en tampon requises pour l'étage start-split du pipeline et, au maximum, 2 microtrames de mise en tampon pour l'étage complete-split du pipeline. Il y a, au maximum, 16 transactions à vitesse intégrale (dimension minimum) possibles dans une microtrame.

La portion non périodique du TT nécessite au moins:

- 2x (64 données + 4 statuts de transaction) octets

Différentes mises en œuvre peuvent nécessiter plus ou moins d'espace de mise en tampon et de suivi d'état.

11.20 Présentation de la traduction de transactions d'interruption

Les figures des séquences de flux et des diagrammes d'états montrent les transitions requises pour les transactions fractionnées à grande vitesse pour les types de transfert d'interruption à vitesse intégrale/basse vitesse pour un seul point d'extrémité. Ces figures ne doivent pas être interprétées comme illustrant une synchronisation spécifique particulière. En particulier, les transactions à grande vitesse ou à vitesse intégrale/basse vitesse pour les autres points d'extrémité peuvent se produire avant ou après ces transactions fractionnées. Des détails spécifiques sont décrits si approprié.

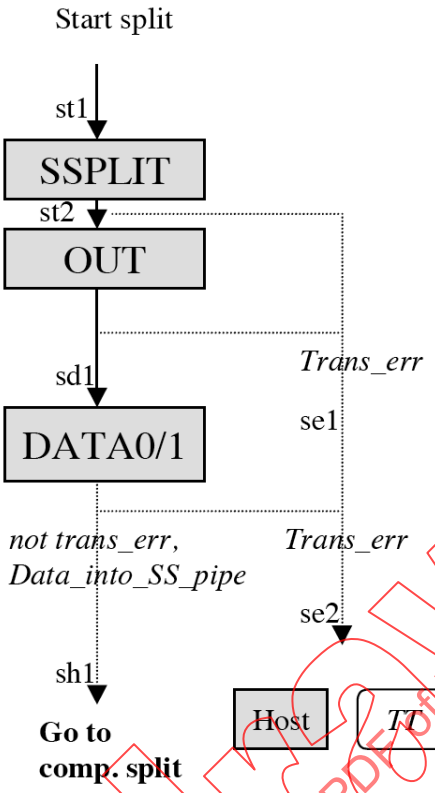
Contrairement au traitement en masse/de contrôle, le gestionnaire à vitesse intégrale/basse vitesse ne doit pas effectuer de traitement de nouvelle tentative local sur le bus à vitesse intégrale/basse vitesse en réponse à une erreur de transaction pour les transactions d'interruption à vitesse intégrale/basse vitesse.

11.20.1 Séquences de transactions fractionnées d'interruption

Les figures des séquences de flux IN et OUT d'interruption utilisent la même notation et ont des descriptions similaires aux figures en masse/de contrôle.

Contrairement au traitement en masse/de contrôle, le gestionnaire à vitesse intégrale ne doit pas effectuer de traitement de nouvelle tentative local sur le bus à vitesse intégrale en réponse à une erreur de transaction (notamment la temporisation) d'une transaction d'interruption.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

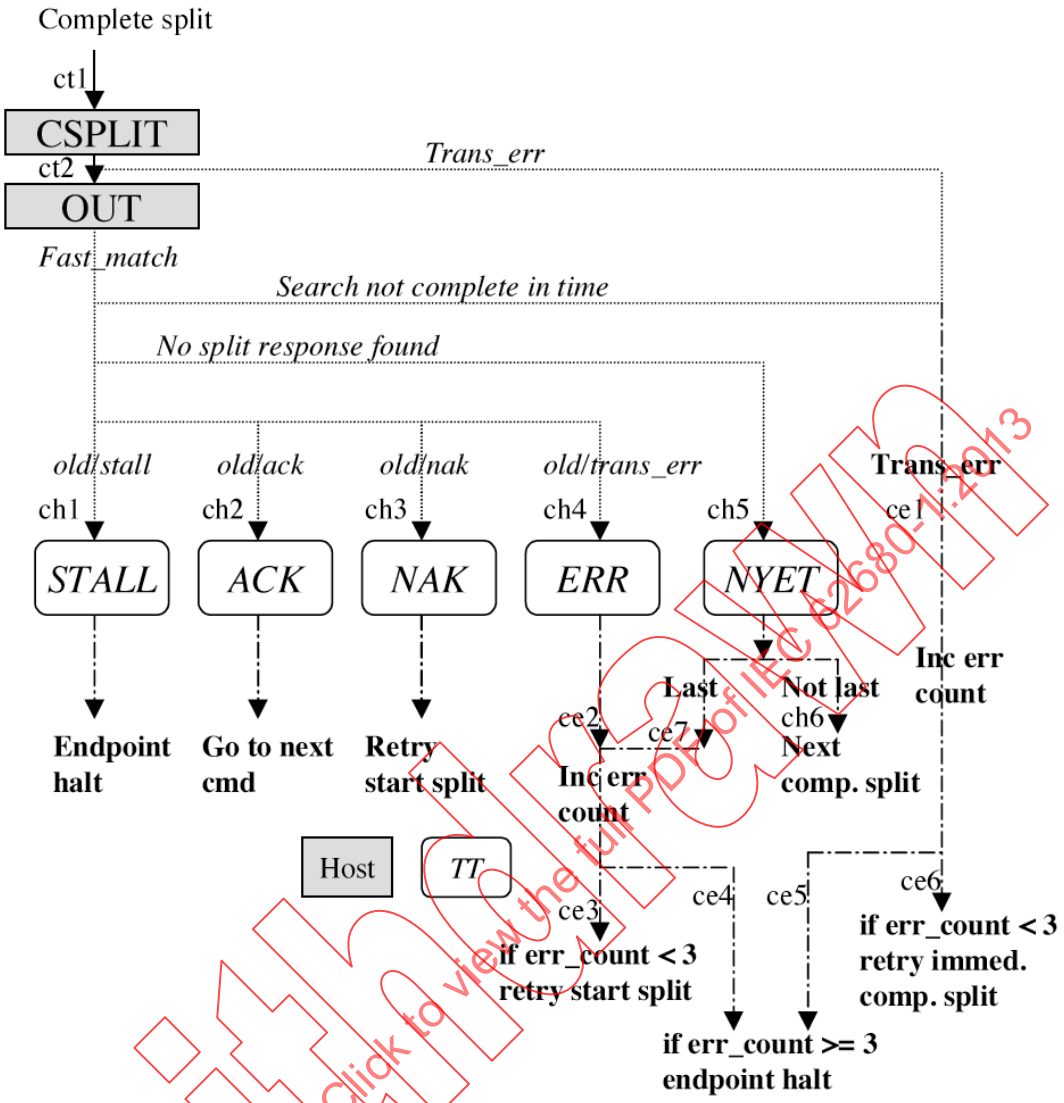


Légende

Anglais	Français
Go to comp. split	Aller à comp. Split
Host	Hôte

Figure 11-68. Séquence de transaction start-split OUT d'interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

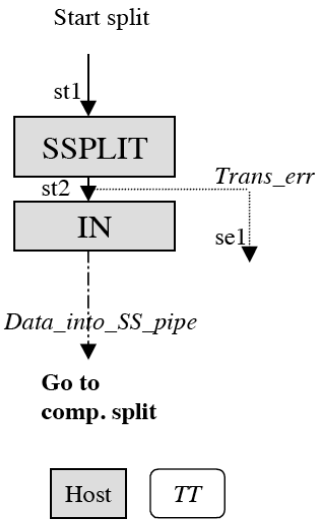


Légende

Anglais	Français
Search not complete in time	Recherche pas terminée à temps
No split response found	Pas de réponse fractionnée trouvée
old/stall	ancien/arrêt
old/nak	ancien/nak
old/ack	ancien/ack
old/trans_err	ancien/trans_err
Endpoint halt	Arrêt point d'extrémité
Go to next cmd	Aller à cmd suivante
Retry start split	Réessayer start split
Last	Dernier
Inc err count	Inc décompte err
Not last	Pas le dernier
Next comp. split	Comp. split suivant
if err_count < 3 retry start split	si err_count < 3 réessayer start split
if err_count >= 3 endpoint halt	si err_count >= 3 arrêt point d'extrémité
if err_count < 3 retry immed. compl. split	si err_count < 3 réessayer imméd. compl. split
Complete split	Complete split

Figure 11-69. Séquence de transaction complete-split OUT d'interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

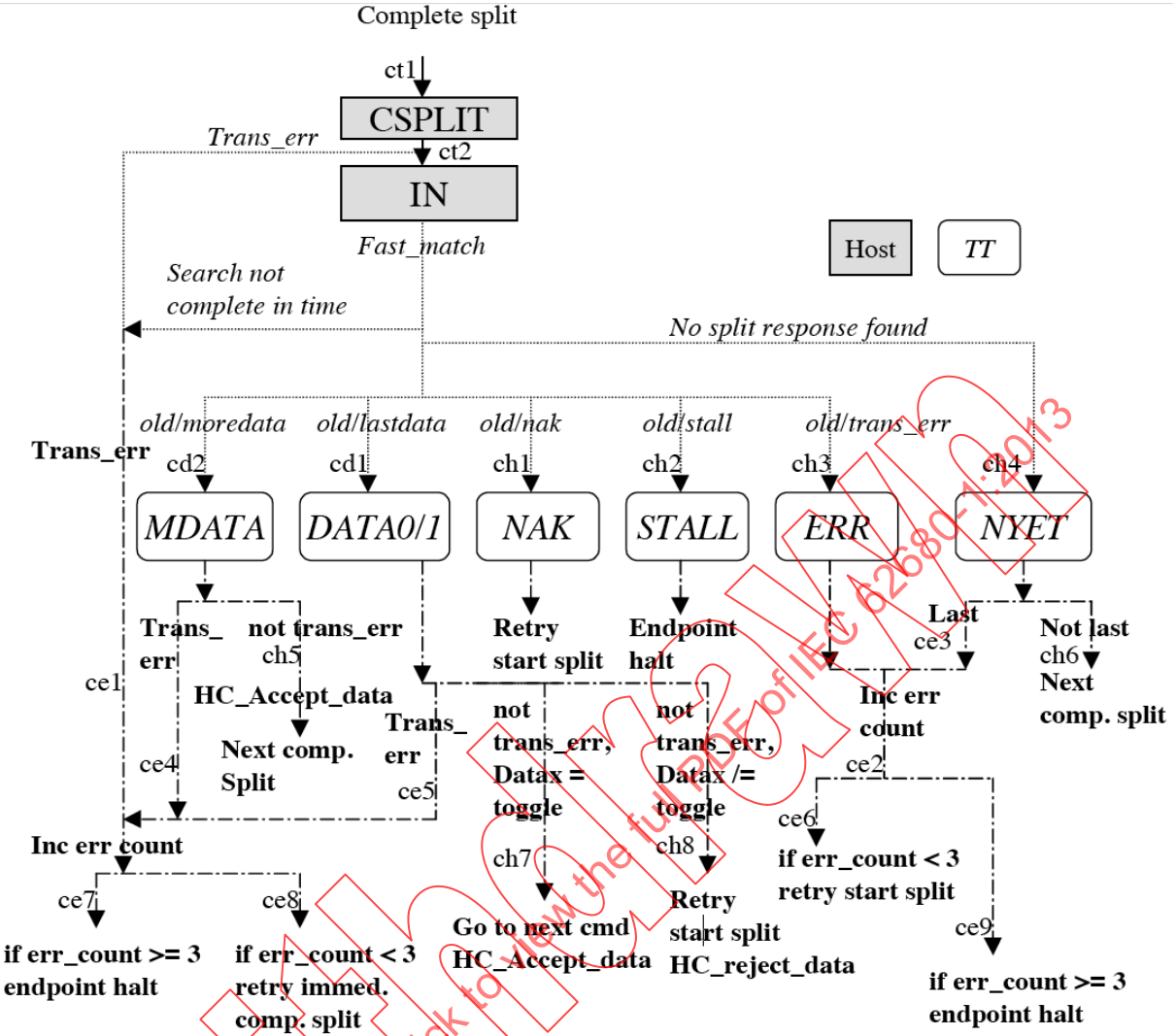


Légende

Anglais	Français
Go to comp. split	Aller à comp. Split
Host	Hôte

Figure 11-70. Séquence de transaction start-split IN d'interruption

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
if err_count < 3 retry start split	si err_count < 3 réessayer start split
if err_count >= 3 endpoint halt	si err_count >= 3 arrêt point d'extrémité
if err_count < 3 retry immed. compl. split	si err_count < 3 réessayer imméd. compl. split
Go to next cmd	Aller à cmd suivante
Retry start split	Réessayer start split
Inc err count	Inc décompte err
Last	Dernier
Not last	Pas le dernier
Next comp. split	Comp. split suivant
Endpoint halt	Arrêt point d'extrémité
Retry start split	Réessayer start split
not trans_err, Datax = toggle	pas trans_err, Datax = basculement
not trans_err, Datax /= toggle	pas trans_err, Datax /= basculement
Next comp. Split	Comp. split suivant
not trans_err	pas trans_err
old/moredata	ancien/plus de données
old/lastdata	ancien/dernières données
old/nak	ancien/nak
old/stall	ancien/stall
old/trans_err	ancien/trans_err
No split response found	Pas de réponse fractionnée trouvée
Search not complete in time	Recherche pas terminée à temps
Complete split	Complete split

Figure 11-71. Séquence de transaction complete-split IN d'interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.20.2 Diagrammes d'états des transactions fractionnées d'interruption

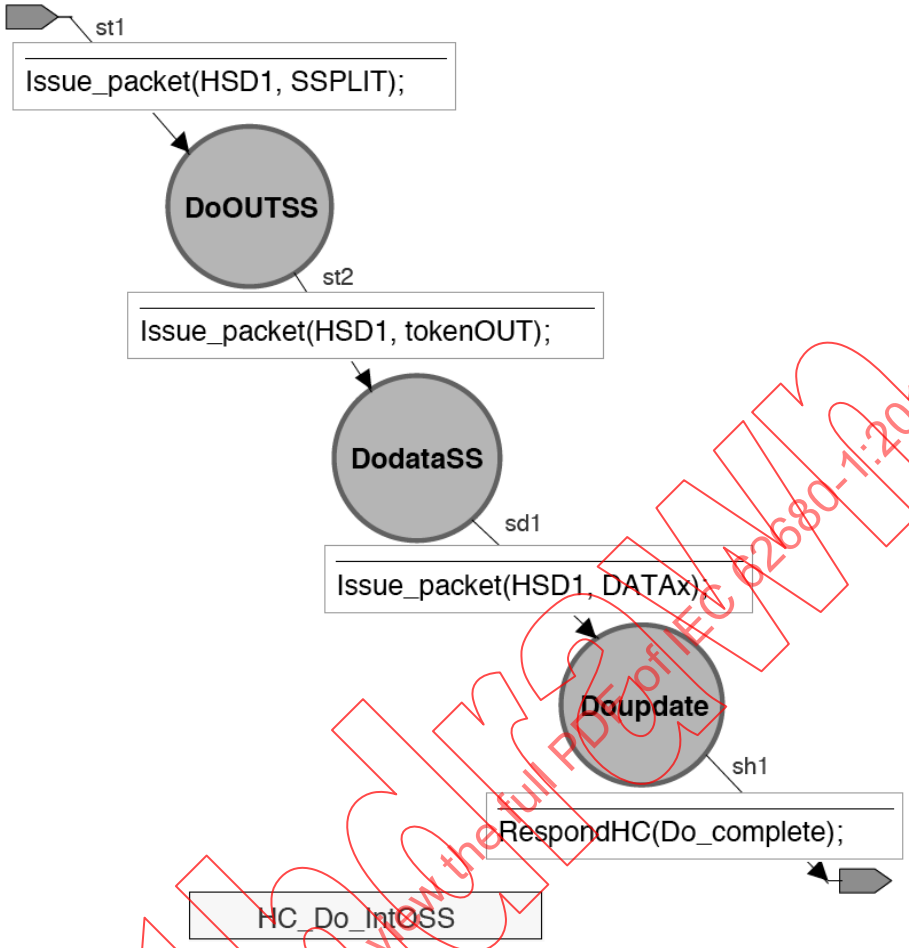
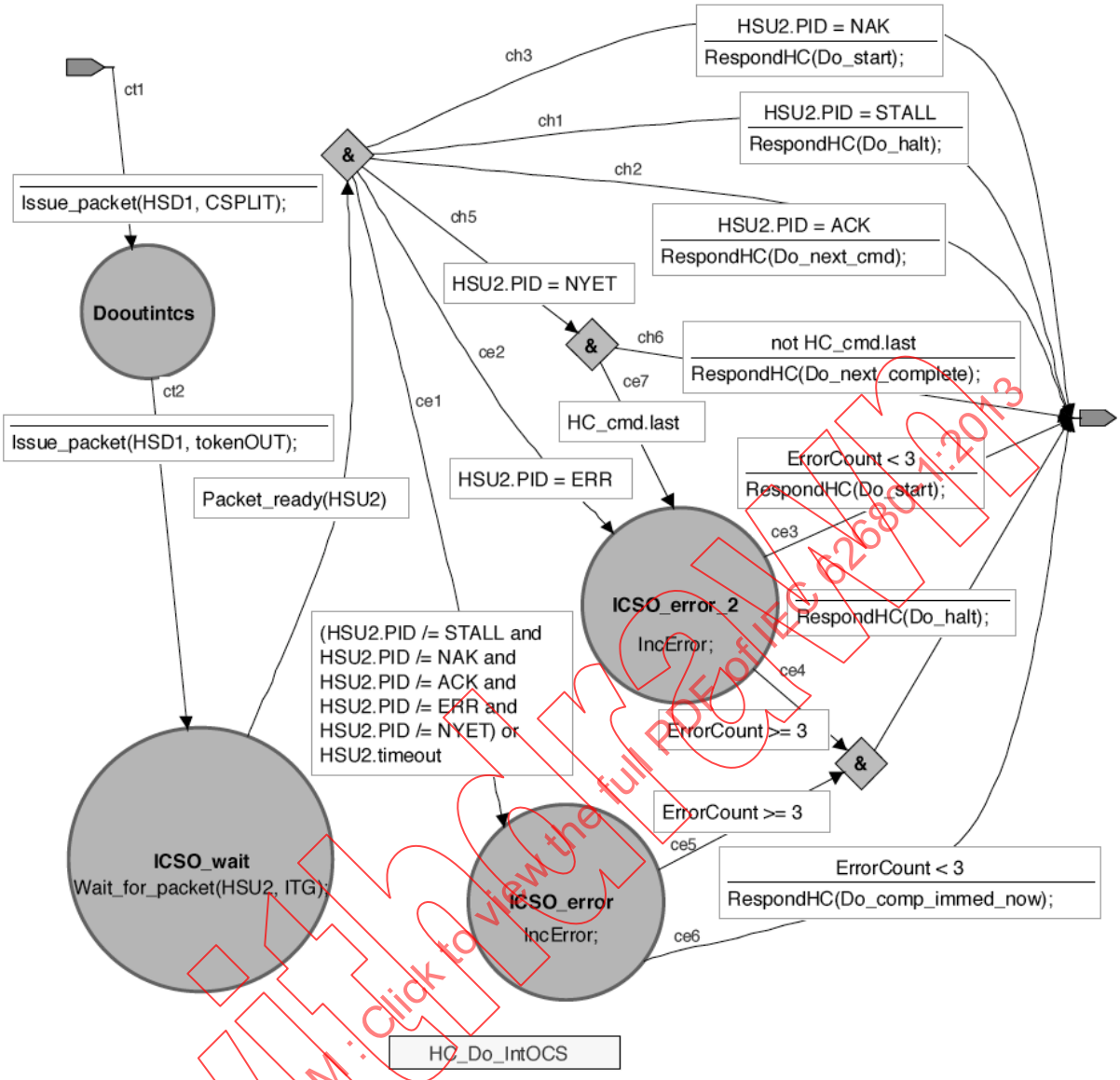


Figure 11-72. Diagramme d'états de l'hôte de transaction start-split OUT d'interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

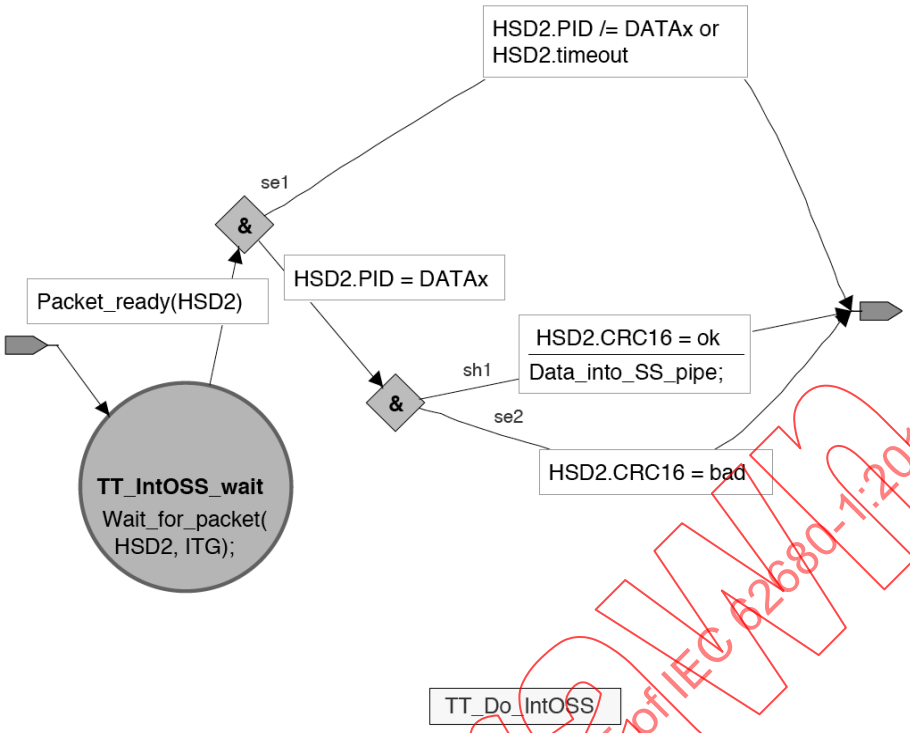


Légende

Anglais	Français
(HSU2.PID /= STALL and HSU2.PID /= NAK and HSU2.PID /= ACK and HSU2.PID /= ERR and HSU2.PID /= NYET) or HSU2.timeout	(HSU2.PID /= STALL et HSU2.PID /= NAK et HSU2.PID /= ACK et HSU2.PID /= ERR et HSU2.PID /= NYET) ou HSU2.timeout

Figure 11-73. Diagramme d'états de l'hôte de transaction complete-split OUT d'interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

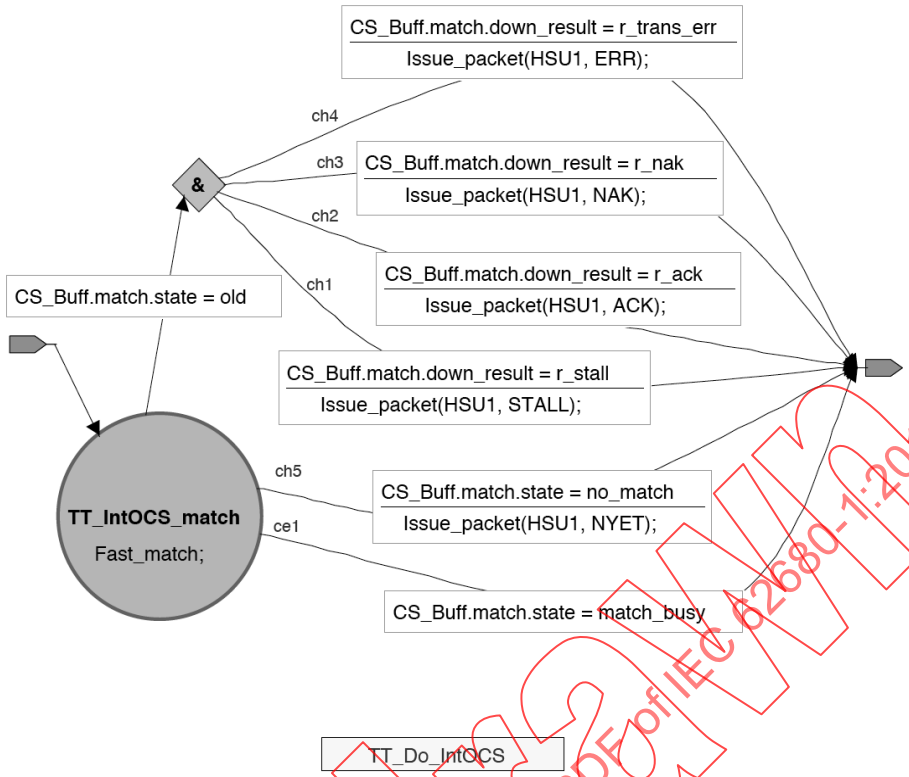


Légende

Anglais	Français
HSD2.PID/= DATAx or HSD2.timeout	HSD2.PID/= DATAx ou HSD2.timeout
HSD2.CRC16 = bad	HSD2.CRC16 = incorrect

Figure 11-74. Diagramme d'états du TT de transaction start-split OUT d'interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
CS_Buff.match.state = old	CS_Buff.match.state = ancien

Figure 11-75. Diagramme d'états du TT de transaction complete-split OUT d'interruption

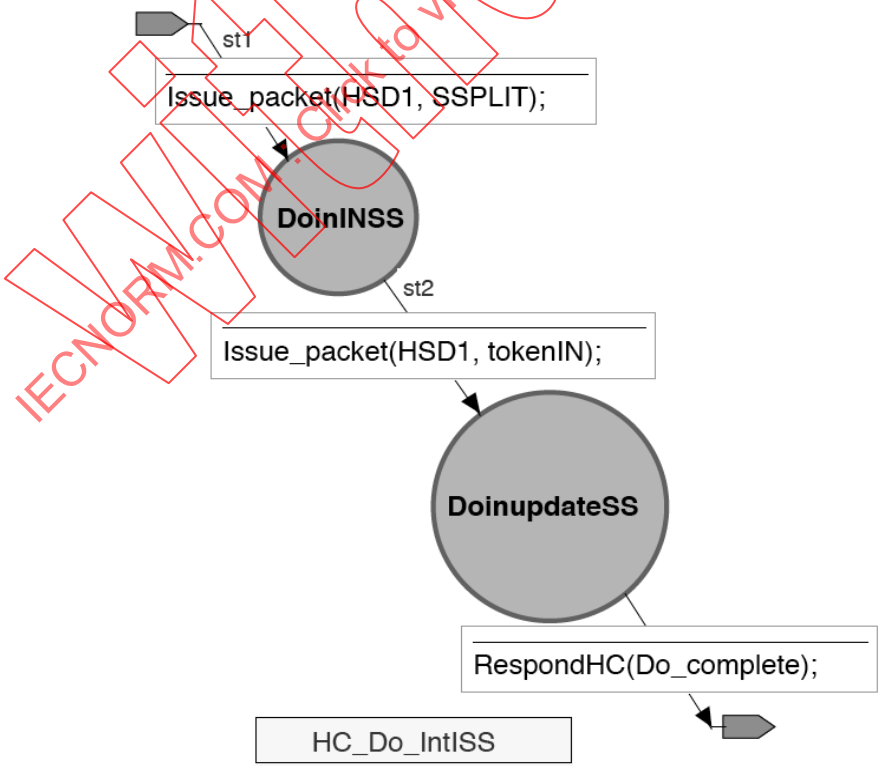
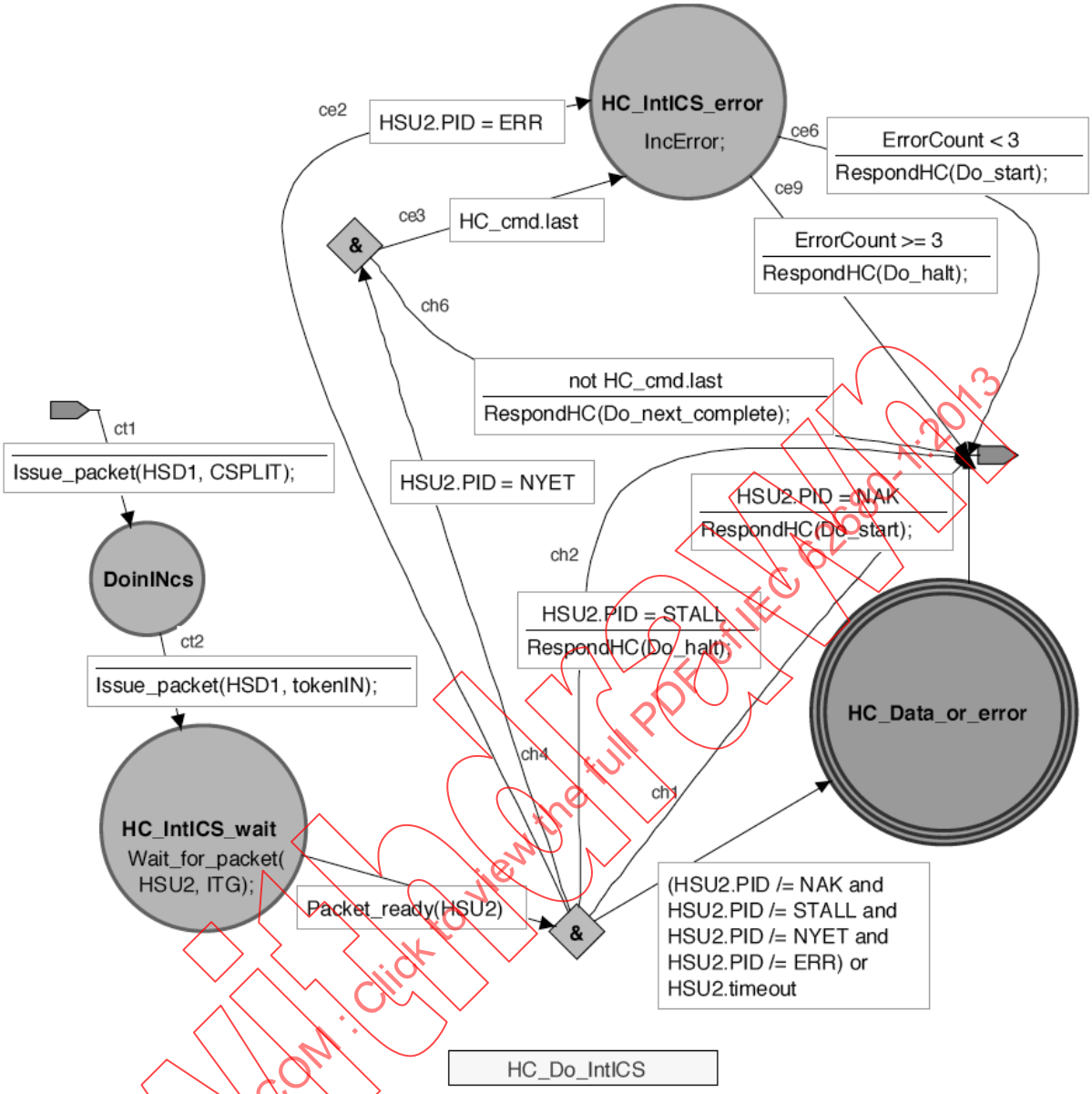


Figure 11-76. Diagramme d'états de l'hôte de transaction start-split IN d'interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

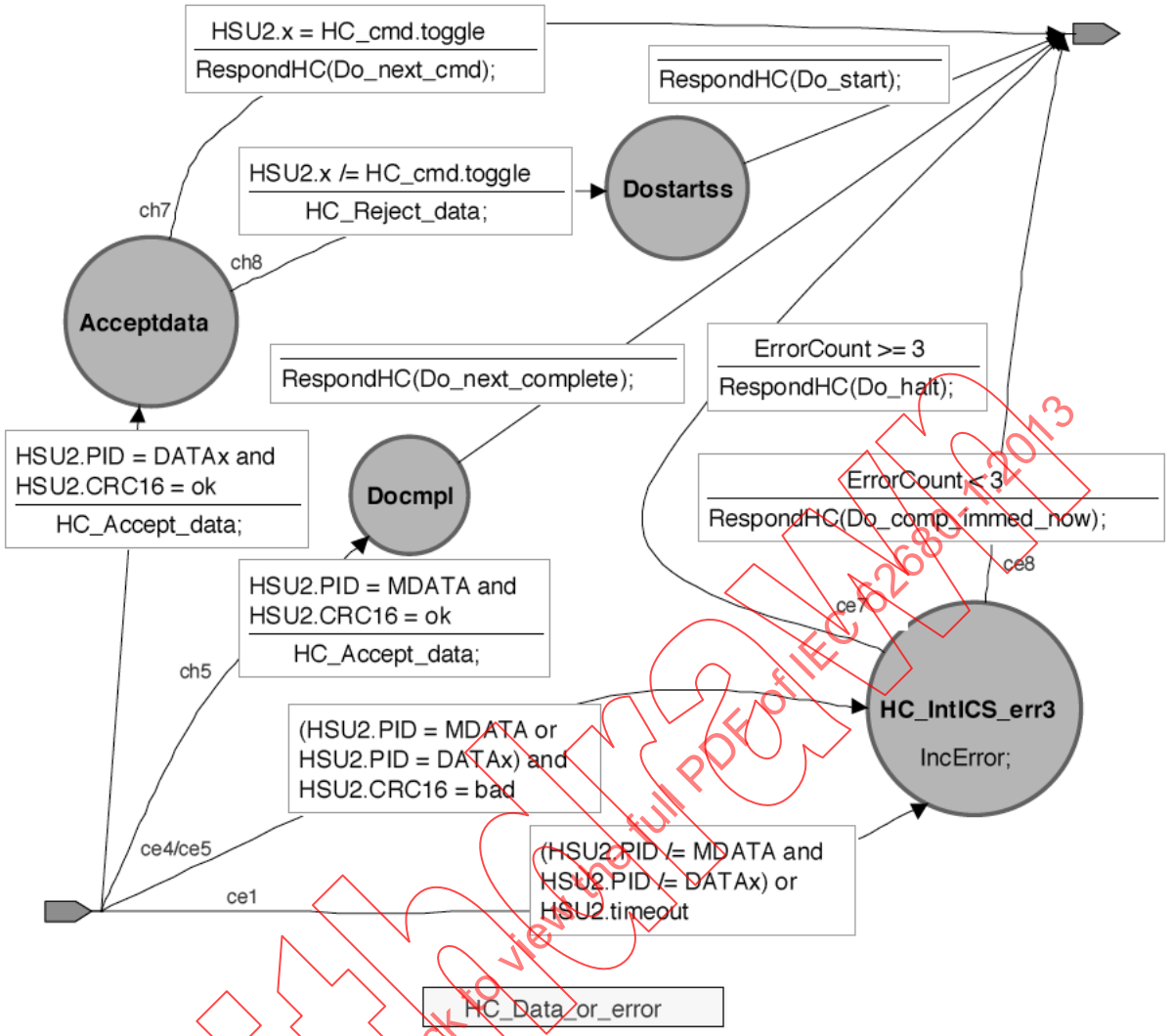


Légende

Anglais	Français
(HSU2.PID /= NAK and HSU2.PID /= STALL and HSU2.PID /= NYET and HSU2.PID /= ERR) or HSU2.timeout	(HSU2.PID /= NAK et HSU2.PID /= STALL et HSU2.PID /= NYET et HSU2.PID /= ERR) ou HSU2.timeout

Figure 11-77. Diagramme d'états de l'hôte de transaction complete-split IN d'interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
(HSU2.PID = MDATA or HSU2.PID = DATAx) and HSU2.CRC16 = bad	(HSU2.PID = MDATA ou HSU2.PID = DATAx) et HSU2.CRC16 = incorrect
(HSU2.PID /= MDATA and HSU2.PID /= DATAx) or HSU2.timeout	(HSU2.PID /= MDATA et HSU2.PID /= DATAx) ou HSU2.timeout

Figure 11-78. Diagramme d'états HC_Data_or_Error

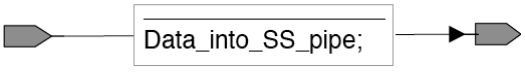
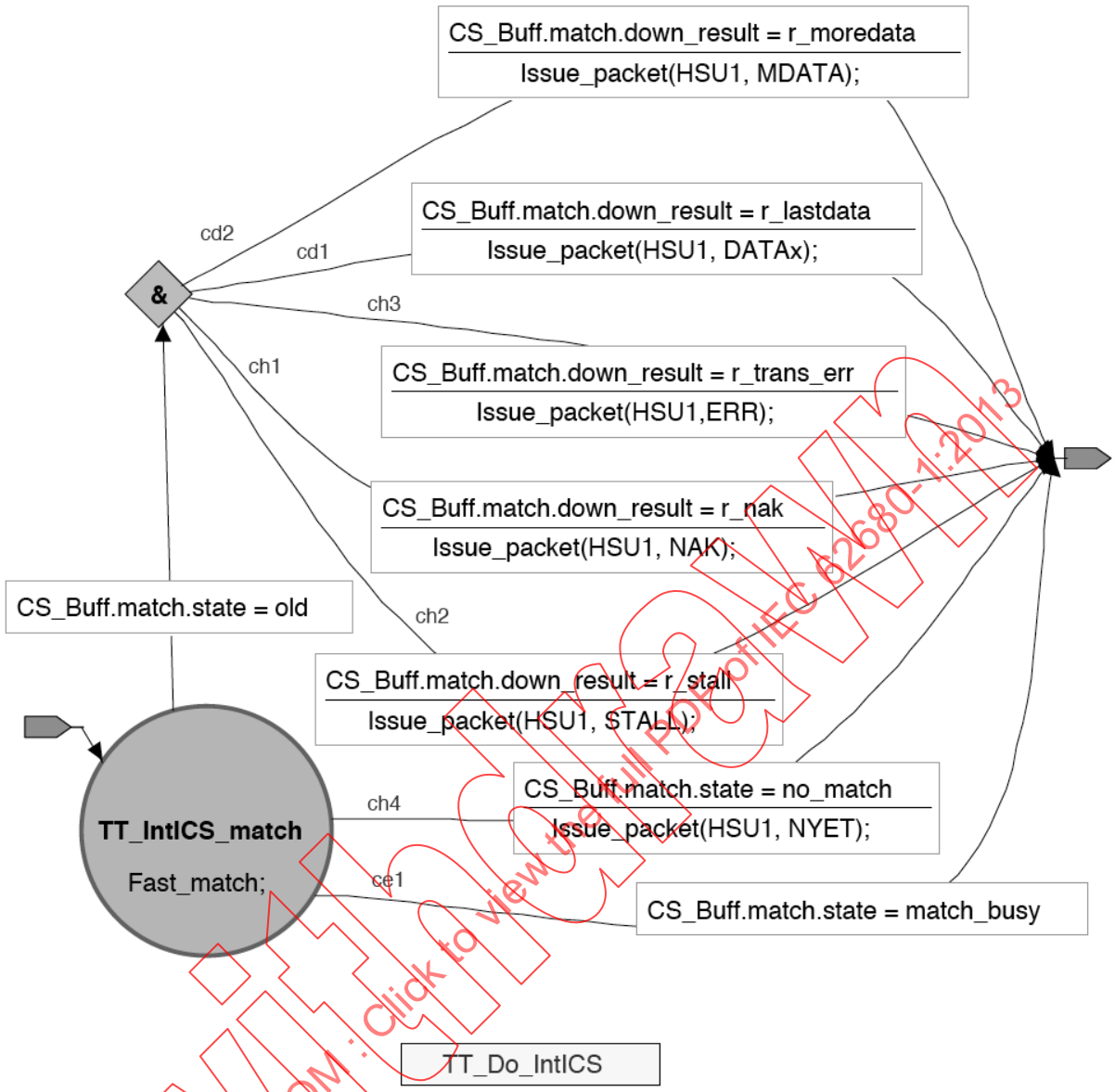




Figure 11-79. Diagramme d'états du TT de transaction start-split IN d'interruption

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
CS_Buff.match.state = old	CS_Buff.match.state = ancien

Figure 11-80. Diagramme d'états du TT de transaction complete-split IN d'interruption

11.20.3 Séquençage OUT d'interruption

Les transactions fractionnées OUT d'interruption sont programmées par le contrôleur hôte comme des transactions à grande vitesse normales avec les start- et complete-splits programmés comme décrit précédemment.

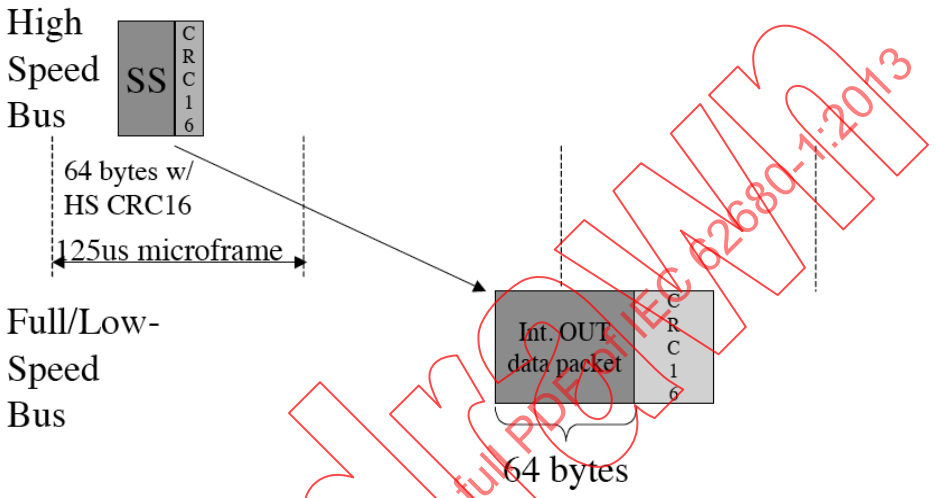
Lorsqu'il y a plusieurs transactions à vitesse intégrale/basse vitesse allouées pour une microtrame donnée, elles sont enregistrées par le gestionnaire à grande vitesse dans le TT dans l'étage de pipeline start-split. Les start-splits sont enregistrés dans l'ordre dans lequel ils sont reçus jusqu'à la fin de la microtrame. A la fin de la microtrame, ces transactions sont disponibles pour être émises par le gestionnaire à vitesse intégrale/basse vitesse sur le bus à vitesse intégrale/basse vitesse dans l'ordre dans lequel elles sont reçues.

Dans une microtrame suivante (comme décrit précédemment), le gestionnaire à vitesse intégrale/basse vitesse émet les transactions qui ont été enregistrées dans l'étage de pipeline

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

start-split sur le bus à vitesse intégrale/basse vitesse orienté en aval. Certaines transactions peuvent être laissées d'une microtrame précédente comme le programme à grande vitesse a été conçu en supposant que le meilleur bourrage de bits et les transactions à vitesse intégrale/basse vitesse pourraient durer plus longtemps sur le bus à vitesse intégrale/basse vitesse. Comme le gestionnaire à vitesse intégrale/basse vitesse émet des transactions sur le bus à vitesse intégrale/basse vitesse orienté en aval, il enregistre les résultats dans l'étage de pipeline complete-split périodique puis passe à la transaction suivante dans le pipeline start-split.

Dans une microtrame suivante (comme décrit précédemment), le contrôleur hôte émet une transaction complete-split à grande vitesse et le TT répond de manière appropriée.



Légende

Anglais	Français
High Speed Bus	Bus à grande vitesse
64 bytes w/ HS CRC16	64 octets avec HS CRC16
125µs microframe	Microtrame 125 µs
Full/Low-Speed Bus	Bus à vitesse intégrale/basse vitesse
Int. OUT data packet	Paquet de données OUT d'int.
64 bytes	64 octets

Figure 11-81. Exemple de traitement CRC16 pour OUT d'interruption

La transaction start-split pour une transaction OUT d'interruption inclut un champ CRC16 normal pour le paquet de données à grande vitesse de la phase de données de la transaction start-split. Mais, la charge utile de données du paquet de données contient uniquement la charge utile de données du paquet de données à vitesse intégrale/basse vitesse correspondant; à savoir, il y a un seul CRC16 dans le paquet de données de la transaction start-split. Le gestionnaire à grande vitesse TT doit contrôler le CRC sur le start-split et ignorer le start-split en cas de défaillance dans le contrôle CRC du paquet de données. Si le start-split a une défaillance de contrôle CRC, la transaction à vitesse intégrale ne doit pas être démarrée sur le bus en aval. La Figure 11-81 montre un exemple de traitement CRC16 pour une transaction OUT d'interruption et son start-split.

11.20.4 Séquençage IN d'interruption

Lorsque le gestionnaire à grande vitesse reçoit une transaction start-split d'interruption, il enregistre le paquet dans l'étage de pipeline start-split. De cette manière, il accumule un certain nombre de transactions start-split pour une microtrame suivante.

Au début de la microtrame suivante (comme décrit précédemment), ces transactions sont disponibles pour être émises par le gestionnaire à vitesse intégrale/basse vitesse sur le bus à vitesse intégrale/basse vitesse en aval dans l'ordre dans lequel elles ont été enregistrées dans l'étage de pipeline start-split. Le gestionnaire à vitesse intégrale/basse vitesse émet chaque transaction sur le bus orienté en aval. Le gestionnaire à vitesse intégrale/basse

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

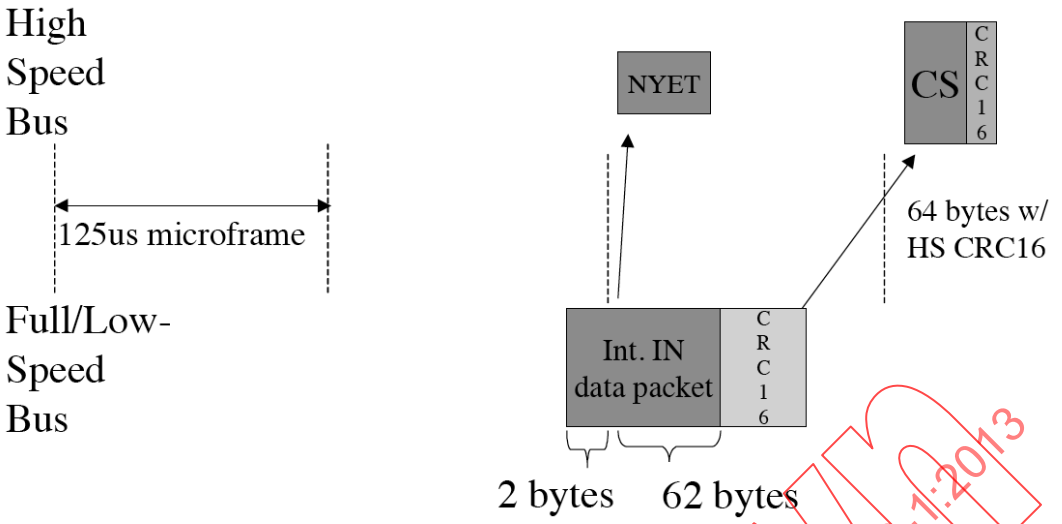
vitesse répond à la transaction à vitesse intégrale/basse vitesse avec un établissement de liaison approprié comme décrit au Chapitre 8. Le gestionnaire à vitesse intégrale/basse vitesse enregistre les résultats de la transaction (données, NAK, STALL, trans_err) dans l'étage de pipeline complete-split.

Pendant les microtrames suivantes, le contrôleur hôte émet des transactions complete-split à grande vitesse pour récupérer les données/l'établissement de liaison du gestionnaire à grande vitesse. Lorsque le gestionnaire à grande vitesse reçoit une transaction complete-split, le TT retourne les données qu'il a reçues pendant une microtrame. Si la transaction à vitesse intégrale/basse vitesse a été démarrée et terminée dans une seule microtrame, le TT retourne toutes les données pour la transaction dans la réponse complete-split se produisant dans la microtrame suivante. Si le contrôle CRC à vitesse intégrale/basse vitesse réussit, le PID DATA0/1 approprié pour le paquet de données est utilisé. Si le contrôle CRC à vitesse intégrale/basse vitesse échoue, un établissement de liaison ERR est utilisé et il n'y a pas de paquet de données dans le cadre de la transaction complete-split.

Si la transaction à vitesse intégrale/basse vitesse s'est étendue à une microtrame, le TT nécessite deux complete-splits (dans deux microtrames suivantes) pour retourner toutes les données pour la transaction à vitesse intégrale/basse vitesse. Le PID de paquet de données pour le premier complete-split doit être un MDATA pour indiquer au contrôleur hôte qu'un autre complete-split est requis pour ce point d'extrémité. Cette réponse MDATA est donnée sans effectuer de contrôle CRC (car le champ CRC16 n'a pas encore été reçu sur le bus à vitesse intégrale/basse vitesse). Le complete-split dans la microtrame suivante doit utiliser un PID DATA0/1 si le contrôle CRC réussit. Si le contrôle CRC échoue, une réponse d'établissement de liaison ERR est donnée et il n'y a pas de paquet de données dans le cadre de la transaction complete-split. Comme les transactions d'interruption à vitesse intégrale sont limitées à 64 octets de données ou moins (et les transactions d'interruption à basse vitesse sont limitées à 8 octets de données ou moins), aucune transaction d'interruption à vitesse intégrale/basse vitesse ne peut s'étendre sur plus d'une limite de microtrame, à savoir, deux microtrames au maximum doivent être présentes pour réaliser la transaction.

La transaction complete-split pour une transaction IN d'interruption ne doit pas inclure le champ CRC16 reçu du paquet de données à vitesse intégrale/basse vitesse (à savoir, seul un champ CRC16 à grande vitesse est utilisé dans les transactions fractionnées). Le TT doit utiliser un CRC16 à grande vitesse sur chaque paquet de données complete-split. Si le gestionnaire à vitesse intégrale détecte un contrôle CRC qui a échoué, il doit utiliser une réponse d'établissement de liaison ERR dans la transaction complete-split pour refléter cette erreur sur le contrôleur hôte à grande vitesse. Le contrôleur hôte doit vérifier le CRC16 sur chaque paquet de données complete-split retourné. Une défaillance CRC (ou un établissement de liaison ERR) sur un complete-split (partiel) est reflétée sous forme de défaillance CRC sur la transaction à vitesse intégrale/basse vitesse totale. Cela signifie que pour un cas où une interruption à vitesse intégrale/basse vitesse s'étend sur une limite de microtrame, le contrôleur hôte peut accepter le premier complete-split sans erreurs, puis le second complete-split peut indiquer que les données du premier complete-split doivent être rejetées comme si elles n'avaient jamais été reçues par le contrôleur hôte. La Figure 11-82 montre un exemple d'un IN d'interruption et de son traitement CRC16 avec les réponses complete-split correspondantes.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
High Speed Bus	Bus à grande vitesse
64 bytes w/ HS CRC16	64 octets avec HS CRC16
125µs microframe	Microtrame 125 µs
Full/Low-Speed Bus	Bus à vitesse intégrale/basse vitesse
Int. IN data packet	Paquet de données IN d'int.
64 bytes	64 octets
2 bytes	2 octets

Figure 11-82. Exemple de traitement CRC16 pour IN d'interruption

11.21 Présentation de la traduction de transactions isochrones

Les transactions fractionnées isochrones sont traitées par l'hôte en programmant les transactions start- et complete-split comme décrit précédemment. Les transactions fractionnées IN isochrones ont plus de deux entrées de programme:

- Une entrée pour la transaction start-split dans la microtrame avant la transaction à vitesse intégrale qui peut se produire le plus tôt
- D'autres entrées pour les complete-splits dans les microtrames après que les données peuvent apparaître sur le bus à vitesse intégrale (similaire à la programmation d'un IN d'interruption)

De plus, les transactions isochrones sont fractionnées en éléments de la taille de microtrames; ex.: une transaction à vitesse intégrale de 300 octets est budgétisée pour que de multiples transactions fractionnées à grande vitesse se déplacent vers/du TT. Cela permet l'alignement des données pour chaque microtrame.

Les transactions OUT isochrones à vitesse intégrale émises par un TT n'ont pas de transactions complete-split correspondantes. Elles doivent seulement avoir des transactions start-split.

Le contrôleur hôte doit conserver le même ordre pour les transactions complete-split (comme pour les transactions start-split) pour le traitement IN.

Les IN isochrones ont des transactions start- et complete- split. La "première" transaction fractionnée à grande vitesse pour un point d'extrémité à vitesse intégrale est toujours une transaction start-split et la seconde (et les autres si nécessaire) est toujours une transaction complete-split quelle que soit la réponse du gestionnaire à grande vitesse du TT.

Le gestionnaire à vitesse intégrale/basse vitesse recombine les données OUT dans ses tampons locaux pour recréer la seule transaction de données à vitesse intégrale et traiter les cas d'erreur de microtrame. Le gestionnaire à vitesse intégrale/basse vitesse fractionne les données de réponse IN sur les limites de microtrame.

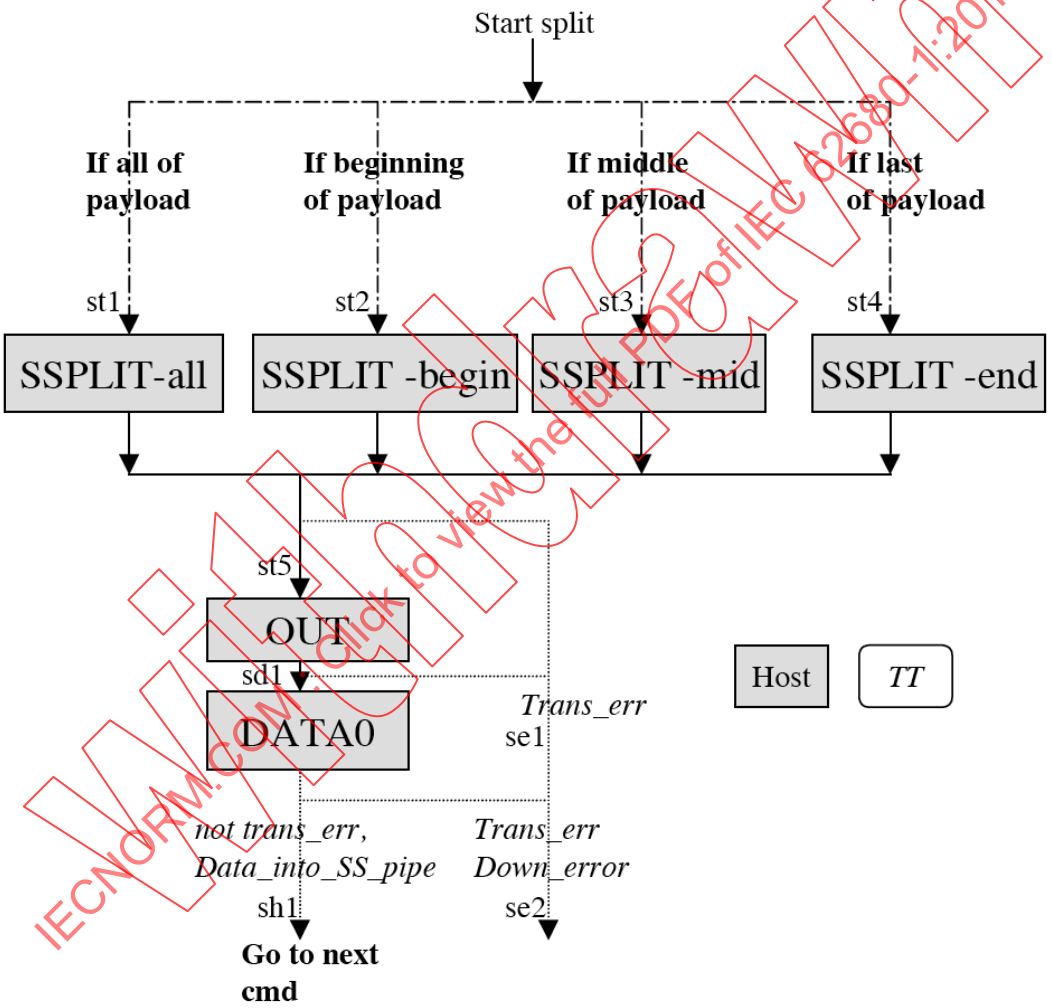
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Les tampons de microtrame avancent toujours quelles que soient les interactions avec le contrôleur hôte ou le gestionnaire à vitesse intégrale.

11.21.1 Séquences de transactions fractionnées isochrones

Les figures des séquences de flux et des diagrammes d'états montrent les transitions requises pour les transactions fractionnées à grande vitesse d'un type de transfert isochrone à vitesse intégrale pour un seul point d'extrémité. Ces figures ne doivent pas être interprétées comme une synchronisation spécifique particulière. Notamment, les transactions à grande vitesse ou à vitesse intégrale des autres points d'extrémité peuvent se produire avant ou après ces transactions fractionnées. Des détails spécifiques sont décrits si appropriés.

Contrairement au traitement en masse/de contrôle, le gestionnaire à vitesse intégrale ne doit pas effectuer de traitement de nouvelle tentative local sur le bus à vitesse intégrale en réponse à des erreurs de transaction (dont la temporisation) d'une transaction isochrone.



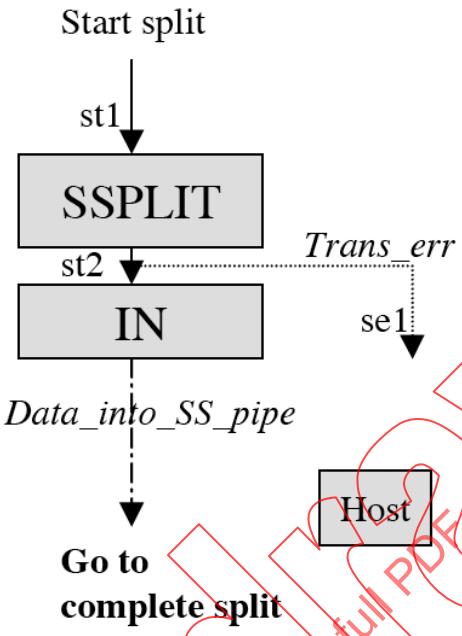
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Légende

Anglais	Français
If all of payload	Si tout de la charge utile
If beginning of payload	Si début de la charge utile
If middle of payload	Si milieu de la charge utile
If last of payload	Si dernier de la charge utile
SSPLIT - all	SSPLIT - tout
SSPLIT - begin	SSPLIT - début
SSPLIT - mid	SSPLIT - mil
SSPLIT - end	SSPLIT - fin

Anglais	Français
Host	Hôte
Go to next cmd	Aller à cmd suivante
Start split	Start split

Figure 11-83. Séquence de transaction start-split OUT isochrone



Légende

Anglais	Français
Go to complete split	Aller à complete split
Host	Hôte

Figure 11-84. Séquence de transaction start-split IN isochrone

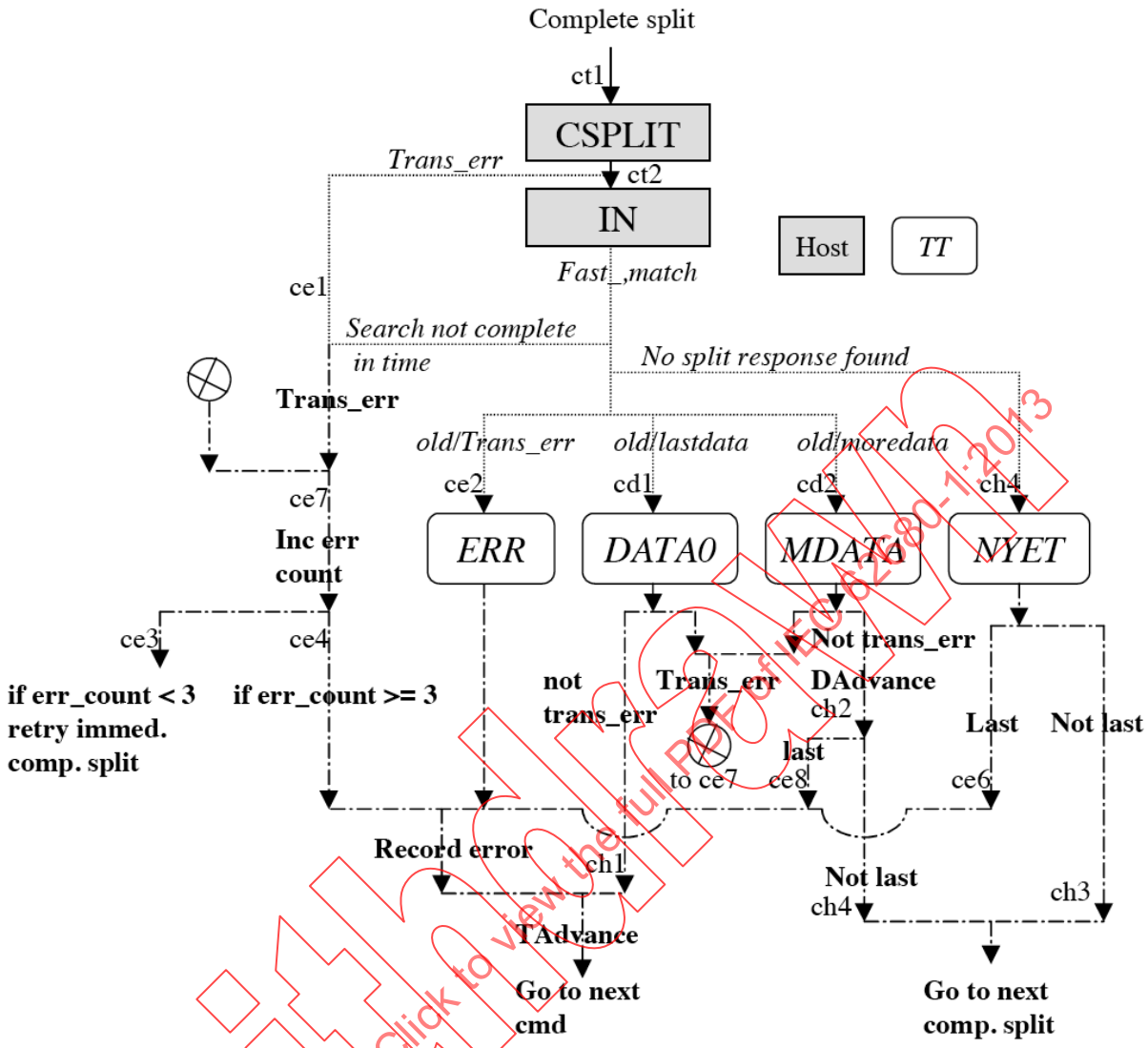
Dans la Figure 11-85, le gestionnaire à grande vitesse retourne un établissement de liaison ERR pour une "erreur de transaction" de la transaction à vitesse intégrale.

Le gestionnaire à grande vitesse retourne un établissement de liaison NYET lorsqu'il n'arrive pas à trouver une entrée correspondante dans l'étage de pipeline complete-split. Cela traite le cas où le contrôleur hôte a émis la première transaction complete-split à grande vitesse, mais le gestionnaire à vitesse intégrale/basse vitesse n'a pas encore démarré la transaction ou n'a pas encore reçu de données de l'appareil à vitesse intégrale. Cela peut être dû à un retard de démarrage des transactions à vitesse intégrale précédentes.

La transition identifiée "TAdvance" indique que l'hôte avance jusqu'à la transaction suivante pour ce point d'extrémité à vitesse intégrale.

La transition identifiée "DAdvance" indique que l'hôte avance jusqu'à la zone de données suivante pour le point d'extrémité à vitesse intégrale actuel.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
if err_count < 3 retry start split	si err_count < 3 réessayer start split
if err_count >= 3	si err_count >= 3
Go to next compl. Split	Aller au compl. Split suivant
Go to next cmd	Aller à cmd suivante
Record error	Enregistrer erreur
retry immed.	nouvel essai imméd.
last	dernier
Last	Dernier
Not last	Pas le dernier
not trans_err	pas trans_err
Not trans_err	Pas trans_err
Inc err count	Inc décompte err
old/moredata	ancien/plus de données
old/lastdata	ancien/dernières données
old/Trans_err	ancien/Trans_err
No split response found	Pas de réponse fractionnée trouvée
Search not complete in time	Recherche pas terminée à temps
Complete split	Complete split

Figure 11-85. Séquence de transaction complete-split IN isochrone

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.21.2 Diagrammes d'états de transaction fractionnée isochrone

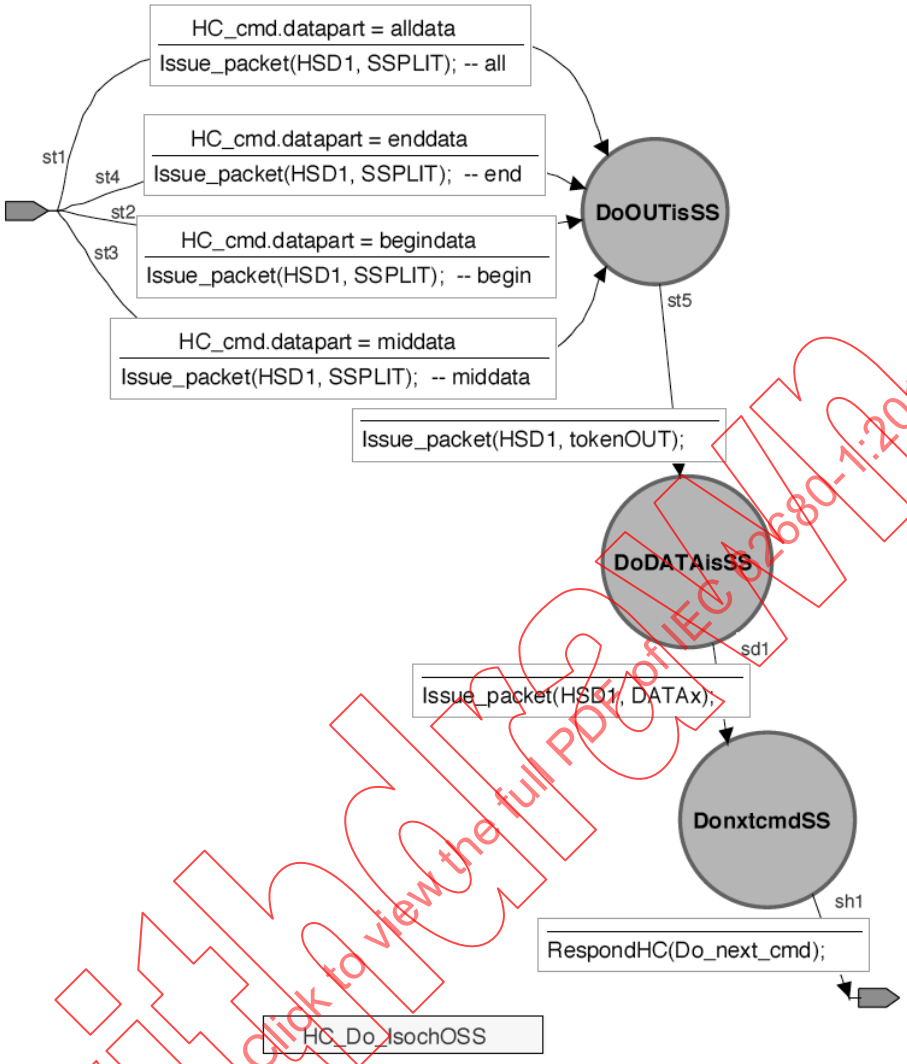
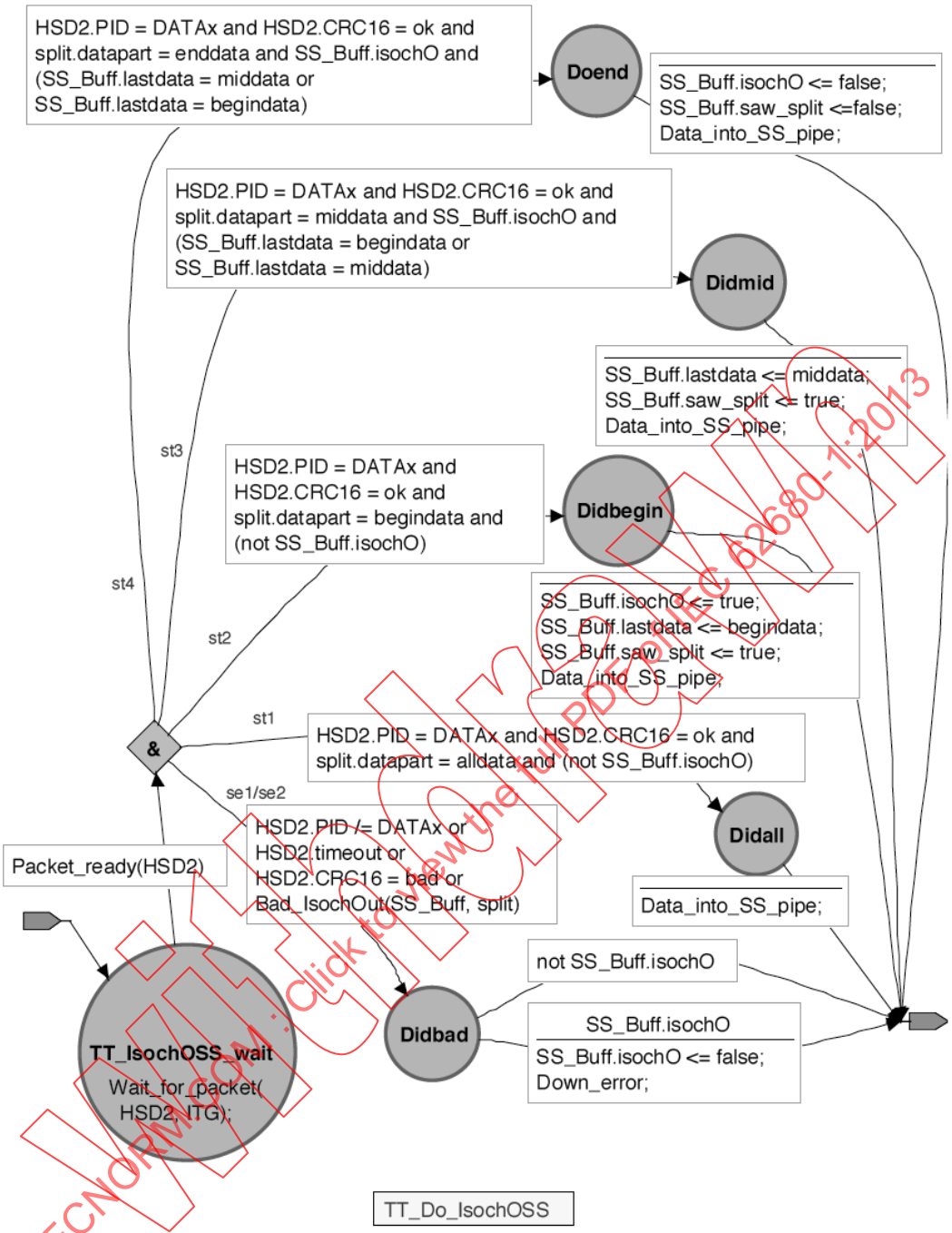


Figure 11-86. Diagramme d'états d'hôte de transaction start-split OUT isochrone

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
HSD2.PID = DATAx and HSD2.CRC16 = ok and split.datapart = enddata and SS Buff.isochO and (SS_Buff.lastdata = middata or SS_Buff.lastdata = begindata)	HSD2.PID = DATAx et HSD2.CRC16 = ok et split.datapart = enddata et SS Buff.isochO et (SS_Buff.lastdata = middata ou SS_Buff.lastdata = begindata)
HSD2.PID = DATAx and HSD2.CRC16 = ok and split.datapart = middata and SS Buff.isochO and (SS_Buff.lastdata = begindata or SS_Buff.lastdata =middata)	HSD2.PID = DATAx et HSD2.CRC16 = ok et split.datapart = middata et SS Buff.isochO et (SS_Buff.lastdata = begindata ou SS_Buff.lastdata =middata)
HSD2.PID = DATAx and HSD2.CRC16 = ok and split.datapart = begindata and (not SS_Buff.isochO)	HSD2.PID = DATAx et HSD2.CRC16 = ok et split.datapart = begindata et (pas SS_Buff.isochO)

Anglais	Français
HSD2.PID = DATAx and HSD2.CRC16 = ok and split.datapart = alldata and (not SS_Buff.isochO)	HSD2.PID = DATAx et HSD2.CRC16 = ok et split.datapart = alldata et (pas SS_Buff.isochO)
HSD2.PID = DATAx or HSD2.timeout or HSD2.CRC16 = bad or Bad.IsochOut(SS_Buff, split)	HSD2.PID = DATAx ou HSD2.timeout ou HSD2.CRC16 = bad ou Bad.IsochOut(SS_Buff, split)

Figure 11-87. Diagramme d'états de TT de transaction start-split OUT isochrone

Il y a une condition dans la Figure 11-87 sur la transition se1/se2 identifiée "Bad_IsochOut". Cette condition est vraie lorsqu'aucune des conditions sur les transitions st1 à st4 n'est vraie. L'action identifiée "Down_error" enregistre une erreur à indiquer sur le bus à vitesse intégrale orientée en aval pour la transaction correspondant à ce start-split.

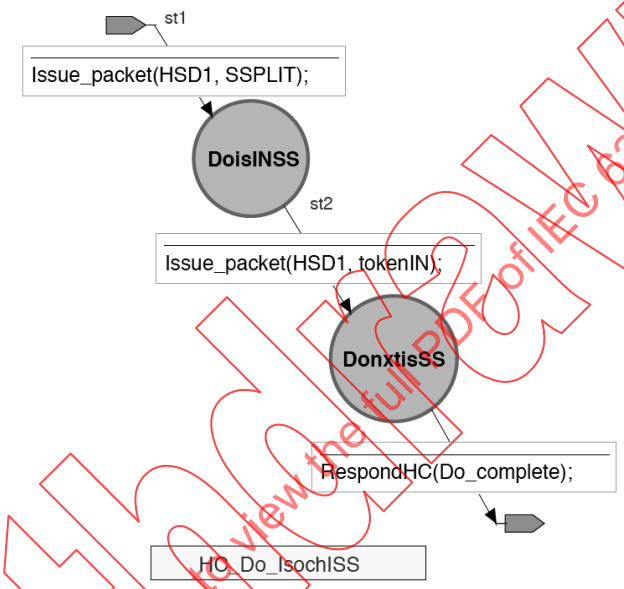
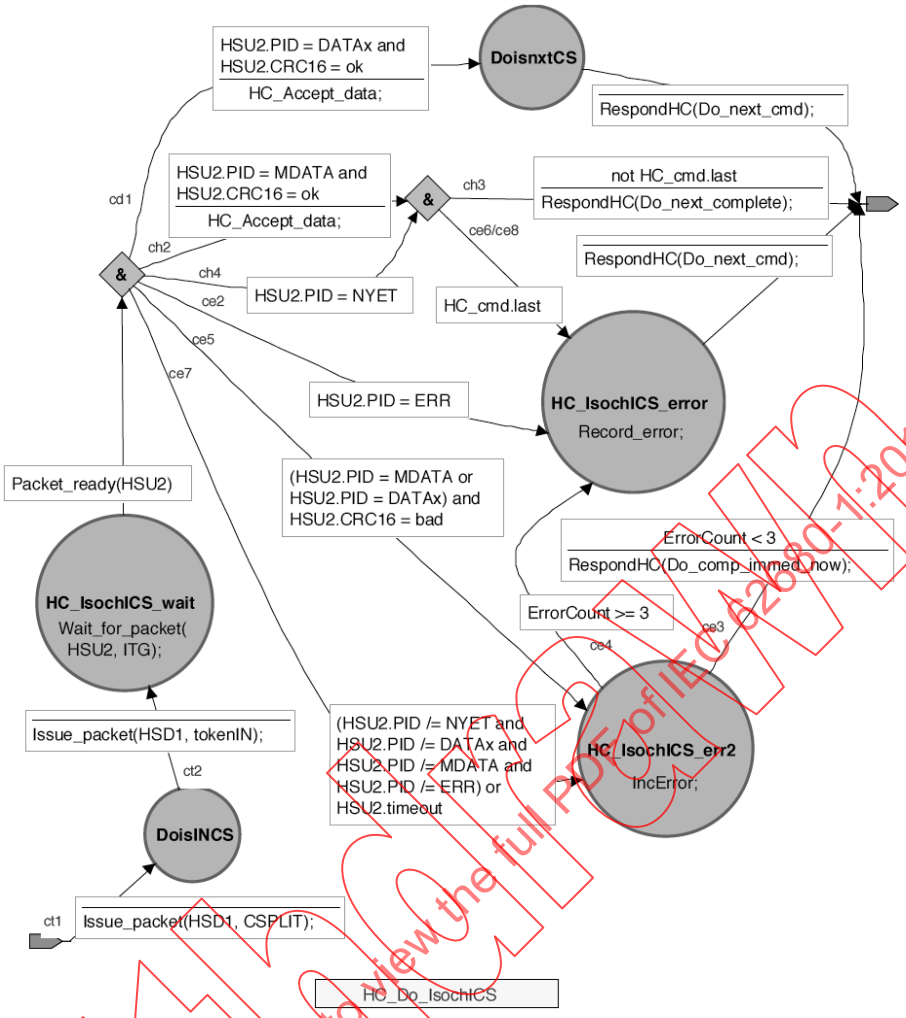


Figure 11-88. Diagramme d'états d'hôte de transaction start-split IN isochrone

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

Anglais	Français
HSU2.PID = DATAx and HSU2.CRC16 = ok	HSU2.PID = DATAx et HSU2.CRC16 = ok
HSU2.PID = MDATA and HSU2.CRC16 = ok	HSU2.PID = MDATA et HSU2.CRC16 = ok
Not HC_cmd.last	Not HC_cmd.last
(HSU2.PID = MDATA or HSU2.PID = DATAx) and HSU.CRC16 = bad	(HSU2.PID = MDATA ou HSU2.PID = DATAx) et HSU.CRC16 = incorrect
(HSU2.PID /= NYET and HSU2.PID /= DATAx and HSU2.PID /= MDATA and HSU2.PID /= ERR) or HSU2.timeout	(HSU2.PID /= NYET et HSU2.PID /= DATAx et HSU2.PID /= MDATA et HSU2.PID /= ERR) ou HSU2.timeout

Figure 11-89. Diagramme d'états d'hôte de transaction complete-split IN isochrone

Dans la Figure 11-89, la transition "ce8" se produit lorsque le gestionnaire à grande vitesse répond avec un MDATA pour indiquer qu'il y a plus de données pour la transaction à vitesse intégrale, mais le contrôleur hôte sait qu'il s'agit du dernier complete-split programmé pour ce point d'extrémité pour cette trame. Si aucune réponse DATA0 du gestionnaire à grande vitesse n'est reçue avant le dernier complete-split programmé, le contrôleur hôte enregistre une erreur et passe à la transaction suivante pour ce point d'extrémité (trame suivante).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

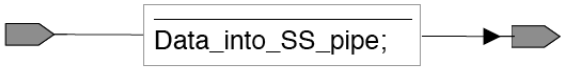
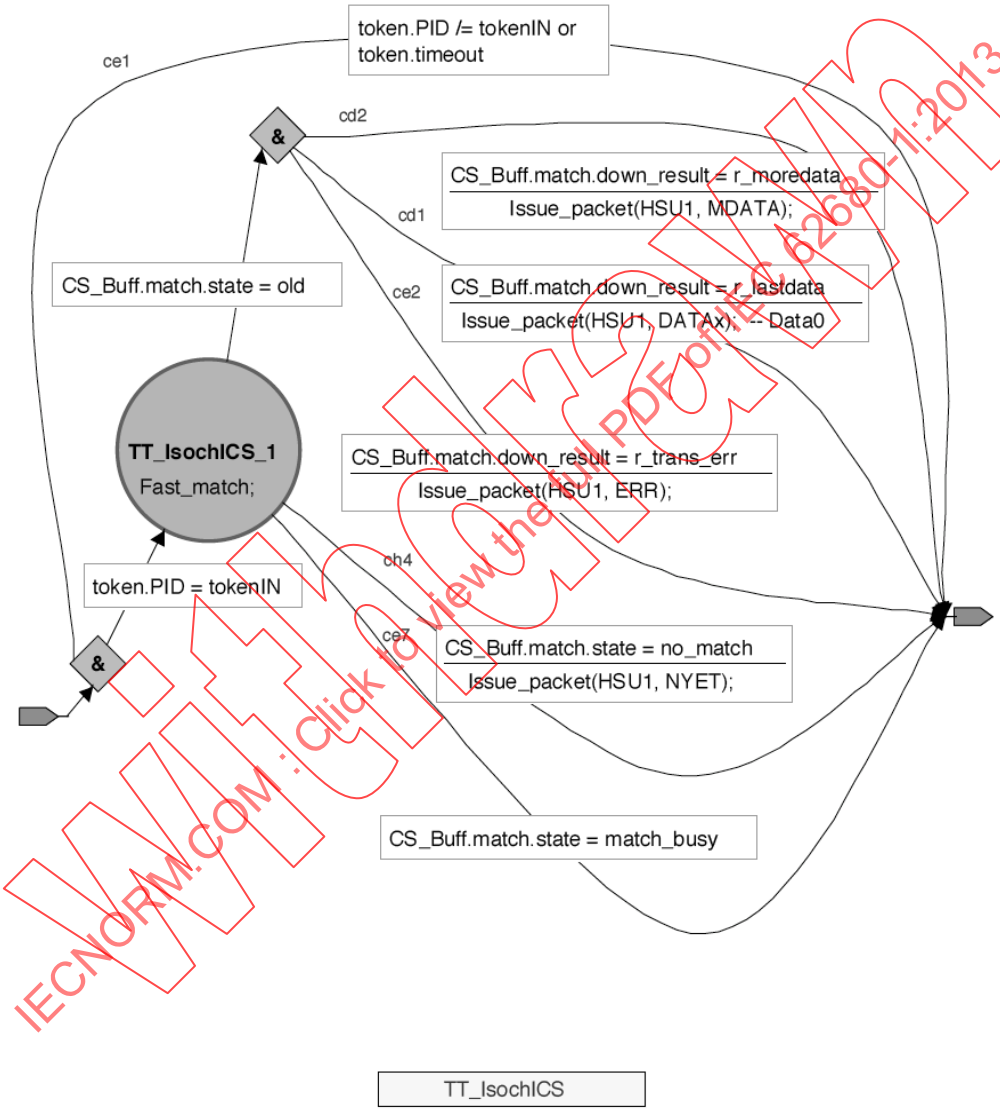




Figure 11-90. Diagramme d'états de TT de transaction start-split IN isochrone



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
token.PID /= tokenIN or token.timeout	token.PID /= tokenIN ou token.timeout
CS_Buff.match.state = old	CS_Buff.match.state = ancien

Figure 11-91. Diagramme d'états de TT de transaction complete-split IN isochrone

11.21.3 Séquençage OUT isochrone

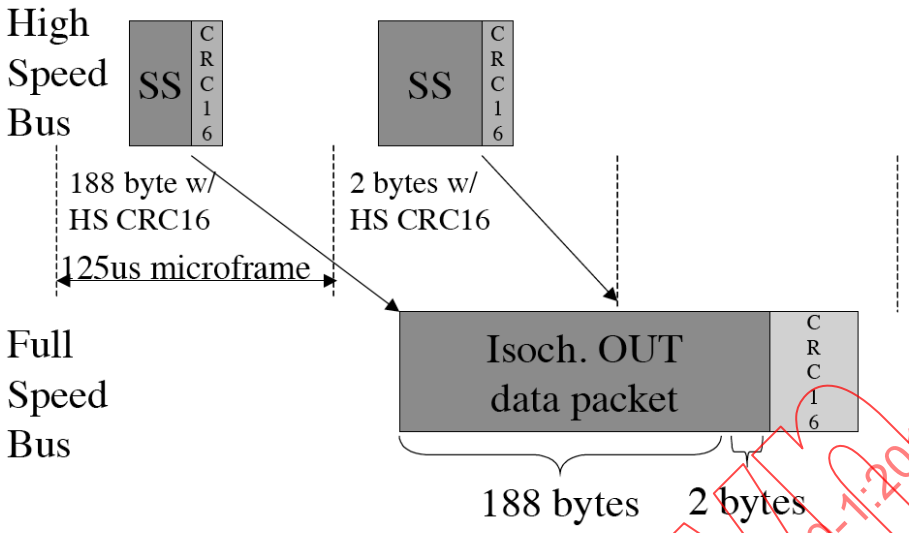
Le contrôleur hôte et le TT doivent s'assurer que les erreurs qui peuvent se produire dans les transactions fractionnées d'une transaction à vitesse intégrale isochrone sont traduites en une erreur détectable. Pour les transactions fractionnées OUT isochrones, lorsque le gestionnaire à grande vitesse a reçu un paquet de jeton de transaction start-split "SSPLIT-begin", le gestionnaire à grande vitesse doit suivre les transactions start-split qui sont reçues pour ce point d'extrémité. Le gestionnaire à grande vitesse doit suivre qu'une transaction start-split est reçue pour chaque microtrame jusqu'à ce qu'un paquet de jeton de transaction fractionnée "SSPLIT-end" soit reçu pour ce point d'extrémité. Si une microtrame passe sans que le gestionnaire à grande vitesse n'ait reçu un start-split pour ce point d'extrémité à vitesse intégrale, il doit garantir que le gestionnaire à vitesse intégrale force une erreur de bourrage de bits sur la transaction à vitesse intégrale. Tout start-split "SPLIT-middle" ou "SPLIT-end" suivant pour le même point d'extrémité doit être ignoré jusqu'à ce que le non "SPLIT-middle" et le non "SPLIT-end" suivant soit reçu (pour tout point d'extrémité pris en charge par ce TT).

La transaction start-split pour une transaction OUT isochrone ne doit pas inclure le champ CRC16 pour le paquet de données à vitesse intégrale. Pour une transaction à vitesse intégrale, l'hôte calcule le CRC16 du paquet de données pour le paquet de données entier (ex.: un paquet de données 1 023 octets utilise un seul champ CRC16 qui est calculé une fois par le contrôleur hôte). Pour une transaction fractionnée, toute transaction à vitesse intégrale OUT isochrone est divisée en de multiples start-splits, chacun ayant une charge utile de données de 188 octets ou moins. Pour chacun de ces start-splits, l'hôte calcule un champ CRC16 à grande vitesse pour chaque paquet de données start-split. Le gestionnaire à grande vitesse TT doit vérifier chaque valeur CRC16 à grande vitesse sur chaque start-split. Le gestionnaire à grande vitesse TT doit générer localement la valeur CRC16 pour le paquet de données à vitesse intégrale entier. La Figure 11-92 montre un exemple de paquet OUT isochrone à vitesse intégrale et les start-splits à grande vitesse avec leurs champs CRC16.

En cas de défaillance de contrôle CRC sur le start-split à grande vitesse, le gestionnaire à grande vitesse doit indiquer au gestionnaire à vitesse intégrale qu'il y avait une erreur dans le start-split pour la transaction à vitesse intégrale. Si la transaction a été indiquée comme ayant une défaillance CRC (ou en cas de start-split raté), le gestionnaire à vitesse intégrale utilise le mécanisme défini pour forcer un paquet corrompu en aval. Si le premier start-split a une défaillance de contrôle CRC, la transaction à vitesse intégrale ne doit pas être démarrée sur le bus en aval.

Les transactions start-split à grande vitesse supplémentaires pour le même point d'extrémité doivent être ignorées après l'échec d'un contrôle CRC, jusqu'à ce que le gestionnaire à grande vitesse reçoive un paquet de jeton de transaction start-split "SSPLIT-end" pour ce point d'extrémité ou un start-split pour un point d'extrémité différent.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

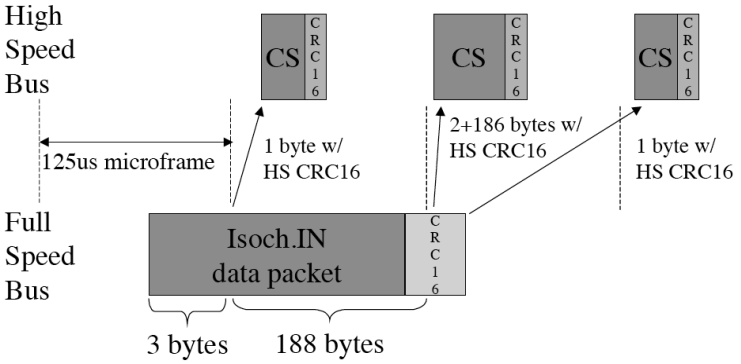
Anglais	Français
High Speed Bus	Bus à grande vitesse
188 bytes w/ HS CRC16	188 octets avec HS CRC16
2 bytes w/ HS CRC16	2 octets avec HS CRC16
125µs microframe	Microtrame 125 µs
Full Speed Bus	Bus à vitesse intégrale
Isoch. OUT data packet	Paquet de données OUT isoch.
188 bytes	188 octets
2 bytes	2 octets

Figure 11-92. Exemple de traitement de paquet de données OUT isochrone CRC16

11.21.4 Séquençage IN isochrone

La transaction complete-split pour une transaction IN isochrone ne doit pas inclure le champ CRC16 pour le paquet de données à vitesse intégrale (à savoir, seul un champ CRC16 à grande vitesse est utilisé dans les transactions fractionnées). Le TT ne doit pas passer la valeur à vitesse intégrale reçue de l'appareil et doit seulement utiliser les valeurs CRC16 à grande vitesse pour les transactions complete-split. Si le gestionnaire à vitesse intégrale détecte un contrôle CRC qui a échoué à la fin du paquet de données (ex.: après plusieurs transactions complete-split potentielles à grande vitesse), le gestionnaire doit utiliser une réponse d'établissement de liaison ERR pour refléter cette erreur au contrôleur hôte à grande vitesse. Le contrôleur hôte doit vérifier le CRC16 sur chaque complete-split à grande vitesse retourné. Une défaillance CRC (ou un établissement de liaison ERR) sur un complete-split (partiel) est reflétée par le contrôleur hôte sous forme de défaillance CRC sur la transaction à vitesse intégrale totale. La Figure 11-93 montre un exemple de relations du paquet de données à vitesse intégrale et des complete-splits à grande vitesse avec leurs champs CRC16.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
High Speed Bus	Bus à grande vitesse
1 byte w/ HS CRC16	1 octet avec HS CRC16
2 + 186 bytes w/ HS CRC16	2 + 186 octets avec HS CRC16
125µs microframe	Microtrame 125 µs
Full Speed Bus	Bus à vitesse intégrale
Isoch. IN data packet	Paquet de données IN isoch.
188 bytes	188 octets
3 bytes	3 octets

Figure 11-93. Exemple de traitement de paquet de données IN isochrone CRC16

11.22 Traitement des erreurs TT

Le TT a les mêmes exigences concernant le traitement des erreurs qu'un contrôleur hôte ou un hub. En particulier:

- Si le TT reçoit un paquet au niveau de EOF2 du bus orienté en aval, il doit désactiver le port orienté en aval qui transmet actuellement.
- Si le TT transmet un paquet à côté de EOF1 du bus orienté en aval, il doit forcer une séquence de terminaison anormale comme défini dans la Section 11.3.3 et arrêter la transmission.
- Si le TT va transmettre une transaction à vitesse intégrale/basse vitesse non périodique, il doit déterminer qu'il a suffisamment de temps avant EOF1 pour terminer la transaction. Cette détermination est basée sur le séquençage normal des paquets dans la transaction. Comme le TT n'a pas d'informations sur la charge utile de données pour les IN, il doit utiliser la taille maximum autorisée pour le type de transfert dans sa détermination. Il n'est pas nécessaire d'inclure les es transactions périodiques dans cet essai, car le pipeline de microtrame est maintenu séparément.

11.22.1 Perte de synchronisation TT avec SOF HS

Le hub a un minuteur qu'il utilise pour la maintenance des (micro)trames. Il a un minuteur 1 ms lorsqu'il fonctionne à vitesse intégrale/basse vitesse pour appliquer l'EOF avec les appareils connectés en aval. Il a un minuteur de microtrame 125 µs lorsqu'il fonctionne à grande vitesse pour appliquer l'EOF avec les appareils à grande vitesse. Il utilise également le minuteur de microtrame 125 µs pour créer un minuteur de trame 1 ms pour appliquer l'EOF avec les appareils à vitesse intégrale/basse vitesse en aval lorsqu'il fonctionne à grande vitesse. Le minuteur de (micro)trame de hub doit toujours rester synchronisé avec les SOF générés par l'hôte pour laisser le bus fonctionner correctement.

Lors du fonctionnement normal du répéteur de hub (à vitesse intégrale ou à grande vitesse) (ex.: n'impliquant pas un TT), le minuteur de (micro)trame perd la synchronisation lorsqu'il a raté les SOF pendant trois microtrames consécutives. Lorsque la synchronisation du minuteur est perdue, le hub n'établit pas la connectivité en amont. La connectivité en aval est établie normalement, même si la synchronisation du minuteur est perdue. Lorsque le minuteur est synchronisé, le hub permet l'établissement de la connectivité en amont si nécessaire. Le hub est chargé de garantir qu'il n'y a pas de signalisation répétée/transmise en amont d'un appareil après le point EOF2 dans une (micro)trame. Le hub ne doit pas établir la connectivité

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

en amont s'il a perdu la synchronisation du minuteur de (micro)trame, car il ne sait plus précisément où se trouve le point EOF2.

11.22.2 Exigences vis-à-vis de la synchronisation du minuteur de trame et de microtrame TT

Lorsque le hub fonctionne à grande vitesse et a des appareils à vitesse intégrale/basse vitesse connectés sur ses ports orientés en aval (ex.: un TT est actif), le hub a des responsabilités supplémentaires en plus de l'application du point (à grande vitesse) EOF2 sur son port orienté en amont dans toutes les microtrames. Le TT doit également générer des SOF à vitesse intégrale en aval et garantir que le TT fonctionne correctement (en pontant le fonctionnement à grande vitesse et à vitesse intégrale/basse vitesse).

Un hub fonctionnant à grande vitesse synchronise son minuteur de microtrame à des SOF de 125 µs. Mais, pour générer des SOF en aval à vitesse intégrale, il doit également avoir un minuteur de trame 1 ms. Il génère ce minuteur de trame 1 ms en reconnaissant les SOF de microtrame zéro, ex.: un SOF à grande vitesse lorsque la valeur du numéro de trame change par rapport au SOF de la microtrame immédiatement précédente.

Pour créer le minuteur de trame 1 ms, le hub doit recevoir avec succès un SOF de microtrame zéro après avoir synchronisé son minuteur de microtrame. Pour reconnaître un SOF de microtrame zéro, le hub doit recevoir avec succès les SOF pour deux microtrames consécutives où le numéro de trame est incrémenté de 1 (mod 2^11). Lorsque le hub a effectué cela, il sait que le second SOF est un SOF de microtrame zéro et établit ainsi une heure de début du minuteur de trame 1 ms. Il faut noter qu'un hub peut synchroniser les deux minuteurs avec deux SOF si les SOF sont pour la microtrame 7 et la microtrame 0, à savoir, si le second SOF est un SOF de microtrame zéro.

Lorsque le hub a synchronisé son minuteur de trame 1 ms, il peut maintenir ce minuteur synchronisé tant qu'il maintient son minuteur de microtrame 125 µs synchronisé (car il sait que toutes les 8 microtrames à partir du SOF de microtrame zéro, il y a une trame 1 ms). En particulier, le hub peut maintenir son minuteur de trame synchronisé même s'il rate les SOF de microtrame zéro (tant que le minuteur de microtrame reste synchronisé).

Ainsi en résumé, le hub peut synchroniser son minuteur de microtrame 125 µs après avoir reçu les SOF de deux microtrames consécutives. Il synchronise son minuteur de trame 1 ms lorsqu'il reçoit un SOF de microtrame zéro (et le minuteur de microtrame est synchronisé). Le minuteur de microtrame 125 µs perd la synchronisation après avoir raté trois SOF pour microtrames consécutives. Cela entraîne également la perte de synchronisation du minuteur de trame 1 ms.

Le TT doit uniquement générer des SOF à vitesse intégrale en aval lorsque son minuteur de trame 1 ms est synchronisé.

Le fonctionnement interne correct du TT dépend des deux minuteurs. Le TT doit savoir avec précision quand les microtrames se produisent pour appliquer ses règles d'abandon/de libération de pipeline de microtrame. Il sait cela en fonction d'un minuteur de microtrame synchronisé (pour incrémenter généralement le numéro de microtrame) et d'un minuteur de trame synchronisé (pour savoir quand la microtrame zéro se produit).

Comme la perte de synchronisation du minuteur de microtrame entraîne immédiatement la perte de la synchronisation du minuteur de trame, le TT arrête le fonctionnement normal lorsque le minuteur de microtrame perd la synchronisation. Dans un environnement sans erreur, la synchronisation du minuteur de microtrame peut être restaurée après avoir reçu les deux SOF pour les deux microtrames consécutives suivantes (ex.: la synchronisation est restaurée au moins 250 µs après la perte de synchronisation). Tant que les SOF ne sont pas ratés, la synchronisation du minuteur de trame est restaurée en moins d'1 ms après la synchronisation de microtrame. Il faut noter que la synchronisation du minuteur de trame peut être restaurée dans un cas de fonctionnement à grande vitesse en moins de temps (0,250-1,250 ms) que les 2-3 ms requises dans le fonctionnement à vitesse intégrale. Lorsque le

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

minuteur de trame est synchronisé, les SOF peuvent être émis sur les ports à vitesse intégrale orientés en aval pour le début de la trame suivante.

Lorsque le hub détecte la perte de synchronisation du minuteur de microtrame, son TT(s):

1. Doit répondre aux complete-splits périodiques avec les réponses mises en tampon dans le pipeline périodique (uniquement correct pour au moins 1 microtrame de complete-splits).
2. Doit abandonner les transactions start-split périodiques mises en tampon dans le pipeline périodique.
3. Doit ignorer les start-splits périodiques à grande vitesse.
4. Doit arrêter l'émission de SOF à vitesse intégrale sur les ports à vitesse intégrale orientés en aval (et les maintiens actifs à basse vitesse sur les ports à basse vitesse).
5. Ne doit pas démarrer l'émission de transactions à vitesse intégrale/basse vitesse périodiques suivantes sur les ports à vitesse intégrale/basse vitesse orientés en aval.
6. Doit répondre aux transactions en masse/de contrôle start-split à grande vitesse.
7. Les résultats en masse/de contrôle mis en tampon doivent répondre aux transactions complete-split à grande vitesse.
8. Les transactions en masse/de contrôle en cours ne doivent pas être émises sur les ports orientés en aval à vitesse intégrale/basse vitesse. Les tampons TT utilisés pour contenir les transactions en masse/de contrôle doivent être préservés jusqu'à ce que le minuteur de microtrame soit resynchronisé. (Ou jusqu'à ce qu'une demande Clear_TT_Buffer soit reçue pour la transaction).

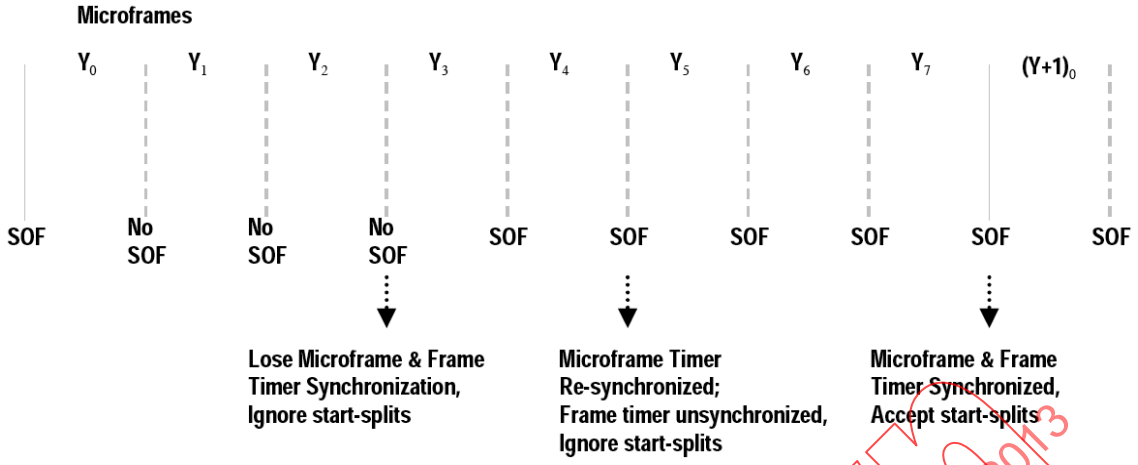
Il faut noter que dans tous les cas un TT ne doit pas émettre les transactions à n'importe quelle vitesse sur les ports orientés en aval lorsque son port orienté en amont est suspendu.

Un TT restaure uniquement le fonctionnement normal sur les ports à vitesse intégrale/basse vitesse orientés en aval après avoir synchronisé les minuteurs de microtrame et de trame. La Figure 11-93 résume la relation entre les SOF à grande vitesse et les exigences de synchronisation de minuteur de trame et de microtrame TT sur vis-à-vis des start-splits.

Pour le séquençage de suspension d'un hub, un hub perd d'abord la synchronisation du minuteur de microtrame/trame en même temps. Cela fait que son TT arrête l'émission de SOF (il convient que ce soient les seules transactions maintenant les ports à vitesse intégrale/basse vitesse orientés en aval en dehors de la suspension). Le hub (avec les appareils en aval) entre alors en suspension.

Lors d'une reprise, le hub restaure d'abord sa synchronisation de minuteur de microtrame (après les transactions à grande vitesse continuent). En moins de 1 ms (en supposant qu'il n'y ait pas d'erreurs), le minuteur de trame est synchronisé et le TT peut démarrer le fonctionnement normal (y compris les SOF/maintiens actifs sur les ports à vitesse intégrale/basse vitesse orientés en aval).

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



Légende

Anglais	Français
Microframes	Microtrames
No SOF	Pas SOF
SOF	SOF
Lose Microframe & Frame Timer Synchronization, Ignore start-splits	Perte Microtrame & Trame Synchronisation de microtrame, Ignorer start-splits
Microframe Time Re-synchronized; Frame timer unsynchronized; Ignore start-splits	Temps de microtrame Resynchronisé; Minuteur de trame non synchronisé; Ignorer start-splits
Microframe & Frame Timer Synchronization, Accept start-splits	Microtrame & Trame Synchronisation du minuteur, Accepter start-splits

Figure 11-94. Exemple d'évènements de synchronisation de trame/microtrame

11.23 Descripteurs

Les descripteurs de hub sont dérivés du cadre général de l'appareil USB général. Les descripteurs de hub définissent un appareil de hub et les ports sur ce hub. L'hôte accès aux descripteurs de hub via le canal par défaut du hub.

La spécification USB (voir Chapitre 9) définit les descripteurs suivants:

- Appareil
- Configuration
- Interface
- Point d'extrémité
- Chaîne (en option)

La classe de hub définit les descripteurs supplémentaires (voir Section 11.23.2). De plus, les descripteurs spécifiques aux fournisseurs sont autorisés dans le cadre général de l'appareil USB. Les hubs prennent en charge les commandes d'appareil USB normalisées comme défini au Chapitre 9.

11.23.1 Descripteurs normalisés pour classe de hub

La classe de hub prédéfinit certains champs dans les descripteurs USB normalisés. Les autres champs dépendent de la mise en œuvre ou ne s'appliquent à cette classe.

Un hub retourne différents descripteurs en fonction de leur fonctionnement à grande vitesse ou à vitesse intégrale/basse vitesse. Un hub peut signaler trois ensembles différents de descripteurs: un descripteur pour le fonctionnement à vitesse intégrale/basse vitesse et deux ensembles pour le fonctionnement à grande vitesse.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Un hub fonctionnant à vitesse intégrale/basse vitesse a un descripteur d'appareil avec un champ bDeviceProtocol défini sur zéro(0) et un descripteur d'interface avec un champ bInterfaceProtocol défini sur zéro(0). Le reste des descripteurs est le même pour toutes les vitesses.

Un hub fonctionnant à grande vitesse peut avoir une des deux organisations TT: un seul TT ou de multiples TT. Tous les hubs doivent en prendre en charge l'organisation d'un seul TT. Un hub à multiples TT a un descripteur d'interface supplémentaire (avec un descripteur de point d'extrémité correspondant). Le premier ensemble de descripteurs illustré ci-après doit être fourni par tous les hubs. Un hub qui a un seul TT doit définir le champ bDeviceProtocol du descripteur d'appareil sur un(1) et le descripteur d'interface du champ bInterfaceProtocol est défini sur 0.

Un hub à multiples TT doit définir le champ bDeviceProtocol du descripteur d'appareil sur deux (2). Le premier descripteur d'interface a le champ bInterfaceProtocol défini sur un(1). Un tel hub a également un second descripteur d'interface où le bInterfaceProtocol est défini sur deux(2). Lorsque le hub est configuré avec un protocole d'interface d'un(1), il fonctionne comme un hub organisé avec un seul TT. Lorsque le hub est configuré avec un protocole d'interface de deux(2), il fonctionne comme un hub organisé avec de multiples TT. L'organisation du TT ne doit pas être modifiée alors que le hub a des transactions à vitesse intégrale/basse vitesse en cours.

Note: Pour les descripteurs et champs illustrés ci-après, les bits dans un champ sont organisés de manière petit-boutiste; ainsi, la position binaire 0 est le bit de poids faible et la position binaire 7 est le bit de poids fort d'une valeur d'octet.

Hub fonctionnant à vitesse intégrale/basse vitesse

Descripteur d'appareil (informations à vitesse intégrale):

bLength	12H
bDescriptorType	1
bcdUSB	0200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
bNumConfigurations	1

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Descripteur Device_Qualifier (informations à grande vitesse):

bLength	0AH
bDescriptorType	6
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	1 (pour un seul TT) ou 2 (pour de multiples TT)
bMaxPacketSize0	64
bNumConfigurations	1

Descripteur de configuration (informations à vitesse intégrale):

bLength	09H
bDescriptorType	2
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	La quantité maximum de puissance de bus que le hub va consommer dans la configuration à vitesse intégrale/basse vitesse

Descripteur d'interface:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	i

Descripteur de point d'extrémité (pour point d'extrémité de changement d'état):

bLength	07H
bDescriptorType	5
bEndpointAddress	En fonction de la mise en œuvre; Bit 7: Direction = In(1)
bmAttributes	Type de transfert = Interruption (00000011B)
wMaxPacketSize	En fonction de la mise en œuvre
bInterval	FFH (intervalle maximum autorisé)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Descripteur Other_Speed_Configuration (informations à grande vitesse):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1 (pour un seul TT) ou 2 (pour de multiples TT)
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	La quantité maximum de puissance de bus que le hub va consommer dans la configuration à grande vitesse

Descripteur d'interface:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0 (pour un seul TT), 1 (pour plusieurs TT)
iInterface	i

Descripteur de point d'extrémité (pour point d'extrémité de changement d'état):

bLength	07H
bDescriptorType	5
bEndpointAddress	En fonction de la mise en œuvre; Bit 7: Direction = In(1)
bmAttributes	Type de transfert = Interruption (00000011B)
wMaxPacketSize	En fonction de la mise en œuvre
bInterval	FFH (intervalle maximum autorisé)

Descripteur d'interface (présent si un hub avec de multiples TT):

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	2
iInterface	i

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Descripteur de point d'extrémité (présent si un hub avec de multiples TT):

bLength	07H
bDescriptorType	5
bEndpointAddress	En fonction de la mise en œuvre; Bit 7: Direction = In(1)
bmAttributes	Type de transfert = Interruption (00000011B)
wMaxPacketSize	En fonction de la mise en œuvre
bInterval	FFH (intervalle maximum autorisé)

Hub fonctionnant à grande vitesse avec un seul TT

Descripteur d'appareil (informations à grande vitesse):

bLength	12H
bDescriptorType	1
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	1
bMaxPacketSize0	64
bNumConfigurations	1

Descripteur Device_Qualifier (informations à vitesse intégrale):

bLength	0AH
bDescriptorType	6
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
bNumConfigurations	1

Descripteur de configuration (informations à grande vitesse):

bLength	09H
bDescriptorType	2
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	La quantité maximale de puissance de bus que le hub va consommer dans cette configuration

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Descripteur d'interface:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0 (un seul TT)
iInterface	i

Descripteur de point d'extrémité (pour point d'extrémité de changement d'état):

bLength	07H
bDescriptorType	5
bEndpointAddress	En fonction de la mise en œuvre; Bit 7: Direction = In(1)
bmAttributes	Type de transfert = Interruption (00000011B)
wMaxPacketSize	En fonction de la mise en œuvre
bInterval	FFH (intervalle maximum autorisé)

Descripteur Other_Speed_Configuration (informations à vitesse intégrale):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	La quantité maximum de puissance de bus que le hub va consommer dans la configuration à grande vitesse

Descripteur d'interface:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	i

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Descripteur de point d'extrémité (pour point d'extrémité de changement d'état):

bLength	07H
bDescriptorType	5
bEndpointAddress	En fonction de la mise en œuvre; Bit 7: Direction = In(1)
bmAttributes	Type de transfert = Interruption (00000011B)
wMaxPacketSize	En fonction de la mise en œuvre
bInterval	FFH (intervalle maximum autorisé)

Hub fonctionnant à grande vitesse avec de multiples TT

Descripteur d'appareil (informations à grande vitesse):

bLength	12H
bDescriptorType	1
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	2 (de multiples TT)
bMaxPacketSize0	64
bNumConfigurations	1

Descripteur Device_Qualifier (informations à vitesse intégrale):

bLength	0AH
bDescriptorType	6
bcdUSB	200H
bDeviceClass	HUB_CLASSCODE (09H)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
bNumConfigurations	1

Descripteur de configuration (informations à grande vitesse):

bLength	09H
bDescriptorType	2
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	La quantité maximum de puissance de bus que le hub va consommer dans cette configuration

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Descripteur d'interface:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	1 (un seul TT)
iInterface	i

Descripteur de point d'extrémité (pour point d'extrémité de changement d'état):

bLength	07H
bDescriptorType	5
bEndpointAddress	En fonction de la mise en œuvre; Bit 7: Direction = In(1)
bmAttributes	Type de transfert = Interruption (00000011B)
wMaxPacketSize	En fonction de la mise en œuvre
bInterval	FFH (intervalle maximum autorisé)

Descripteur d'interface:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	1
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	2 (de multiples TT)
iInterface	i

Descripteur de point d'extrémité:

bLength	07H
bDescriptorType	5
bEndpointAddress	En fonction de la mise en œuvre; Bit 7: Direction = In(1)
bmAttributes	Type de transfert = Interruption (00000011B)
wMaxPacketSize	En fonction de la mise en œuvre
bInterval	FFH (intervalle maximum autorisé)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Descripteur Other_Speed_Configuration (informations à vitesse intégrale):

bLength	09H
bDescriptorType	7
wTotalLength	N
bNumInterfaces	1
bConfigurationValue	X
iConfiguration	Y
bmAttributes	Z
bMaxPower	La quantité maximale de puissance de bus que le hub va consommer dans la configuration à grande vitesse

Descripteur d'interface:

bLength	09H
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	1
bInterfaceClass	HUB_CLASSCODE (09H)
bInterfaceSubClass	0
bInterfaceProtocol	0
iInterface	i

Descripteur de point d'extrémité (pour point d'extrémité de changement d'état):

bLength	07H
bDescriptorType	5
bEndpointAddress	En fonction de la mise en œuvre; Bit 7: Direction = In(1)
bmAttributes	Type de transfert = Interruption (00000011B)
wMaxPacketSize	En fonction de la mise en œuvre
bInterval	FFH (intervalle maximum autorisé)

Le pilote de classe de hub extrait une configuration d'appareil du logiciel du système USB à l'aide de demande d'appareil GetDescriptor(). Le seul descripteur de point d'extrémité qui est retourné par la demande GetDescriptor() est le descripteur de point d'extrémité de changement d'état.

11.23.2 Descripteurs spécifiques à la classe

11.23.2.1 Descripteur de hub

Le Tableau 11-13 montre les divers champs contenus par le descripteur de hub.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 11-13. Descripteur de hub

Décalage	Champ	Taille	Description
0	bDescLength	1	Nombre d'octets dans ce descripteur, y compris cet octet
1	bDescriptorType	1	Type de descripteur, valeur: 29H pour le descripteur de hub
2	bNbrPorts	1	Nombre de ports orientés en aval que ce hub prend en charge
3	wHubCharacteristics	2	D1...D0: Mode de commutation d'alimentation logique 00: Commutation d'alimentation associée (alimentation de tous' les ports en même temps) 01: Commutation d'alimentation de port individuel 1X: Réservé. Utilisé uniquement sur les hubs conformes 1.0 qui ne mettent pas en œuvre la commutation d'alimentation D2: Identifie un appareil combiné 0: Le hub ne fait pas partie d'un appareil combiné. 1: Le hub fait partie d'un appareil combiné. D4...D3: Mode de protection contre les surintensités 00: Protection globale contre les surintensités. Le hub signale la surintensité sous forme de résumé de la consommation de courant de tous les ports, sans ventilation du statut de surintensité de port individuel. 01: Protection contre les surintensités d'un port individuel. Le hub signale une surintensité sur une base par port. Chaque port a un statut de surintensité. 1X: Pas de protection contre les surintensités. Cette option est admise uniquement pour les hubs alimentés par bus qui ne mettent pas en œuvre la protection contre les surintensités. D6...D5: TT Think Time 00: Le TT nécessite au moins 8 temps bit FS d'écart inter-transaction sur un bus en aval à vitesse intégrale/basse vitesse. 01: Le TT nécessite au moins 16 temps bit FS. 10: Le TT nécessite au moins 24 temps bit FS. 11: Le TT nécessite au moins 32 temps bit FS. D7: Indicateurs de port pris en charge 0: Les indicateurs de port ne sont pas pris en charge sur ses ports en aval et la demande PORT_INDICATOR n'a pas d'effet. 1: Les indicateurs de port sont pris en charge sur ses ports en aval et la demande PORT_INDICATOR commande les indicateurs. Voir Section 11.5.3. D15...D8: Réservé
5	bPwrOn2PwrGood	1	Temps (en intervalles de 2 ms) à partir du moment où la séquence de mise sous tension commence sur un port jusqu'à ce que l'alimentation soit bonne sur ce port. Le logiciel du système USB utilise cette valeur pour déterminer le temps d'attente avant d'accéder au port sous tension.
6	bHubContrCurrent	1	Exigences de courant maximum de l'électronique du contrôleur de hub en mA.
7	DeviceRemovable	Variable, selon le nombre de ports sur le hub	Indique si un port a un appareil amovible raccordé. Ce champ est signalé sur la granularité de l'octet. Dans un octet, si aucun port n'existe pour une position donnée, le champ représentant les caractéristiques de port retourne 0. Définition de la valeur binaire: 0B - L'appareil est amovible. 1B - L'appareil n'est pas amovible. Topogramme binaire correspondant aux ports du hub: Bit 0: Réservé à un usage ultérieur. Bit 1: Port 1 Bit 2: Port 2 .... Bit *n*: Port *n* (en fonction de la mise en œuvre jusqu'au maximum 255 ports).
Variable	*PortPwrCtrlMask*	Variable, selon le nombre de ports sur le hub	Ce champ existe pour des raisons de compatibilité avec le logiciel écrit pour les appareils compatibles 1.0. Il convient que tous les bits de ce champ soient définis sur 1B. Ce champ a un bit pour chaque port du hub avec des bits tampons supplémentaires, si nécessaire, pour que le nombre de bits dans le champ soit un multiple entier de 8.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.24 Demandes

11.24.1 Demandes normalisées

Les hubs ont des contraintes plus strictes concernant la synchronisation de traitement des demandes par rapport à ce qui est spécifié dans la Section 9.2.6 pour les appareils normalisés, car ils sont cruciaux pour le "temps jusqu'à disponibilité" de tous les appareils raccordés à l'USB. Les pires exigences de synchronisation de demande sont listées ci-après (s'applique aux demandes normalisées et de classe de hub):

1. Temps d'exécution des demandes sans étage de données: 50 ms
2. Temps d'exécution des demandes normalisées avec étage(s) de données

   Temps entre le paquet de configuration et le premier étage de données: 50 ms

   Temps entre chaque étage de données suivant: 50 ms

   Temps entre le dernier étage de données et l'étage de statut: 50 ms

Comme les hubs jouent un rôle crucial dans l'énumération de bus, il convient que les temps de réponse moyens des hubs soient inférieurs à 5 ms pour toutes les demandes.

Le Tableau 11-14 montre les diverses demandes d'appareil normalisées.

Tableau 11-14. Réponses de hub aux demandes d'appareil normalisées

bRequest	Réponse du hub
CLEAR_FEATURE	Normalisée
GET_CONFIGURATION	Normalisée
GET_DESCRIPTOR	Normalisée
GET_INTERFACE	Indéfinie Les hubs peuvent prendre en charge une seule interface.
GET_STATUS	Normalisée
SET_ADDRESS	Normalisée
SET_CONFIGURATION	Normalisée
SET_DESCRIPTOR	Optionnelle
SET_FEATURE	Normalisée
SET_INTERFACE	Indéfinie Les hubs peuvent prendre en charge une seule interface.
SYNCH_FRAME	Indéfinie Les hubs ne peuvent pas avoir des points d'extrémité isochrones.

Les demandes en option qui ne sont pas mises en œuvre doivent retourner un STALL dans l'étage Données ou l'étage Etat de la demande.

11.24.2 Demandes spécifiques à la classe

La classe de hub définit les demandes auxquelles les hubs répondent, comme indiqué dans le Tableau 11-15. Le Tableau 11-16 définit les codes de demande de classe de hub. Toutes les demandes dans le tableau ci-après sauf SetHubDescriptor() sont obligatoires.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 11-15. Demandes de classe de hub

Demande	bmRequestType	bRequest	wValue	wIndex	wLength	Données
ClearHubFeature	00100000B	CLEAR_ FEATURE	Sélecteur de caractéristiques	Zéro	Zéro	Néant
ClearPortFeature	00100011B	CLEAR_ FEATURE	Sélecteur de caractéristiques	Sélecteur, Port	Zéro	Néant
ClearTTBuffer	00100011B	CLEAR_TT_BUFFER	Dev_Addr, EP_Num	TT_port	Zéro	Néant
GetHubDescriptor	10100000B	GET_DESCRIPTOR	Type de descripteur et Index de descripteur	Zéro ou ID de langage	Longueur de descripteur	Descripteur
GetHubStatus	10100000B	GET_ STATUS	Zéro	Zéro	Quatre	Etat de hub et Etat de changement
GetPortStatus	10100011B	GET_ STATUS	Zéro	Port	Quatre	Etat de port et Etat de changement
ResetTT	00100011B	RESET_TT	Zéro	Port	Zéro	Néant
SetHubDescriptor	00100000B	SET_DESCRIPTOR	Type de descripteur et Index de descripteur	Zéro ou ID de langage	Longueur de descripteur	Descripteur
SetHubFeature	00100000B	SET_ FEATURE	Sélecteur de caractéristiques	Zéro	Zéro	Néant
SetPortFeature	00100011B	SET_ FEATURE	Sélecteur de caractéristiques	Sélecteur, Port	Zéro	Néant
GetTTState	10100011B	GET_TT_STATE	TT_Flags	Port	Longueur d'état TT	Etat TT
StopTT	00100011B	STOP_TT	Zéro	Port	Zéro	Néant

Tableau 11-16. Codes de demande de classe de hub

bRequest	Valeur
GET_ STATUS	0
CLEAR_ FEATURE	1
Réservé (UTILISE DANS LES SPECIFICATIONS PRECEDENTES POUR GET_STATE)	2
SET_ FEATURE	3
Réservé à un usage ultérieur	*4-5*
GET_DESCRIPTOR	6
SET_DESCRIPTOR	7
CLEAR_TT_BUFFER	8
RESET_TT	9
GET_TT_STATE	10
STOP_TT	11

Le Tableau 11-17 indique les sélecteurs de caractéristiques valides pour la classe de hub. Voir Section 11.24.2.6 et Section 11.24.2.7 pour une description des caractéristiques.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Tableau 11-17. Sélecteurs de caractéristiques de classe de hub

	Destinataire	Valeur
C_HUB_LOCAL_POWER	Hub	0
C_HUB_OVER_CURRENT	Hub	1
PORT_CONNECTION	Port	0
PORT_ENABLE	Port	1
PORT_SUSPEND	Port	2

Tableau 11-17. Sélecteurs de caractéristiques de classe de hub (suite)

	Destinataire	Valeur
PORT_OVER_CURRENT	Port	3
PORT_RESET	Port	4
PORT_POWER	Port	8
PORT_LOW_SPEED	Port	9
C_PORT_CONNECTION	Port	16
C_PORT_ENABLE	Port	17
C_PORT_SUSPEND	Port	18
C_PORT_OVER_CURRENT	Port	19
C_PORT_RESET	Port	20
PORT_TEST	Port	21
PORT_INDICATOR	Port	22

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.24.2.1 Clear Hub Feature

Cette demande réinitialise une valeur indiquée dans le statut de hub.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
00100000B	CLEAR_FEATURE	Sélecteur de caractéristiques	Zéro	Zéro	Néant

L'effacement d'une caractéristique désactive cette caractéristique; voir Tableau 11-17 pour les définitions de sélecteur de caractéristiques qui s'appliquent au hub en tant que destinataire. Si le sélecteur de caractéristiques est associé à un changement d'état, l'effacement du changement d'état accepte le changement. Ce format de demande est utilisé pour effacer les caractéristiques C_HUB_LOCAL_POWER ou C_HUB_OVER_CURRENT.

Il s'agit d'une erreur de demande si *wValue* n'est pas un sélecteur de caractéristiques listé dans le Tableau 11-17 ou si *wIndex* ou *wLength* ne sont pas comme spécifié ci-dessus.

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

11.24.2.2 Clear Port Feature

Cette demande réinitialise une valeur indiquée dans l'état de port.

bmRequestType	bRequest	wValue	wIndex		wLength	Données
00100011B	CLEAR_FEATURE	Sélecteur de caractéristiques	Sélecteur	Port	Zéro	Néant

Le numéro de port doit être un numéro de port valide pour ce hub, supérieur à zéro. Le champ de port est situé dans les bits 7..0 du champ *wIndex*.

L'effacement d'une caractéristique désactive cette caractéristique ou démarre un processus associé à la caractéristique; voir Tableau 11-17 pour les définitions de sélecteur de caractéristiques. Si le sélecteur de caractéristiques est associé à un changement d'état, l'effacement de ce changement d'état accepte le changement. Ce format de demande est utilisé pour effacer les caractéristiques suivantes:

- PORT_ENABLE
- PORT_SUSPEND
- PORT_POWER
- PORT_INDICATOR
- C_PORT_CONNECTION
- C_PORT_RESET
- C_PORT_ENABLE
- C_PORT_SUSPEND
- C_PORT_OVER_CURRENT

L'effacement de la caractéristique PORT_SUSPEND entraîne une reprise initiée par l'hôte sur le port spécifié. Si le port n'est pas dans l'état Suspendu, il convient que le hub traite cette demande comme un non-fonctionnement fonctionnel.

L'effacement de la caractéristique PORT_ENABLE entraîne le placement du port dans l'état Désactivé. Si le port est dans l'état Hors tension, il convient que le hub traite cette demande comme un non-fonctionnement fonctionnel.

L'effacement de la caractéristique PORT_POWER entraîne le placement du port dans l'état Hors tension et peut, en raison des contraintes dues à la méthode de commutation d'alimentation du port, entraîner la coupure de l'alimentation du port. Voir Section 11.11 concernant les règles relatives à la manière dont cette demande est utilisée avec les ports qui sont alimentés pas association.

Le champ de sélecteur identifie le sélecteur d'indicateur de port lors de l'effacement d'un indicateur de port. Le champ de sélecteur est situé dans les bits 15..8 du champ *wIndex*.

Il s'agit d'une erreur de demande si *wValue* n'est pas un sélecteur de caractéristiques listé dans le Tableau 11-17, si *wIndex* spécifie un port qui n'existe pas ou si *wLength* ne sont pas comme spécifié ci-dessus. Il n'est pas une erreur pour cette demande d'essayer d'effacer une caractéristique qui est déjà effacée (il convient que le hub traite cela comme un non-fonctionnement fonctionnel).

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

11.24.2.3 Clear TT Buffer

Cette demande efface l'état d'un tampon en masse/de contrôle du traducteur de transactions (TT) après qu'il ait été laissé dans un état occupé en raison d'erreurs à grande vitesse. Cette demande est uniquement définie pour les points d'extrémité non périodiques; ex.: si elle est émise pour un point d'extrémité périodique, la réponse est indéfinie. Après l'exécution réussie de cette demande, le tampon peut être réutilisé par le TT avec les transactions fractionnées à grande vitesse pour les transactions à vitesse intégrale/basse vitesse sur les appareils à vitesse intégrale/basse vitesse raccordés.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
00100011B	CLEAR_TT_BUFFER	Device_Address, Endpoint_Number	TT_port	Zéro	Néant

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Si le hub prend en charge un TT par port, wIndex doit alors spécifier le numéro de port du TT qui a rencontré les erreurs à grande vitesse (ex.: avec le tampon TT occupé). Si le hub fournit uniquement un seul TT, *wIndex* doit alors être défini sur un.

device_address, endpoint_number et endpoint_type du point d'extrémité à vitesse intégrale/basse vitesse (comme spécifié dans la transaction fractionnée correspondante) qui peuvent avoir un tampon TT occupé doivent être spécifiés dans le champ wValue. Les champs binaires spécifiques utilisés sont indiqués dans le Tableau 11-18.

Il s'agit d'une erreur de demande si *wIndex* spécifie un port qui n'existe pas, ou si *wLength* n'est pas comme spécifié ci-dessus. Il n'est pas une erreur pour cette demande d'essayer d'effacer un tampon pour une transaction qui n'est pas mise en tampon par le TT (il convient que cela soit traité comme un non-fonctionnement fonctionnel).

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

Tableau 11-18. Champ wValue pour Clear_TT_Buffer

Bits	Champ
3..0	Endpoint Number
10..4	Device Address
12..11	Endpoint Type
14..13	Reserved, must be zero
15	Direction, 1 = IN, 0 = OUT

11.24.2.4 Get Bus State

Versions précédentes de l'USB qui ont défini une demande GetBusState. Cette demande n'est plus définie.

11.24.2.5 Get Hub Descriptor

Cette demande retourne le descripteur de hub.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
10100000B	GET_DESCRIPTOR	Type de descripteur et Index de descripteur	Zéro	Longueur de descripteur	Descripteur

La demande GetDescriptor() pour le descripteur de classe de hub suit le même modèle d'utilisation que celle de la demande GetDescriptor() normalisée (voir Chapitre 9). Le descripteur de hub normalisé est indiqué en utilisant la valeur *bDescriptorType* définie dans la Section 11.23.2.1. Tous les hubs doivent appliquer un descripteur de hub, avec un index de descripteur zéro.

Si *wLength* est plus grand que la longueur réelle du descripteur, seule la longueur réelle est alors retournée. Si *wLength* est inférieur à la longueur réelle du descripteur, seuls les premiers octets *wLength* du descripteur sont alors retournés; cela n'est pas considéré comme une erreur même si *wLength* est égal à zéro.

Il s'agit d'une erreur de demande si *wValue* ou *wIndex* sont différents de ce qui est spécifié ci-dessus.

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

11.24.2.6 Get Hub Status

Cette demande retourne le statut de hub actuel et les états qui ont changé depuis le précédent acquittement.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

bmRequestType	bRequest	wValue	wIndex	wLength	Données
10100000B	GET_ STATUS	Zéro	Zéro	Quatre	Etat de hub et Etat de changement

Le premier mot de données contient wHubStatus (voir Tableau 11-19). Le second mot de données contient wHubChange (voir Tableau 11-20).

Il s'agit d'une erreur de demande si *wValue*, *wIndex* ou *wLength* sont différents de ce qui est spécifié ci-dessus.

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

Tableau 11-19. Champ d'état de hub, wHubStatus

Bit	Description
0	**Local Power Source:** Il s'agit de la source de l'alimentation électrique locale. Ce champ indique si l'alimentation du hub (pour autre que le SIE) est fournie par une source externe ou par l'USB. Ce champ permet au logiciel du système USB de déterminer la quantité d'alimentation disponible d'un hub vers les appareils en aval. 0 = Alimentation électrique locale bonne 1 = Alimentation électrique locale perdue (inactive)
1	**Over-current:** Si le hub prend en charge la notification de surintensité pour chaque hub, ce champ indique que la somme de courant de tous les ports a dépassé le maximum spécifié et que tous les ports ont été placés dans l'état Hors tension. Si le hub signale une surintensité pour chaque port et n'a pas de fonctionnalités de détection de surintensité, ce champ est toujours égal à zéro. Pour plus de détails sur la protection contre les surintensités, voir Section 7.2.1.2.1. 0 = Aucune condition de surintensité n'existe actuellement. 1 = Une condition de surintensité de hub existe.
2-15	**Reserved** Ces bits retournent 0 en cas de lecture.

Il n'y a pas de valeurs de sélecteur de caractéristiques définies pour ces bits d'état et elles ne peuvent pas être définies ou effacées par le logiciel du système USB.

Tableau 11-20. Changement d'état de hub, wHubChange

Bit	Description
0	**Local Power Status Change:** (c_hub_local_power) Ce champ indique qu'un changement s'est produit dans le champ Local Power Source du hub dans *wHubStatus*. Ce champ est remis à zéro lorsque le hub reçoit une réinitialisation de bus. 0 = Aucun changement d'état d'alimentation locale n'a eu lieu. 1 = L'état d'alimentation locale a changé.
1	**Over-Current Change**: (c_hub_over_current) Ce champ indique qu'un changement s'est produit dans le champ Over-Current dans *wHubStatus*. Ce champ est remis à zéro lorsque le hub reçoit une réinitialisation de bus. 0 = Aucun changement d'état de surintensité n'a eu lieu. 1 = L'état de surintensité a changé.
2-15	**Reserved** Ces bits retournent 0 en cas de lecture.

Les hubs peuvent autoriser le paramétrage de ces bits de changement avec les demandes SetHubFeature() pour prendre en charge le diagnostic. Si le hub ne prend pas en charge le paramétrage de ces bits, il convient de traiter la demande SetHubFeature() comme une erreur de demande ou comme un non-fonctionnement fonctionnel. Lorsqu'ils sont définis, ces bits peuvent être effacés par une demande ClearHubFeature(). Une demande de configurer une caractéristique qui est déjà définie ou d'effacer une caractéristique qui est déjà effacée n'a pas d'effet et le hub n'échoue pas à la demande.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

11.24.2.7 Get Port Status

Cette demande retourne l'état du port actuel et la valeur actuelle des bits de changement d'état de port.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
10100011B	GET_STATUS	Zéro	Port	Quatre	Etat de port et Etat de changement

Le numéro de port doit être un numéro de port valide pour ce hub, supérieur à zéro.

Le premier mot de données contient wPortStatus (voir Tableau 11-21). Le second mot de données contient wPortChange (voir Tableau 11-20).

Les positions binaires dans les champs wPortStatus et wPortChange correspondent d'une manière biunivoque si applicable.

Il s'agit d'une erreur de demande si *wValue* ou *wLength* sont différents de ce qui est spécifié ci-dessus ou si *wIndex* spécifie un port qui n'existe pas.

Si le hub n'est pas configuré, le comportement du hub en réponse à cette demande est indéfini.

11.24.2.7.1 Bits d'état de port

Tableau 11-21. Champ d'état de port, wPortStatus

Bit	Description
0	**Current Connect Status:** (port_connection) Ce champ reflète si un appareil est ou non actuellement connecté à ce port. 0 = Aucun appareil n'est présent. 1 = Un appareil est présent sur ce port.
1	**Port Enabled/Disabled:** (port_enable) Les ports peuvent être activés par le logiciel du système USB uniquement. Les ports peuvent être désactivés par une condition de défaut (évènement de déconnexion ou autre condition de défaut) ou par le logiciel du système USB. 0 = Le port est désactivé. 1 = Le port est activé.
2	**Suspend:** (port_suspend) Ce champ indique si l'appareil sur ce port est ou non suspendu. La définition de ce champ entraîne la suspension de l'appareil par non-propagation du trafic du bus en aval. Ce champ peut être réinitialisé par une demande ou par une signalisation de reprise à partir de l'appareil raccordé au port. 0 = Non suspendu. 1 = Suspendu ou reprise.
3	**Over-current:** (port_over_current) Si le hub signale les conditions de surintensité par port, ce champ indique que le courant débité sur le port dépasse le maximum spécifié. Pour plus de détails, voir Section 7.2.1.2.1. 0 = Aucune condition de surintensité n'existe sur ce port. 1 = Une condition de surintensité existe sur ce port.
4	**Reset**: (port_reset) Ce champ est défini lorsque l'hôte souhaite réinitialiser l'appareil raccordé. Il reste défini jusqu'à ce que la signalisation de réinitialisation soit désactivée par le hub. 0 = Signalisation de réinitialisation pas affirmée. 1 = Signalisation de réinitialisation affirmée.
5-7	**Reserved** Ces bits retournent 0 en cas de lecture.
8	**Port Power:** (port_power) Ce champ reflète un état de commande d'alimentation logique d'un port. Comme les hubs peuvent appliquer différentes méthodes de commutation d'alimentation de port, ce champ peut ou non indiquer si l'alimentation est appliquée au port. Le descripteur d'appareil signale le type de commutation d'alimentation appliquée par le hub. 0 = Ce port est dans l'état Hors tension. 1 = Ce port n'est pas dans l'état Hors tension.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Bit	Description
9	**Low- Speed Device Attached:** (port_low_speed) Cela est pertinent uniquement si un appareil est raccordé. 0 = Appareil à vitesse intégrale ou à grande vitesse raccordé à ce port (déterminé par le bit 10). 1 = Appareil à basse vitesse raccordé à ce port.
10	**High-speed Device Attached:** (PORT_HIGH_SPEED) Cela est pertinent uniquement si un appareil est raccordé. 0 = Appareil à vitesse intégrale raccordé à ce port. 1 = Appareil à grande vitesse raccordé à ce port.
11	**Port Test Mode**: (PORT_TEST) Ce champ reflète le statut du mode essai du port. Le logiciel utilise les demandes SetPortFeature() et ClearPortFeature() pour manipuler le mode essai du port. 0 = Ce port n'est pas dans le mode Essai Port. 1 = Ce port est dans le mode Essai Port.
12	**Port Indicator Control**: (PORT_INDICATOR) Ce champ est défini pour refléter la commande logicielle de l'indicateur de port. Pour plus de détails, voir Sections 11.5.3, 11.24.2.7.1.10 et 11.24.2.13. 0 = L'indicateur de port affiche les couleurs par défaut. 1 = L'indicateur de port affiche la couleur commandée par logiciel.
13-15	**Reserved** Ces bits retournent 0 en cas de lecture.

11.24.2.7.1.1 PORT_CONNECTION

Lorsque le bit Alimentation Port est un, ce bit indique si un appareil est raccordé ou non. Ce champ indique un lorsqu'un appareil est raccordé; il indique zéro lorsqu'aucun appareil n'est raccordé. Ce bit est remis à zéro lorsque le port est dans l'état Hors tension ou Déconnecté. Il est défini sur un lorsque le port est dans l'état Hors tension, un raccordement d'appareil est détecté (voir Section 7.1.7.3), et le port passe de l'état Déconnecté à l'état Désactivé.

Les demandes SetPortFeature(PORT_CONNECTION) et ClearPortFeature(PORT_CONNECTION) ne doivent pas être utilisées par le logiciel du système USB et doivent être traitées comme des demandes de non-fonctionnement par les hubs.

11.24.2.7.1.2 PORT_ENABLE

Ce bit est défini lorsque le port peut envoyer ou recevoir des données en paquet ou la signalisation de reprise.

Ce bit peut être défini uniquement comme résultat d'une demande SetPortFeature(PORT_RESET). Lorsque le hub quitte l'état exits Réinitialisation ou, si présent, l'état Speed_eval, ce bit est défini et le trafic du bus peut être transmis au port. Ce bit peut être effacé en résultat de ce qui suit:

- Port étant dans l'état Hors tension
- Réception d'une demande ClearPortFeature(PORT_ENABLE)
- Détection Port_Error
- Détection de déconnexion
- Lorsque le port entre dans l'état Réinitialisation suite à la réception de la demande SetPortFeature(PORT_RESET)

La réponse du hub à une demande SetPortFeature(PORT_ENABLE) n'est pas spécifiée. Le comportement préféré est que le hub réponde avec une erreur de demande. Cela ne peut pas être utilisé par le logiciel du système USB. La demande ClearPortFeature(PORT_ENABLE) est prise en charge comme spécifié dans la Section 11.5.1.4.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.24.2.7.1.3 PORT_SUSPEND

Ce bit est défini lorsque le port est suspendu de manière sélective par le logiciel du système USB. Alors que ce bit est défini, le hub ne propage pas le trafic orienté en aval vers ce port, mais il répond à la signalisation de reprise du port. Ce bit peut être défini uniquement si le bit PORT_ENABLE du port est défini et le hub reçoit une demande SetPortFeature(PORT_SUSPEND). Ce bit est remis à zéro lors du passage de l'état SendEOP à l'état Activé, ou lors du passage de l'état Restart_S à l'état Transmettre, ou en cas d'évènement entraînant l'effacement du bit PORT_ENABLE alors que le bit PORT_SUSPEND est défini.

La demande SetPortFeature(PORT_SUSPEND) peut être émise par le logiciel du système USB à tout moment, mais s'applique uniquement comme spécifié dans la Section 11.5.

11.24.2.7.1.4 PORT_OVER-CURRENT

Ce bit est défini sur un lorsqu'une condition de surintensité existe sur le port. Ce bit est effacé lorsqu'aucune condition de surintensité n'existe sur le port.

Si la tension sur ce port est affectée par une condition de surintensité sur un autre port, ce bit est défini et reste défini tant que la condition de surintensité sur le port concerné n'est pas éliminée. Lorsque la condition de surintensité sur le port concerné est éliminée, ce bit est remis à zéro si aucune condition de surintensité n'existe sur ce port.

La protection contre les surintensités est requise sur les hubs autoalimentés (en option sur les hubs alimentés par bus) comme indiqué dans la Section 7.2.1.2.1.

Les demandes SetPortFeature(PORT_OVER_CURRENT) et ClearPortFeature(PORT_OVER_CURRENT) ne doivent pas être utilisées par le logiciel du système USB et peuvent être traitées comme demandes de non-fonctionnement par les hubs.

11.24.2.7.1.5 PORT_RESET

Ce bit est défini lorsque le port est dans l'état Réinitialisation. Une demande SetPortFeature(PORT_RESET) initie l'état Réinitialisation si les conditions dans la Section 11.5.1.5 sont remplies. Ce bit est remis à zéro lorsque le port est dans l'état Hors tension.

La demande ClearPortFeature(PORT_RESET) ne doit pas être utilisée par le logiciel du système USB et peut être traitée comme une demande de non-fonctionnement par les hubs.

11.24.2.7.1.6 PORT_POWER

Ce bit reflète l'état d'alimentation actuel d'un port. Ce bit est appliqué sur tous les ports si des appareils de commutation d'alimentation de port réels sont présents ou non.

Lorsque ce bit est égal à zéro, le port est dans l'état Hors tension. De manière similaire, tout ce qui fait que ce port entre dans l'état Hors tension entraîne la remise à zéro de ce bit.

Une demande SetPortFeature(PORT_POWER) place ce bit sur un à moins que C_HUB_LOCAL_POWER et Local Power Status (dans *wHubStatus*) ne soient définis sur un, dans quel cas la demande est traitée comme un non-fonctionnement fonctionnel.

Ce bit peut être effacé dans les circonstances suivantes:

- Le hub reçoit une demande ClearPortFeature(PORT_POWER).
- Une condition de surintensité existe sur le port.
- Une condition de surintensité sur un autre port entraîne l'arrêt d'alimentation de ce port.

Les demandes SetPortFeature(PORT_POWER) et ClearPortFeature(PORT_POWER) peuvent être émises par le logiciel du système USB lorsque le port n'est pas dans l'état Non configuré, mais s'applique uniquement comme spécifié dans la Section 11.11.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.24.2.7.1.7 PORT_LOW_SPEED

Ce bit a uniquement du sens lorsque le bit PORT_ENABLE est défini. Ce bit est défini sur un si l'appareil raccordé est à basse vitesse. Si ce bit est défini sur zéro, l'appareil raccord est à vitesse intégrale ou grande vitesse comme déterminé par le bit 10 (PORT_HIGH_SPEED, voir ci-après).

Les demandes SetPortFeature(PORT_LOW_SPEED) et ClearPortFeature(PORT_LOW_SPEED) ne doivent pas être utilisées par le logiciel du système USB et peuvent être traitées comme demandes de non-fonctionnement par les hubs.

11.24.2.7.1.8 PORT_HIGH_SPEED

Ce bit a uniquement du sens lorsque le bit PORT_ENABLE est défini et le bit PORT_LOW_SPEED est défini sur zéro. Ce bit est défini sur un si l'appareil raccordé est à grande vitesse. Le bit est défini sur zéro si l'appareil raccordé est à vitesse intégrale.

Les demandes SetPortFeature(PORT_HIGH_SPEED) et ClearPortFeature(PORT_HIGH_SPEED) ne doivent pas être utilisées par le logiciel du système USB et peuvent être traitées comme demandes de non-fonctionnement par les hubs.

11.24.2.7.1.9 PORT_TEST

Quand le bit Port Test Mode est défini sur un (1B), le port est en mode d'essai. Le mode d'essai spécifique est spécifié dans la demande SetPortFeature(PORT_TEST) par le sélecteur d'essai. Le hub ne fournit pas de mécanisme normalisé pour signaler le mode d'essai spécifique; par conséquent, le logiciel système doit suivre quel sélecteur d'essai a été utilisé. Voir Section 7.1.20 pour plus de détails sur chaque mode d'essai. Voir Section 11.24.2.13 pour plus d'informations sur l'utilisation de SetPortFeature pour contrôler le mode d'essai.

Ce champ peut uniquement être défini suite à une demande SetPortFeature(PORT_TEST). Une demande PORT_TEST du port est uniquement valide sur un port qui est dans l'état Désactivé, Déconnecté ou Suspendu. Si le port n'est pas dans un de ces états, le hub doit répondre avec une erreur de demande.

Ce champ peut être effacé suite à la réinitialisation du hub.

11.24.2.7.1.10 PORT_INDICATOR

Lorsque le statut du port Indicateur de port est défini sur un (1B), le sélecteur d'indicateur de port est différent de zéro. L'indicateur spécifique est spécifié dans la demande SetPortFeature(PORT_INDICATOR) par le sélecteur d'indicateur. Le GetPortStatus(PORT_INDICATOR) ne fournit pas de mécanisme normalisé pour signaler un mode d'indicateur spécifique; par conséquent, le logiciel système doit suivre quel mode d'indicateur a été utilisé. Voir Sections 11.5.3 et 11.24.2.13 pour plus de détails sur chaque mode d'indicateur.

Ce champ ne peut être défini que suite à une demande SetPortFeature(PORT_INDICATOR).

Ce champ peut être effacé suite à une demande SetPortFeature(PORT_INDICATOR) avec sélecteur d'indicateur = Par défaut ou une demande ClearPortFeature(PORT_INDICATOR).

Cette caractéristique doit être définie lorsque le logiciel hôte détecte une erreur sur un port qui nécessite l'intervention de l'utilisateur. Cette caractéristique doit être utilisée par le logiciel système s'il détermine qu'une des conditions suivantes est vraie:

- Un appareil grande puissance est branché dans un port basse puissance.
- Un appareil défectueux est branché dans un port (conditions de babillage, erreurs excessives, etc.).
- Une condition de surintensité se produit et fait que le logiciel ou le matériel définit l'indicateur.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Le bit d'état PORT_OVER_CURRENT définit la couleur de l'indicateur de port par défaut sur orange. La définition de la caractéristique PORT_POWER désactive l'indicateur.

Cette caractéristique est également utilisée lorsque le logiciel hôte détermine qu'un port nécessite l'attention de l'utilisateur. De nombreuses conditions d'erreur peuvent être résolues si l'utilisateur déplace un appareil d'un port vers un autre qui a les fonctionnalités appropriées.

Un scénario type est quand un utilisateur branche un appareil grande puissance dans un hub alimenté par bus. Si un port grande puissance est disponible, il peut alors être demandé à l'utilisateur de déplacer l'appareil du port basse puissance vers le port grande puissance.

1. Le logiciel hôte cycle la caractéristique PORT_INDICATOR du port basse puissance pour faire clignoter l'indicateur et afficher un message demandant à l'utilisateur de débrancher l'appareil du port avec l'indicateur clignotant.
2. A l'aide de la caractéristique de changement d'état C_PORT_CONNECTION, le logiciel hôte peut déterminer quand l'utilisateur a retiré physiquement l'appareil du port basse puissance.
3. Le logiciel hôte émet ensuite un ClearPortFeature(PORT_INDICATOR) au port basse tension (restaurant la couleur par défaut), commence à cycler le PORT_INDICATOR du port grande puissance, et affiche un message indiquant à l'utilisateur de brancher l'appareil dans le port avec l'indicateur clignotant.
4. A l'aide de la caractéristique de changement d'état C_PORT_CONNECTION, le logiciel hôte peut déterminer quand l'utilisateur a inséré physiquement l'appareil dans le port grande puissance.

Le logiciel hôte doit cycler la caractéristique PORT_INDICATOR pour faire clignoter la couleur actuelle à une vitesse d'environ 0,5 Hz avec un cycle de service de 30-50 %.

11.24.2.7.2 Bits de changement d'état de port

Les bits de changement d'état de port sont utilisés pour indiquer les changements de bits d'état de port qui ne sont pas le résultat direct de demandes. Les bits de changement d'état de port peuvent être effacés avec une demande ClearPortFeature() ou par une réinitialisation du hub. Les hubs peuvent permettre la configuration des bits de changement d'état avec une demande SetPortFeature() à des fins de diagnostic. Si un hub ne prend pas en charge le paramétrage des bits de changement d'état, il peut traiter la demande comme une erreur de demande ou comme un non-fonctionnement fonctionnel. Le Tableau 11-22 décrit les divers bits dans le champ *wPortChange*.

Tableau 11-22. Champ de changement de port, wPortChange

Bit	Description
0	**Connect Status Change:** (c_port_connection) Indique qu'un changement s'est produit dans le statut de connexion actuel du port. L'appareil du hub définit ce champ comme décrit dans la Section 11.24.2.7.2.1. 0 = Aucun changement du statut de connexion actuel n'a eu lieu. 1 = Le statut de connexion actuel a changé.
1	**Port Enable/Disable Change:** (c_port_enable) Ce champ est défini sur un lorsqu'un port est désactivé en raison d'une condition Port_Error (voir Section 11.8.1).
2	**Suspend Change:** (c_port_suspend) Ce champ indique un changement de l'état suspension visible de l'hôte de l'appareil raccordé. Il indique que l'appareil a quitté l'état Suspension. Ce champ est défini uniquement lorsque le processus de reprise entier est terminé. Le hub a ainsi cessé la reprise de signalisation sur ce port. 0 = Aucun changement. 1 = Reprise terminée.
3	**Over-Current Indicator Change:** (c_port_over_current) Ce champ s'applique uniquement aux hubs qui signalent des conditions de surintensité par port (comme signalé dans le descripteur du hub). 0 = Aucun changement de l'indicateur de surintensité n'a eu lieu. 1 = L'indicateur de surintensité a changé. Si le hub ne signale pas les surintensités pour chaque port, ce champ est alors toujours égal à zéro.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Bit	Description
4	**Reset Change:** (c_port_reset) Ce champ est défini lorsque le traitement de réinitialisation sur ce port est terminé. 0 = Aucun changement. 1 = Réinitialisation terminée.
5-15	**Reserved** Ces bits retournent 0 en cas de lecture.

11.24.2.7.2.1 C_PORT_CONNECTION

Ce bit est défini lorsque le bit PORT_CONNECTION change en raison d'un évènement de détection de raccordement ou de déconnexion (voir Section 7.1.7.3). Ce bit est remis à zéro par une demande ClearPortFeature(C_PORT_CONNECTION) ou lorsque le port est dans l'état Hors tension.

11.24.2.7.2.2 C_PORT_ENABLE

Ce bit est défini lorsque le bit PORT_ENABLE passe de un à zéro suite à une condition Port Error (voir Section 11.8.1). Ce bit n'est pas défini sur d'autres changements vers PORT_ENABLE.

Ce bit peut être défini si, sur un SetPortFeature(PORT_RESET), le port reste dans l'état Désactivé car un état inactif invalide existe sur le bus (voir Section 11.8.2).

Ce bit est effacé par une demande ClearPortFeature(C_PORT_ENABLE) ou lorsque le port est dans l'état Hors tension.

11.24.2.7.2.3 C_PORT_SUSPEND

Ce bit est défini lors des transitions suivantes:

- Lors du passage de l'état Reprise à l'état SendEOP
- Lors du passage de l'état Restart_S à l'état Transmission

Ce bit est effacé par une demande ClearPortFeature(C_PORT_SUSPEND) ou lorsque le port est dans l'état Hors tension.

11.24.2.7.2.4 C_PORT_OVER-CURRENT

Ce bit est défini lorsque le bit PORT_OVER_CURRENT passe de zéro à un ou de un à zéro. Ce bit est également défini si le port est placé dans l'état Hors tension en raison d'une condition de surintensité sur un autre port.

Ce bit est effacé lorsque le port est dans l'état Pas configuré ou par une demande ClearPortFeature(C_PORT_OVER_CURRENT).

11.24.2.7.2.5 C_PORT_RESET

Ce bit est défini lorsque le port passe de l'état Réinitialisation (ou, si présent, l'état Speed_eval) à l'état Activé.

Ce bit est effacé par une demande ClearPortFeature(C_PORT_RESET) ou lorsque le port est dans l'état Hors tension.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.24.2.8 Get_TT_State

Cette demande retourne l'état interne du traducteur de transactions dans un format spécifique au fournisseur. Un TT recevant cette demande doit d'abord être arrêté via la demande Stop_TT. Cette demande est fournie à des fins de débogage.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
10100011B	Get_TT_State	TT_Flags	TT_Port	Longueur d'état TT	Etat TT

Les TT_Flags bits 7..0 sont réservés pour une définition ultérieure de l'USB et doivent être mis à zéro. Les TT_Flags bits 15..8 sont réservés à un usage spécifique au fournisseur.

L'état TT retourné dans l'étage de données du transfert de commande pour cette demande est illustré dans le Tableau 11-23.

Tableau 11-23. Format de l'état TT retourné

Décalage	Champ	Taille (octets)	Commentaires
0	TT_Return_Flags	4	Les bits 15..0 sont réservés pour une définition ultérieure de l'USB et doivent être mis à zéro. Les bits 31..16 sont réservés à un usage spécifique au fournisseur.
4	TT_specific_state	En fonction de la mise en œuvre	

Si le hub prend en charge de multiples TT, *wIndex* doit alors spécifier le numéro du port du TT qui va retourner TT_state. Si le hub fournit uniquement un seul TT, le port doit alors être défini sur un.

L'état du TT après le traitement de cette demande est indéfini.

Il s'agit d'une erreur de demande, si *wIndex* spécifie un port qui n'existe pas. Si *wLength* est plus grand que la longueur réelle de cette demande, seule la longueur réelle est alors retournée. Si *wLength* est inférieur à la longueur réelle de cette demande, seuls les premiers octets *wLength* de cette demande sont alors retournés; cela n'est pas considéré comme une erreur même si *wLength* est égal à zéro.

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

11.24.2.9 Reset_TT

Cette demande remet le traducteur de transactions dans un hub dans un état connu.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
00100011B	RESET_TT	Zéro	TT_Port	Zéro	Néant

Dans certaines circonstances, un traducteur de transactions (TT) dans un hub peut être dans un état inconnu de sorte qu'il ne fonctionne plus correctement. La demande Reset_TT permet au TT d'être retourné dans l'état dans lequel il est juste après avoir configuré le hub. Reset_TT réinitialise uniquement les structures de données internes du TT (tampons) et le pipeline, ainsi que ses diagrammes d'états associés. Une fois la réinitialisation terminée, le TT peut reprendre son fonctionnement normal. La réinitialisation du TT est découplée des autres parties du hub (notamment les ports orientés en aval du hub, le répéteur de hub, le contrôleur de hub, etc.). Les autres parties du hub ne sont pas réinitialisées et peuvent poursuivre leur fonctionnement normal. Les ports orientés en aval ne sont pas réinitialisés de sorte que lorsque le TT reprend son fonctionnement normal, les appareils raccordés correspondants continuent de fonctionner; à savoir, aucun nouveau processus d'énumération n'est requis. Le travail des appareils à vitesse intégrale/basse vitesse en aval est interrompu uniquement pendant la réinitialisation du TT auquel ils appartiennent.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Si le hub prend en charge de multiples TT, *wIndex* doit alors spécifier le numéro du port du TT qui doit être réinitialisé. Si le hub fournit uniquement un seul TT, le port doit alors être défini sur un. Pour un hub à un seul TT, le hub peut ignorer le numéro de port.

Il s'agit d'une erreur de demande si *wIndex* spécifie un port qui n'existe pas, ou si *wLength* n'est pas comme spécifié ci-dessus.

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

11.24.2.10 Set Hub Descriptor

Cette demande écrase le descripteur de hub.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
00100000B	set_descriptor	Type de descripteur et Index de descripteur	Zéro	Longueur de descripteur	Descripteur

La demande SetDescriptor pour le descripteur de classe de hub suit le même modèle d'utilisation que celle de la demande SetDescriptor normalisée (voir Chapitre 9). Le descripteur de hub normalisé est indiqué en utilisant la valeur *bDescriptorType* définie dans la Section 11.23.2.1. Tous les hubs doivent appliquer un descripteur de hub, avec un index de descripteur zéro.

Cette demande est facultative. Cette demande écrit les données sur un descripteur spécifique à la classe. L'hôte fournit les données qui doivent être transférées au hub pendant la phase de transfert de données de la transaction de contrôle. Cette demande décrit le descripteur de hub entier en même temps.

Les hubs doivent mettre en tampon tous les octets reçus de cette demande pour garantir que le descripteur entier a été transmis avec succès par l'hôte. Lors de l'exécution réussie du transfert de bus, le hub met à jour le contenu du descripteur spécifié.

Il s'agit d'une erreur de demande si *wIndex* est différent de zéro ou si *wLength* ne correspond pas à la quantité de données envoyées par l'hôte. Les hubs qui ne prennent pas en charge cette demande répondent avec un STALL pendant l'étage de données de la demande.

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

11.24.2.11 Stop_TT

Cette demande arrête l'exécution normale du traducteur de transactions de sorte que l'état TT interne peut être extrait via Get_TT_State. Cette demande est fournie à des fins de débogage.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
00100011B	STOP_TT	Zéro	TT_Port	Zéro	Néant

La seule méthode normalisée pour redémarrer un TT après une demande Stop_TT est via la demande Reset_TT.

Si le hub prend en charge de multiples TT, *wIndex* doit alors spécifier le numéro du port du TT qui est arrêté. Si le hub fournit uniquement un seul TT, le port doit alors être défini sur un. Pour un hub à un seul TT, le hub peut ignorer le numéro de port.

Il s'agit d'une erreur de demande si *wIndex* spécifie un port qui n'existe pas, ou si *wLength* n'est pas comme spécifié ci-dessus.

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

11.24.2.12 Set Hub Feature

Cette demande définit une valeur indiquée dans l'état du hub.

bmRequestType	bRequest	wValue	wIndex	wLength	Données
00100000B	SET_ FEATURE	Sélecteur de caractéristiques	Zéro	Zéro	Néant

La définition d'une caractéristique active cette caractéristique; voir Tableau 11-17 pour les définitions de sélecteur de caractéristiques qui s'appliquent au hub en tant que destinataire. Les changements d'état ne peuvent pas être acquittés avec cette demande.

Il s'agit d'une erreur de demande si *wValue* n'est pas un sélecteur de caractéristiques listé dans le Tableau 11-17 ou si *wIndex* ou *wLength* ne sont pas comme spécifié ci-dessus.

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

11.24.2.13 Set Port Feature

Cette demande définit une valeur indiquée dans l'état du port.

bmRequestType	bRequest	wValue	wIndex		wLength	Données
00100011B	SET_ FEATURE	Sélecteur de caractéristiques	Sélecteur	Port	Zéro	Néant

Le numéro de port doit être un numéro de port valide pour ce hub, supérieur à zéro. Le numéro de port est dans l'octet de poids faible (bits 7..0) du champ *wIndex*. L'octet de poids le plus fort de *wIndex* est zéro, sauf lorsque le sélecteur de caractéristique est PORT_TEST.

La définition d'une caractéristique active cette caractéristique ou démarre un processus associé à cette caractéristique; voir Tableau 11-17 pour les définitions de sélecteur de caractéristique qui s'appliquent à un port en tant que destinataire. Le changement d'état ne peut pas être acquitté avec cette demande. Les caractéristiques qui peuvent être définies avec cette demande sont les suivantes:

- PORT_RESET
- PORT_SUSPEND
- PORT_POWER
- PORT_TEST
- PORT_INDICATOR
- C_PORT_CONNECTION*
- C_PORT_RESET*
- C_PORT_ENABLE*
- C_PORT_SUSPEND*
- C_PORT_OVER_CURRENT*

* Indique les caractéristiques qui ne sont pas tenues d'être définies par cette demande.

La définition de la caractéristique port_suspend entraîne la cessation du trafic du bus sur ce port et, par conséquent, la suspension de l'appareil. La définition de la caractéristique de réinitialisation PORT_RESET entraîne la réinitialisation du signal du hub sur ce port. Lorsque la signalisation de la réinitialisation est terminée, le hub définit le changement d'état C_PORT_RESET et active immédiatement le port. Voir aussi Section 11.24.2.7.1 pour plus de détails.

Lorsque le sélecteur de caractéristique est PORT_TEST, l'octet de poids fort (bits 15..8) du champ *wIndex* est le sélecteur identifiant le mode d'essai spécifique. Le Tableau 11-24 liste

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

les définitions de sélecteur d'essai. Voir Section 7.1.20 pour les définitions de chaque mode d'essai. Le mode d'essai d'un port orienté en aval peut uniquement être utilisé dans une séquence bien définie d'états de hub. Cette séquence est définie de la manière suivante:

1) Tous les ports orientés en aval activés du hub contenant le port à soumettre à essai doivent être (de manière sélective) suspendus via la demande SetPortFeature(PORT_SUSPEND). Chaque port orienté en aval du hub doit être dans l'état désactivé, déconnecté ou suspendu (voir Figure 11-9).
2) Une demande SetPortFeature(PORT_TEST) doit être émise vers le port orienté en aval à soumettre à essai. Un seul port orienté en aval peut être en test_mode à la fois. Le passage au mode essai doit être terminé au plus tard 3 ms après l'exécution de l'étage d'état de la demande.
3) Le port orienté en aval soumis à essai peut maintenant être soumis à essai.
4) Pendant test_mode, une déconnexion de port ou un changement d'état de reprise sur un des ports suspendus (sauf le port soumis à essai) doit entraîner un signalement de changement d'état (C_PORT_CONNECTION ou C_PORT_SUSPEND) (voir Section 11.12.3 et 11.24.2.7.2) du hub. Note: Les autres changements d'état peuvent ou ne peuvent pas être pris en charge dans un hub avec un port orienté en aval en mode essai. Le signalement de ces changements d'état peut permettre à une application d'essai de restaurer le fonctionnement normal d'un hub racine sans nécessiter de clavier non USB ou de souris pour la saisie de l'utilisateur. Par exemple, un appareil USB raccordé au hub racine peut être déconnecté pour signaler à l'application d'essai de restaurer le fonctionnement normal du hub racine.
5) Pendant test_mode, l'état des ports orientés en aval du hub ne doit pas être modifié par l'hôte (à savoir, les demandes de classe de hub autres que la demande Get_Port_Status() ne doivent pas être émises par l'hôte). Note: Le hub doit également être réinitialisé avant de pouvoir utiliser un SetPortFeature(PORT_TEST) pour placer le port dans un autre mode d'essai.
6) Une fois l'essai terminé, le hub (avec le port soumis à essai) doit être réinitialisé par l'hôte ou l'utilisateur. Cela doit être effectué en manipulant le port du hub parent auquel le hub soumis à essai est raccordé. Cette manipulation peut inclure un des points suivants:
   a) Emission d'un SetPortFeature(PORT_RESET) vers le port du hub parent auquel le hub soumis à essai est raccordé.
   b) Emission d'un ClearPortFeature(PORT_POWER) et SetPortFeature(PORT_POWER) pour arrêter-redémarrer un hub parent qui prend en charge la commande d'alimentation par port.
   c) Déconnexion et rebranchement du hub soumis à essai à partir de son port hub parent.
   d) Pour un hub racine soumis à essai, une réinitialisation du contrôleur hôte peut être requise comme il n'y a pas de hub parent du hub racine.
7) Le comportement du hub soumis à essai et de ses ports orientés en aval est indéfini si ces exigences ne sont pas satisfaites.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Tableau 11-24. Codes du sélecteur de mode essai

Valeur	Description du mode essai
0H	Réservé
1H	Test_J
2H	Test_K
3H	Test_SE0_NAK
4H	Test_Packet
5H	Test_Force_Enable
06H-3FH	Réservé pour les sélections d'essai normalisé
40H-BFH	Réservé
C0H-FFH	Réservé pour les sélections d'essai propres au fournisseur

Lorsque le sélecteur de caractéristique est PORT_INDICATOR, l'octet de poids fort du champ *wIndex* est le sélecteur identifiant le mode indicateur spécifique. Le Tableau 11-25 liste les définitions de sélecteur d'indicateur. Voir Sections 11.5.3 et 11.24.2.7.1.10 pour les détails d'indicateur. Le hub répond à une erreur de demande si la demande contient un sélecteur d'indicateur invalide.

Tableau 11-25. Codes de sélecteur d'indicateur de port

Valeur	Couleur d'indicateur de port	Mode d'indicateur de port
0	Couleur définie automatiquement, comme défini dans le Tableau 11-6	Automatique
1	Orange	Manuel
2	Verte	
3	Off	
4-FFH	Réservé	Réservé

Le hub doit satisfaire aux exigences suivantes:

- Si le port est dans l'état Hors tension, le hub doit traiter une demande SetPortFeature(PORT_RESET) comme un non-fonctionnement fonctionnel.
- Si le port est dans l'état Activé ou Transmission, le hub doit traiter une demande SetPortFeature(PORT_SUSPEND) comme un non-fonctionnement fonctionnel.
- Si le port n'est pas dans l'état Hors tension, le hub doit traiter une demande SetPortFeature(PORT_POWER) comme un non-fonctionnement fonctionnel.

Il s'agit d'une erreur de demande si *wValue* n'est pas un sélecteur de caractéristiques listé dans le Tableau 11-17, si *wIndex* spécifie un port qui n'existe pas ou si *wLength* ne sont pas comme spécifié ci-dessus.

Si le hub n'est pas configuré, la réponse du hub à cette demande est indéfinie.

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Annexe A Exemples de transactions

Cette annexe contient des exemples de transactions pour différents cas de transactions fractionnées. Les cas sont OUT en masse/de contrôle et SETUP, IN en masse/de contrôler, OUT d'interruption, IN d'interruption, OUT isochrone et IN isochrone.

A.1 Exemples de transactions OUT en masse/de contrôle et SETUP

Légende:

(S): Start Split

(C): Complete Split

Résumé des cas de transaction OUT en masse/de contrôle et SETUP

- Cas normaux

Cas	Figure de référence	Figure similaire
Pas de smash	Figure A-1	
Smash HS SSPLIT		Figure A-2
Smash 3 essais HS SSPLIT		Figure A-3
Smash HS OUT/SETUP(S)		Figure A-2
Smash 3 essais HS OUT/SETUP(S)		Figure A-3
Smash HS DATA0/1	Figure A-2	
Smash 3 essais HS DATA0/1	Figure A-3	
Smash HS ACK(S)	Figure A-4 Figure A-5	
Smash 3 essais HS ACK(S)	Figure A-6	
Smash HS CSPLIT	Figure A-7	
Smash 3 essais HS CSPLIT	Figure A-8	
Smash HS OUT/SETUP(C)		Figure A-7
Smash 3 essais HS OUT/SETUP(C)		Figure A-8
Smash HS ACK(C)	Figure A-9	
Smash 3 essais HS ACK(C)	Figure A-10	
Smash FS/LS OUT/SETUP		Figure A-11
Smash 3 essais FS/LS OUT/SETUP		Figure A-12
Smash FS/LS DATA0/1	Figure A-11	
Smash 3 essais FS/LS DATA0/1	Figure A-12	
Smash FS/LS ACK	Figure A-13	
Smash 3 essais FS/LS ACK	Figure A-14	

- Aucun tampon (sur hub) - Cas disponibles

Cas	Figure de référence	Figure similaire
Pas de smash(HS NAK(S))	Figure A-15	
Smash HS NAK(S)	Figure A-16	
Smash 3 essais HS NAK(S)	Figure A-17	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- CS(transaction complete-split) - Cas précédents

Cas	Figure de référence	Figure similaire
Pas de smash(HS NYET)	Figure A-18	
Smash HS NYET	Figure A-19 Figure A-20	
Smash 3 essais HS NYET	Figure A-21	

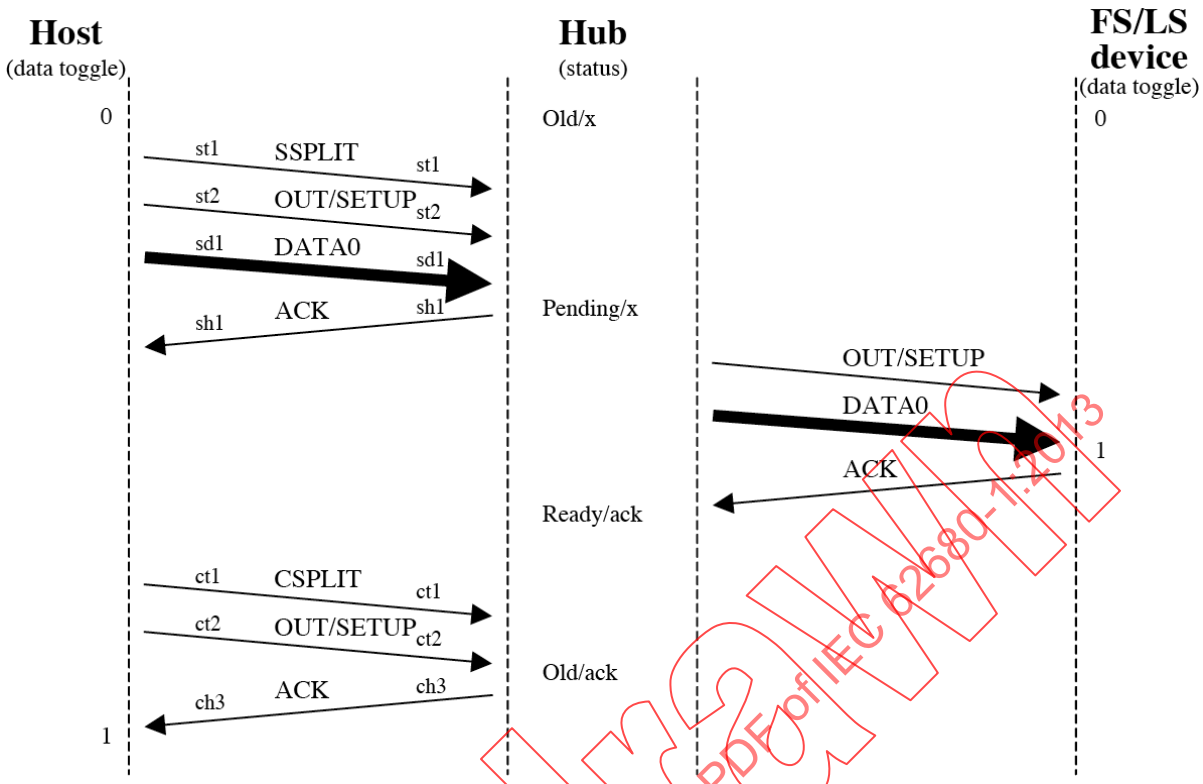
- Cas appareil occupé

Cas	Figure de référence	Figure similaire
Pas de smash(HS NAK(C))	Figure A-22	
Smash HS NAK(C)		Figure A-9
Smash 3 essais HS NAK(C)		Figure A-10
Smash FS/LS NAK		Figure A-13
Smash 3 essais FS/LS NAK		Figure A-14

- Cas appareil arrêté

Cas	Figure de référence	Figure similaire
Pas de smash	Figure A-23	
Smash HS STALL(C)		Figure A-9
Smash 3 essais HS STALL(C)		Figure A-10
Smash FS/LS STALL		Figure A-13
Smash 3 essais FS/LS STALL		Figure A-14

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

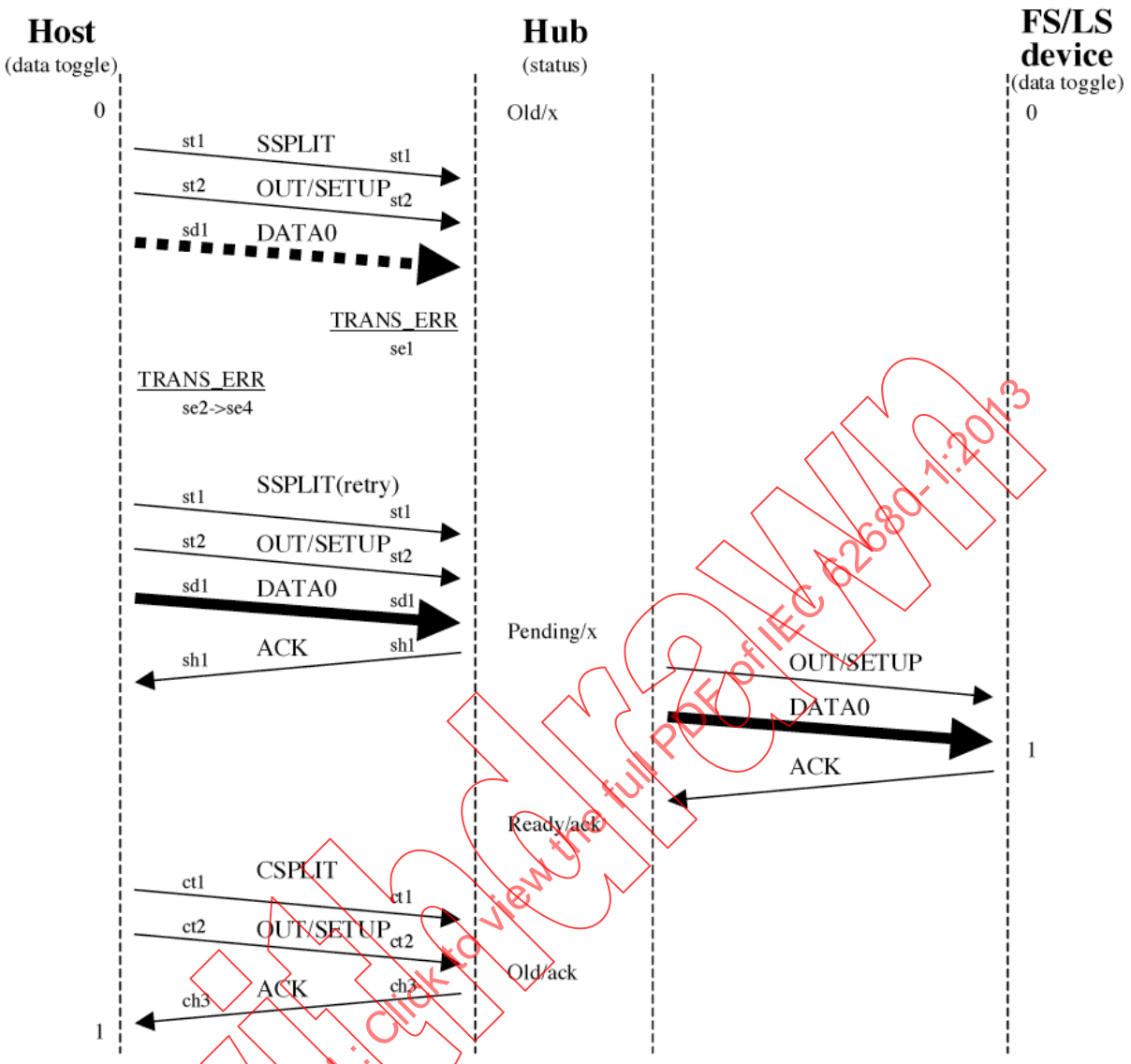


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack

Figure A-1. Normal Pas de smash

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

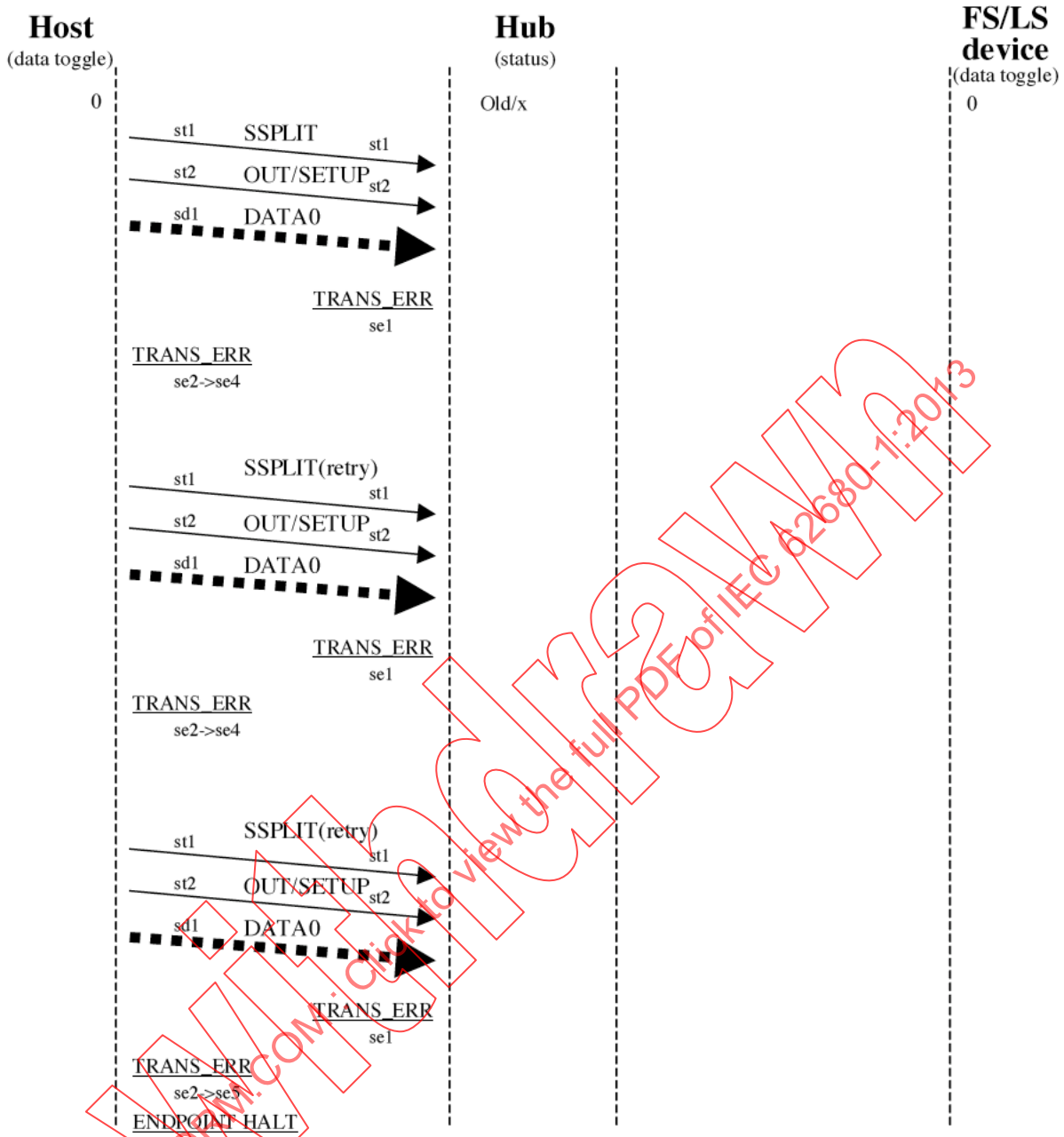


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack

Figure A-2. Smash HS DATA0/1 normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

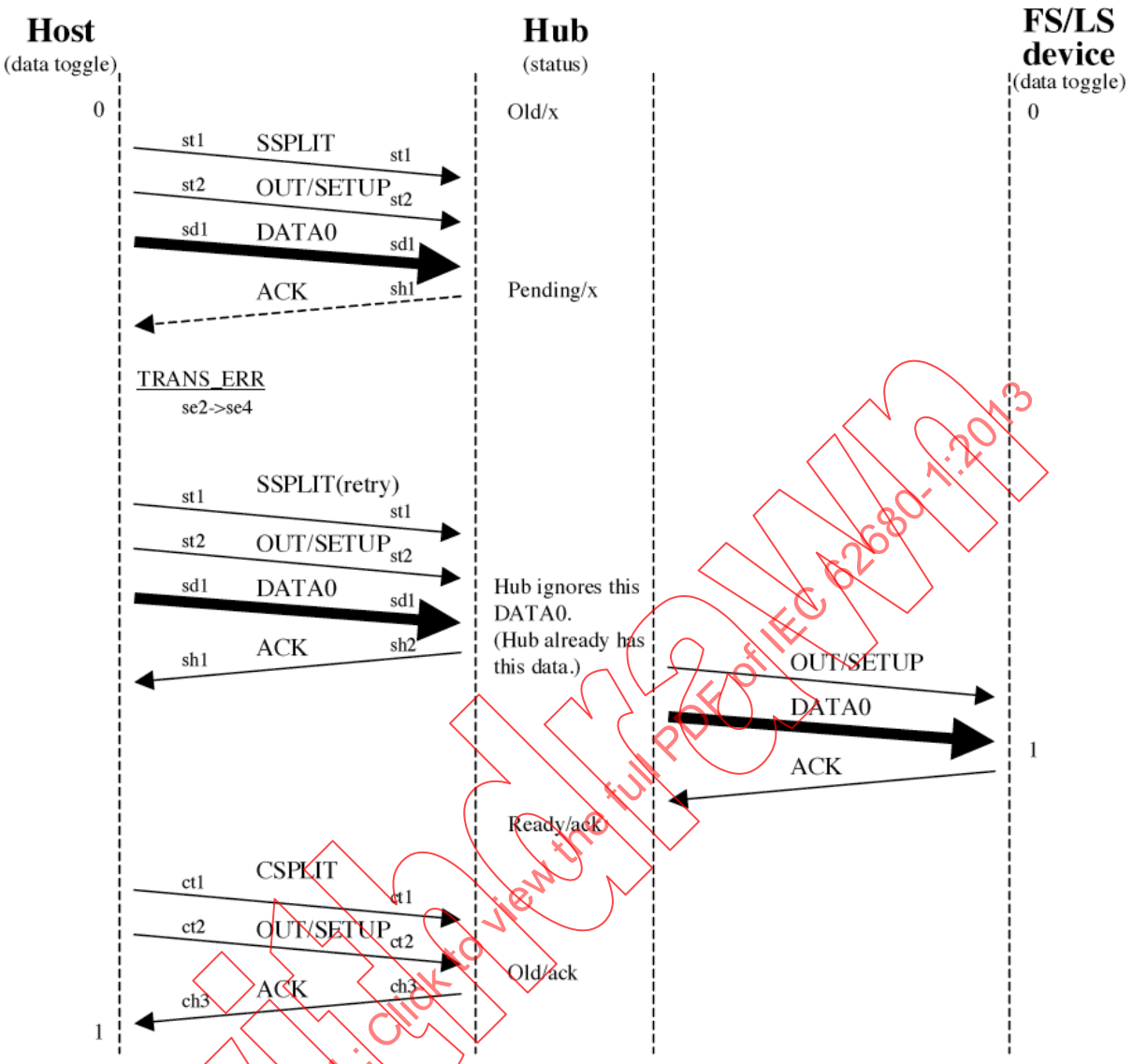


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
SSPLIT (retry)	SSPLIT (nouvel essai)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-3. Smash 3 essais HS DATA0/1 normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

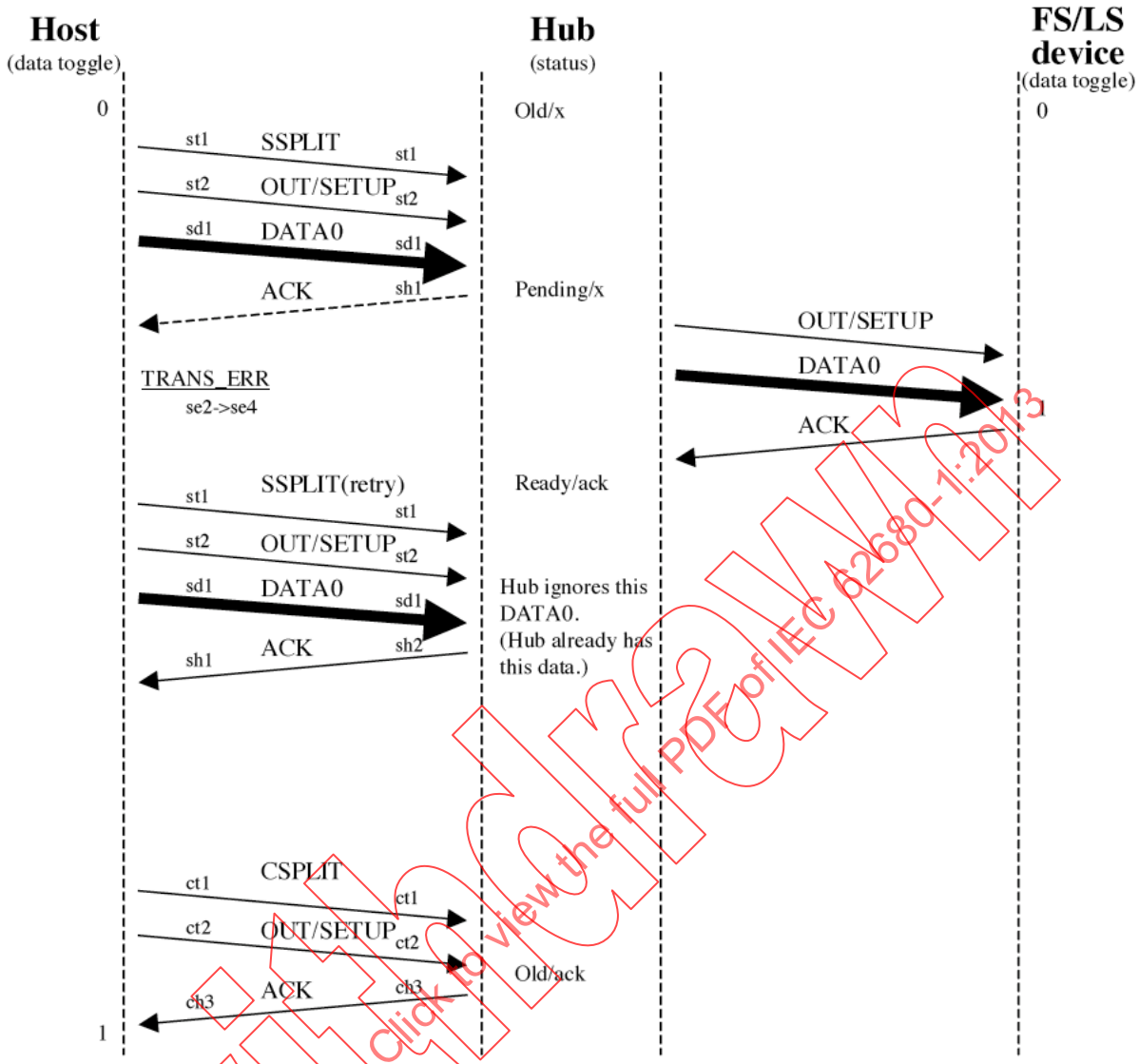


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack
SSPLIT (retry)	SSPLIT (nouvel essai)
Hub ignores this DATA0. (Hub already has this data).	Le hub ignore ce DATA0. (Le hub a déjà ces données).

Figure A-4. Smash HS ACK(S) normal (cas 1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

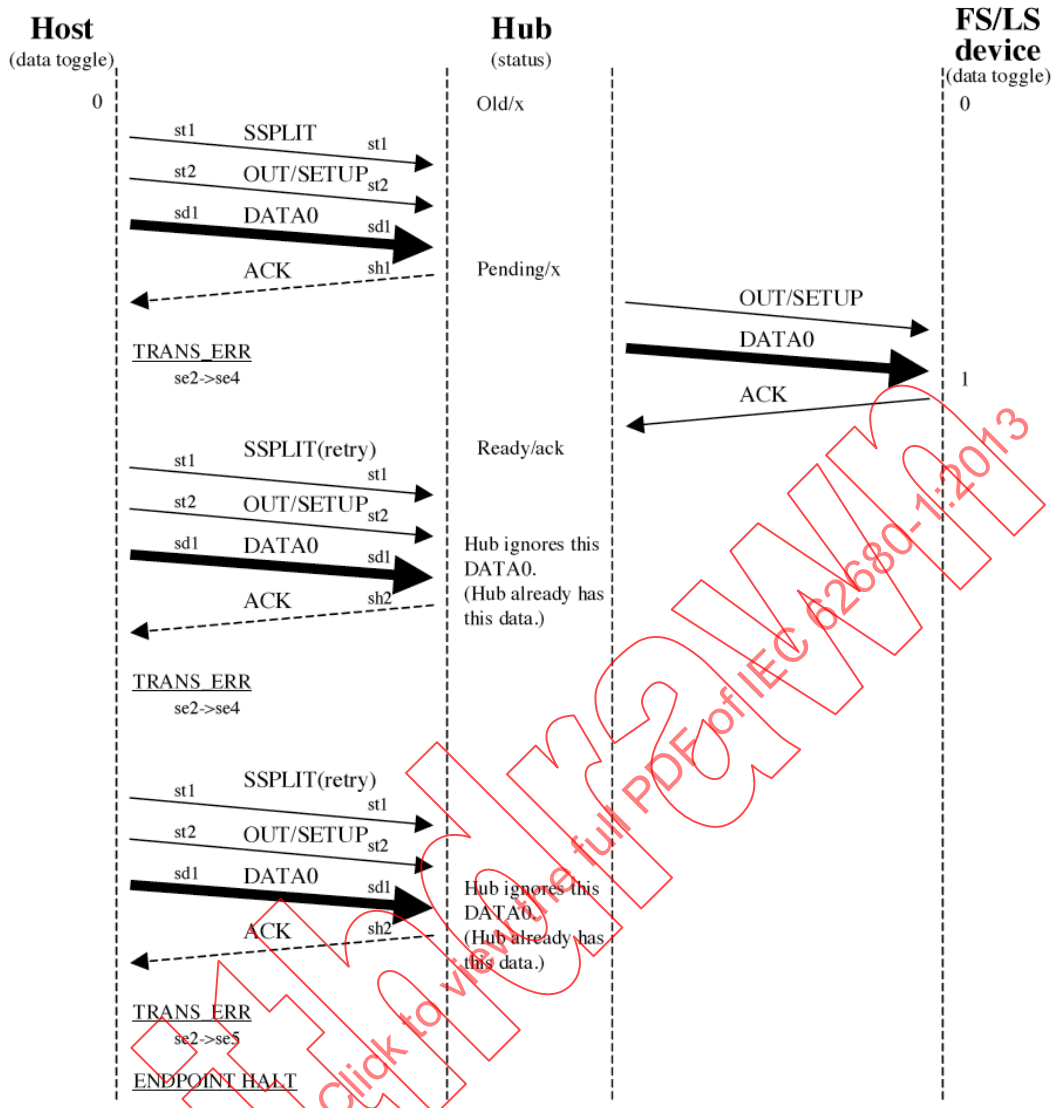


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack
SSPLIT (retry)	SSPLIT (nouvel essai)
Hub ignores this DATA0. (Hub already has this data).	Le hub ignore ce DATA0. (Le hub a déjà ces données).

Figure A-5. Smash HS ACK(S) normal (cas 2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

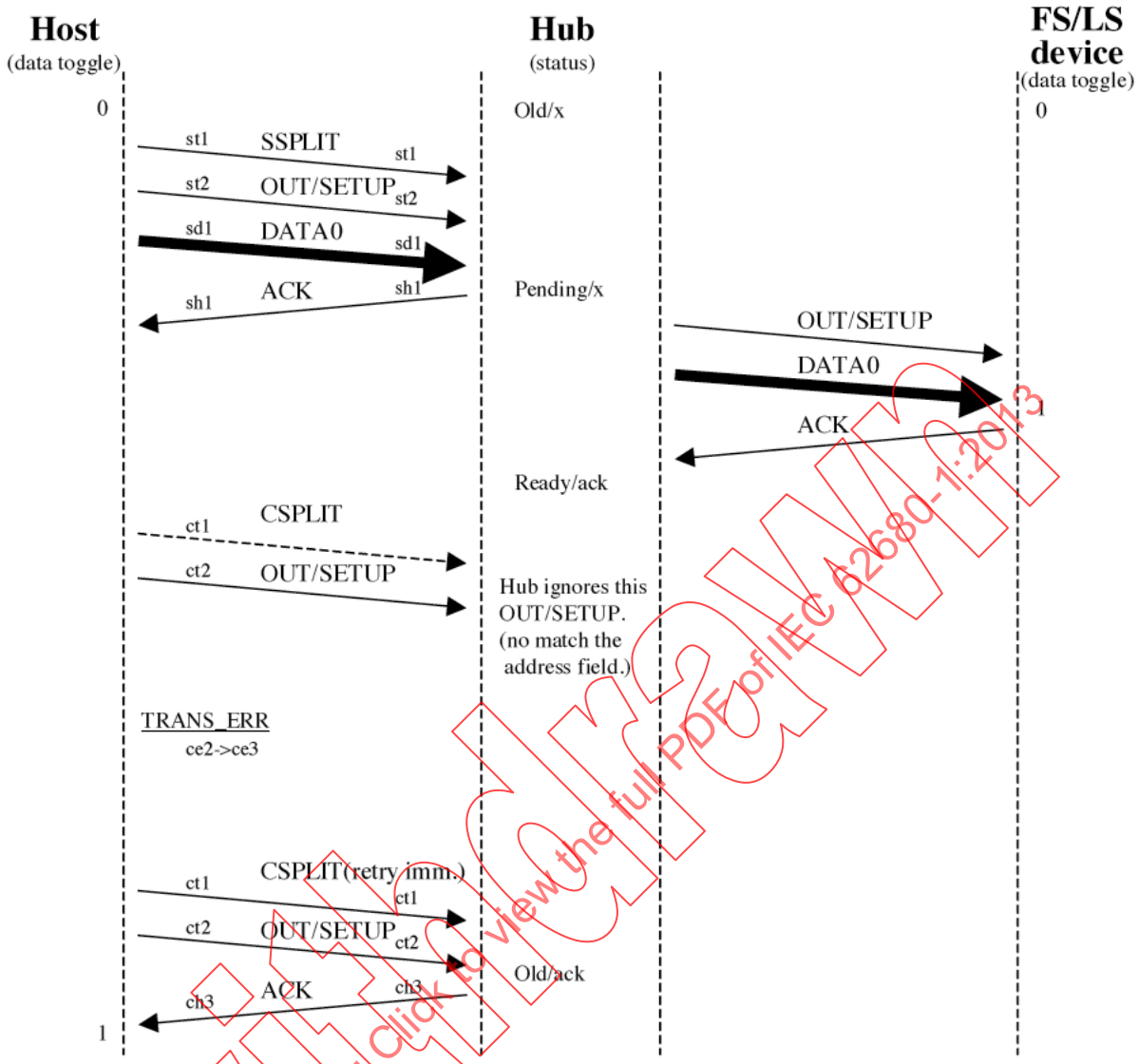


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
SSPLIT (retry)	SSPLIT (nouvel essai)
Hub ignores this DATA0. (Hub already has this data).	Le hub ignore ce DATA0. (Le hub a déjà ces données).
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-6. Smash 3 essais HS ACK(S) normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

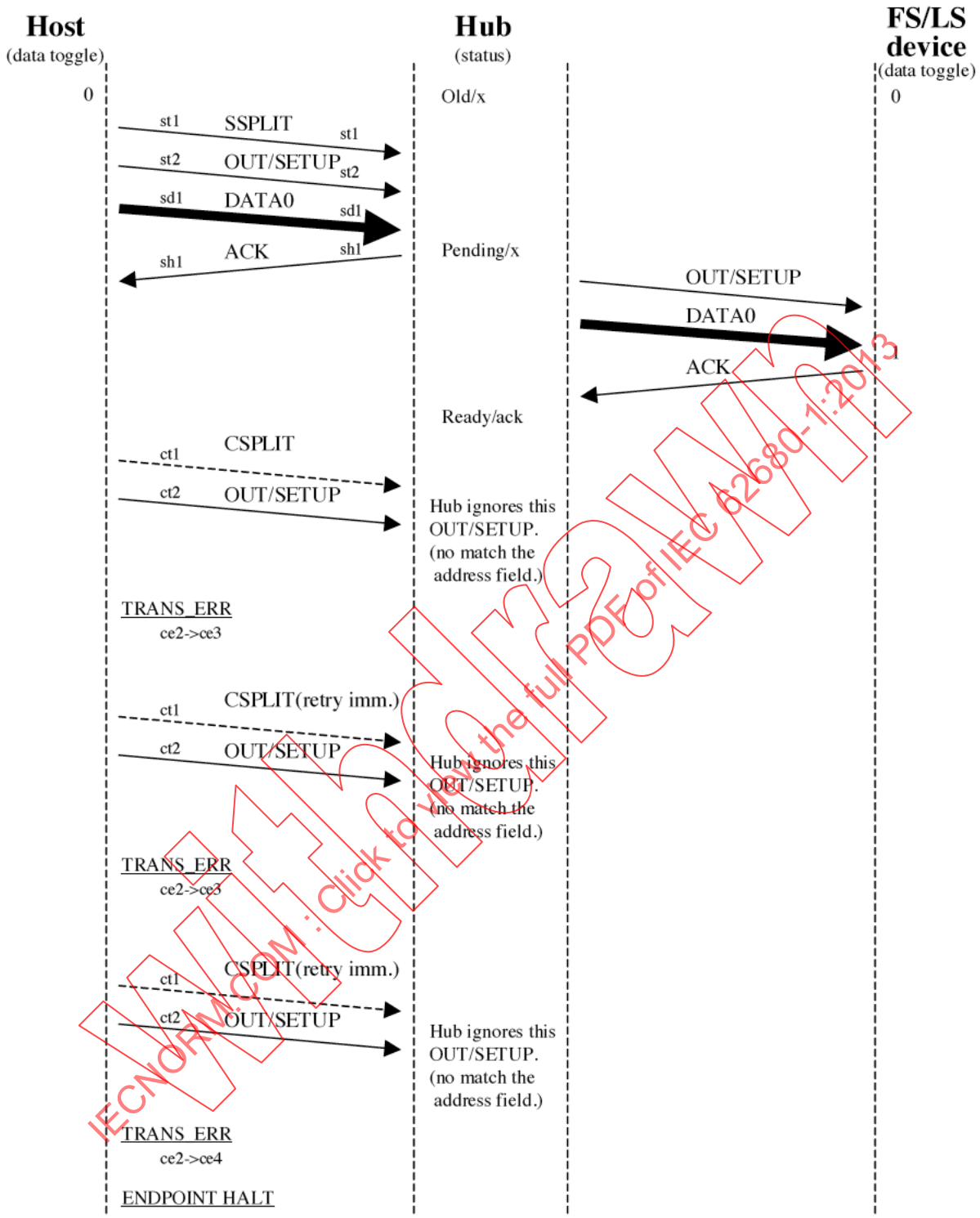


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
Hub ignores this OUT/SETUP. (no match the address field).	Le hub ignore ce OUT/SETUP. (pas de correspondance du champ d'adresse).

Figure A-7. Smash HS CSPLIT normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



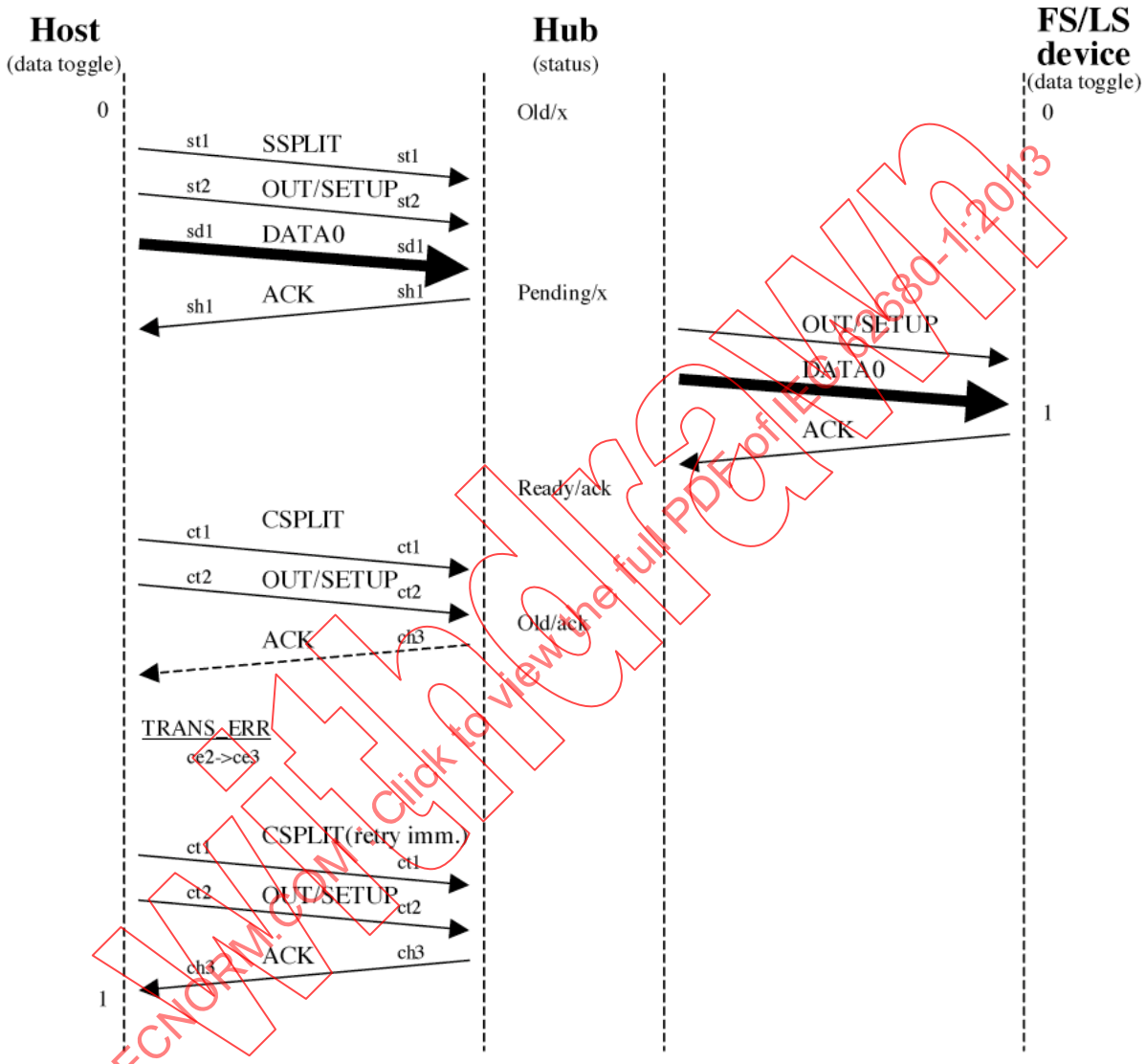
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x

Anglais	Français
Ready/ack	Prêt/ack
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
Hub ignores this OUT/SETUP. (no match the address field).	Le hub ignore ce OUT/SETUP. (pas de correspondance du champ d'adresse).
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-8. Smash 3 essais HS CSPLIT normal

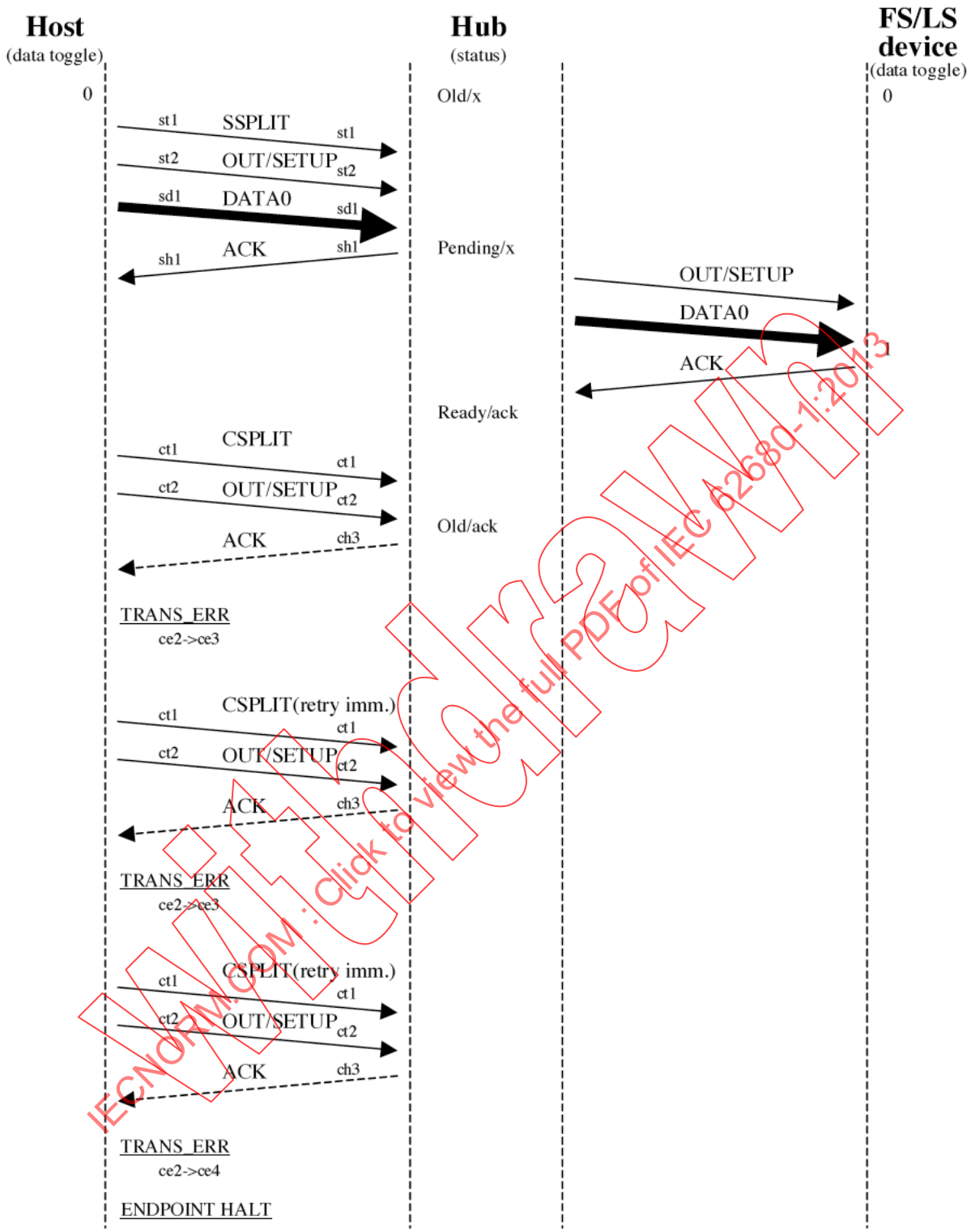


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
CSPLIT(retry imm.)	CSPLIT (nouvel essai imm.)
Old/ack	Ancien/ack

Figure A-9. Smash HS ACK(C) normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



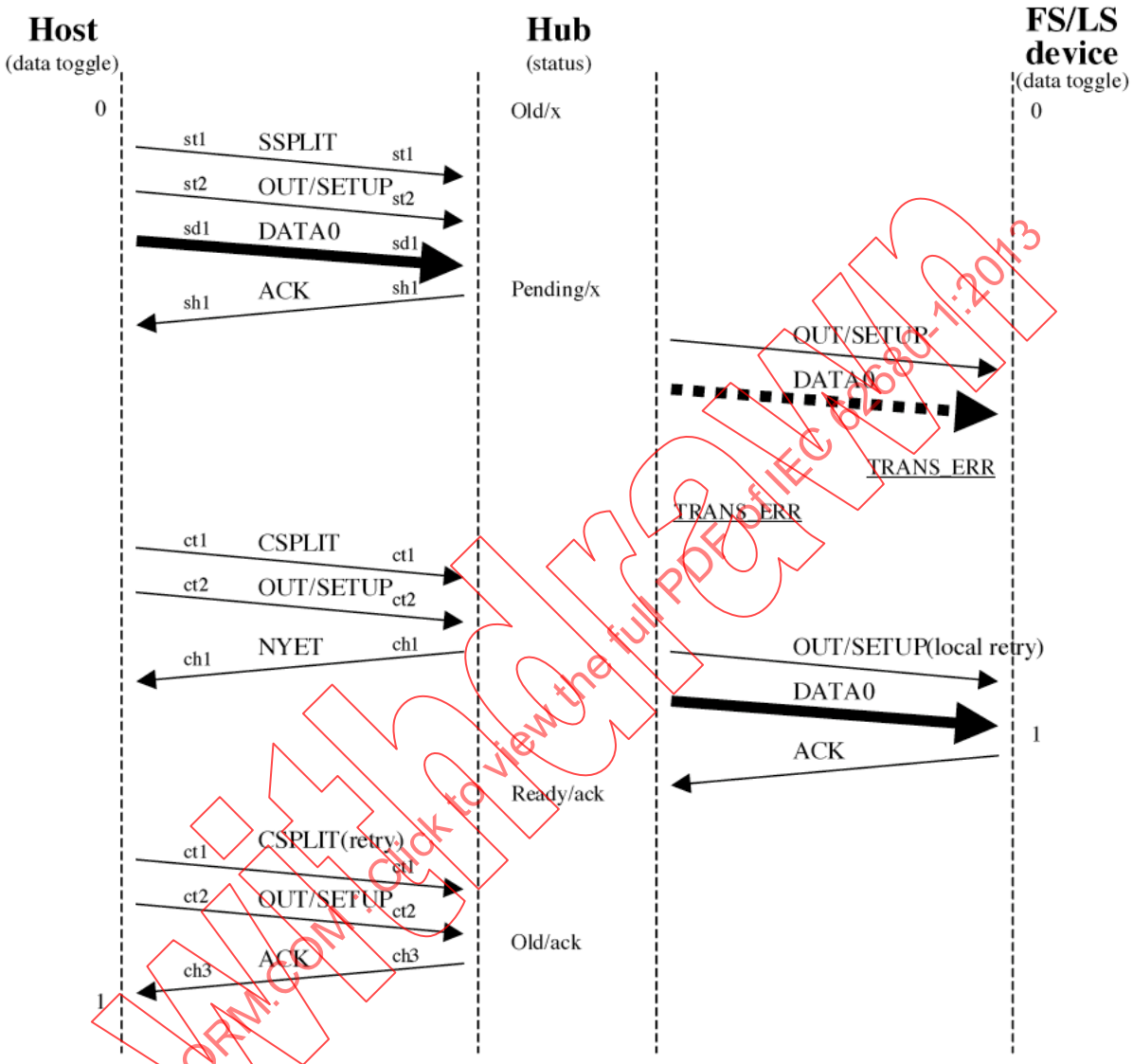
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack

Anglais	Français
Old/ack	Ancien/ack
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-10. Smash 3 essais S ACK(C) normal

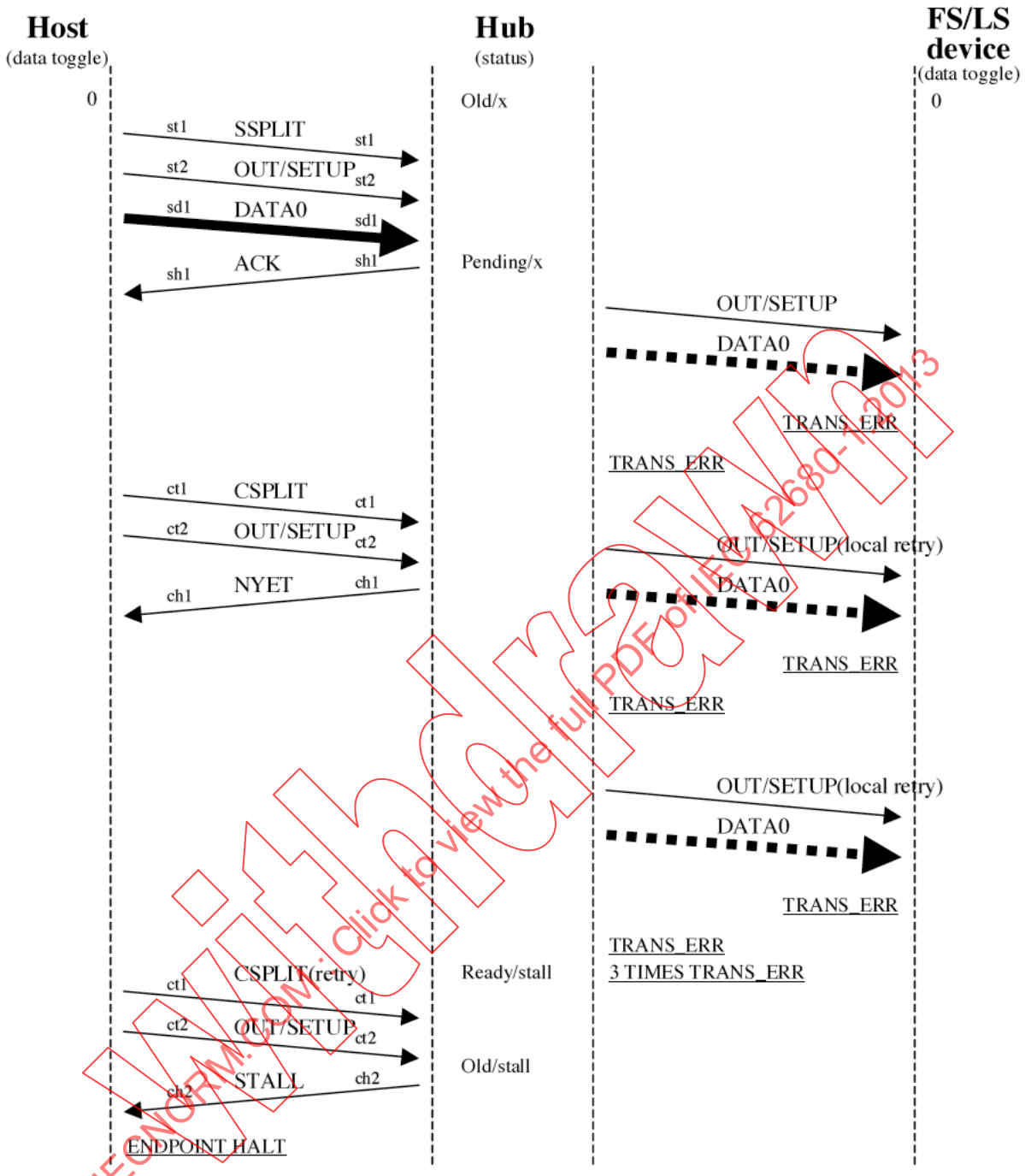


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack
CSPLIT (retry)	CSPLIT (nouvel essai)
OUT/SETUP (local retry)	OUT/SETUP (nouvel essai local)

Figure A-11. Smash FS/LS DATA0/1 normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

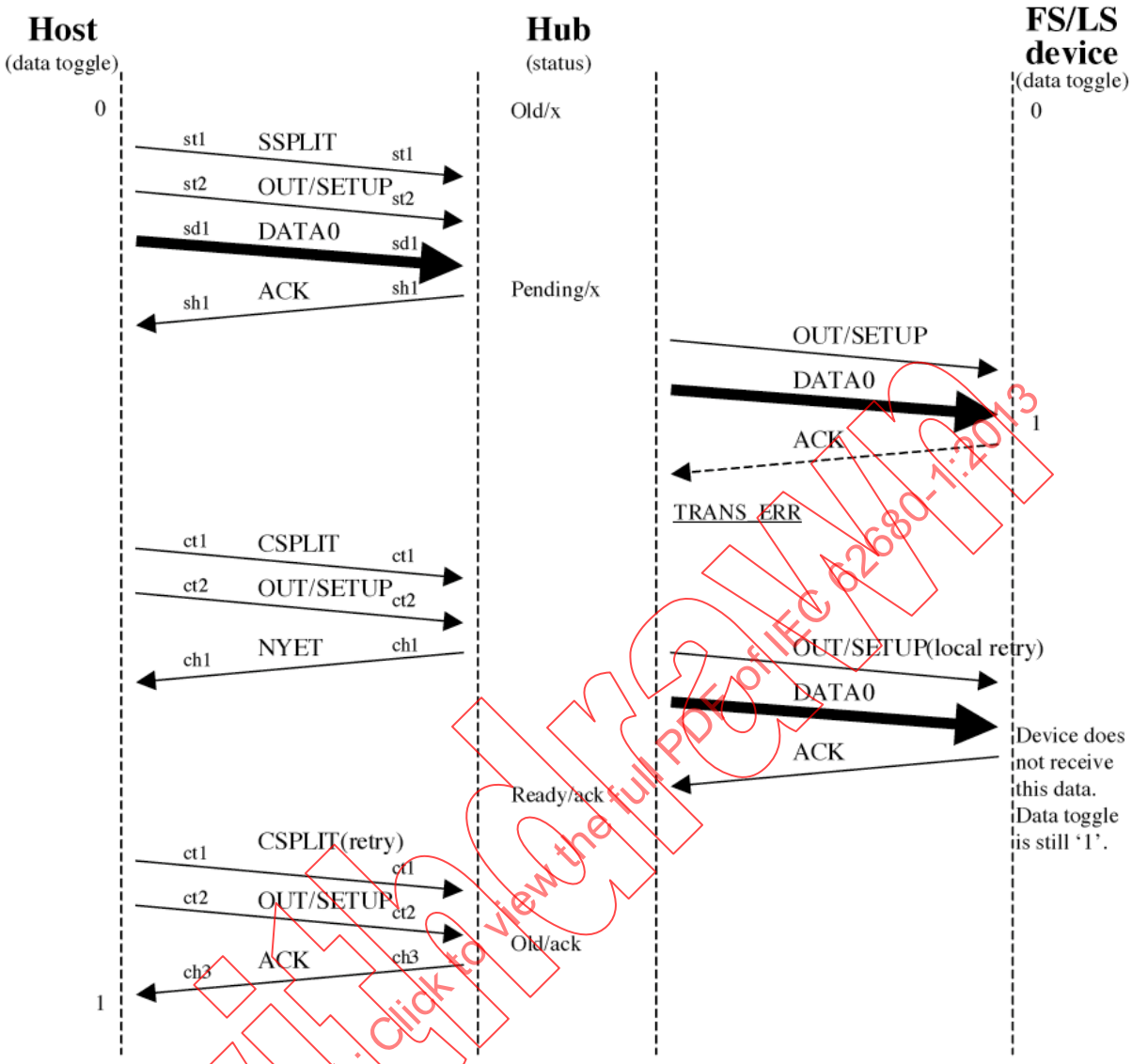


IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/stall	Prêt/arrêt
Old/stall	Ancien/arrêt
CSPLIT (retry)	CSPLIT (nouvel essai)
OUT/SETUP (local retry)	OUT/SETUP (nouvel essai local)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-12. Smash 3 essais FS/LS DATA0/1 normal

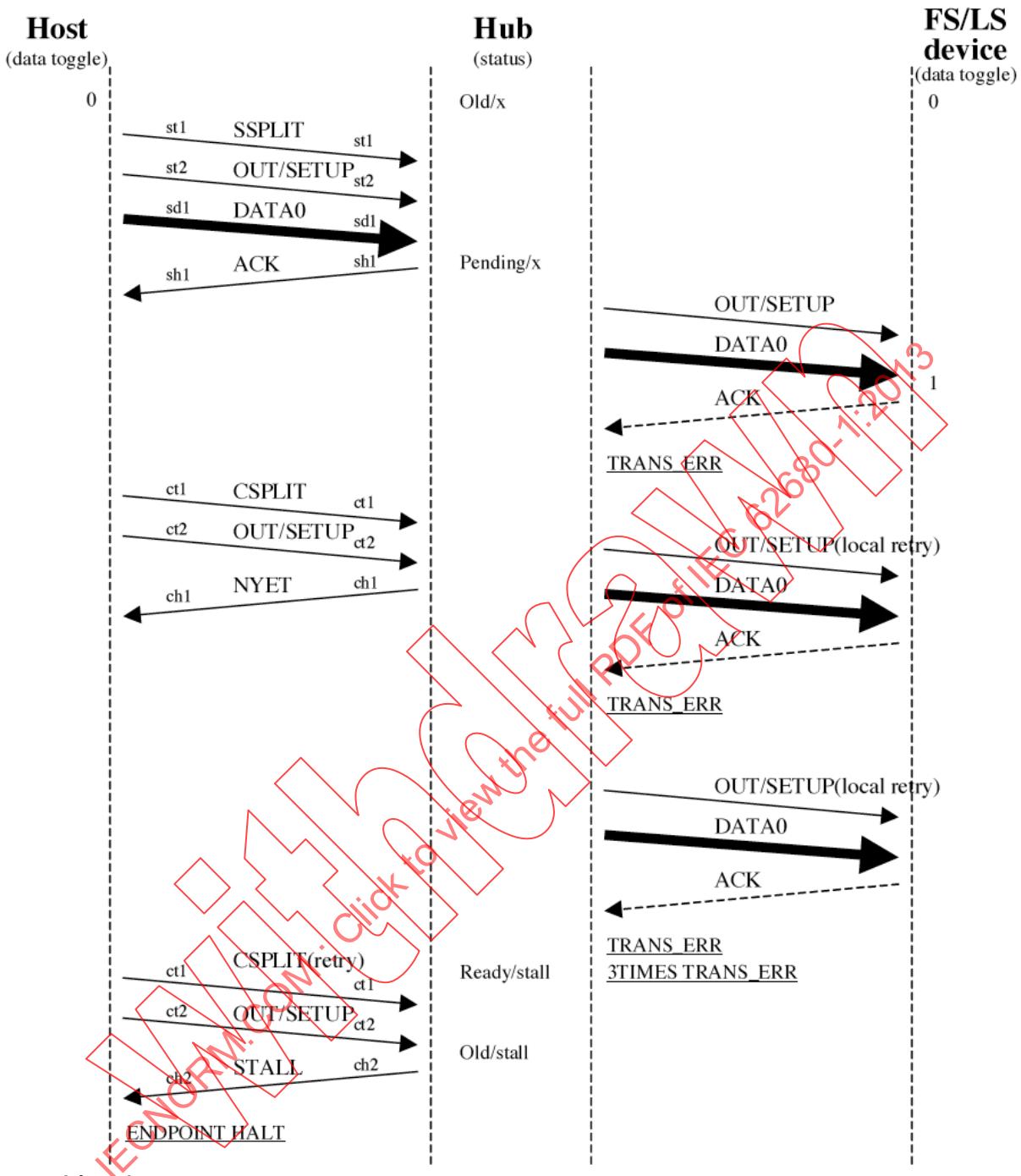


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack
CSPLIT (retry)	CSPLIT (nouvel essai)
OUT/SETUP (local retry)	OUT/SETUP (nouvel essai local)
Device does not receive this data. Data toggle is still '1'.	L'appareil ne reçoit pas ces données. Le basculement des données est encore '1'.

Figure A-13. Smash FS/LS ACK normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

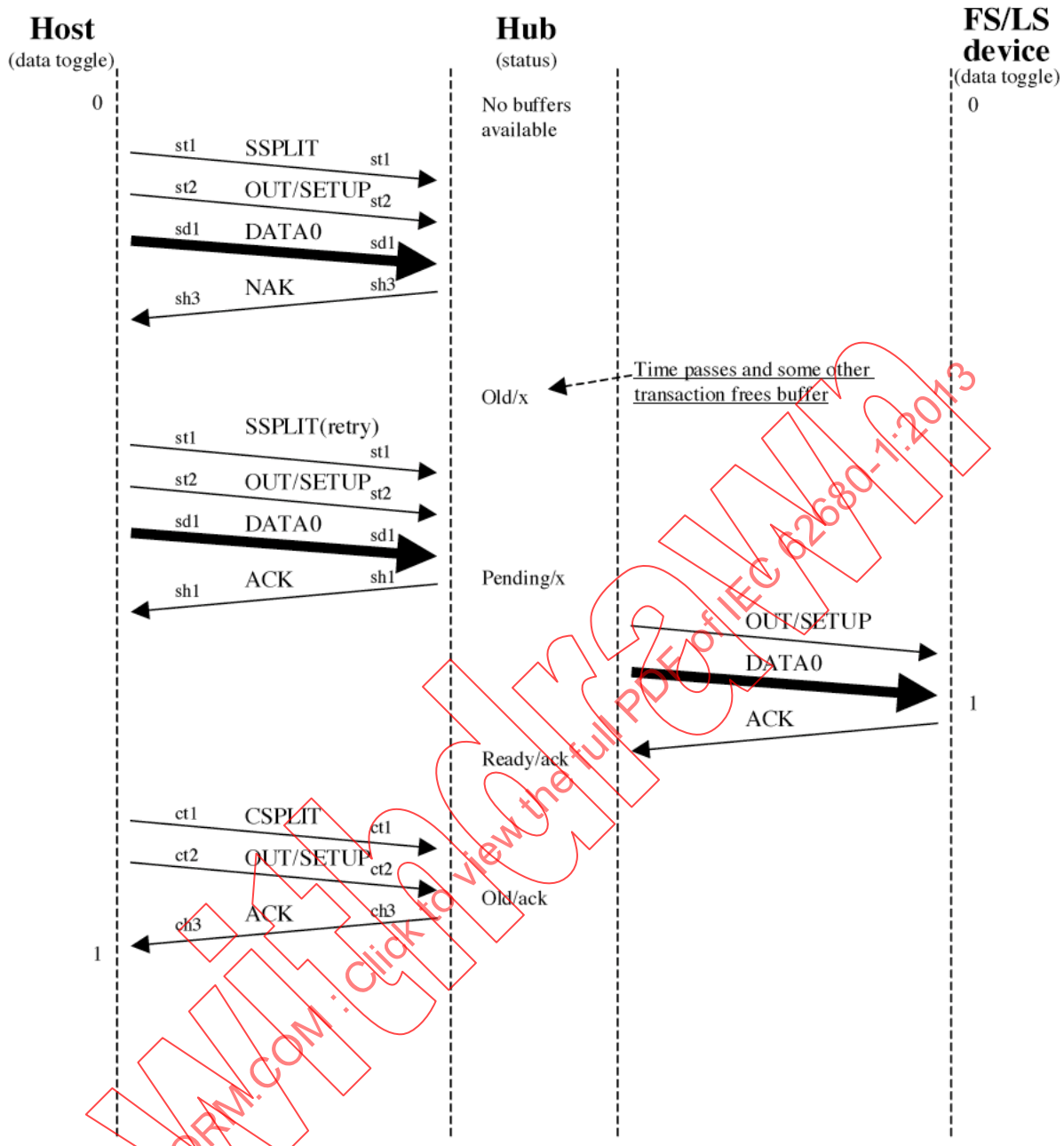


IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/stall	Prêt/arrêt
Old/stall	Ancien/arrêt
CSPLIT (retry)	CSPLIT (nouvel essai)
OUT/SETUP (local retry)	OUT/SETUP (nouvel essai local)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-14. Smash 3 essais FS/LS ACK normal

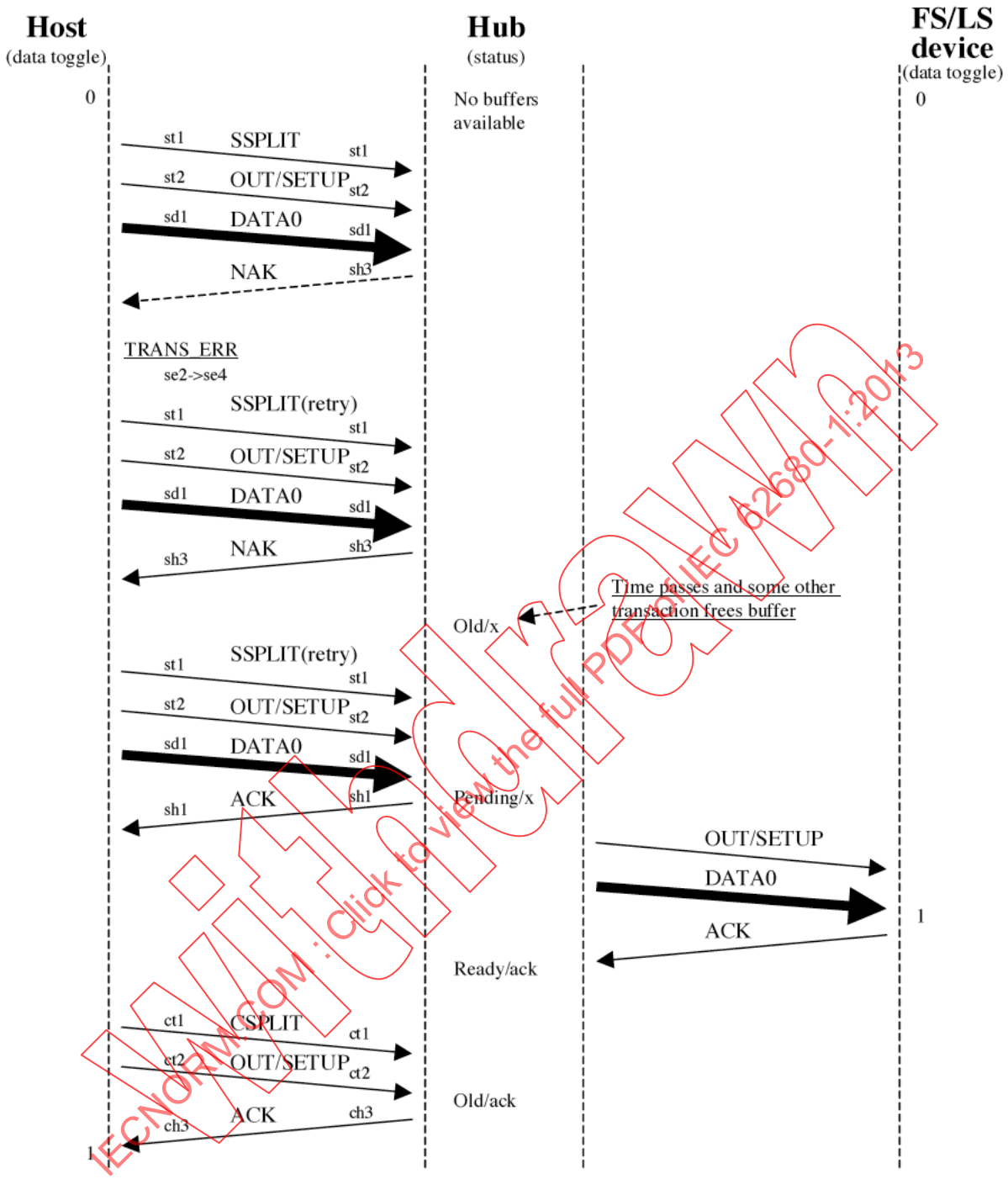


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
No buffers available	Pas de tampons disponibles
Old/x	Ancien/x
Time passes and some other transaction frees buffer	Le temps passe et d'autres transactions libèrent le tampon
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack

Figure A-15. Pas de tampon disponible Pas de smash (HS NAK(S))

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



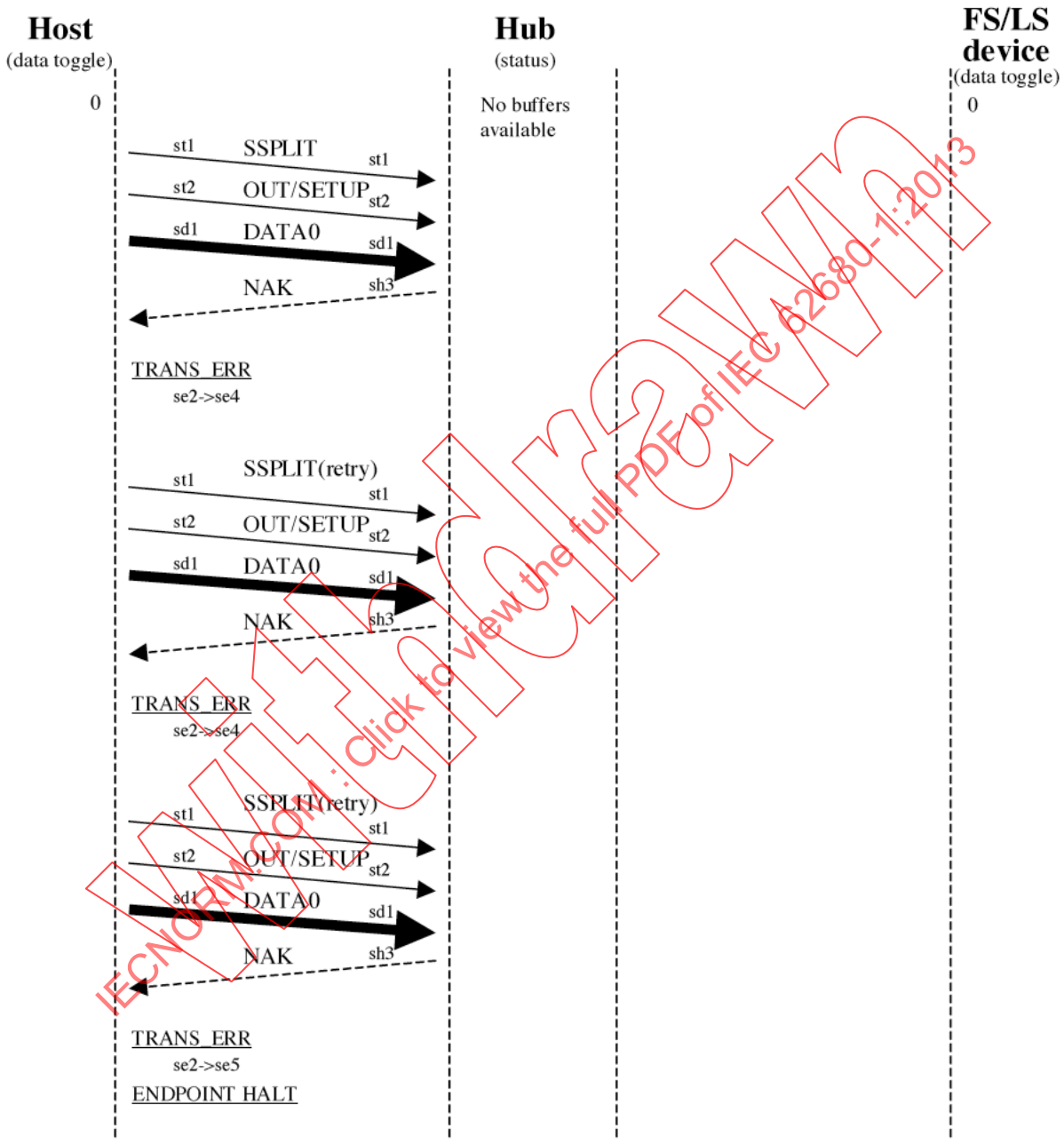
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
No buffers available	Pas de tampons disponibles
Old/x	Ancien/x
Time passes and some other transaction frees buffer	Le temps passe et d'autres transactions libèrent le tampon

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack
SSPLIT (retry)	SSPLIT (nouvel essai)

Figure A-16. Pas de tampon disponible Smash HS NAK(S)



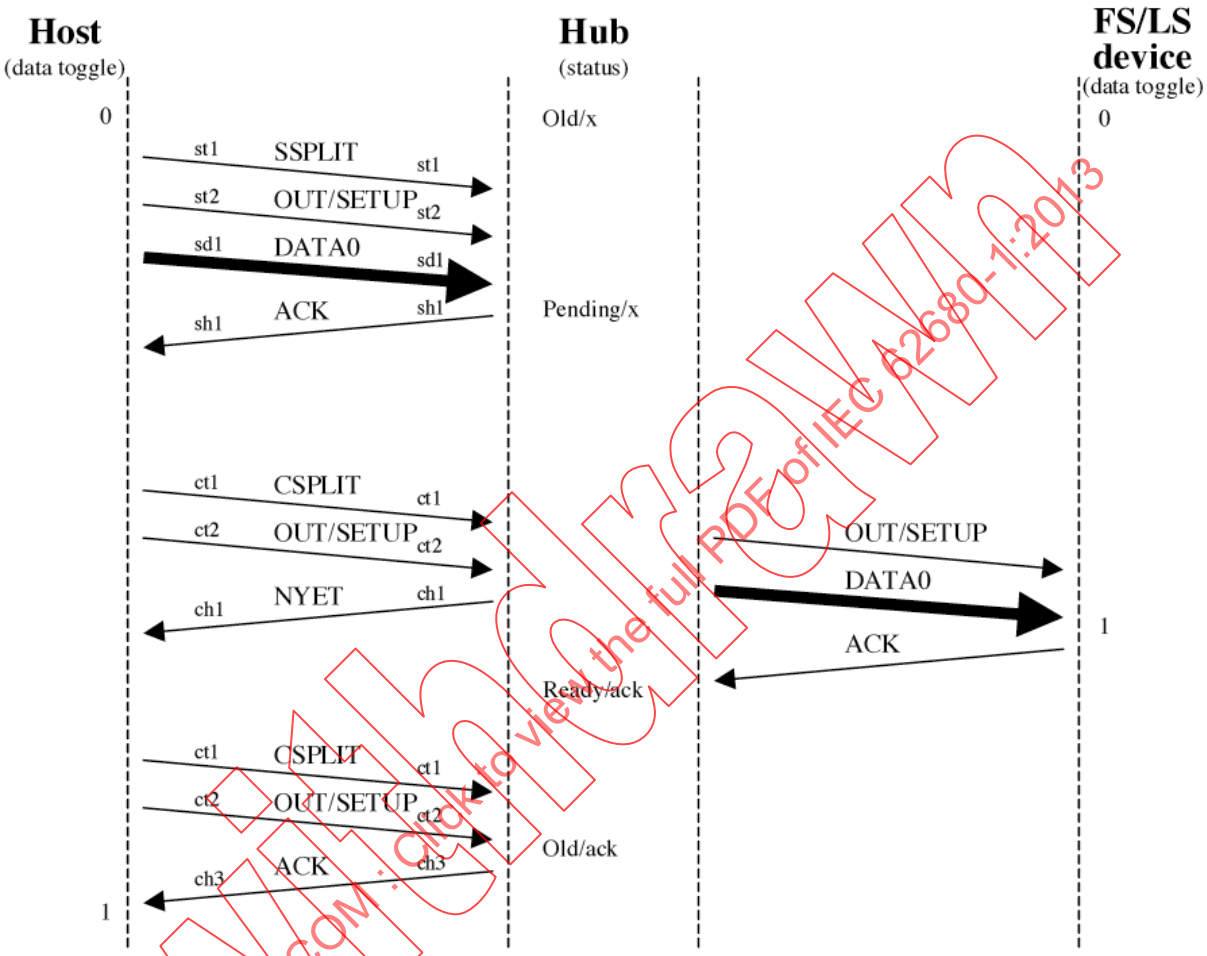
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device	Appareil à vitesse intégrale/basse vitesse

Anglais	Français
(data toggle)	(basculement des données)
No buffers available	Pas de tampons disponibles
SSPLIT (retry)	SSPLIT (nouvel essai)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-17. Pas de tampon disponible Smash 3 essais HS NAK(S)

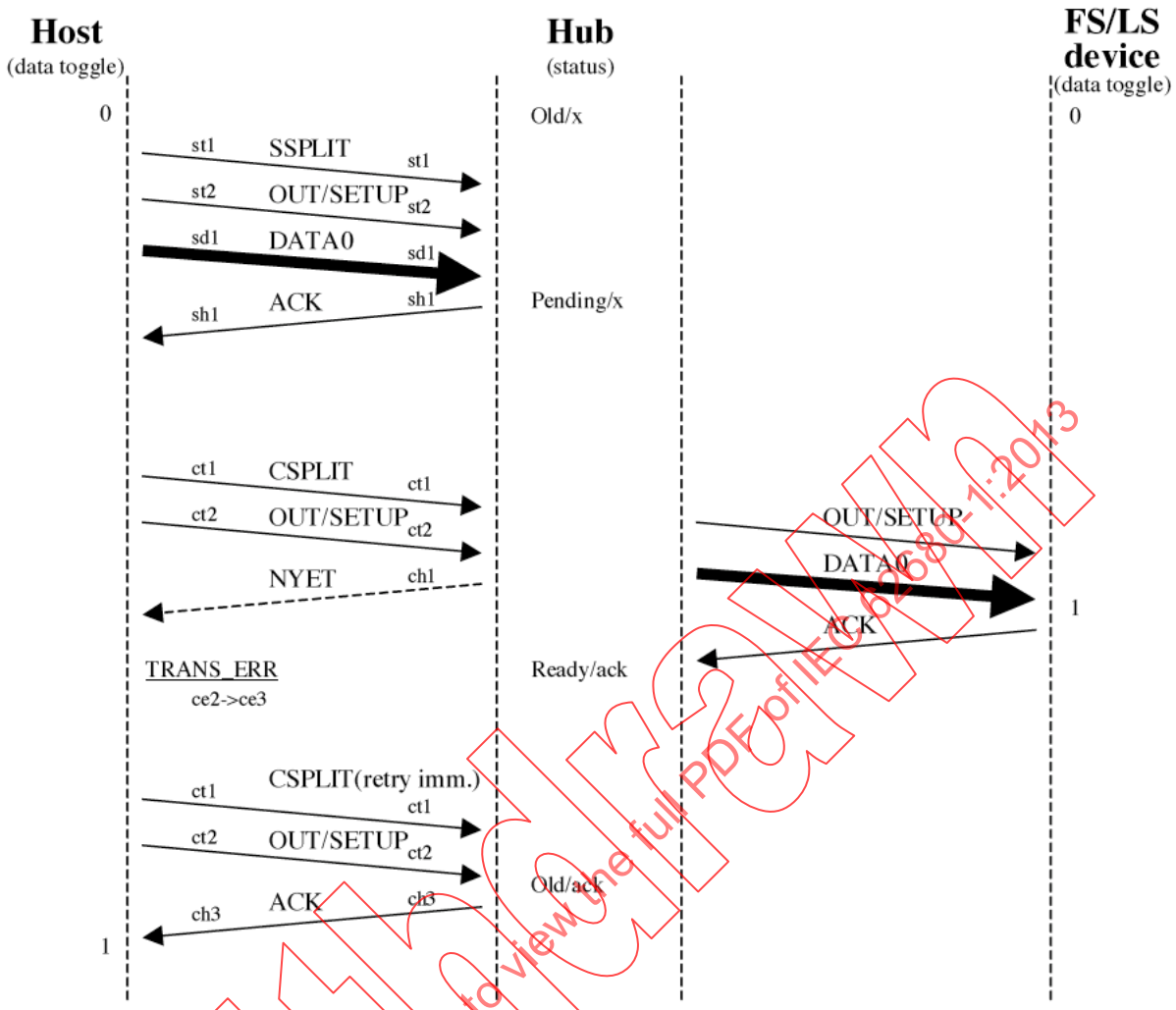


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack

Figure A-18. CS plus tôt Pas de smash (HS NYET)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

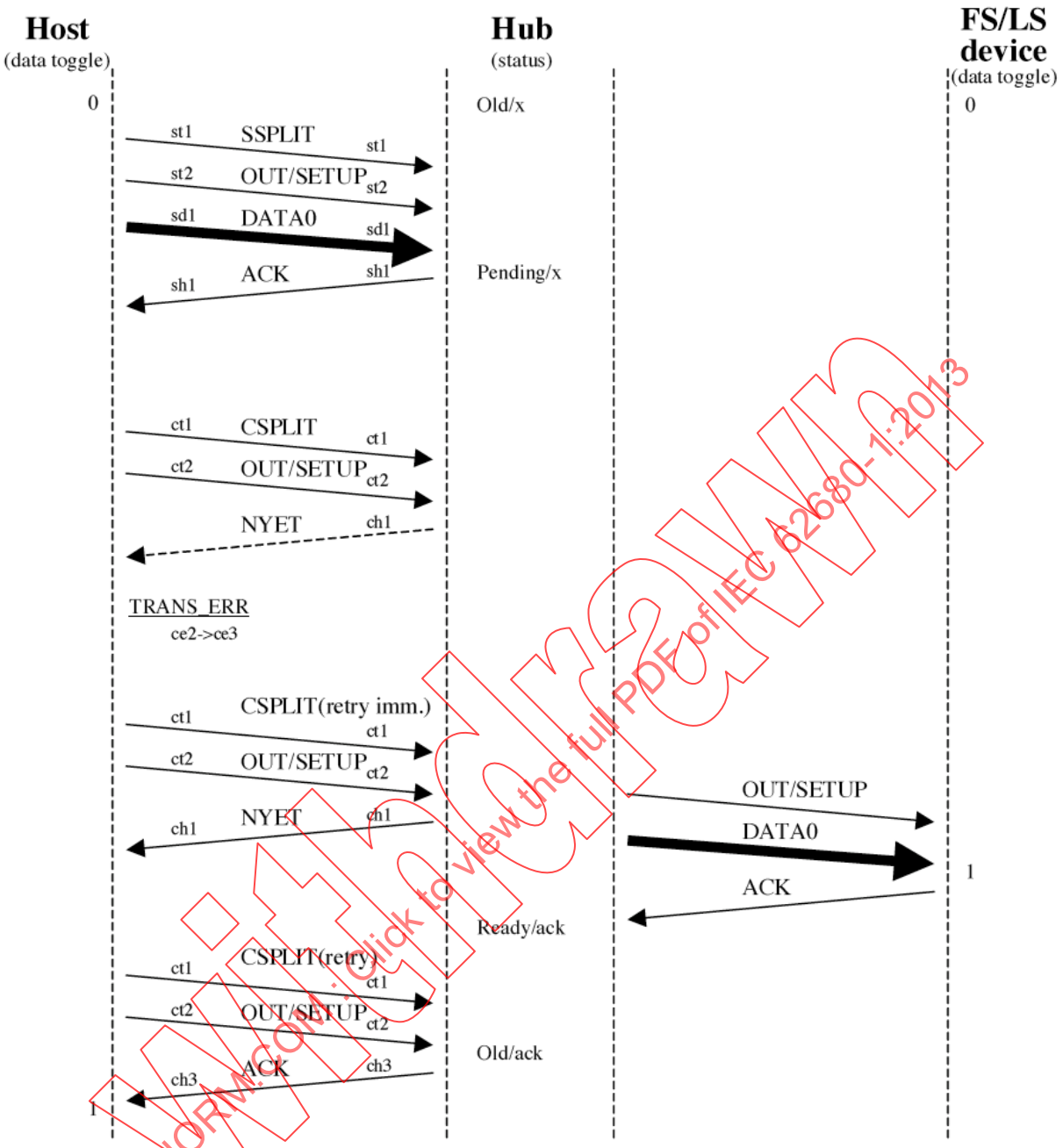


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)

Figure A-19. CS plus tôt Smash HS NYET (cas 1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

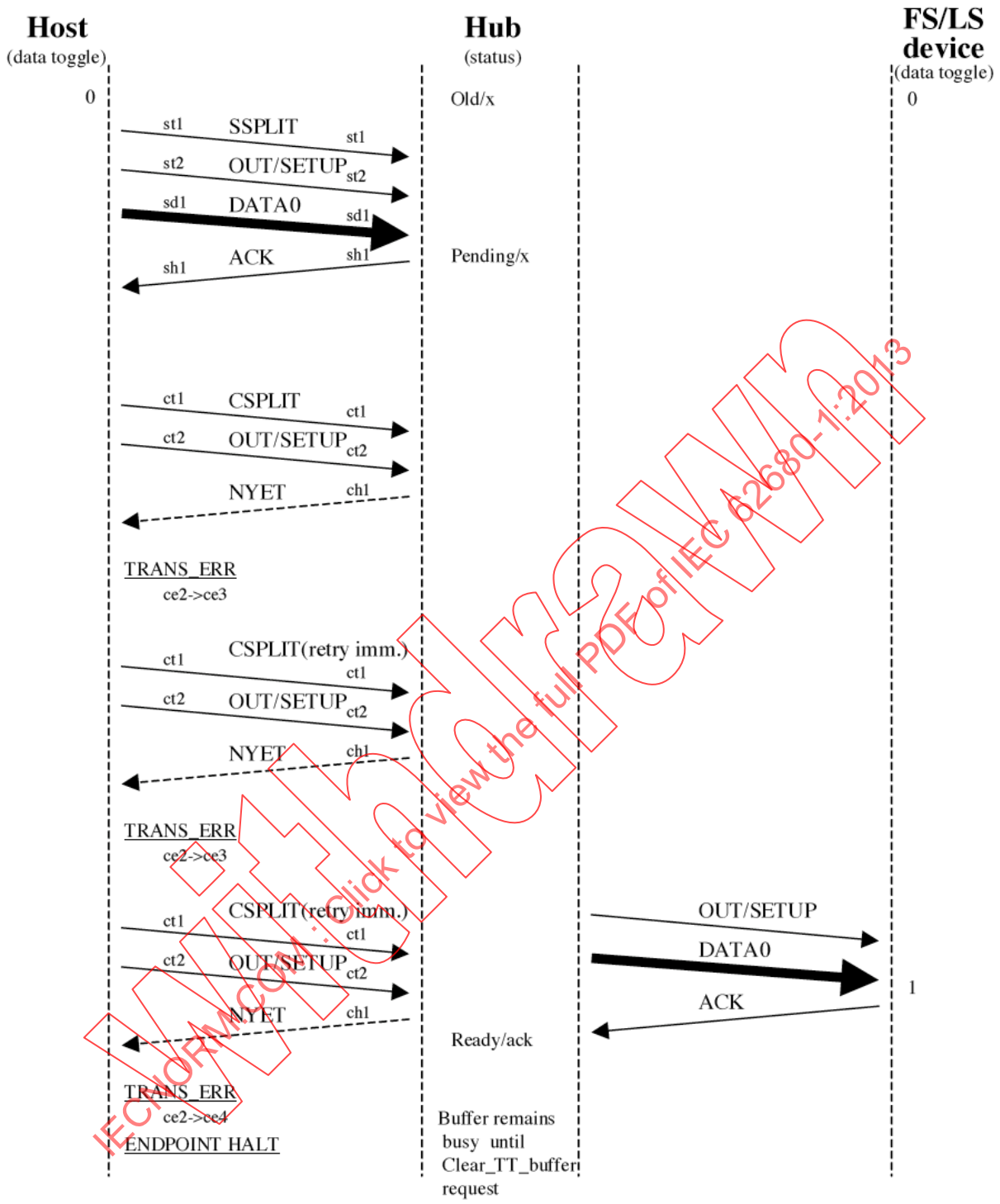


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/ack	Prêt/ack
Old/ack	Ancien/ack
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
CSPLIT (retry)	CSPLIT (nouvel essai)

Figure A-20. CS plus tôt Smash HS NYET (cas 2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

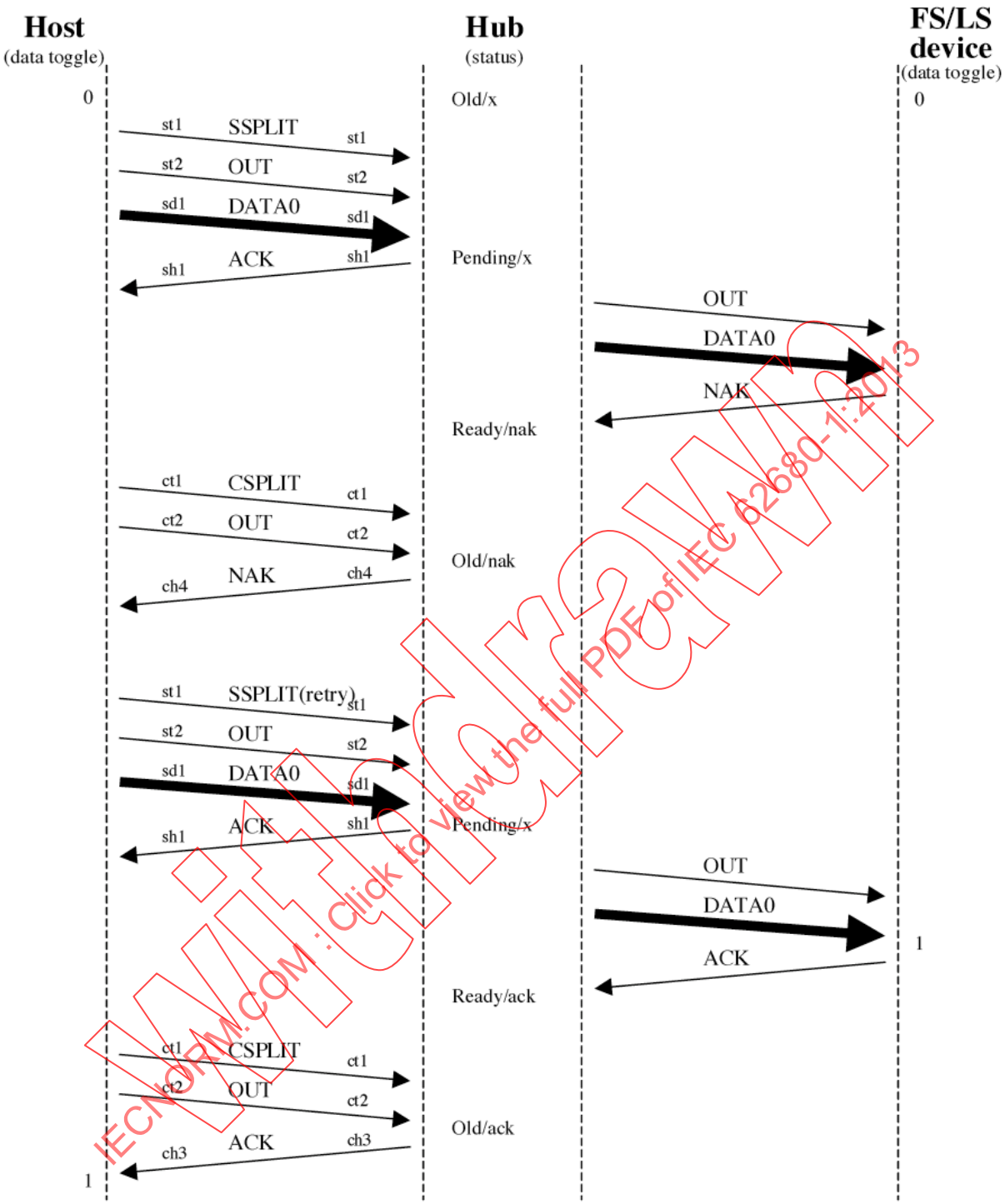
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x

Anglais	Français
Ready/ack	Prêt/ack
Buffer remains busy until Clear_TT_buffer request	Le tampon reste occupé jusqu'à la demande Clear_TT_buffer
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-21. CS plus tôt Smash 3 essais HS NYET

Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



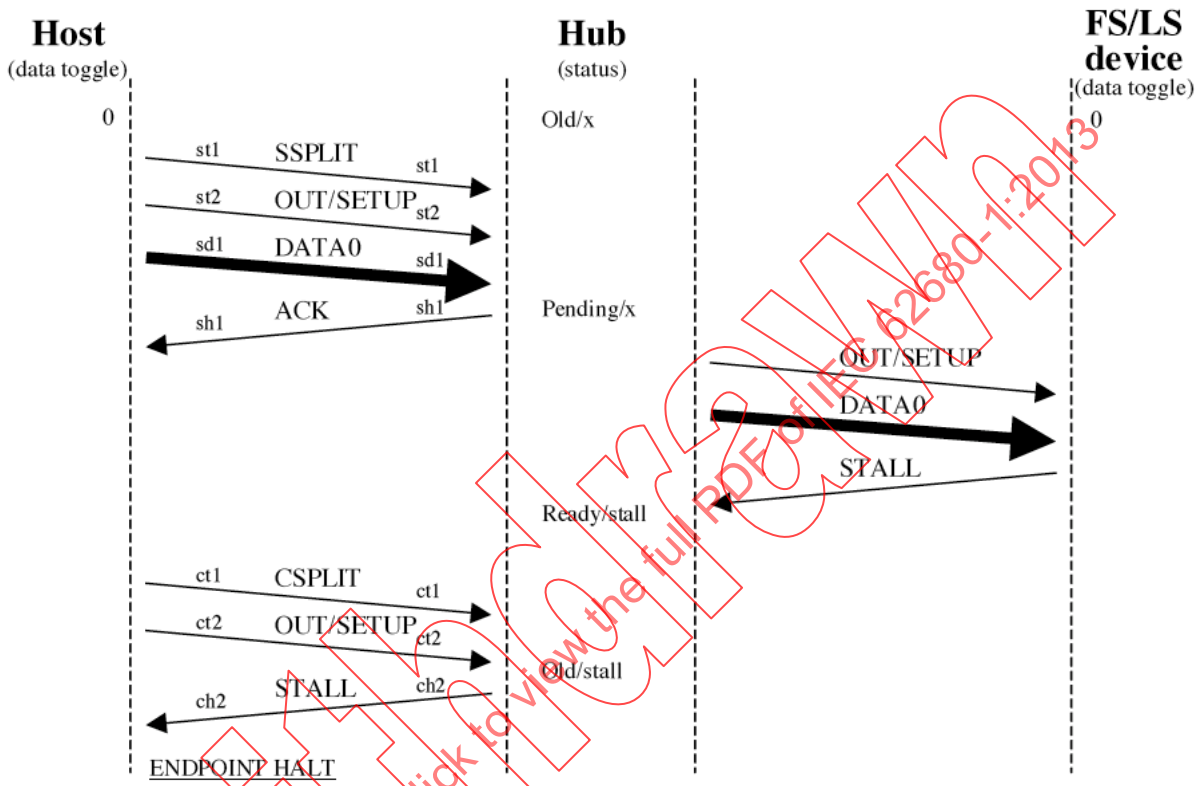
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x

Anglais	Français
Ready/nak	Prêt/nak
Old/nak	Ancien/nak
SSPLIT (retry)	SSPLIT (nouvel essai)
Ready/ack	Prêt/ack
Old/ack	Ancien/ack

Figure A-22. Appareil occupé Pas de smash(FS/LS NAK)



Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/stall	Prêt/arrêt
Old/stall	Ancien/arrêt
ENPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-23. Appareil arrêté Pas de smash(FS/LS STALL)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A.2 Exemples de transactions IN en masse/de contrôle

Légende:

(S): Start Split

(C): Complete Split

Résumé des cas de transaction IN en masse/de contrôle

- Cas normaux

Cas	Figure de référence	Figure similaire
Pas de smash	Figure A-24	
Smash HS SSPLIT	Figure A-25	
Smash 3 essais HS SSPLIT	Figure A-26	
Smash HS IN(S)		Figure A-25
Smash 3 essais HS IN(S)		Figure A-26
Smash HS ACK(S)	Figure A-27 Figure A-28	
Smash 3 essais HS ACK(S)	Figure A-29	
Smash HS CSPLIT	Figure A-30	
Smash 3 essais HS CSPLIT	Figure A-31	
Smash HS IN(C)		Figure A-30
Smash 3 essais HS IN(C)		Figure A-31
Smash HS DATA0/1	Figure A-32	
Smash 3 essais HS DATA0/1	Figure A-33	
Smash FS/LS IN	Figure A-34	
Smash 3 essais FS/LS IN	Figure A-35	
Smash FS/LS DATA0/1	Figure A-36	
Smash 3 essais FS/LS DATA0/1	Figure A-37	
Smash FS/LS ACK	Figure A-38	
Smash 3 essais FS/LS ACK	Pas de figure	

- Aucun tampon (sur hub) - Cas disponibles

Cas	Figure de référence	Figure similaire
Pas de smash(HS NAK(S))	Figure A-39	
Smash HS NAK(S)	Figure A-40	
Smash 3 essais HS NAK(S)	Figure A-41	

- CS(transaction complete-split) - Cas précédents

Cas	Figure de référence	Figure similaire
Pas de smash(HS NYET)	Figure A-42	
Smash HS NYET	Figure A-43 Figure A-44	
Smash 3 essais HS NYET	Pas de figure	

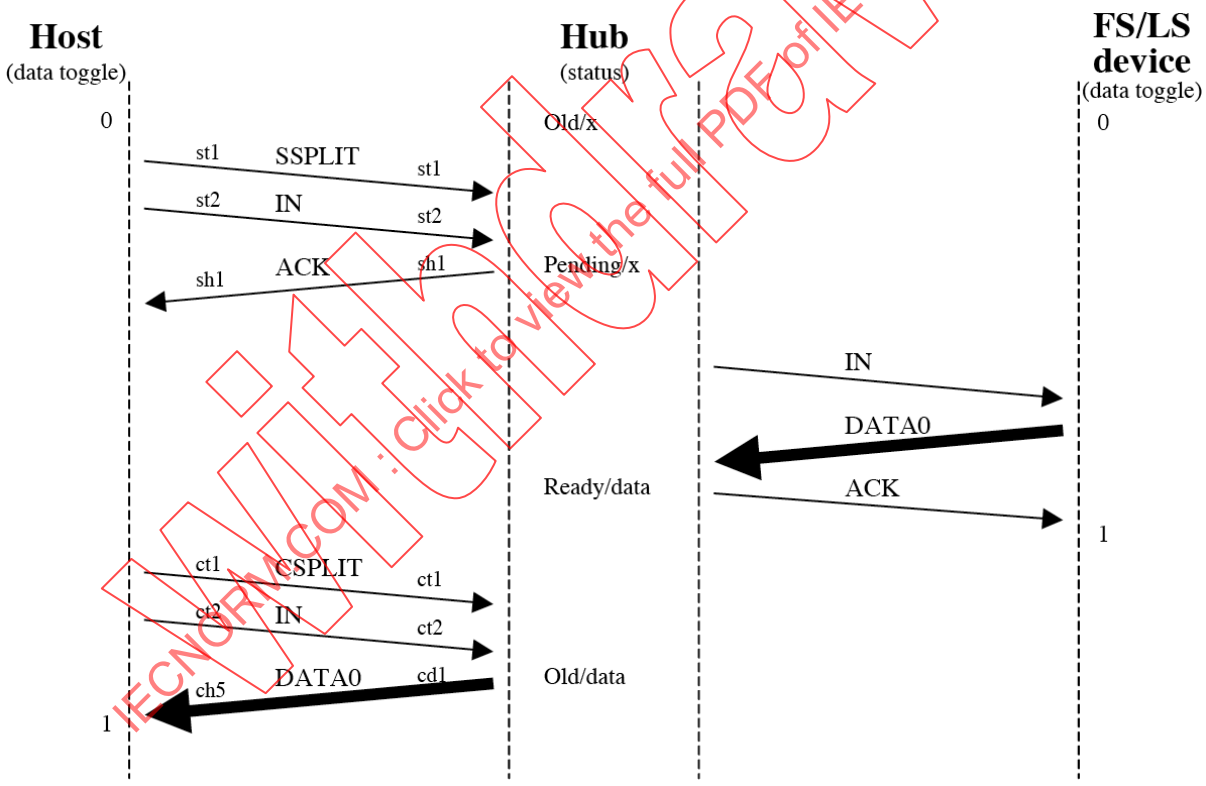
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

- Cas appareil occupé

Cas	Figure de référence	Figure similaire
Pas de smash(HS NAK(C))	Figure A-45	
Smash HS NAK(C)		Figure A-32
Smash 3 essais HS NAK(C)		Figure A-33
Smash FS/LS NAK		Figure A-36
Smash 3 essais FS/LS NAK		Figure A-37

- Cas appareil arrêté

Cas	Figure de référence	Figure similaire
Pas de smash	Figure A-46	
Smash HS STALL(C)		Figure A-32
Smash 3 essais HS STALL(C)		Figure A-33
Smash FS/LS STALL		Figure A-36
Smash 3 essais FS/LS STALL		Figure A-37



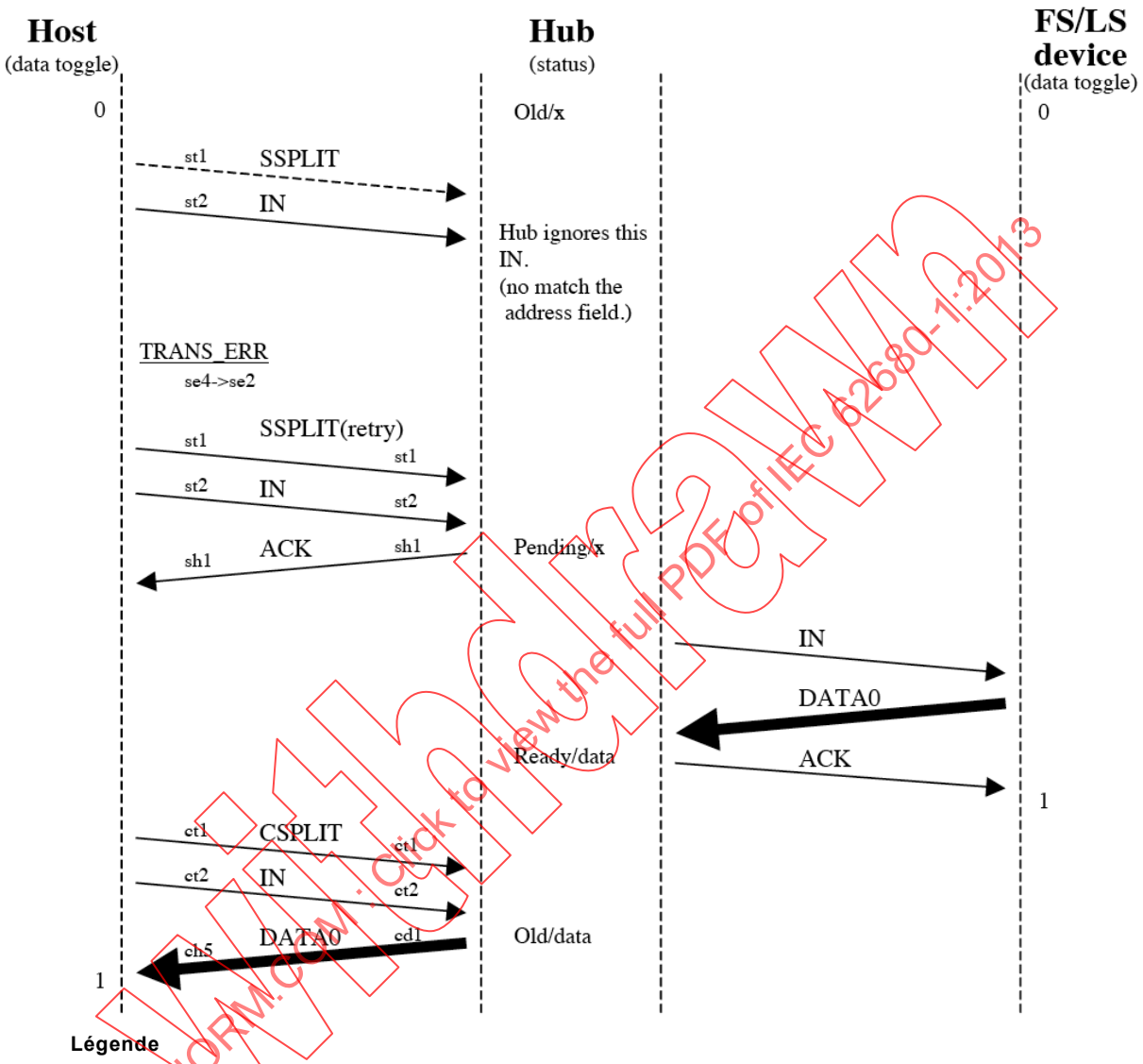
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
Ready/data	Prêt/données
Old/data	Ancien/données

Figure A-24. Normal Pas de smash

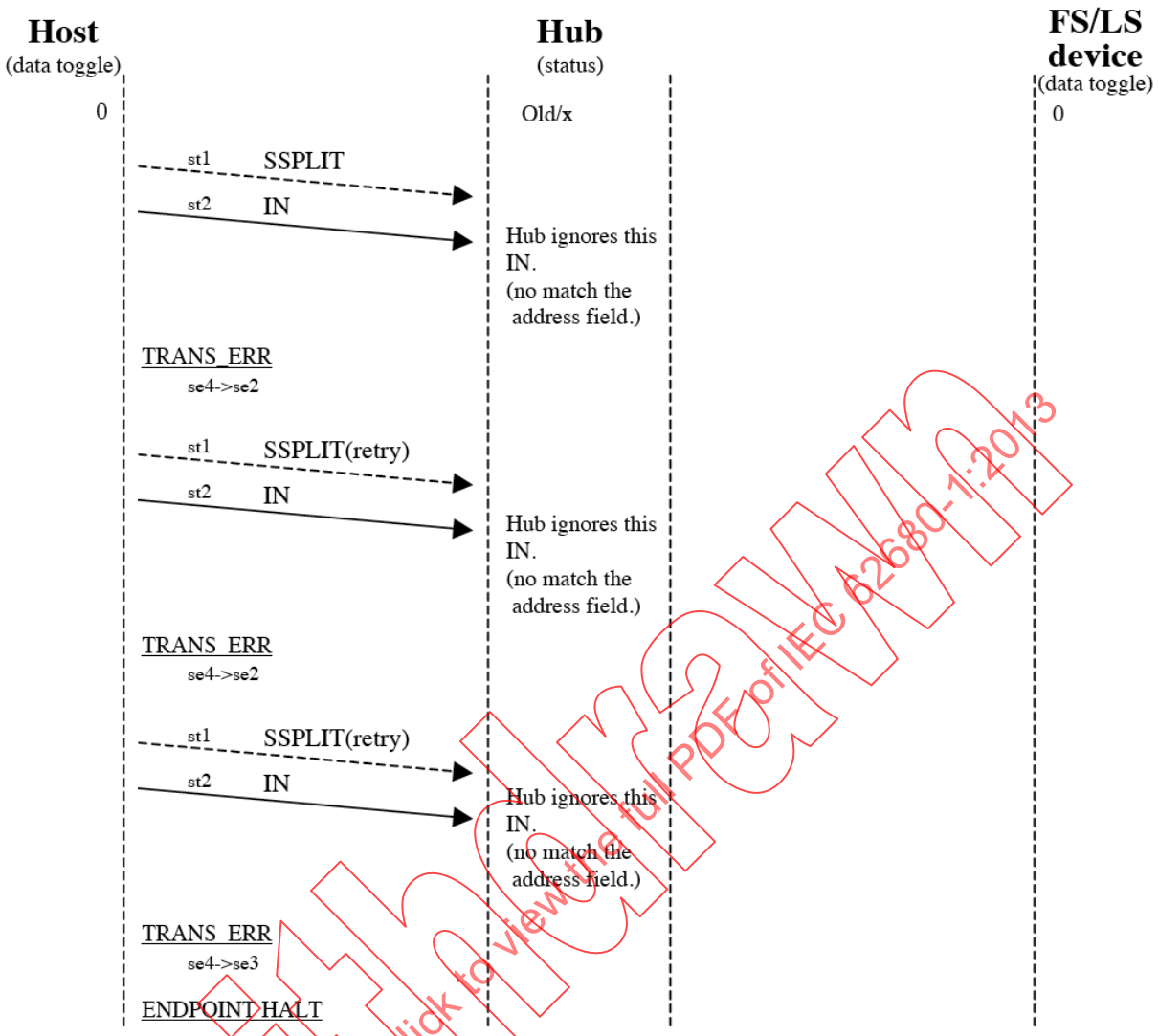


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Hub ignores this IN. (no match the address field)	Le hub ignore cet IN. (pas de correspondance dans le champ d'adresse)
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données

Figure A-25. Smash HS SSPLIT normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

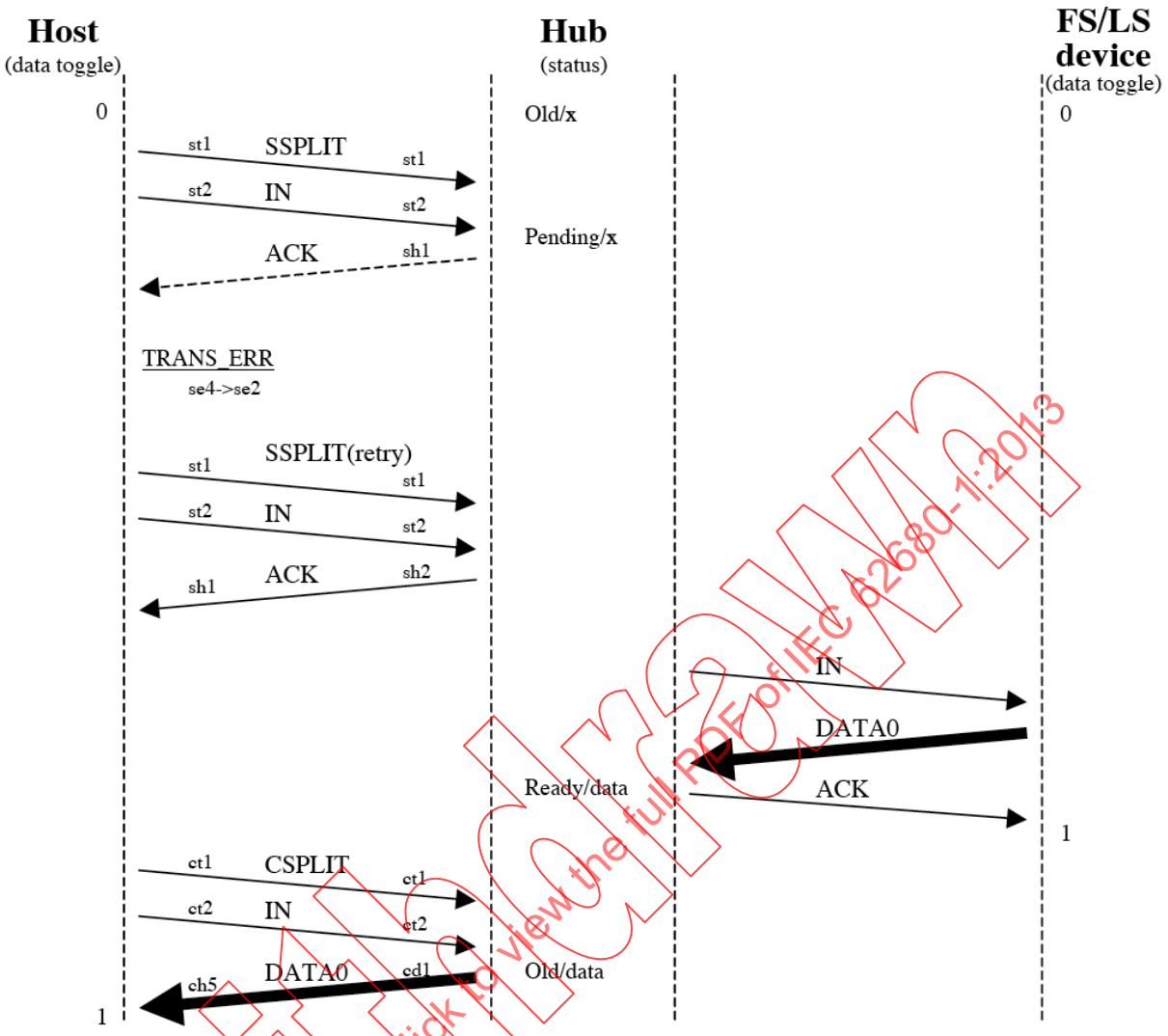


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Hub ignores this IN. (no match the address field)	Le hub ignore cet IN. (pas de correspondance dans le champ d'adresse)
SSPLIT(retry)	SSPLIT (nouvel essai)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-26. Smash 3 essais SSPLIT normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

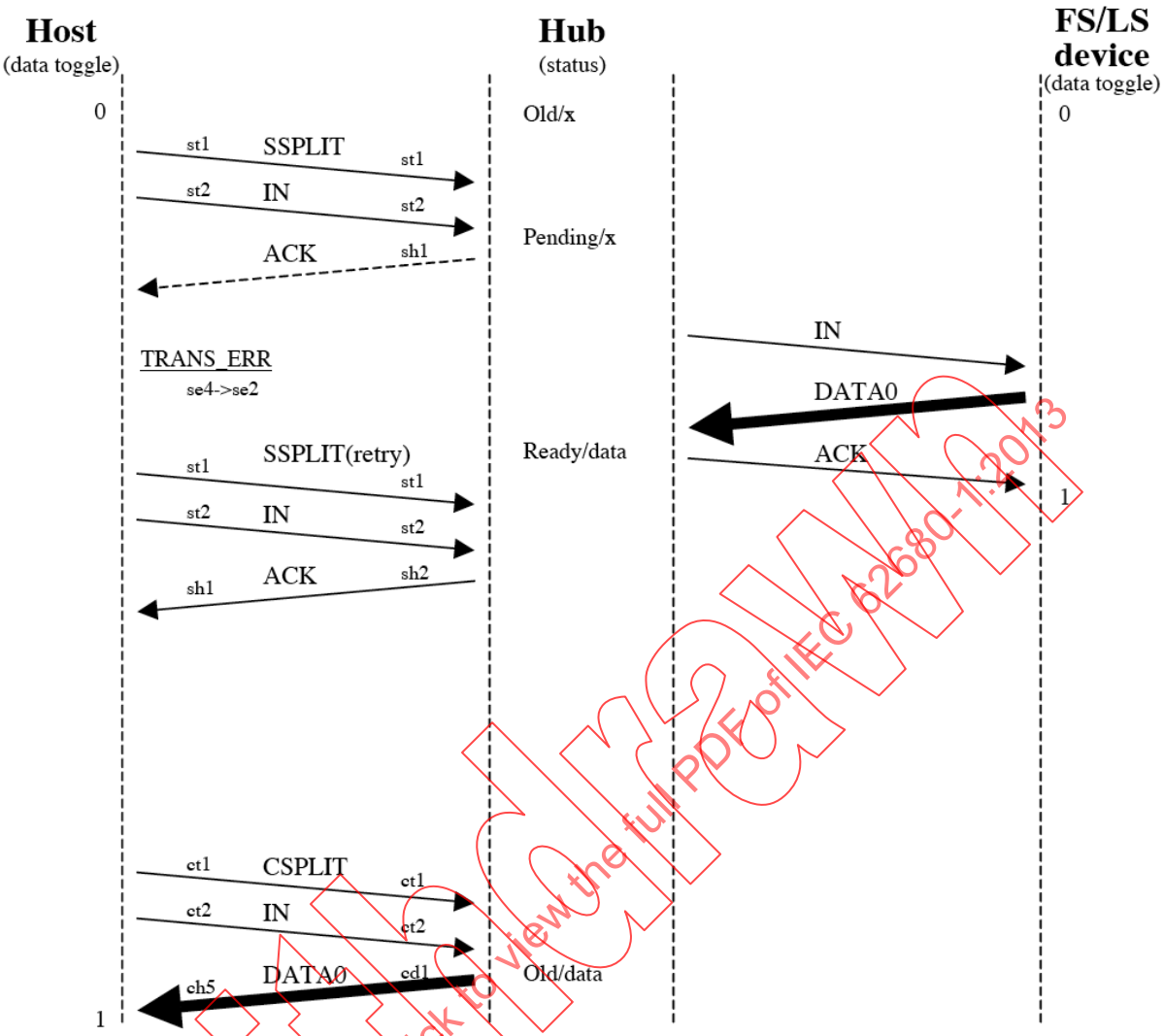


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
SSPLIT(retry)	SSPLIT (nouvel essai)
Ready/data	Prêt/données
Old/data	Ancien/données

Figure A-27. Smash HS ACK(S) normal (cas 1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

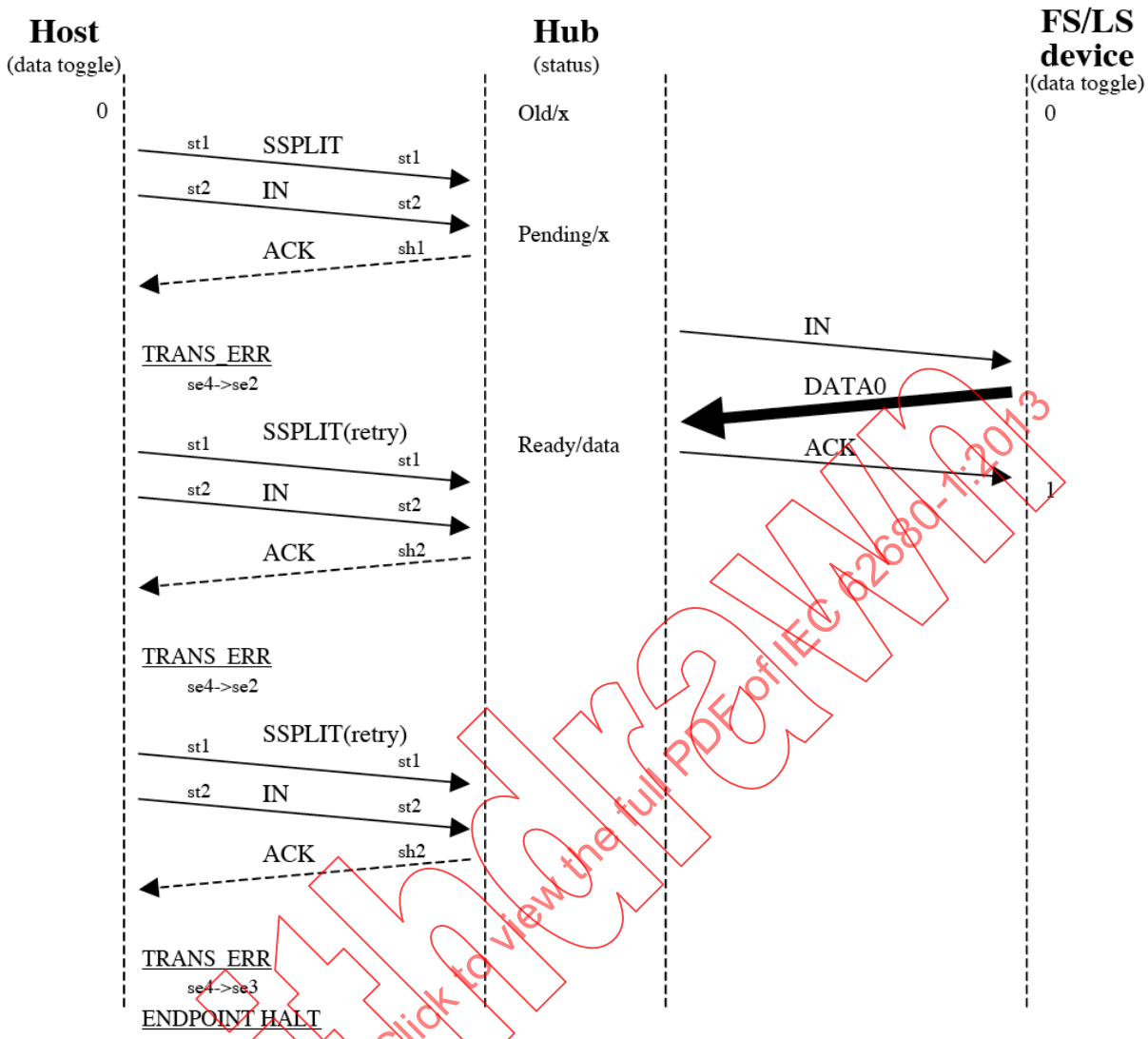


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
SSPLIT (retry)	SSPLIT (nouvel essai)

Figure A-28. Smash HS ACK(S) normal (cas 2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

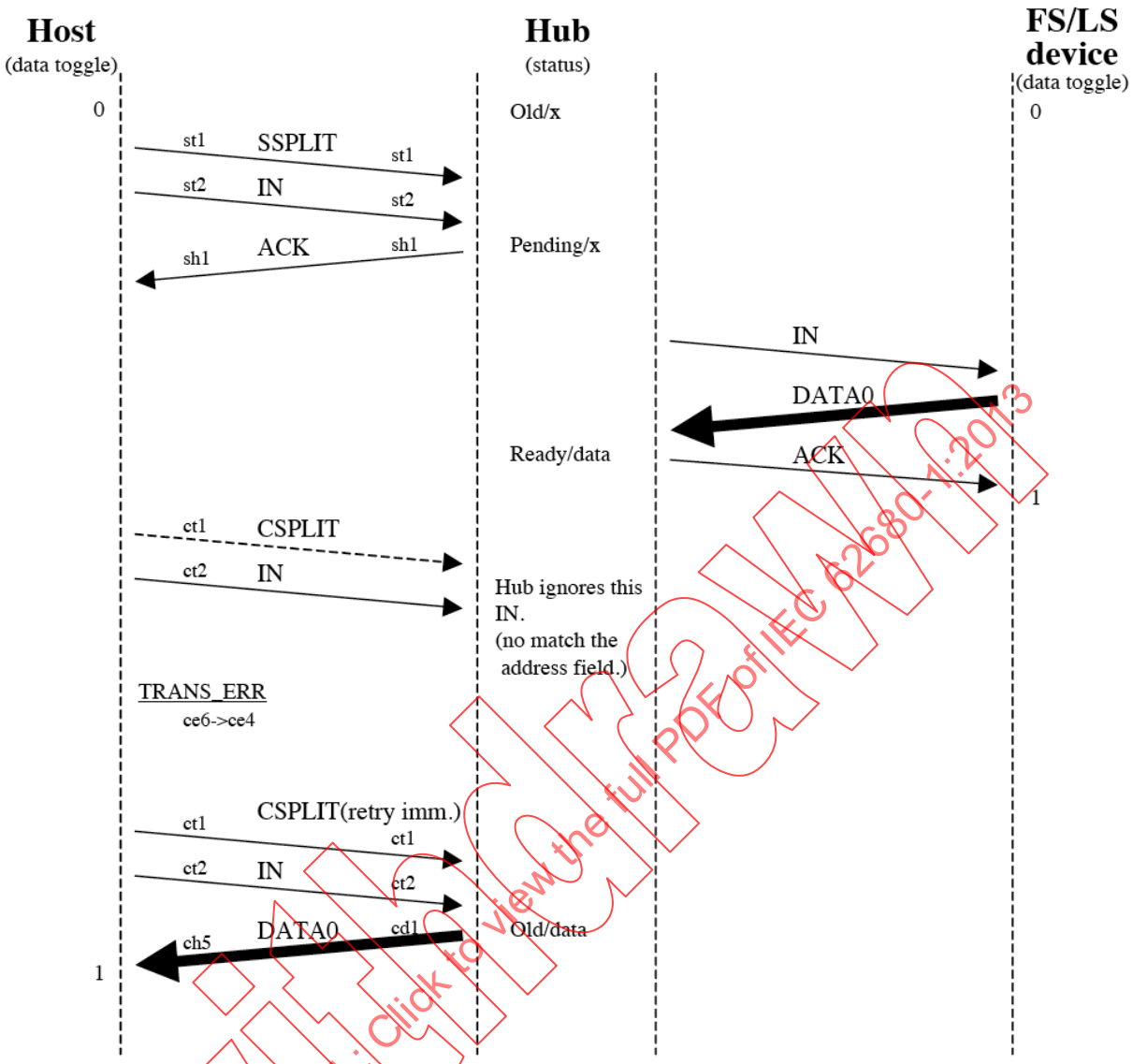


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/data	Prêt/données
SSPLIT (retry)	SSPLIT (nouvel essai)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-29. Smash 3 essais HS ACK(S) normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

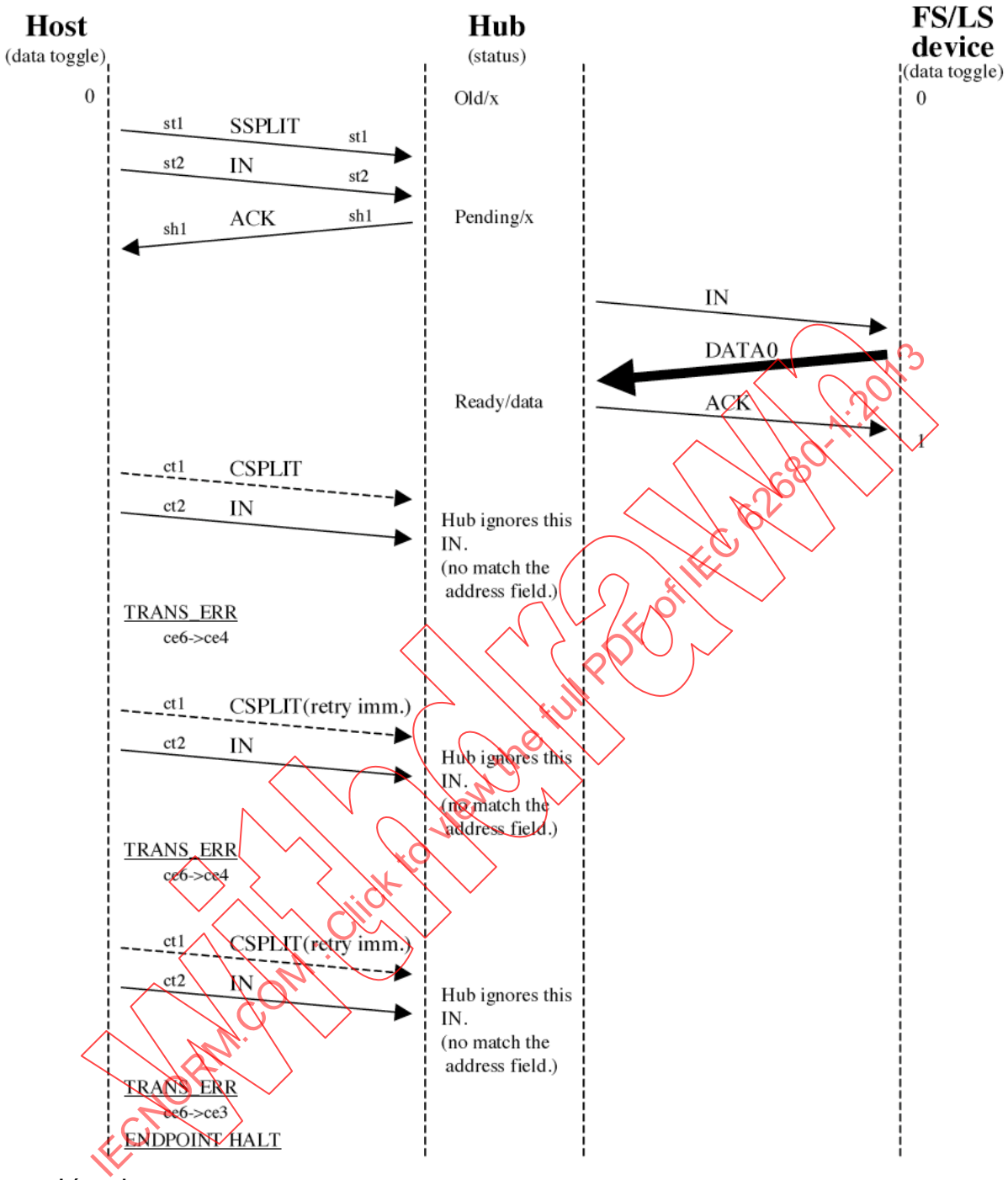


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Hub ignores this IN. (no match the address field)	Le hub ignore cet IN. (pas de correspondance dans le champ d'adresse)
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)

Figure A-30. Smash HS CSPLIT normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



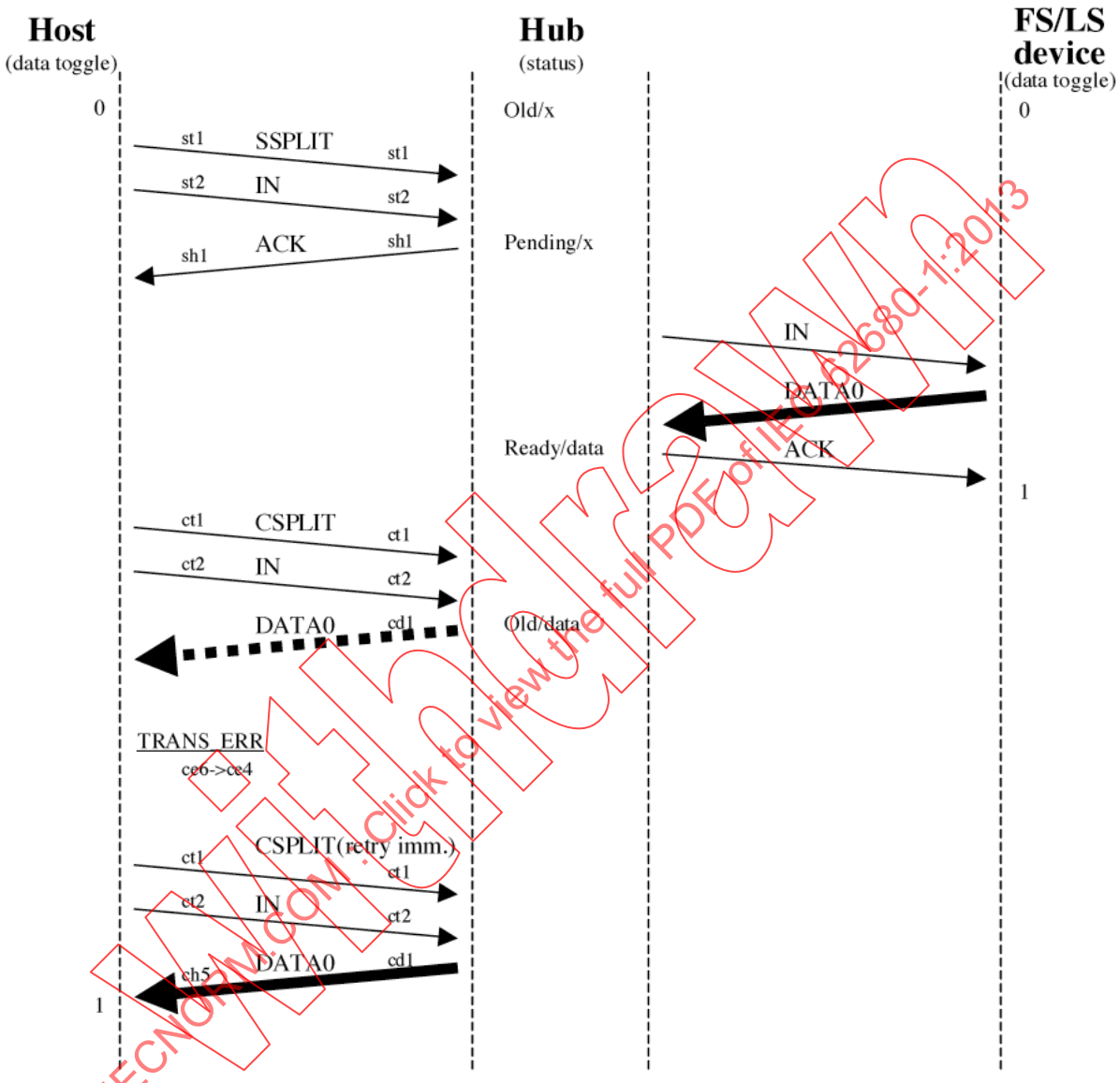
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Hub ignores this IN. (no match the address field)	Le hub ignore cet IN. (pas de correspondance dans le champ d'adresse)
Pending/x	En cours/x

Anglais	Français
Ready/data	Prêt/données
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)

Figure A-31. Smash 3 essais HS CSPLIT normal

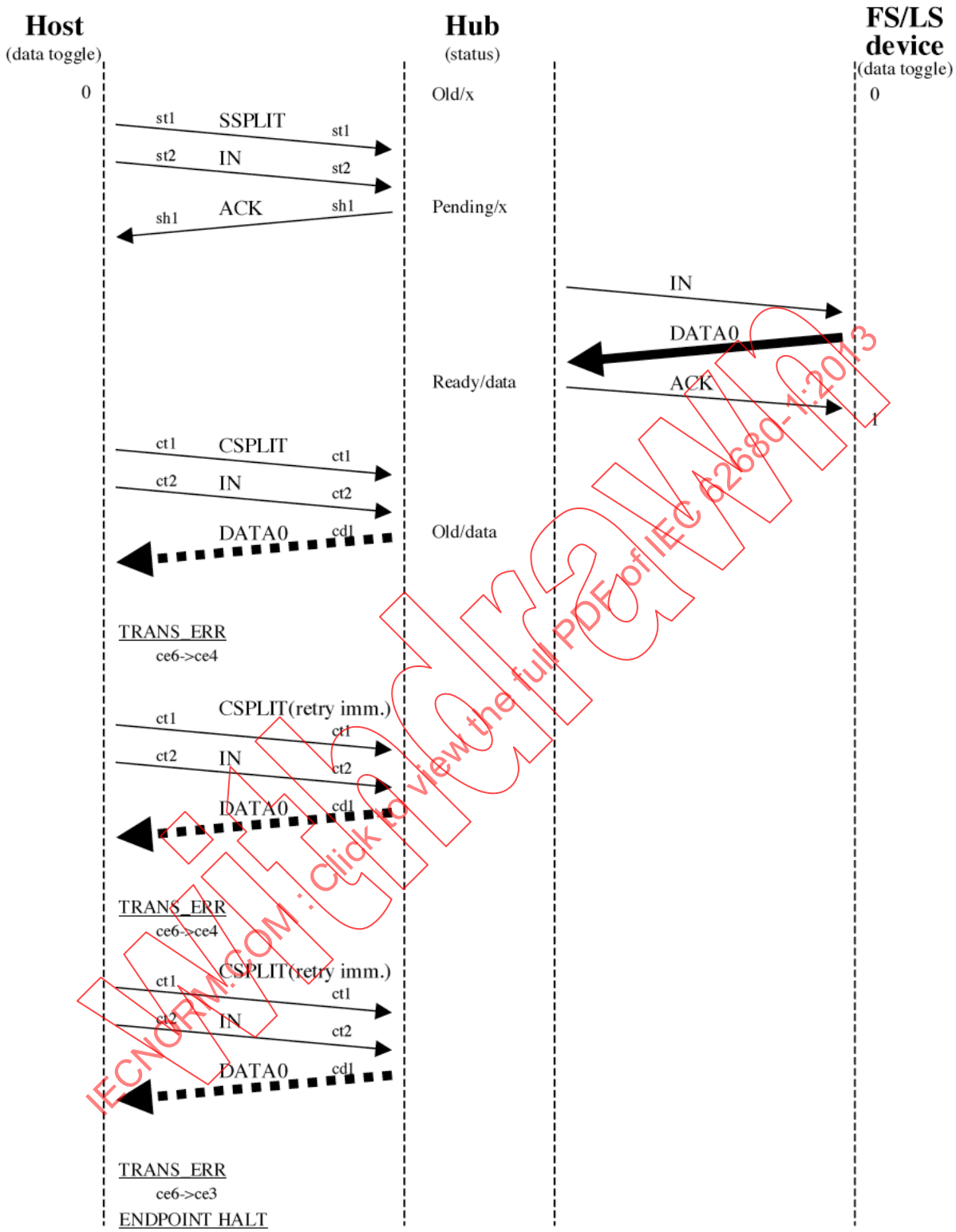


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)

Figure A-32. Smash HS DATA0/1 normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



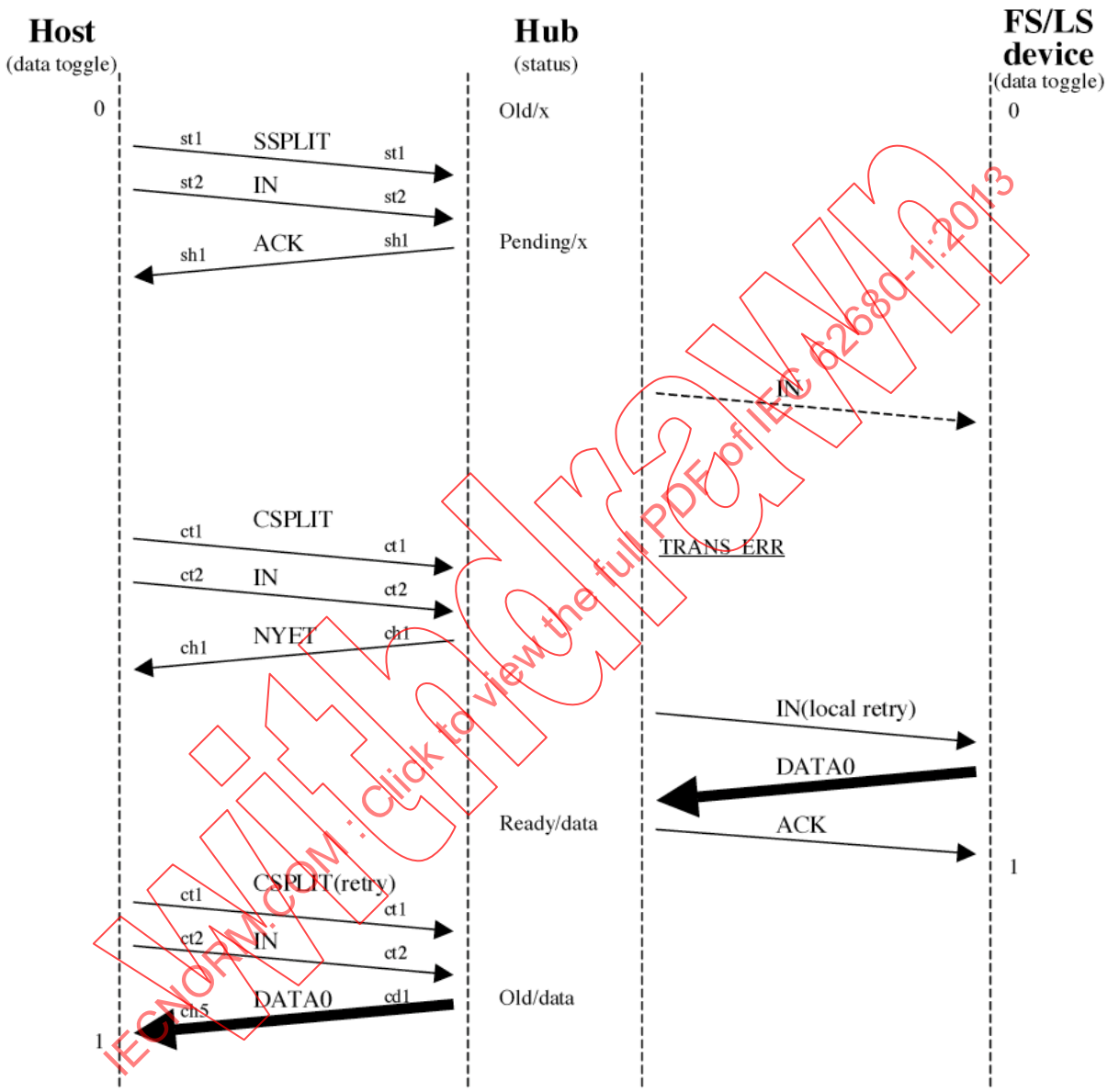
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x

Anglais	Français
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-33. Smash 3 essais HS DATA0/1 normal

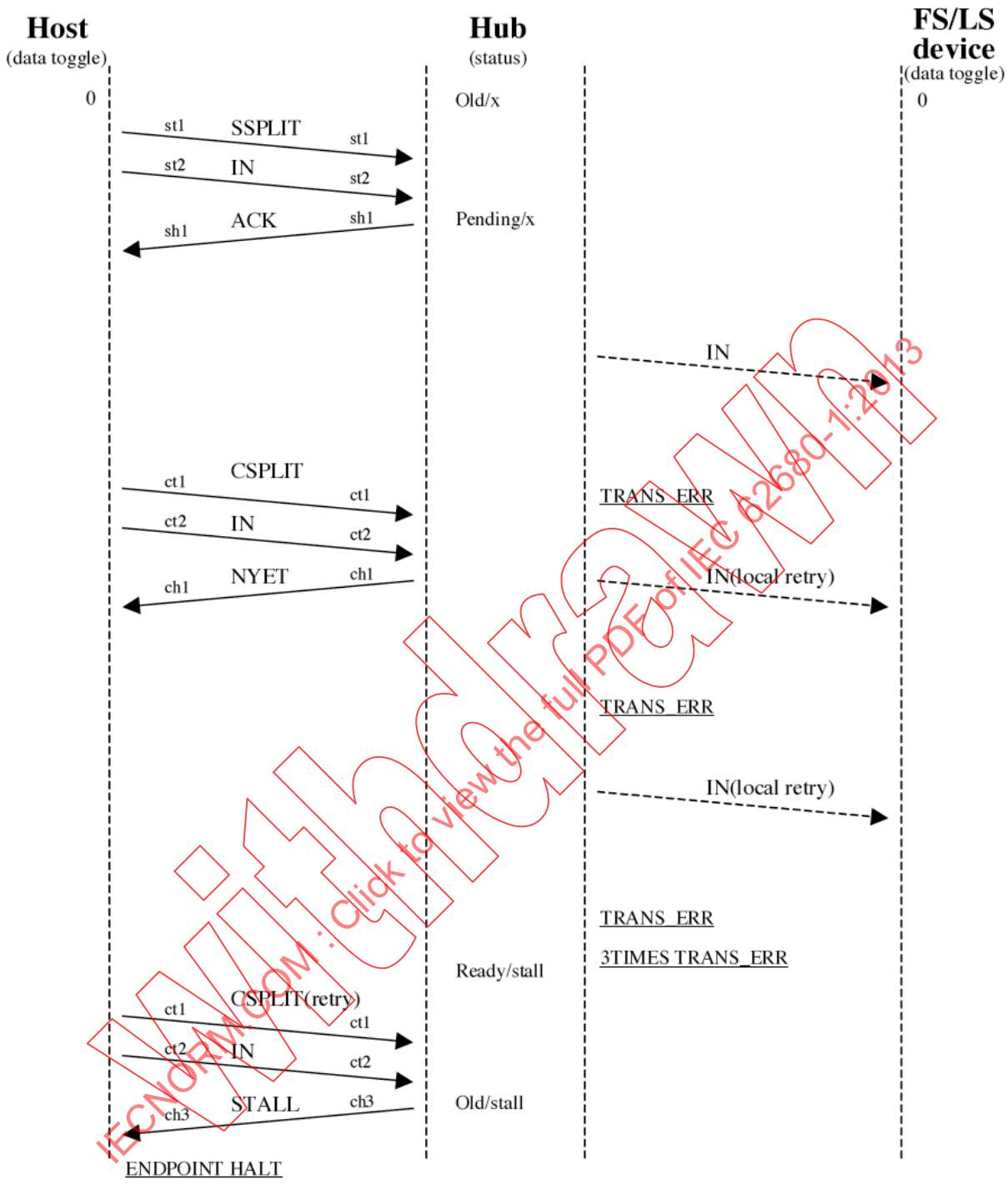


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
CSPLIT (retry)	CSPLIT (nouvel essai)

Figure A-34. Smash FS/LS IN normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



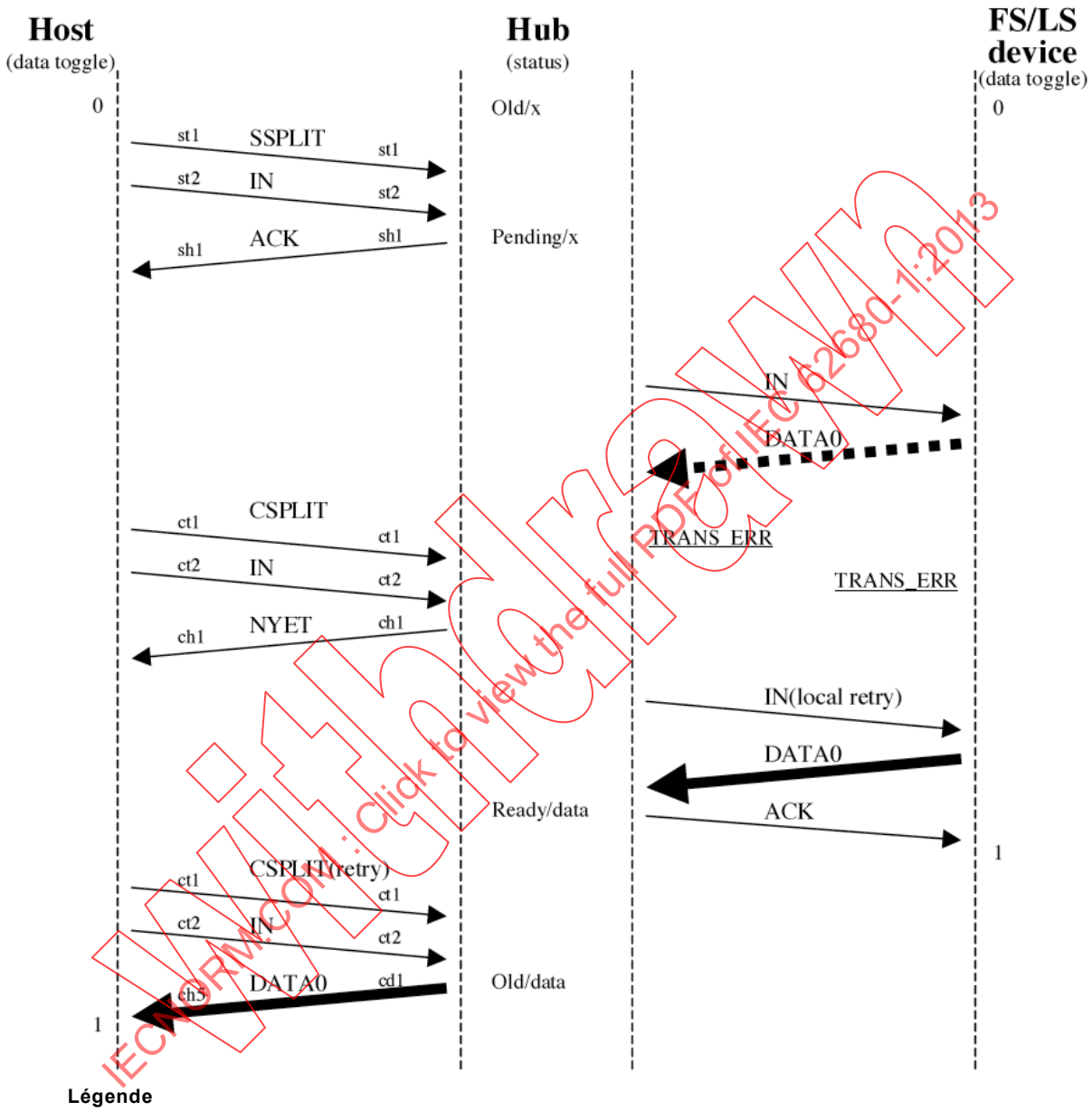
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/stall	Prêt/arrêt
Old/stall	Ancien/arrêt

Anglais	Français
CSPLIT (retry)	CSPLIT (nouvel essai)
IN(local retry)	IN (nouvel essai local)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-35. Smash 3 essais FS/LS IN normal

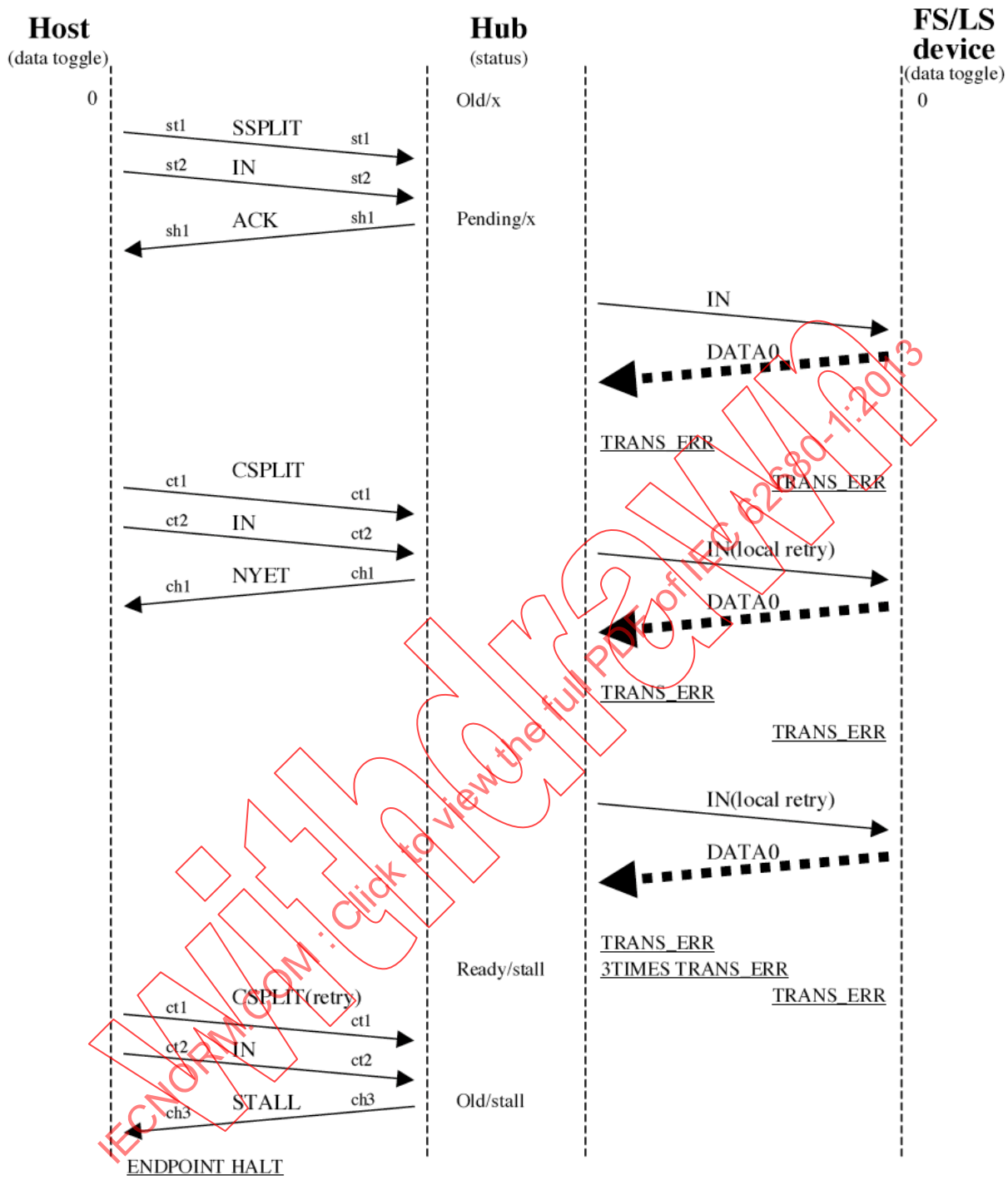


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
CSPLIT (retry)	CSPLIT (nouvel essai)
IN(local retry)	IN(nouvel essai local)

Figure A-36. Smash FS/LS DATA0/1 normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



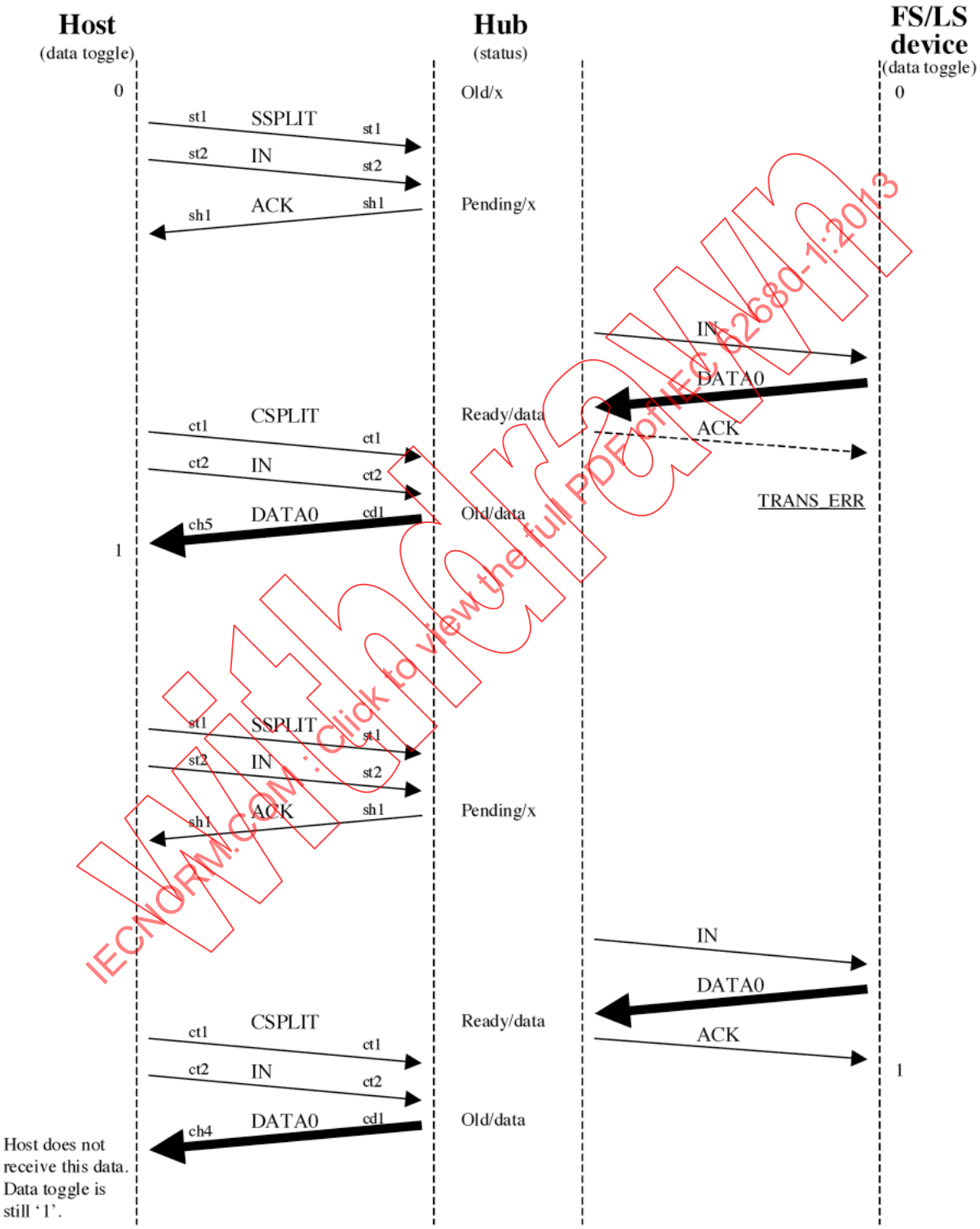
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/stall	Prêt/arrêt
Old/stall	Ancien/arrêt

Anglais	Français
CSPLIT (retry)	CSPLIT (nouvel essai)
IN(local retry)	IN(nouvel essai local)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-37. Smash 3 essais FS/LS DATA0/1 normal



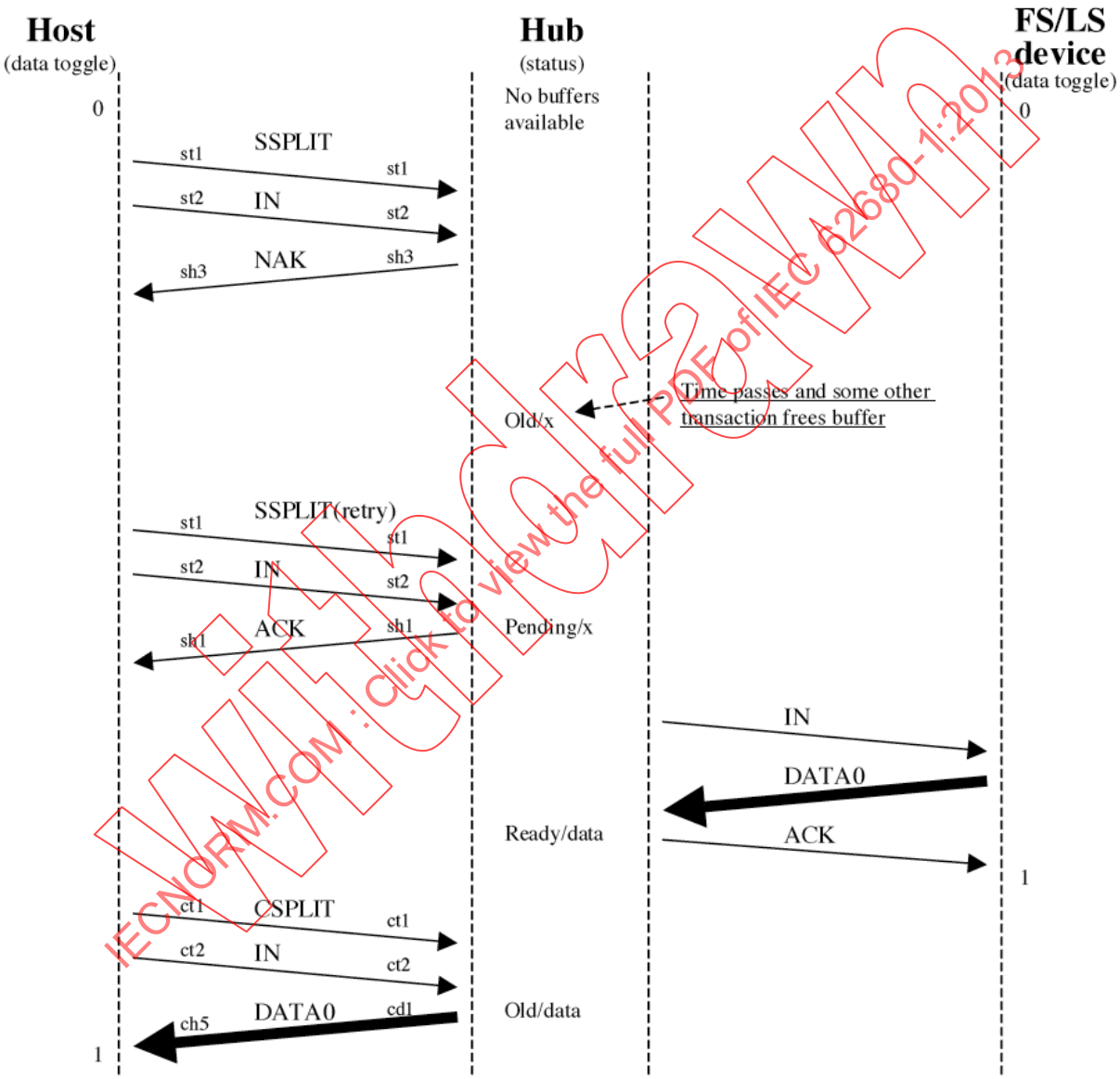
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)

Anglais	Français
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
Host does not receive this data. Data toggle is still '1'.	L'hôte ne reçoit pas ces données. Le basculement des données est encore '1'.

Figure A-38. Smash FS/LS ACK normal



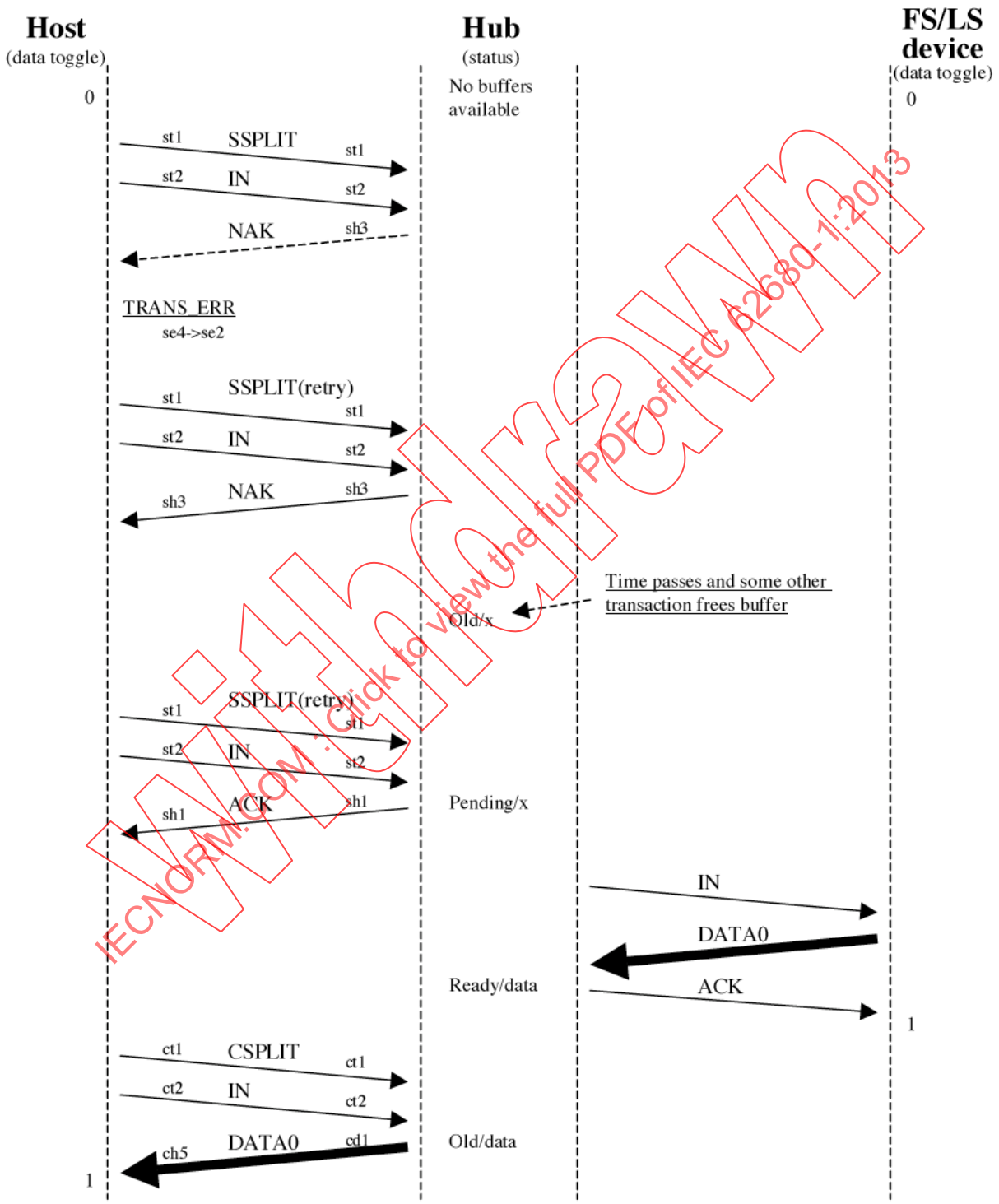
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
No buffers available	Pas de tampons disponibles
Old/x	Ancien/x
Time passes and some other transaction frees	Le temps passe et d'autres transactions libèrent

Anglais	Français
buffer	le tampon
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
SSPLIT (retry)	SSPLIT (nouvel essai)

Figure A-39. Pas de tampon disponible Pas de smash (HS NAK(S))



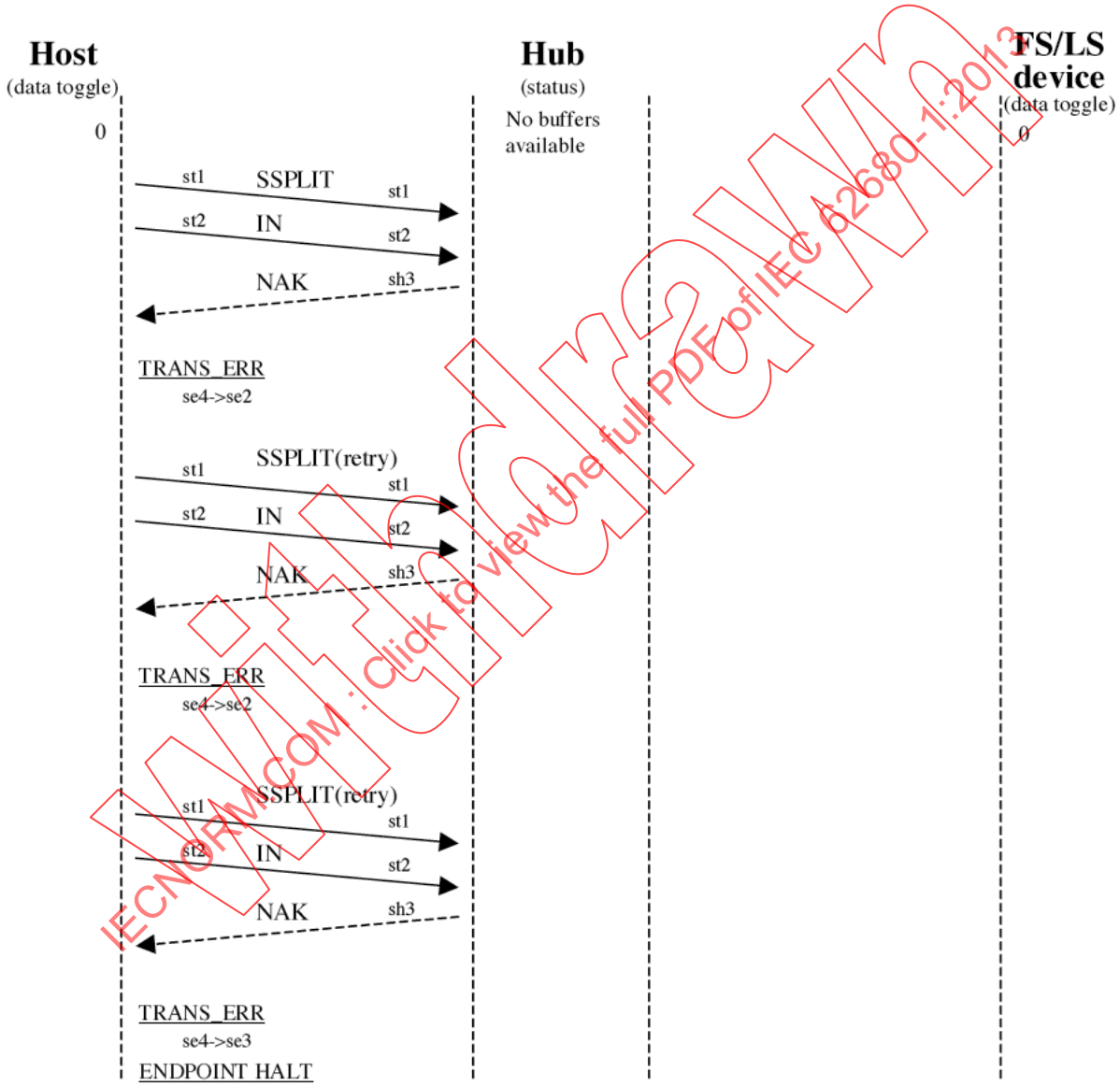
Withdrawn
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)

Anglais	Français
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
No buffers available	Pas de tampons disponibles
Old/x	Ancien/x
Time passes and some other transaction frees buffer	Le temps passe et d'autres transactions libèrent le tampon
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
SSPLIT (retry)	SSPLIT (nouvel essai)

Figure A-40. Pas de tampon disponible Smash HS NAK(S)



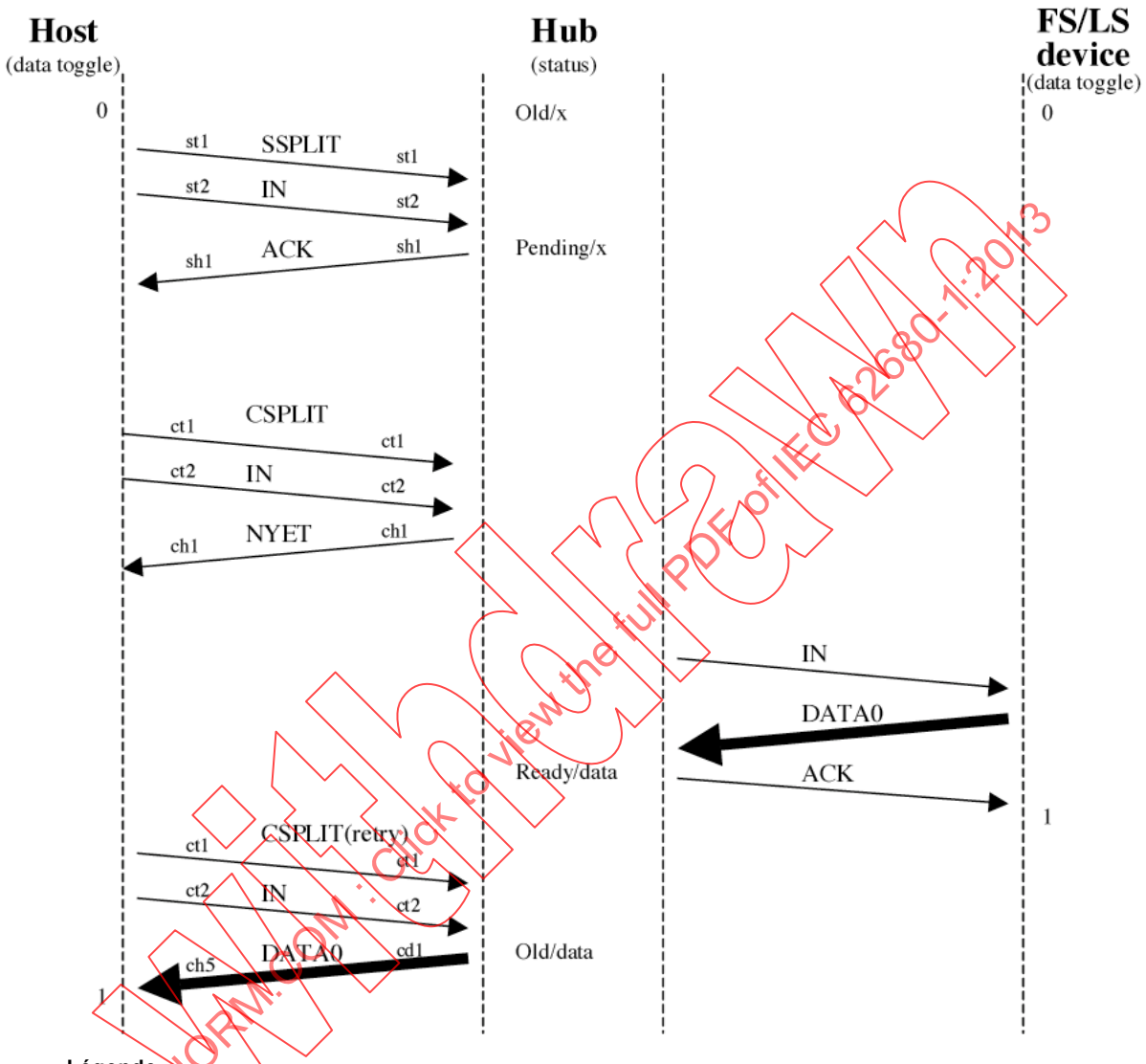
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Anglais	Français
No buffers available	Pas de tampons disponibles
SSPLIT (retry)	SSPLIT (nouvel essai)
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-41. Pas de tampon disponible Smash 3 essais HS NAK(S)

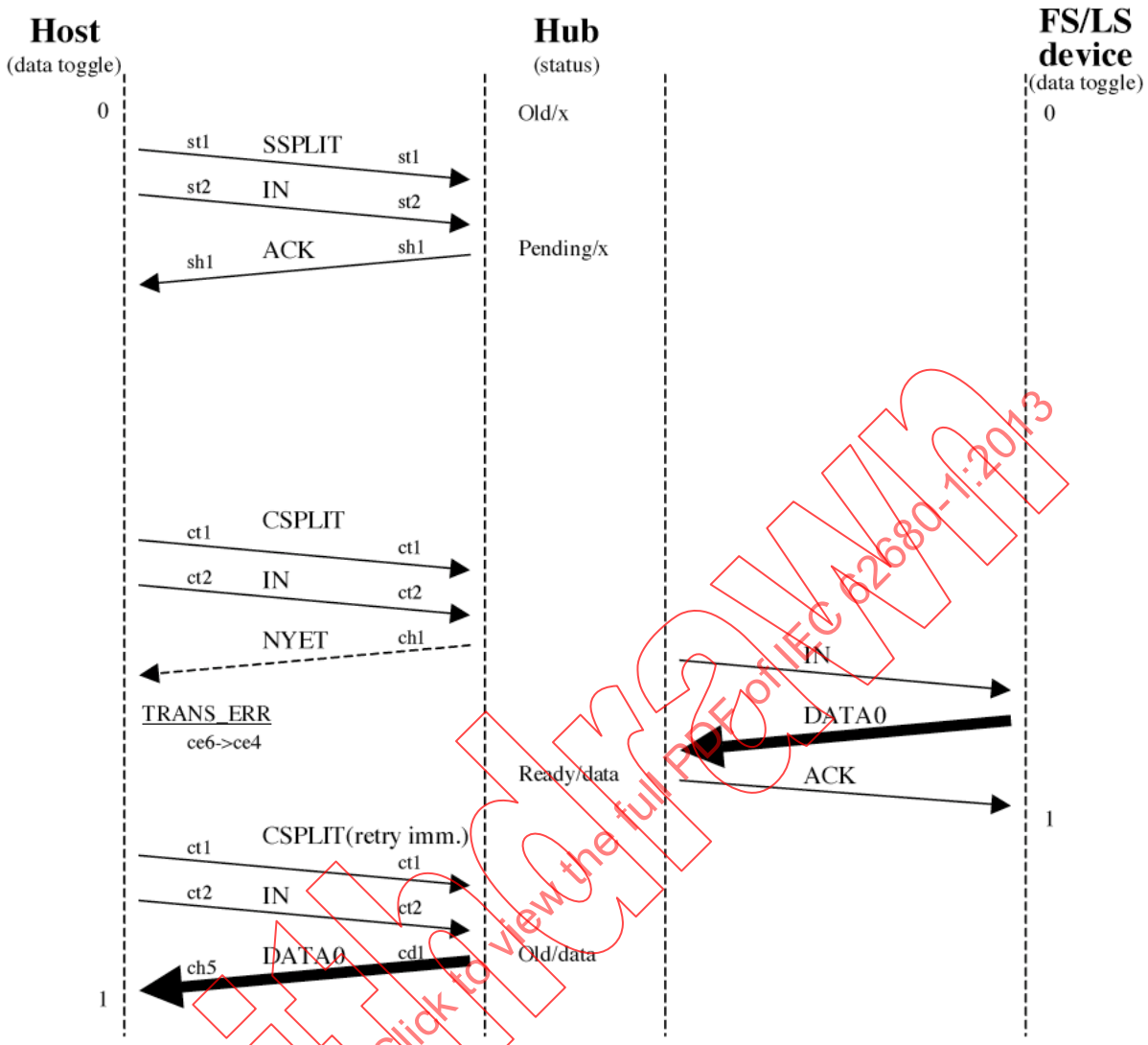


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
CSPLIT (retry)	CSPLIT (nouvel essai)

Figure A-42. CS plus tôt Pas de smash(HS NYET)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

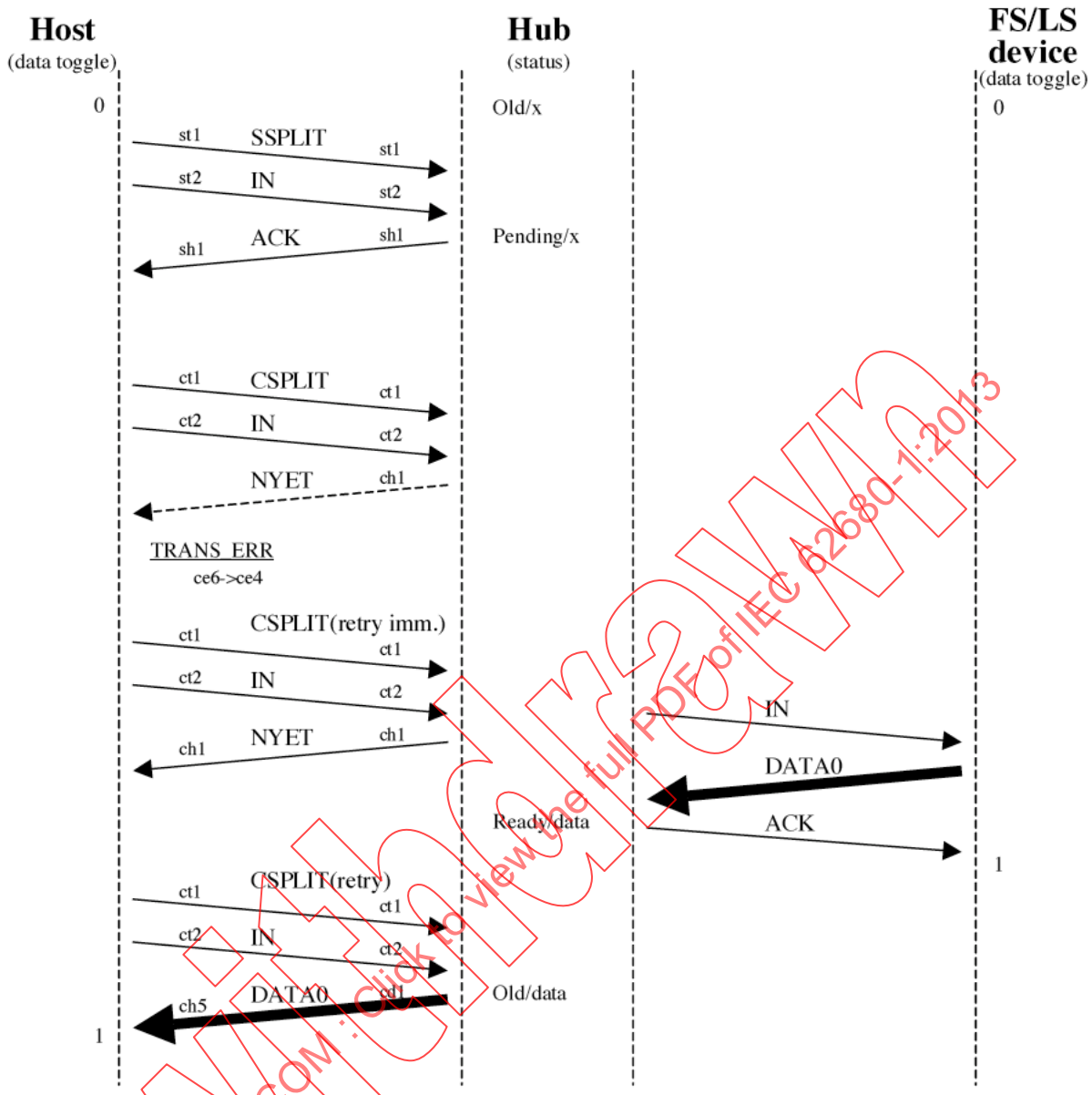


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)

Figure A-43. CS plus tôt Smash HS NYET (cas 1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

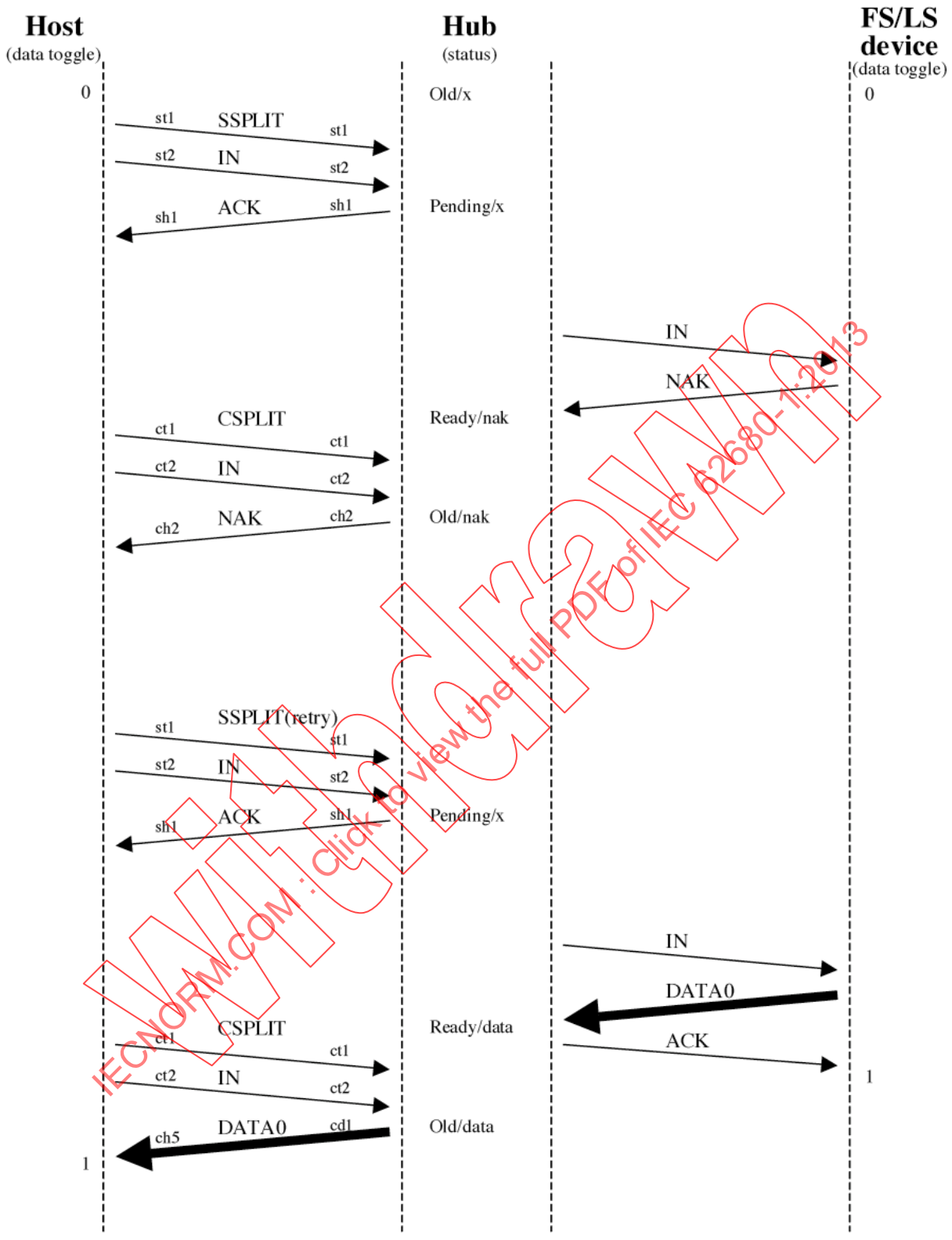


Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/data	Prêt/données
Old/data	Ancien/données
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
CSPLIT (retry)	CSPLIT (nouvel essai)

Figure A-44. CS plus tôt Smash HS NYET (cas 2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



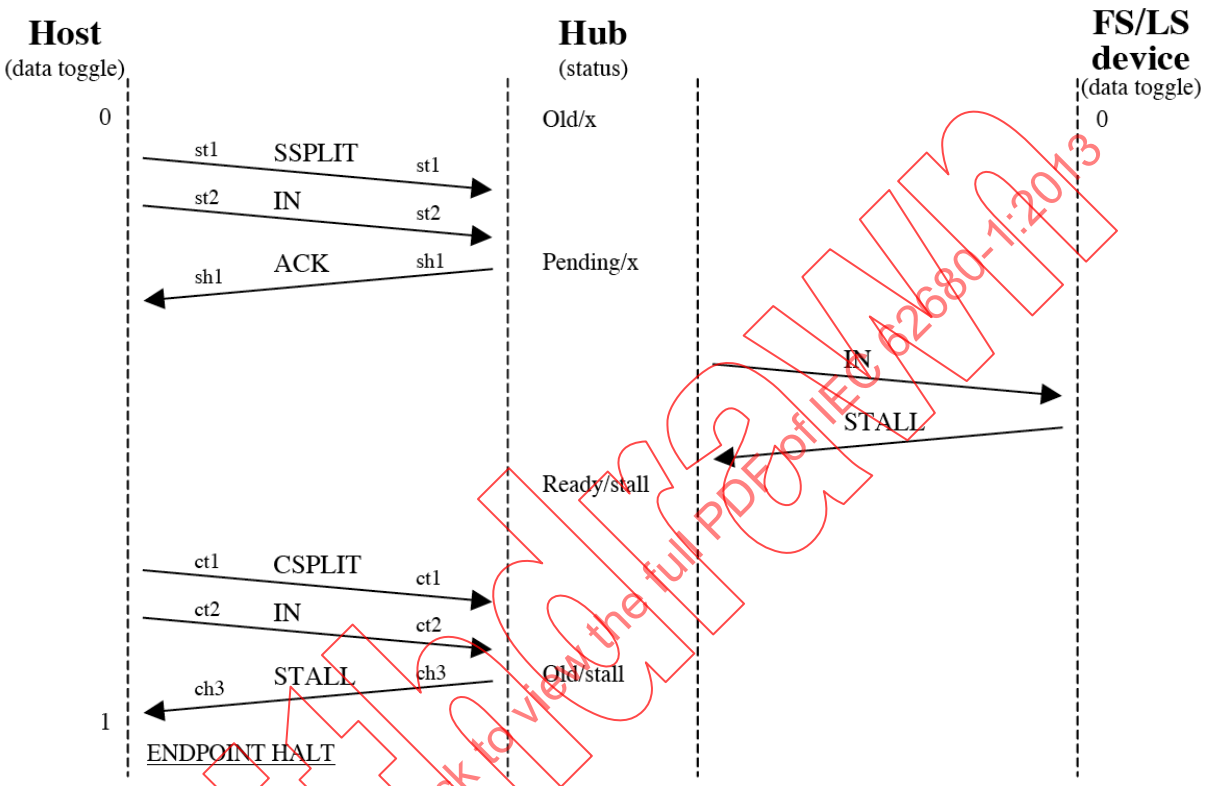
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)

Anglais	Français
Old/x	Ancien/x
Pending/x	En cours/x
Ready/nak	Prêt/nak
Old/nak	Ancien/nak
SSPLIT (retry)	SSPLIT (nouvel essai)
Ready/data	Prêt/données
Old/data	Ancien/données

Figure A-45. Appareil occupé Pas de smash(FS/LS NAK)



Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (status)	Hub (état)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
Old/x	Ancien/x
Pending/x	En cours/x
Ready/stall	Prêt/arrêt
Old/stall	Ancien/arrêt
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-46. Appareil arrêté Pas de smash(FS/LS STALL)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

A.3 Exemples de transactions OUT d'interruption

Légende:

(S): Start Split

(C): Complete Split

Résumé des cas de transaction OUT d'interruption

- Cas normaux

Cas	Figure de référence	Figure similaire
Pas de smash (le paquet d'établissement de liaison FS/LS est effectué par M+1)	Figure A-47	
Smash HS SSPLIT		Figure A-48
Smash 3 essais HS SSPLIT	Pas de figure	
Smash HS OUT(S)		Figure A-48
Smash 3 essais HS OUT(S)	Pas de figure	
Smash HS DATA0/1	Figure A-48	
Smash 3 essais HS DATA0/1	Pas de figure	
Smash HS CSPLIT	Figure A-49	
Smash 3 essais HS CSPLIT	Figure A-50	
Smash HS OUT(C)		Figure A-49
3 essais HS OUT(C)		Figure A-50
Smash HS ACK(C)	Figure A-51	
Smash 3 essais HS ACK(C)	Figure A-52	
Smash FS/LS OUT		Figure A-53
Smash 3 essais FS/LS OUT	Pas de figure	
Smash FS/LS DATA0/1	Figure A-53	
Smash 3 essais FS/LS DATA0/1	Pas de figure	
Smash FS/LS ACK	Figure A-54	
Smash 3 essais FS/LS ACK	Pas de figure	

- Recherche

Cas	Figure de référence	Figure similaire
Pas de smash	Figure A-55	

- CS(transaction complete-split) - Cas précédents

Cas	Figure de référence	Figure similaire
Pas de smash (le paquet d'établissement de liaison HS NYET et FS/LS est effectué par M+2)	Figure A-56	
Pas de smash (le paquet d'établissement de liaison HS NYET et FS/LS est effectué par M+3)	Figure A-57	
Smash HS NYET	Figure A-58	
Smash 3 essais HS NYET	Figure A-59	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

- Cas d'abandon et de libération

Cas	Figure de référence	Figure similaire
Pas de smash et abandon (la transaction HS NYET et FS/LS est poursuivie à la fin de M+3)	Figure A-60	
Pas de smash et libération (la transaction HS NYET et FS/LS n'est pas démarrée à la fin de M+3)	Figure A-61	

- Cas d'erreur de transaction FS/LS

Cas	Figure de référence	Figure similaire
Smash HS ERR		Figure A-51
Smash 3 essais HS ERR		Figure A-52

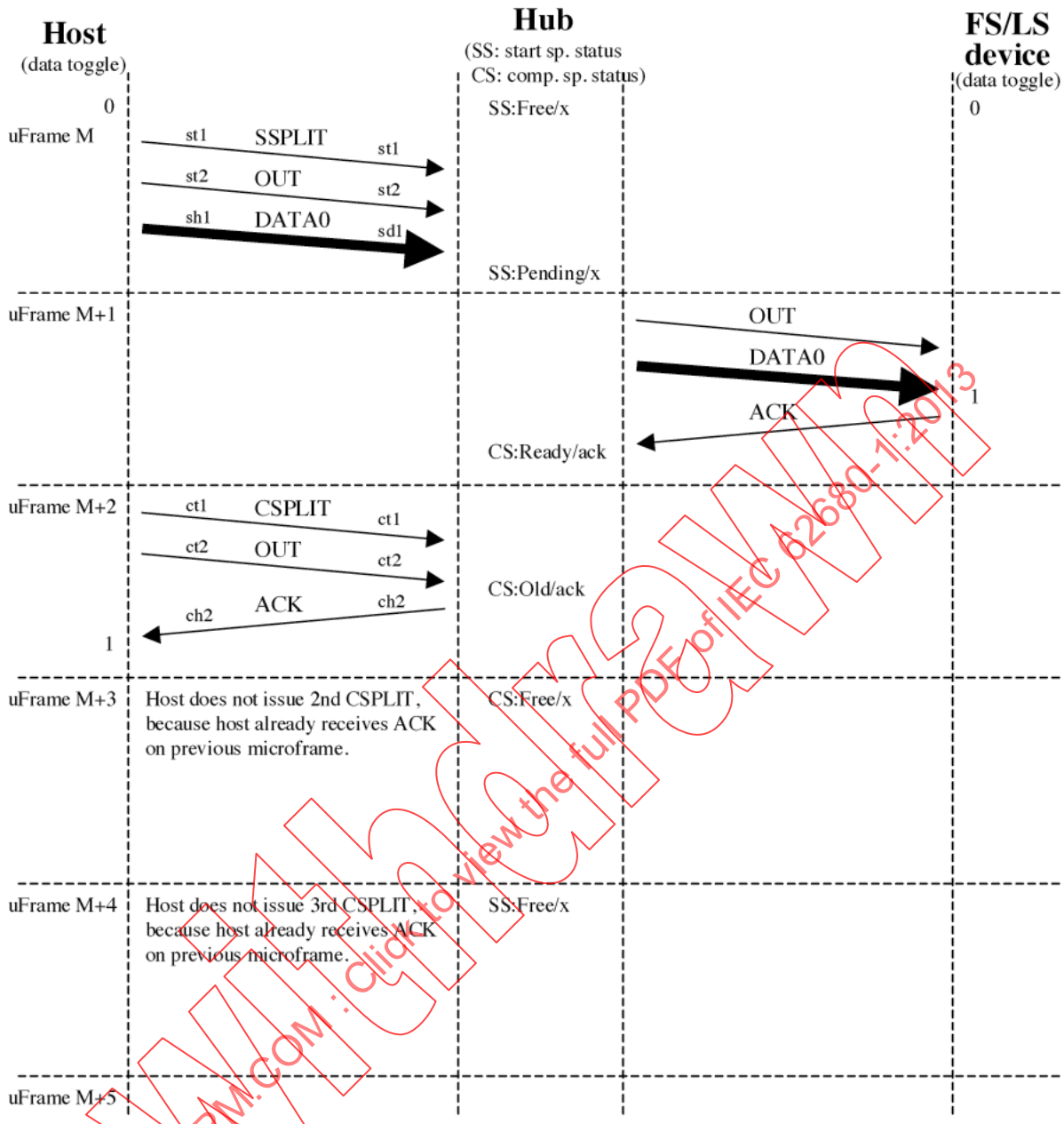
- Cas appareil occupé

Cas	Figure de référence	Figure similaire
Pas de smash(HS NAK(C))	Figure A-62	
Smash HS NAK(C)		Figure A-51
Smash 3 essais HS NAK(C)		Figure A-52
Smash FS/LS NAK		Figure A-53
Smash 3 essais FS/LS NAK	Pas de figure	

- Cas appareil arrêté

Cas	Figure de référence	Figure similaire
Pas de smash	Figure A-63	
Smash HS STALL(C)		Figure A-51
Smash 3 essais HS STALL(C)		Figure A-52
Smash FS/LS STALL		Figure A-53
Smash 3 essais FS/LS STALL	Pas de figure	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

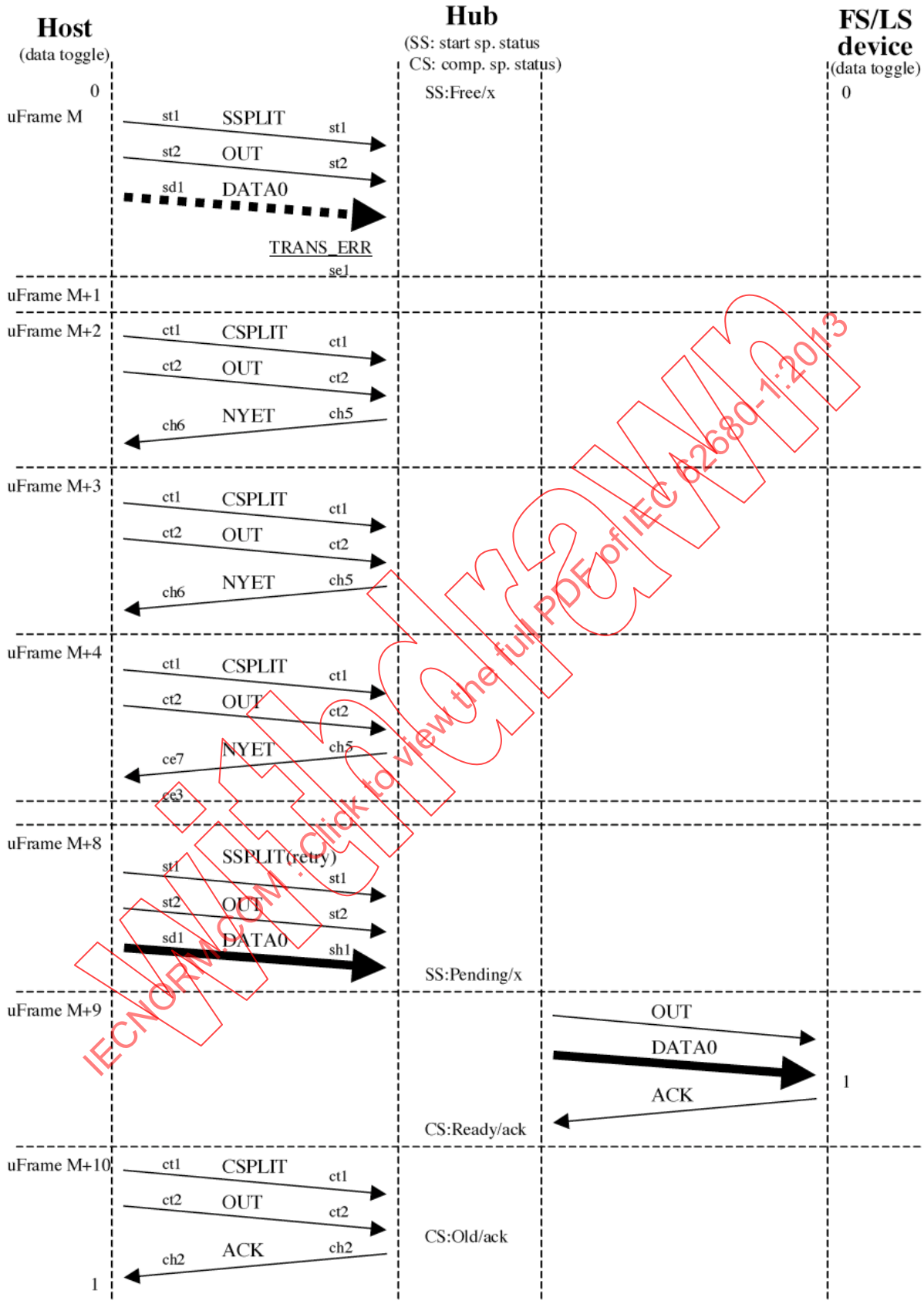
Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
SSPLIT(retry)	SSPLIT (nouvel essai)
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 5	uTrame M + 5
Host does not issue 2nd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 2ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.
Host does not issue 3rd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 3ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.

Figure A-47. Normal Pas de smash (le paquet d'établissement de liaison FS/LS est effectué par M+1)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

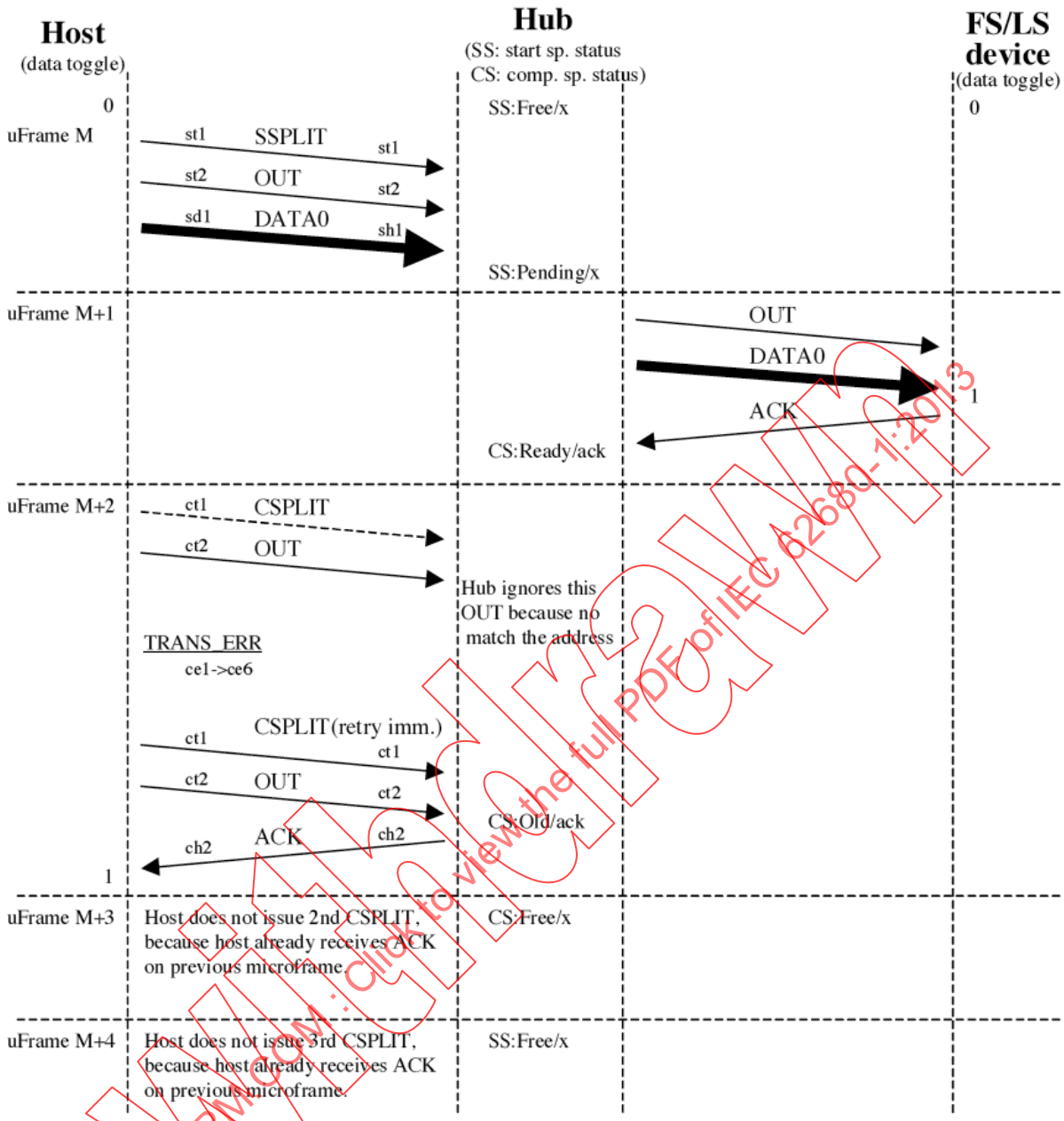
Légende

Anglais	Français
Host	Hôte

Anglais	Français
(data toggle)	(basculement des données)
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 8	uTrame M + 8
uFrame M + 9	uTrame M + 9
uFrame M + 10	uTrame M + 10

Figure A-48. Smash HS DATA0/1 normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

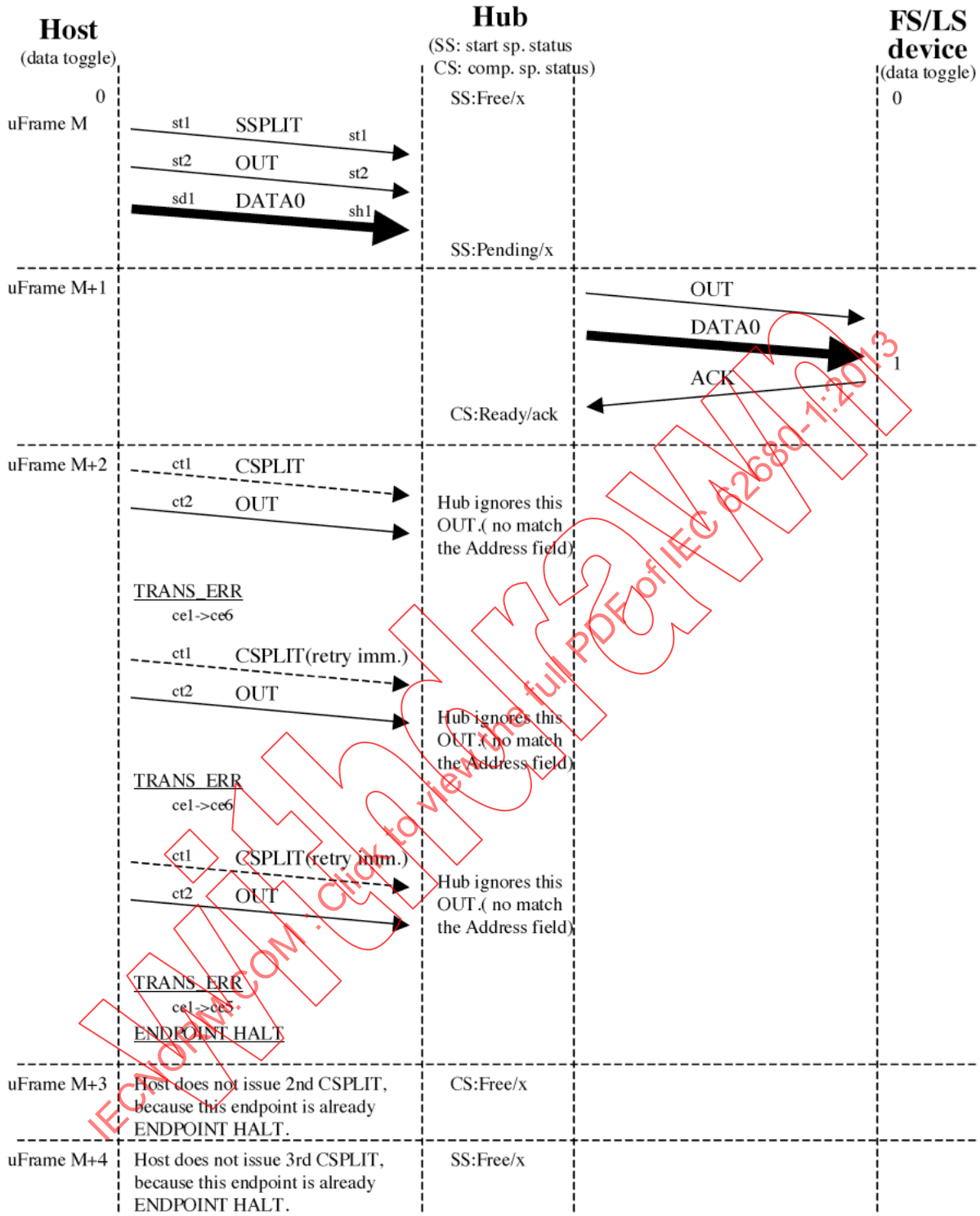
Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
Hub ignores this OUT because no match the address.	L'hôte ignore ce OUT car pas de correspondance d'adresse.
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
Host does not issue 2nd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 2ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.
Host does not issue 3rd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 3ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.

Figure A-49. Smash HS CSPLIT normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

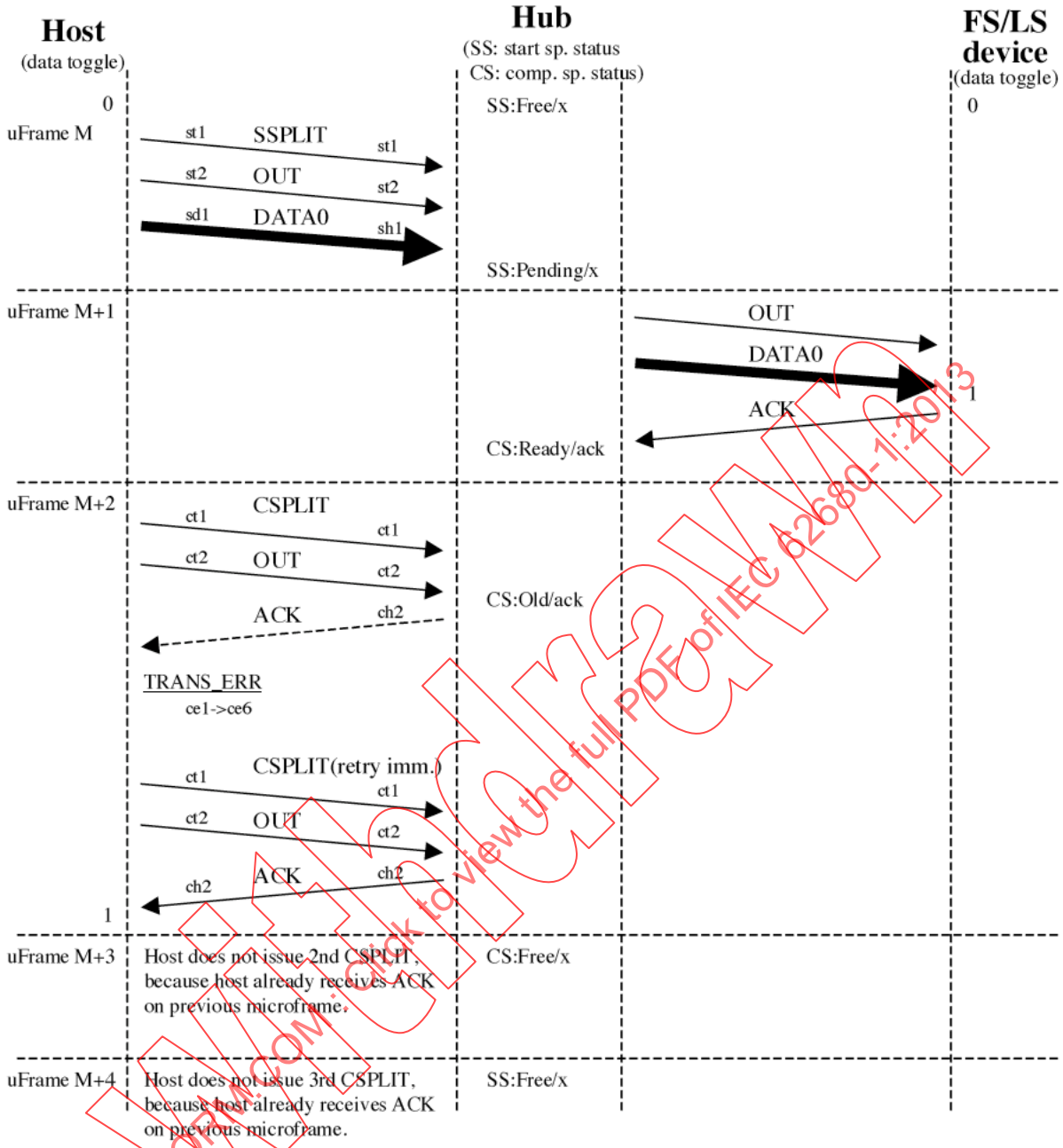
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)

Anglais	Français
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
Hub ignores this OUT (no match the Address field).	L'hôte ignore ce OUT (pas de correspondance de champ d'adresse).
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
Host does not issue 2nd CSPLIT, because this endpoint is already ENDPOINT HALT.	L'hôte n'émet pas un 2ème CSPLIT, car ce point d'extrémité est déjà l'ARRÊT POINT D'EXTRÉMITÉ.
Host does not issue 3rd CSPLIT, because this endpoint is already ENDPOINT HALT.	L'hôte n'émet pas un 3ème CSPLIT, car ce point d'extrémité est déjà l'ARRÊT POINT D'EXTRÉMITÉ.
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ

Figure A-50. Smash 3 essais HS CSPLIT normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



Légende

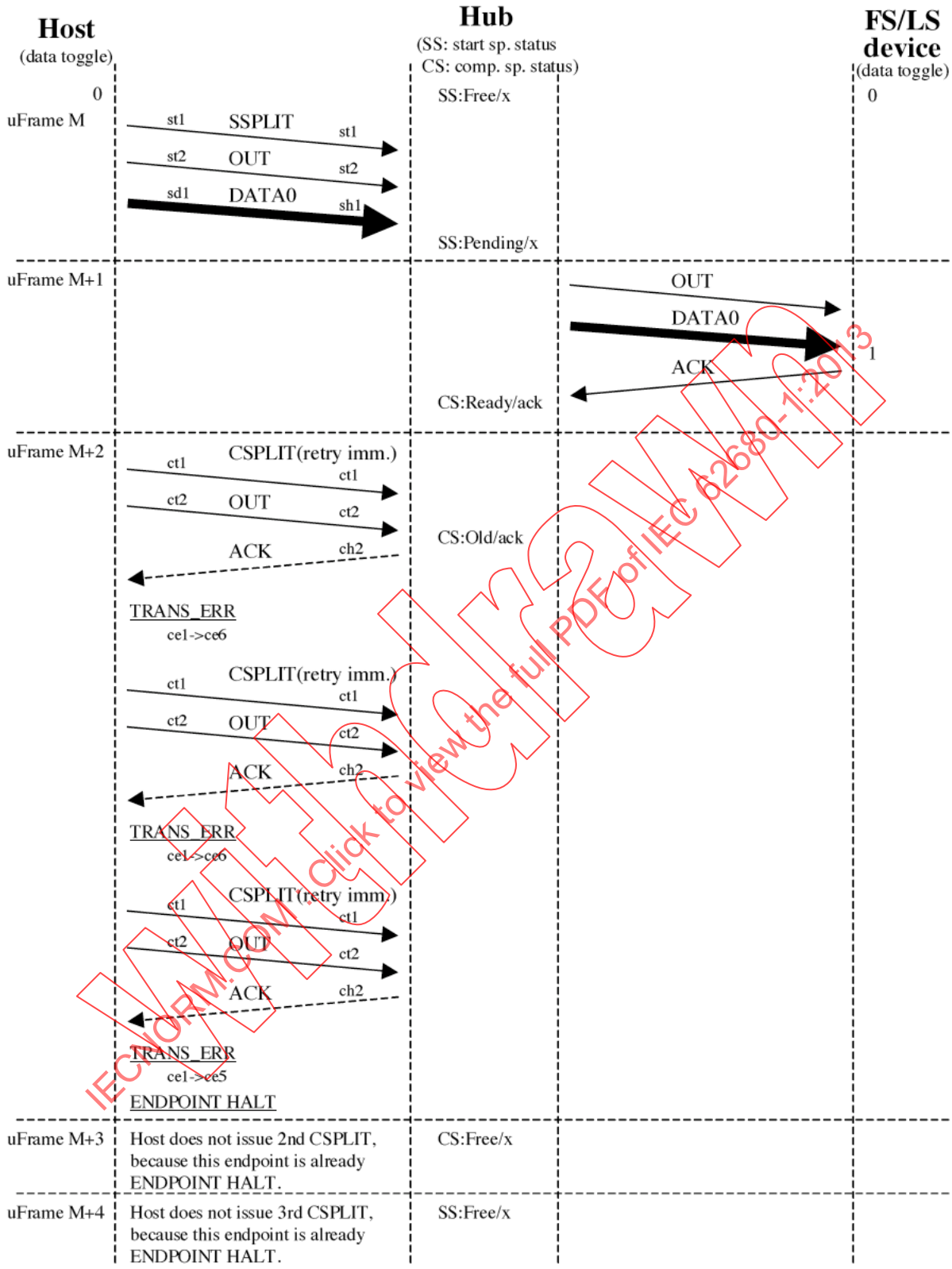
Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
Host does not issue 2nd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 2ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.
Host does not issue 3rd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 3ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.

Figure A-51. Smash HS ACK(C) normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

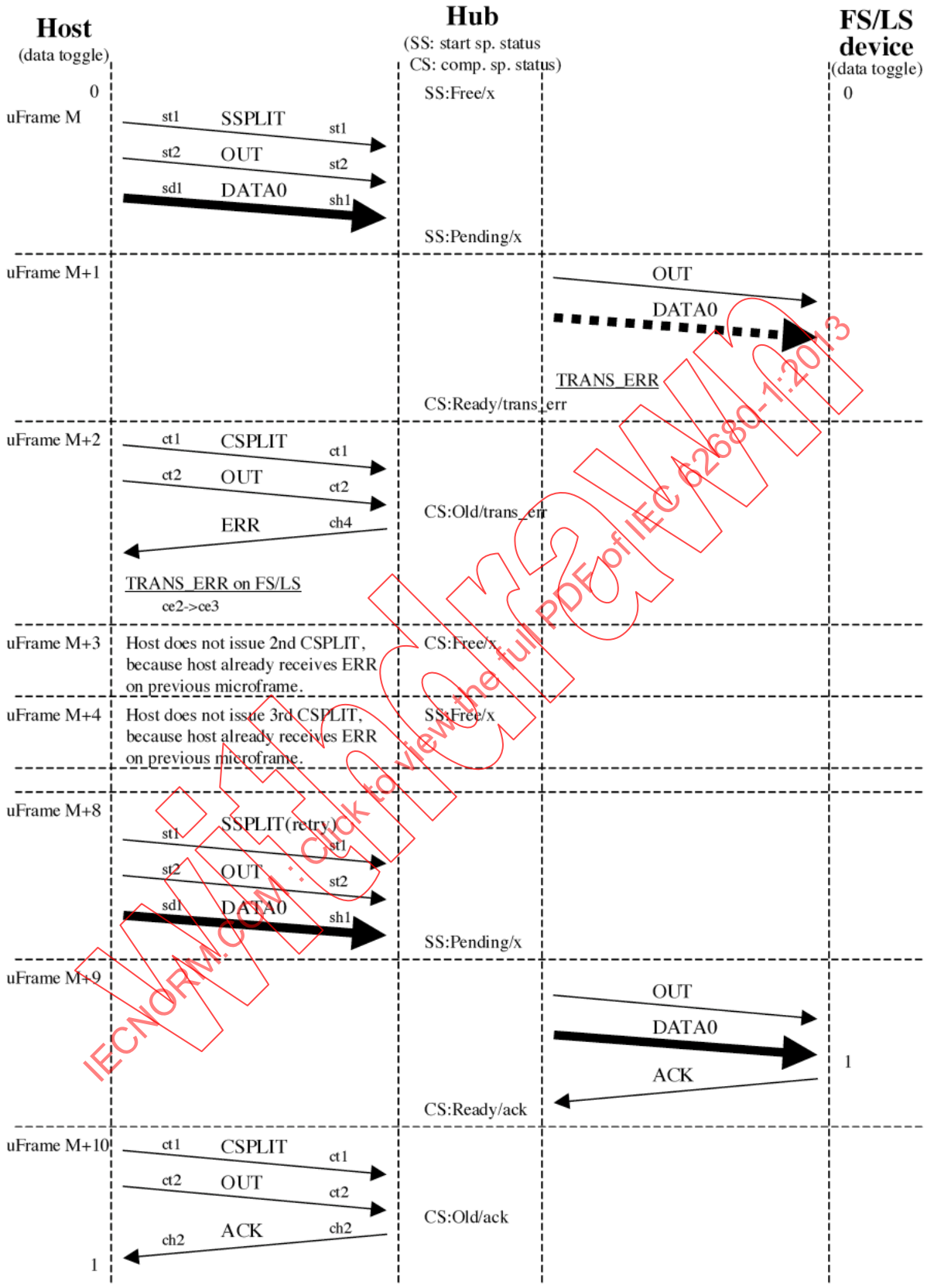
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (SS: start sp. status	Hub (SS: état start sp.

Anglais	Français
CS: comp. sp. status)	CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
Host does not issue 2nd CSPLIT, because this endpoint is already ENDPOINT HALT.	L'hôte n'émet pas un 2ème CSPLIT, car ce point d'extrémité est déjà l'ARRÊT POINT D'EXTRÉMITÉ.
Host does not issue 3rd CSPLIT, because this endpoint is already ENDPOINT HALT.	L'hôte n'émet pas un 3ème CSPLIT, car ce point d'extrémité est déjà l'ARRÊT POINT D'EXTRÉMITÉ.

Figure A-52. Smash 3 essais HS ACK(C) normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

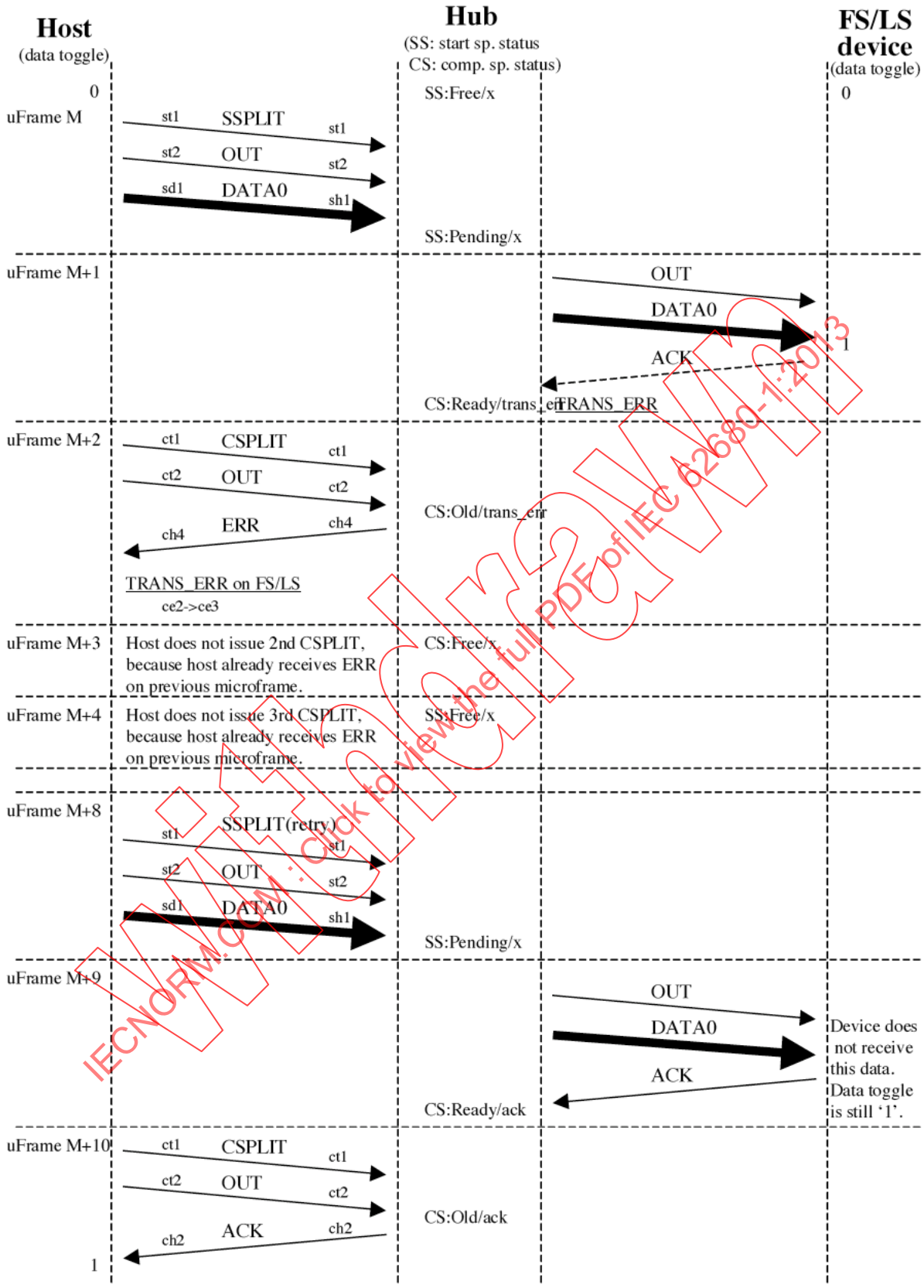
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)

Anglais	Français
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/trans_err	CS:Prêt/trans_err
CS:Old/trans_err	CS:Ancien/trans_err
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 8	uTrame M + 8
uFrame M + 9	uTrame M + 9
uFrame M + 10	uTrame M + 10
SSPLIT (retry)	SSPLIT (nouvel essai)
Host does not issue 2nd CSPLIT, because host already receives ERR on previous microframe.	L'hôte n'émet pas un 2ème CSPLIT, car l'hôte reçoit déjà ERR sur microtrame précédente.
Host does not issue 3rd CSPLIT, because host already receives ERR on previous microframe.	L'hôte n'émet pas un 3ème CSPLIT, car l'hôte reçoit déjà ERR sur microtrame précédente.
TRANS_ERR on FS/LS	TRANS_ERR sur FS/LS

Figure A-53. Smash FS/LS DATA0/1 normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

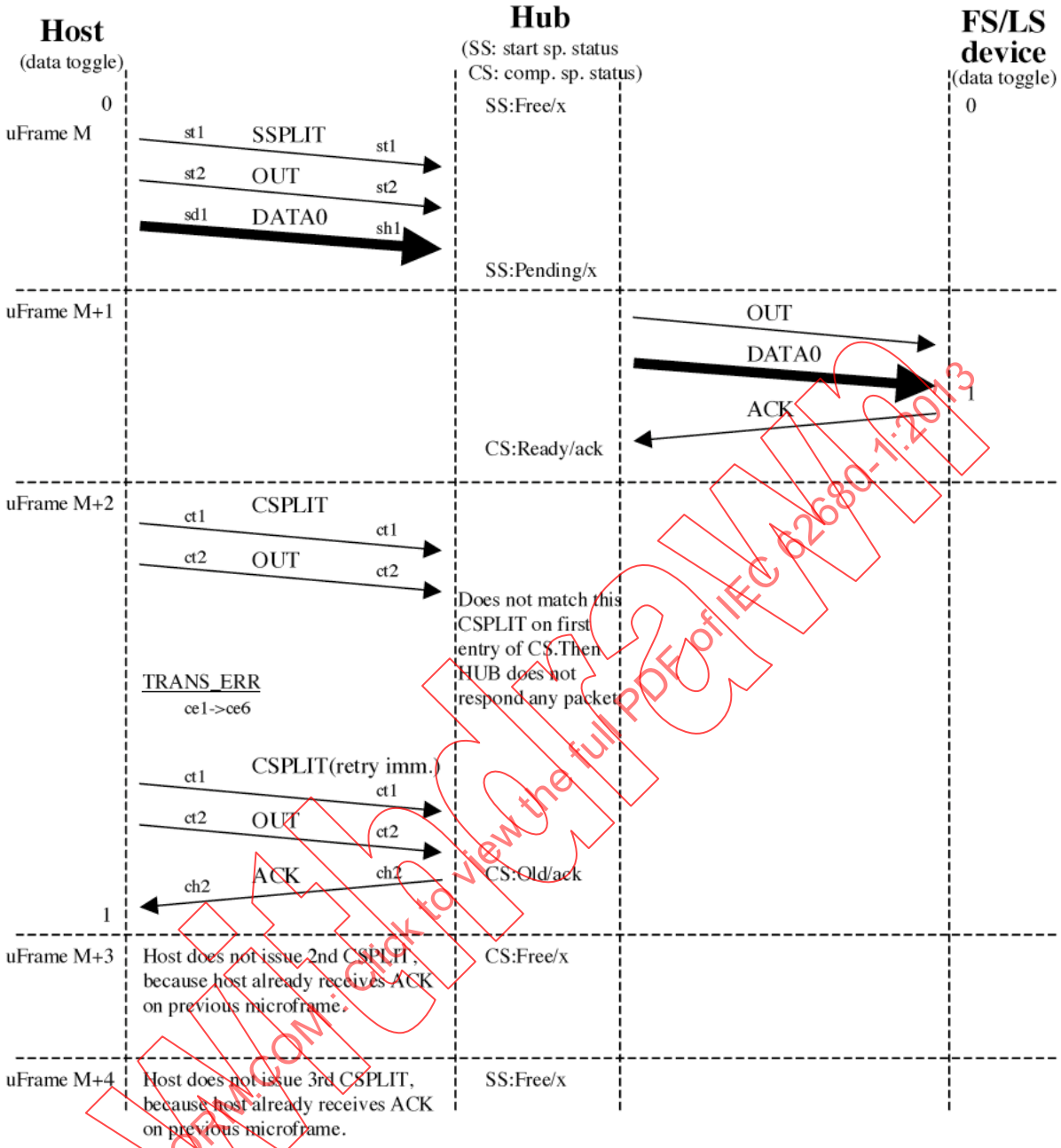
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)

Anglais	Français
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/trans_err	CS:Prêt/trans_err
CS:Old/trans_err	CS:Ancien/trans_err
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 8	uTrame M + 8
uFrame M + 9	uTrame M + 9
uFrame M + 10	uTrame M + 10
SSPLIT (retry)	SSPLIT (nouvel essai)
Host does not issue 2nd CSPLIT, because host already receives ERR on previous microframe.	L'hôte n'émet pas un 2ème CSPLIT, car l'hôte reçoit déjà ERR sur microtrame précédente.
Host does not issue 3rd CSPLIT, because host already receives ERR on previous microframe.	L'hôte n'émet pas un 3ème CSPLIT, car l'hôte reçoit déjà ERR sur microtrame précédente.
TRANS_ERR on FS/LS	TRANS_ERR sur FS/LS
Device does not receive this data. Data toggle is still '1'.	L'appareil ne reçoit pas ces données. Le basculement des données est encore '1'.

Figure A-54. Smash FS/LS ACK normal

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



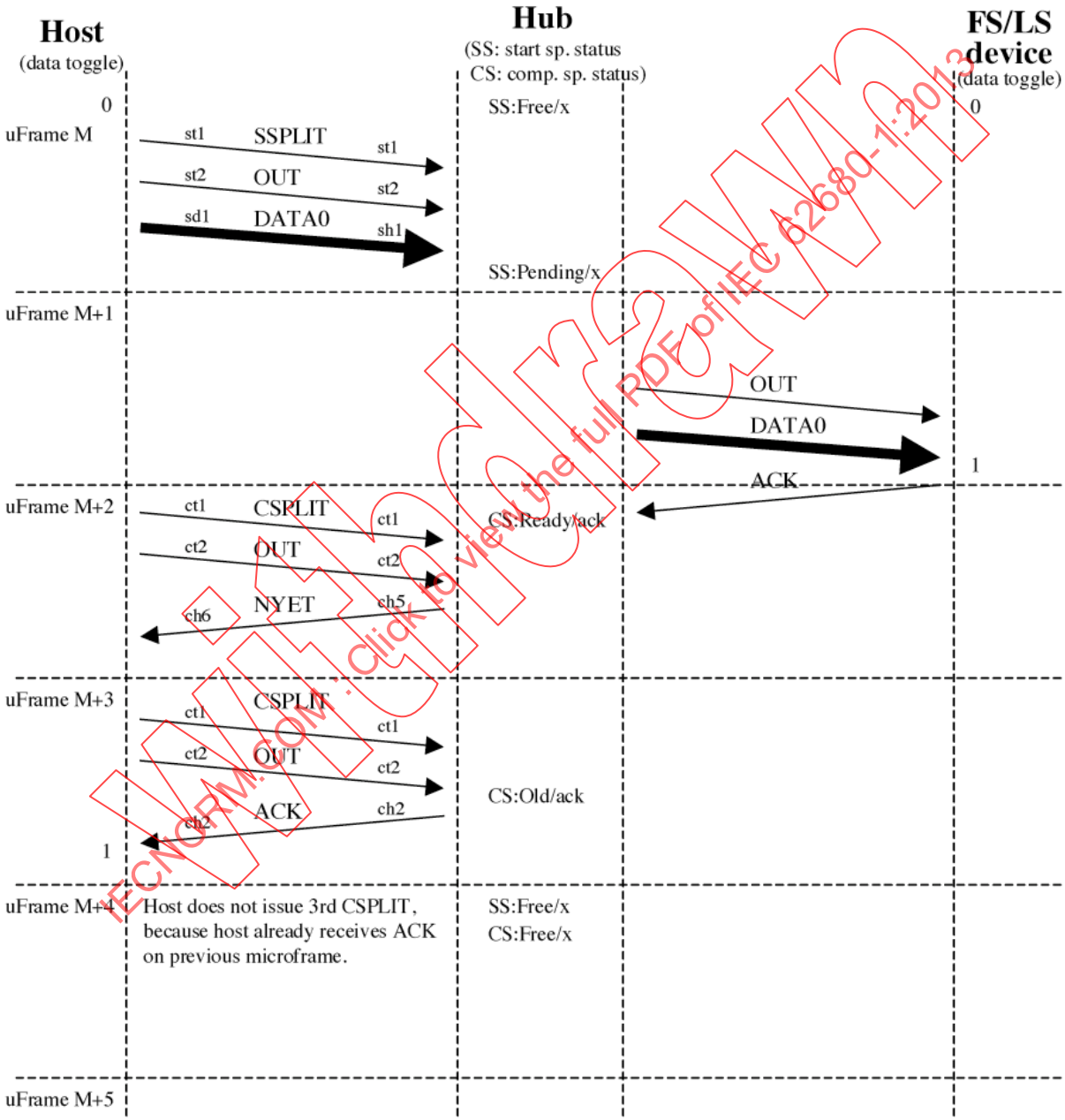
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Anglais	Français
uFrame M + 4	uTrame M + 4
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
Host does not issue 2nd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 2ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.
Host does not issue 3rd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 3ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.
TRANS_ERR on FS/LS	TRANS_ERR sur FS/LS
Does not match this CSPLIT on first entry of CS. Then HUB does not respond any packet	Ne correspond pas à ce CSPLIT sur la première entrée de CS. Puis HUB ne répond à aucun paquet

Figure A-55. Recherche Pas de smash



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

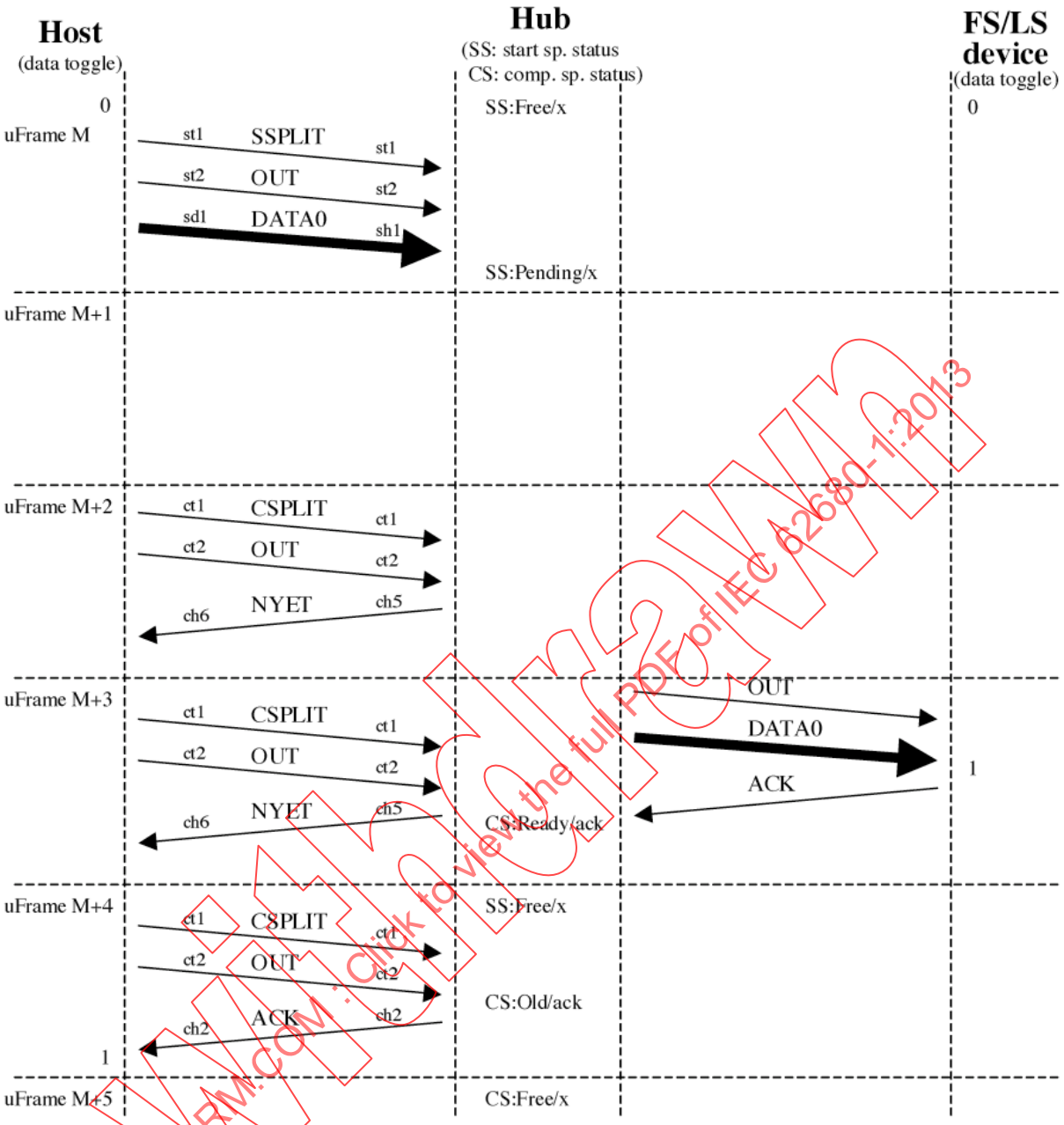
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub	Hub

Anglais	Français
(SS: start sp. status CS: comp. sp. status)	(SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x
CS:Free/x	CS:Libre/x
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 5	uTrame M + 5
Host does not issue 3rd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 3ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.

Figure A-56 CS plus tôt Pas de smash (le paquet d'établissement de liaison HS NYET et FS/LS est effectué par M+2)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



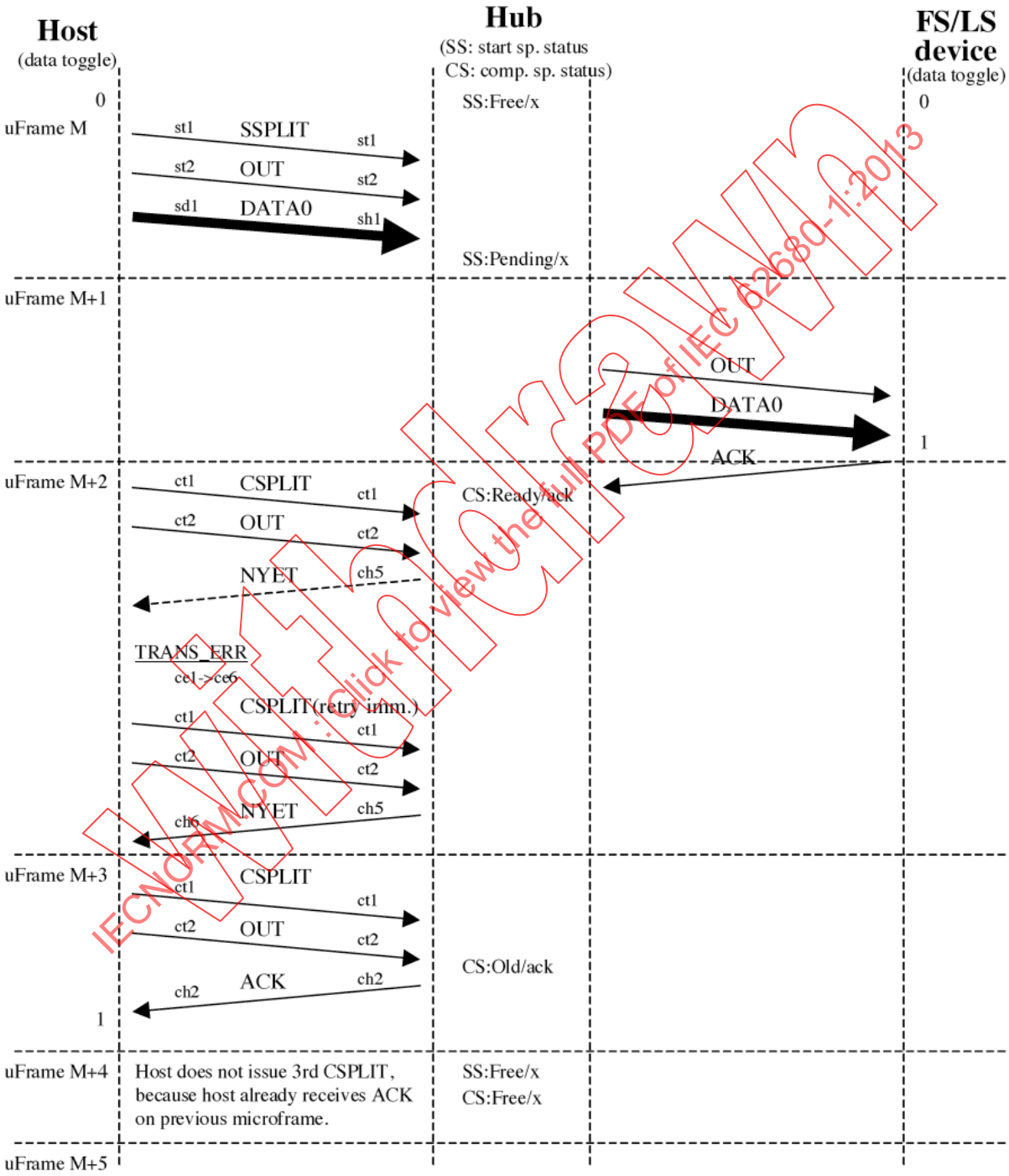
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
SS:Free/x	SS:Libre/x
CS:Old/x	CS:Ancien/x
CS:Free/x	CS:Libre/x
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1

Anglais	Français
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 5	uTrame M + 5

Figure A-57. CS plus tôt Pas de smash (le paquet d'établissement de liaison HS NYET et FS/LS est effectué par M+3)



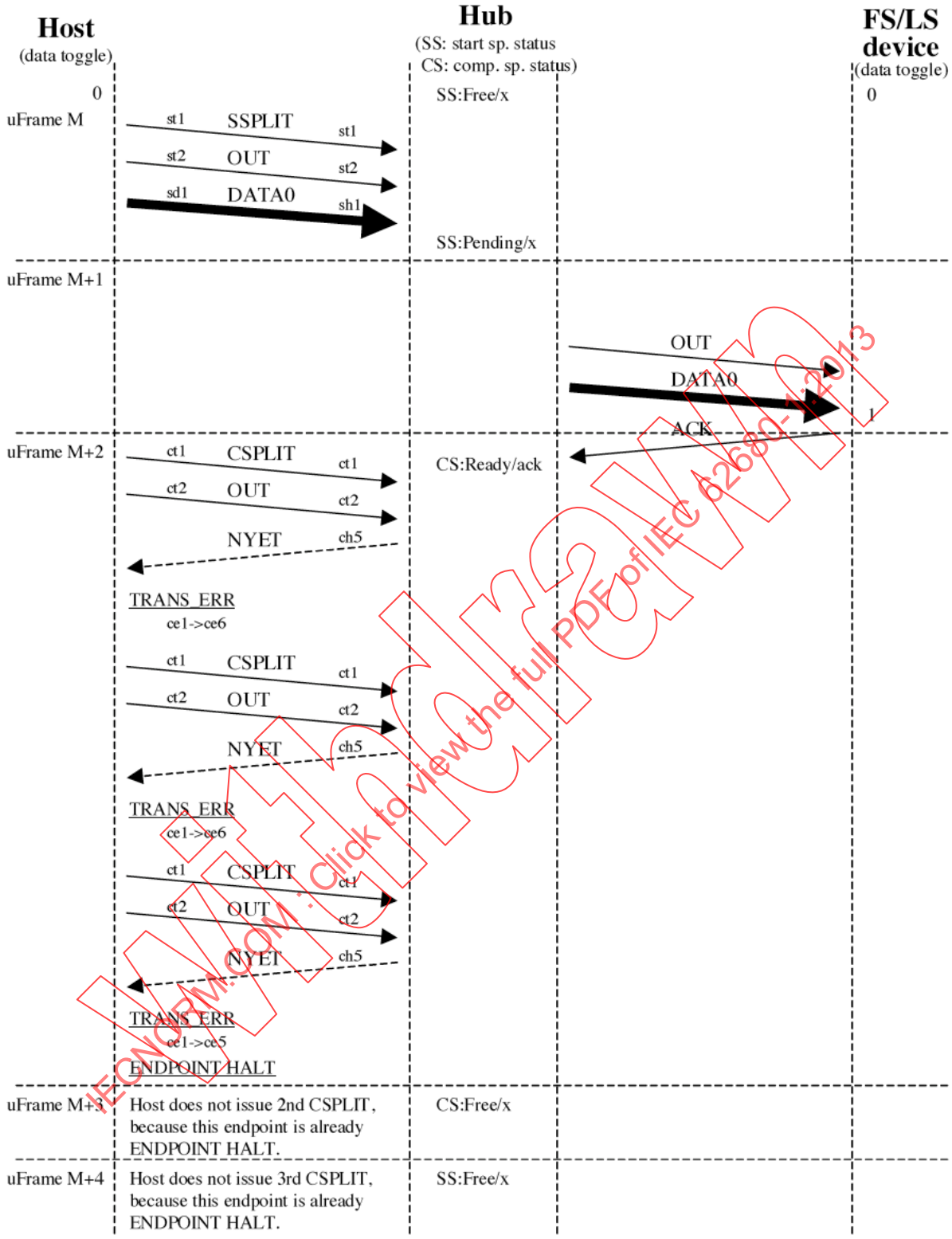
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)

Anglais	Français
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
SS:Free/x	SS:Libre/x
CS:Free/x	CS:Libre/x
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 5	uTrame M + 5
CSPLIT (retry imm.)	CSPLIT (nouvel essai imm.)
Host does not issue 3rd CSPLIT, because host already receives ACK on previous microframe.	L'hôte n'émet pas un 3ème CSPLIT, car l'hôte reçoit déjà ACK sur microtrame précédente.

Figure A-58. CS plus tôt Smash HS NYET

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

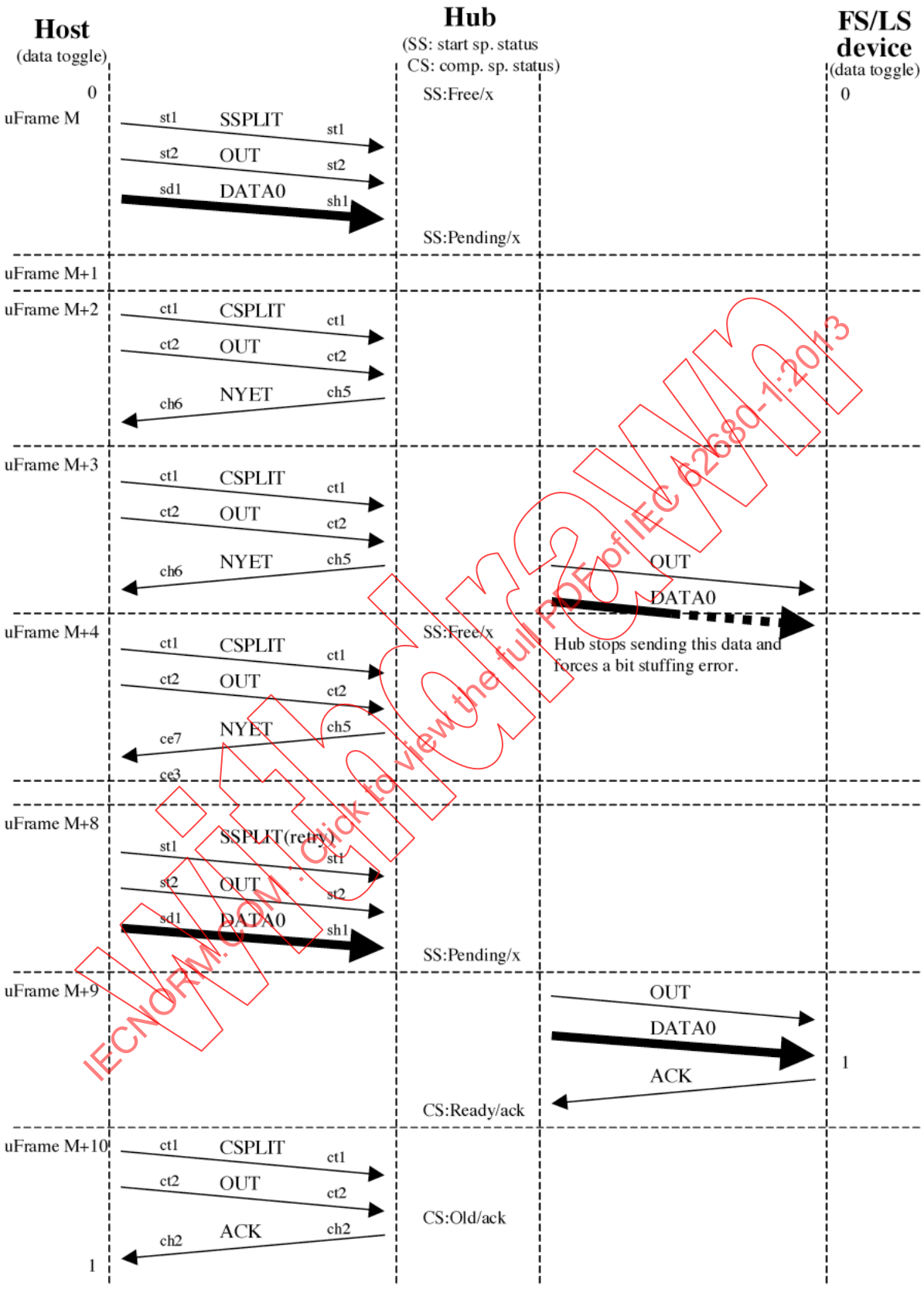
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)

Anglais	Français
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ
Host does not issue 2nd CSPLIT, because this endpoint is already ENDPOINT HALT.	L'hôte n'émet pas un 2ème CSPLIT, car ce point d'extrémité est déjà l'ARRÊT POINT D'EXTRÉMITÉ.
Host does not issue 3rd CSPLIT, because this endpoint is already ENDPOINT HALT.	L'hôte n'émet pas un 3ème CSPLIT, car ce point d'extrémité est déjà l'ARRÊT POINT D'EXTRÉMITÉ.

Figure A-59. CS plus tôt Smash 3 essais HS NYET

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

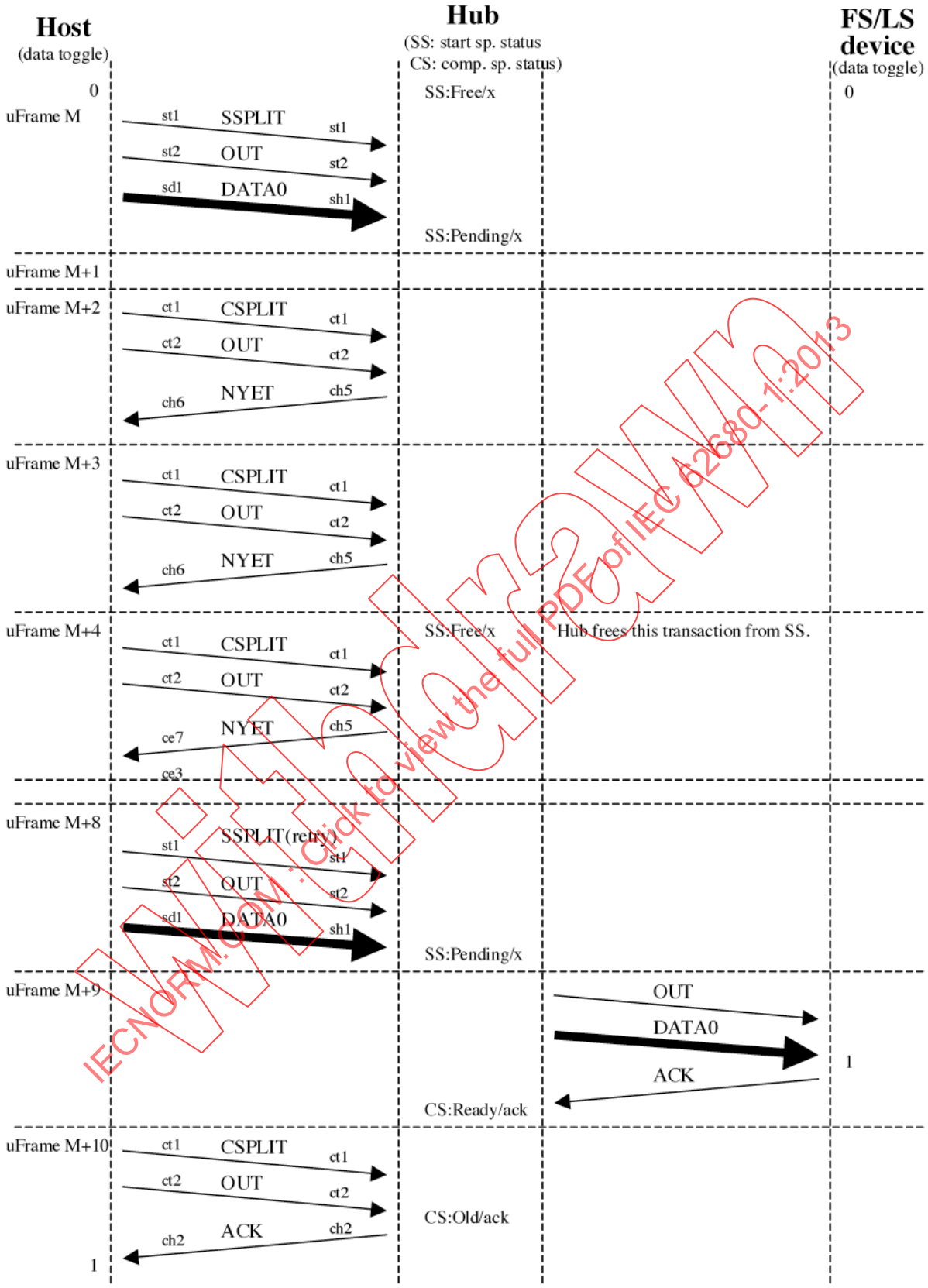
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)

Anglais	Français
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 8	uTrame M + 8
uFrame M + 9	uTrame M + 9
uFrame M + 10	uTrame M + 10
SSPLIT (retry)	SSPLIT (nouvel essai)
Hub stops sending this data and forces a bit stuffing error.	Le hub arrête d'envoyer ces données et force une erreur de bourrage de bits.

Figure A-60. Abandon et Libération Abandon (la transaction FS/LS est poursuivie à la fin de M+3)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

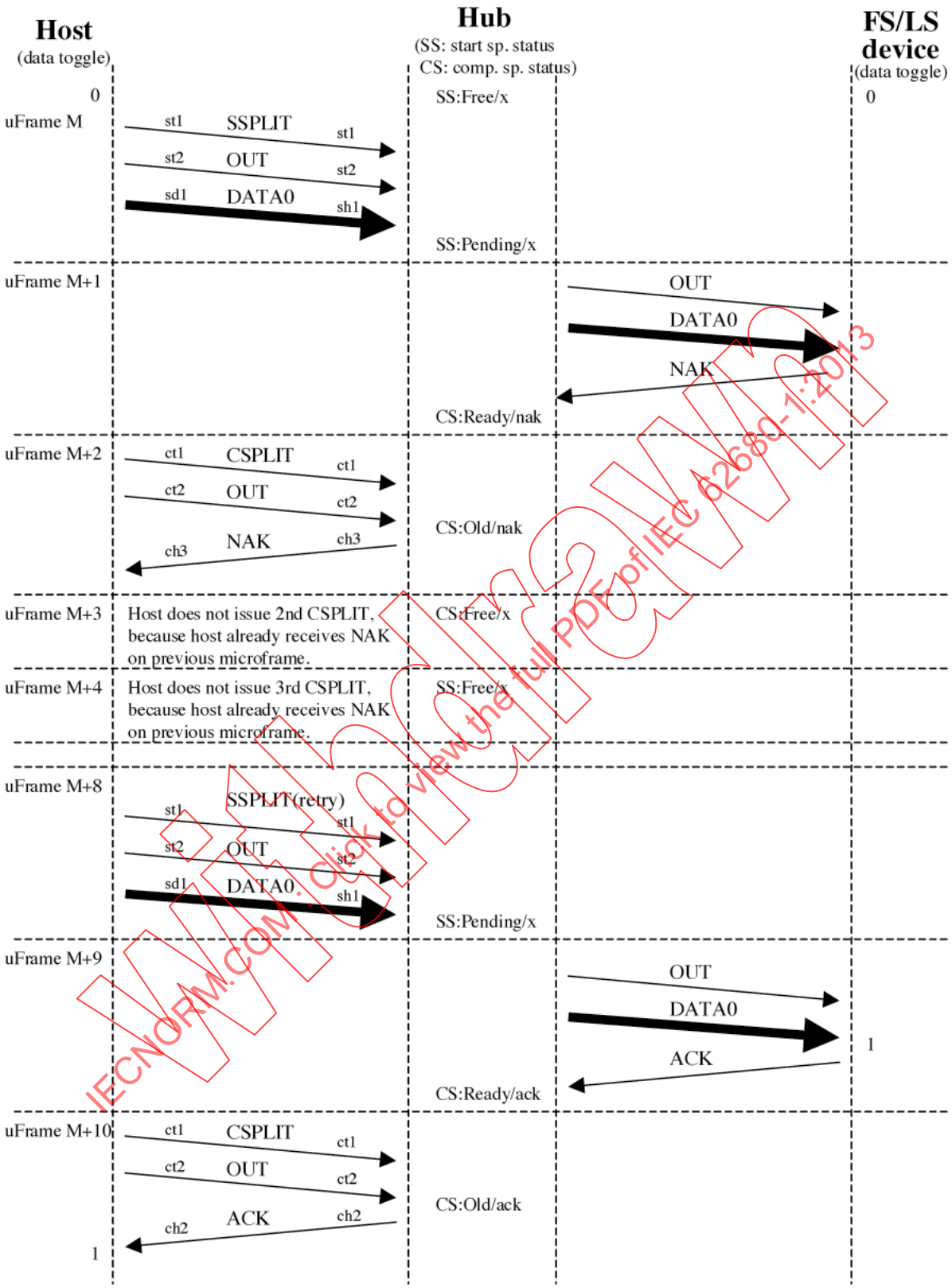
Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)

Anglais	Français
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 8	uTrame M + 8
uFrame M + 9	uTrame M + 9
uFrame M + 10	uTrame M + 10
SSPLIT (retry)	SSPLIT (nouvel essai)
Hub frees this transaction from SS.	Le hub libère cette transaction de SS.

Figure A-61. Abandon et Libération Abandon (la transaction FS/LS n'est pas démarrée à la fin de M+3)

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn



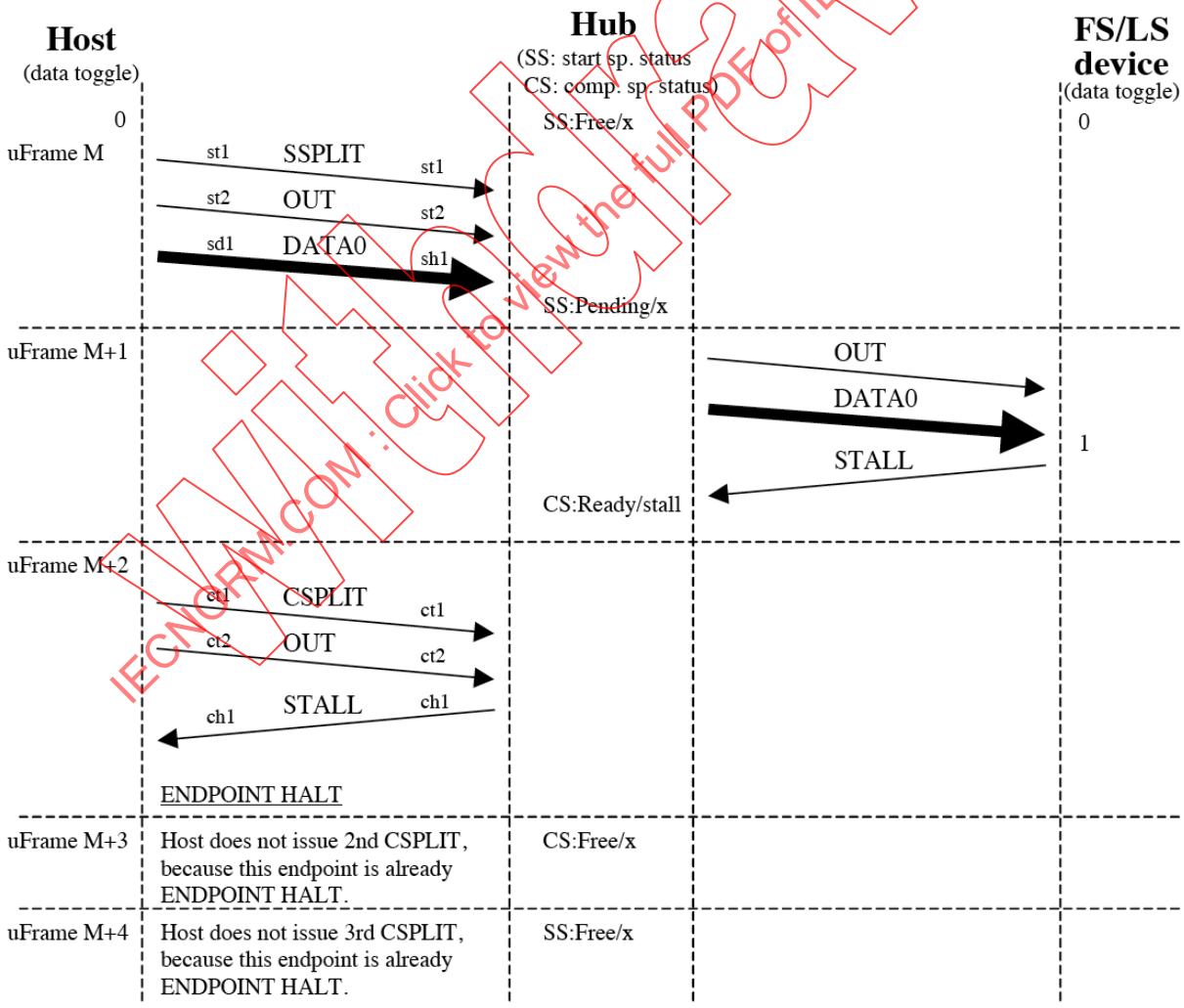
IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)

Anglais	Français
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/nak	CS:Prêt/nak
CS:Old/nak	CS:Ancien/nak
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x
CS:Ready/ack	CS:Prêt/ack
CS:Old/ack	CS:Ancien/ack
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
uFrame M + 8	uTrame M + 8
uFrame M + 9	uTrame M + 9
uFrame M + 10	uTrame M + 10
SSPLIT (retry)	SSPLIT (nouvel essai)
Host does not issue 2nd CSPLIT, because host already receives NAK on previous microframe.	L'hôte n'émet pas un 2ème CSPLIT, car l'hôte reçoit déjà NAK sur microtrame précédente.
Host does not issue 3rd CSPLIT, because host already receives NAK on previous microframe.	L'hôte n'émet pas un 3ème CSPLIT, car l'hôte reçoit déjà NAK sur microtrame précédente.

Figure A-62. Appareil occupé Pas de smash(FS/LS NAK)



IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn

Légende

Anglais	Français
Host (data toggle)	Hôte (basculement des données)

Anglais	Français
Hub (SS: start sp. status CS: comp. sp. status)	Hub (SS: état start sp. CS: état comp. sp.)
FS/LS device (data toggle)	Appareil à vitesse intégrale/basse vitesse (basculement des données)
SS:Free/x	SS:Libre/x
SS:Pending/x	SS:En cours/x
CS:Ready/stall	CS:Prêt/arrêt
CS:Free/x	CS:Libre/x
SS:Free/x	SS:Libre/x
uFrame M	uTrame M
uFrame M + 1	uTrame M + 1
uFrame M + 2	uTrame M + 2
uFrame M + 3	uTrame M + 3
uFrame M + 4	uTrame M + 4
ENDPOINT HALT	ARRÊT POINT D'EXTRÉMITÉ
Host does not issue 2nd CSPLIT, because this endpoint is already ENDPOINT HALT.	L'hôte n'émet pas un 2ème CSPLIT, car ce point d'extrémité est déjà l'ARRÊT POINT D'EXTRÉMITÉ.
Host does not issue 3rd CSPLIT, because this endpoint is already ENDPOINT HALT.	L'hôte n'émet pas un 3ème CSPLIT, car ce point d'extrémité est déjà l'ARRÊT POINT D'EXTRÉMITÉ.

Figure A-63. Appareil arrêté Pas de smash(FS/LS STALL)

A.3 Exemples de transactions IN d'interruption

Légende:

(S): Start Split

(C): Complete Split

Résumé des cas de transaction OUT d'interruption

- Cas normaux

Cas	Figure de référence	Figure similaire
Pas de smash (le paquet d'établissement de liaison FS/LS est sur M+1)	Figure A-64	
Smash HS SSPLIT	Figure A-65	
Smash 3 essais HS SSPLIT	Pas de figure	
Smash HS IN(S)		Figure A-65
Smash 3 essais HS IN(S)	Pas de figure	
Smash HS CSPLIT	Figure A-66	
Smash 3 essais HS CSPLIT	Figure A-67	
Smash HS IN(C)		Figure A-66
3 essais HS IN(C)		Figure A-67
Smash HS DATA0/1	Figure A-68	
Smash 3 essais HS DATA0/1	Figure A-69	
Smash FS/LS IN	Figure A-68	
Smash 3 essais FS/LS IN	Pas de figure	
Smash FS/LS DATA0/1	Figure A-71	
Smash 3 essais FS/LS DATA0/1	Pas de figure	
Smash FS/LS ACK	Figure A-72	
Smash 3 essais FS/LS ACK	Pas de figure	

- Recherche

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013

Cas	Figure de référence	Figure similaire
Pas de smash	Figure A-73	

- CS(transaction complete-split) - Cas précédents

Cas	Figure de référence	Figure similaire
Pas de smash (la transaction HS MDATA et FS/LS est sur M+1 et M+2)	Figure A-74	
Pas de smash (la transaction HS NYET et FS/LS est sur M+2)	Figure A-75	
Pas de smash (la transaction HS NYET et MDATA et FS/LS est sur M+2 et M+3)	Figure A-76	
Pas de smash (la transaction HS NYET et FS/LS est sur M+3)	Figure A-77	
Smash HS NYET	Figure A-78	
Smash 3 essais HS NYET	Figure A-79	

- Cas d'abandon et de libération

Cas	Figure de référence	Figure similaire
Pas de smash et abandon (la transaction HS NYET et FS/LS est poursuivie à la fin de M+3)	Figure A-80	
Pas de smash et libération (la transaction HS NYET et FS/LS n'est pas démarrée à la fin de M+3)	Figure A-81	

- Cas d'erreur de transaction FS/LS

Cas	Figure de référence	Figure similaire
Smash HS ERR		Figure A-68
Smash 3 essais HS ERR		Figure A-69

- Cas appareil occupé

Cas	Figure de référence	Figure similaire
Pas de smash(HS NAK(C))	Figure A-82	
Smash HS NAK(C)		Figure A-68
Smash 3 essais HS NAK(C)		Figure A-69
Smash FS/LS NAK		Figure A-71
Smash 3 essais FS/LS NAK	Pas de figure	

- Cas appareil arrêté

Cas	Figure de référence	Figure similaire
Pas de smash	Figure A-83	
Smash HS STALL(C)		Figure A-68
Smash 3 essais HS STALL(C)		Figure A-69
Smash FS/LS STALL		Figure A-71
Smash 3 essais FS/LS STALL	Pas de figure	

IECNORM.COM : Click to view the full PDF of IEC 62680-1:2013
Withdrawn