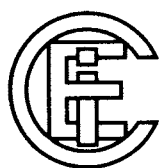


NORME INTERNATIONALE INTERNATIONAL STANDARD

**CEI
IEC
821**

Première édition
First edition
1987



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

BUS CEI 821

**Bus système à microprocesseurs pour données
de 1 à 4 octets**

IEC 821 BUS

Microprocessor system bus for 1 to 4 byte data

Publication
821: 1987

Révision de la présente publication

Le contenu technique des publications de la CEI est constamment revu par la Commission afin d'assurer qu'il reflète bien l'état actuel de la technique.

Les renseignements relatifs à ce travail de révision, à l'établissement des éditions révisées et aux mises à jour peuvent être obtenus auprès des Comités nationaux de la CEI et en consultant les documents ci-dessous:

- **Bulletin de la CEI**
- **Annuaire de la CEI**
- **Catalogue des publications de la CEI**
Publié annuellement

Terminologie

En ce qui concerne la terminologie générale, le lecteur se reportera à la Publication 50 de la CEI: Vocabulaire Electrotechnique International (VEI), qui est établie sous forme de chapitres séparés traitant chacun d'un sujet défini, l'Index général étant publié séparément. Des détails complets sur le VEI peuvent être obtenus sur demande.

Les termes et définitions figurant dans la présente publication ont été soit repris du VEI, soit spécifiquement approuvés aux fins de cette publication.

Symboles graphiques et littéraux

Pour les symboles graphiques, symboles littéraux et signes d'usage général approuvés par la CEI, le lecteur consultera:

- la Publication 27 de la CEI: Symboles littéraux à utiliser en électrotechnique;
- la Publication 617 de la CEI: Symboles graphiques pour schémas.

Les symboles et signes contenus dans la présente publication ont été soit repris des Publications 27 ou 617 de la CEI, soit spécifiquement approuvés aux fins de cette publication.

Publications de la CEI établies par le même Comité d'Etudes

L'attention du lecteur est attirée sur le deuxième feuillet de la couverture, qui énumère les publications de la CEI préparées par le Comité d'Etudes qui a établi la présente publication.

Revision of this publication

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology.

Information on the work of revision, the issue of revised editions and amendment sheets may be obtained from IEC National Committees and from the following IEC sources:

- **IEC Bulletin**
- **IEC Yearbook**
- **Catalogue of IEC Publications**
Published yearly

Terminology

For general terminology, readers are referred to IEC Publication 50: International Electrotechnical Vocabulary (IEV), which is issued in the form of separate chapters each dealing with a specific field, the General Index being published as a separate booklet. Full details of the IEV will be supplied on request.

The terms and definitions contained in the present publication have either been taken from the IEV or have been specifically approved for the purpose of this publication.

Graphical and letter symbols

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to:

- IEC Publication 27: Letter symbols to be used in electrical technology;
- IEC Publication 617: Graphical symbols for diagrams.

The symbols and signs contained in the present publication have either been taken from IEC Publications 27 or 617, or have been specifically approved for the purpose of this publication.

IEC publications prepared by the same Technical Committee

The attention of readers is drawn to the back cover, which lists IEC publications issued by the Technical Committee which has prepared the present publication.

NORME INTERNATIONALE INTERNATIONAL STANDARD

**CEI
IEC
821**

Première édition
First edition
1987



Commission Electrotechnique Internationale

International Electrotechnical Commission

Международная Электротехническая Комиссия

BUS CEI 821

**Bus système à microprocesseurs pour données
de 1 à 4 octets**

IEC 821 BUS

Microprocessor system bus for 1 to 4 byte data

© CEI 1987 Droits de reproduction réservés — Copyright — all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

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 the publisher.

Bureau Central de la Commission Electrotechnique Internationale 3, rue de Varembe Genève, Suisse

SOMMAIRE

	Pages
PREAMBULE	18
PREFACE	18
CHAPITRE 0: DOMAINE D'APPLICATION	
Note au lecteur	20
CHAPITRE 1: INTRODUCTION A LA NORME DU BUS CEI 821	
Sections	
1.1 Objectifs de la norme du BUS CEI 821	22
1.2 Eléments du système d'interface BUS CEI 821.....	22
1.2.1 Définitions générales	22
1.2.1.1 Termes utilisés pour décrire la structure mécanique du BUS CEI 821	22
1.2.1.2 Termes utilisés pour décrire la structure fonctionnelle du BUS CEI 821	24
1.2.1.3 Types de cycles du BUS CEI 821	28
1.2.2 Structure générale du BUS CEI 821	30
1.3 Diagrammes de la norme du BUS CEI 821	38
1.4 Terminologie utilisée dans la norme	38
1.4.1 Etats des lignes de signaux	40
1.4.2 Utilisation de l'astérisque (*)	42
1.5 Spécification du protocole	42
1.5.1 Signaux d'interverrouillage du bus	44
1.5.2 Signal de diffusion du bus	44
1.6 Exemples et explications relatifs au système	46
CHAPITRE 2: BUS DE TRANSFERT DE DONNEES DU BUS CEI 821	
2.1 Introduction	48
2.2 Lignes du bus de transfert de données	48
2.2.1 Lignes d'adresse	52
2.2.2 Lignes de modification d'adresse	58
2.2.3 Lignes de données	62
2.2.4 Lignes de commande du bus de transfert de données	68
2.2.4.1 AS*	68
2.2.4.2 DS0* et DS1*	68
2.2.4.3 DTACK*	68
2.2.4.4 BERR*	70
2.2.4.5 WRITE*	72
2.3 Modules DTB - Description générale	72
2.3.1 MAITRE	72
2.3.2 ESCLAVE	78
2.3.3 LIMITEUR DE TEMPS D'OCCUPATION DU BUS	82
2.3.4 DETECTEUR D'ACCES	86
2.3.5 Modes d'adressage	88
2.3.6 Possibilités de base de transfert de données	92
2.3.7 Possibilités de transferts par bloc	96
2.3.8 Possibilités de lecture-modification-écriture	100
2.3.9 Possibilités de transferts non alignés	106

CONTENTS

	Page
FOREWORD	19
PREFACE	19
CHAPTER 0: SCOPE	
Note to the reader	21
CHAPTER 1: INTRODUCTION TO THE IEC 821 BUS STANDARD	
Section	
1.1 IEC 821 BUS standard objectives	23
1.2 IEC 821 BUS interface system elements	23
1.2.1 Basic definitions	23
1.2.1.1 Terms used to describe the IEC 821 BUS mechanical structure	23
1.2.1.2 Terms used to describe the IEC 821 BUS functional structure	25
1.2.1.3 Types of cycles on the IEC 821 Bus	29
1.2.2 Basic IEC 821 BUS structure	31
1.3 IEC 821 BUS standard diagrams	39
1.4 Standard terminology	39
1.4.1 Signal line states	41
1.4.2 Use of the asterisk (*)	43
1.5 Protocol specification	43
1.5.1 Interlocked bus signals	45
1.5.2 Broadcast bus signal	45
1.6 System examples and explanations	47
CHAPTER 2: IEC 821 BUS DATA TRANSFER BUS	
2.1 Introduction	49
2.2 Data Transfer Bus lines	49
2.2.1 Addressing lines	53
2.2.2 Address modifier lines	59
2.2.3 Data lines	63
2.2.4 Data Transfer Bus control lines	69
2.2.4.1 AS*	69
2.2.4.2 DS0* and DS1*	69
2.2.4.3 DTACK*	69
2.2.4.4 BERR*	71
2.2.4.5 WRITE*	73
2.3 DTB modules - Basic description	73
2.3.1 MASTER	73
2.3.2 SLAVE	79
2.3.3 BUS TIMER	83
2.3.4 LOCATION MONITOR	87
2.3.5 Addressing modes	89
2.3.6 Basic data transfer capabilities	93
2.3.7 Block transfer capabilities	97
2.3.8 Read-modify-write capabilities	101
2.3.9 Unaligned transfer capabilities	107

Sections	Pages
2.3.10 Possibilité UNIQUEMENT D'ADRESSAGE	112
2.3.11 Interaction entre les modules fonctionnels du DTB	112
2.4 Fonctionnement typique	116
2.4.1 Cycles typiques de transfert de données	116
2.4.2 Anticipation d'adresse	126
2.5 Acquisition du bus de transfert de données	128
2.6 Règles et observations de séquençement du DTB	132

CHAPITRE 3: ARBITRAGE DU BUS DE TRANSFERT DE DONNEES DU BUS CEI 821

3.1 Principes de l'arbitrage du bus	192
3.1.1 Types d'arbitrage	192
3.2 Lignes du bus d'arbitrage	196
3.2.1 Lignes de demande et d'allocation du bus	200
3.2.2 Ligne d'occupation du bus (BBSY*)	200
3.2.3 Ligne de libération du bus (BCLR*)	200
3.3 Modules fonctionnels	202
3.3.1 ARBITRE	202
3.3.2 DEMANDEUR	208
3.3.3 MAITRE du bus de transfert de données	214
3.3.3.1 Libération du DTB	214
3.3.3.2 Acquisition du DTB	216
3.3.3.3 Autres informations	216
3.4 Fonctionnement typique	216
3.4.1 Arbitrage de deux niveaux différents de demande du bus	216
3.4.2 Arbitrage de deux demandes du bus sur la même ligne	226
3.5 Course critique entre les demandes du MAITRE et les accords de l'ARBITRE	234

CHAPITRE 4: BUS D'INTERRUPTION PRIORITAIRE DU BUS CEI 821

4.1 Introduction	236
4.1.1 Systèmes à contrôleur unique	236
4.1.2 Systèmes distribués	236
4.2 Lignes du bus d'interruption prioritaire	244
4.2.1 Lignes de demande d'interruption	244
4.2.2 Ligne de reconnaissance d'interruption	244
4.2.3 Chaîne série de reconnaissance d'interruption - IACKIN*/IACKOUT*	244
4.3 Modules du bus d'interruption prioritaire - Description générale	246
4.3.1 CONTROLEURS D'INTERRUPTION	248
4.3.2 GENERATEUR D'INTERRUPTION	252
4.3.3 ÉMETTEUR DE CHAÎNE SÉRIE IACK	260
4.3.4 Possibilités de prise en compte d'une demande d'interruption	262
4.3.5 Possibilités de génération d'une demande d'interruption	264
4.3.6 Possibilités de transferts du MOT D'ÉTAT/IDentificateur	264
4.3.7 Possibilités de libération de la demande d'interruption	266
4.3.8 Interaction entre les modules du bus d'interruption prioritaire	270
4.4 Fonctionnement typique	278
4.4.1 Fonctionnement des interruptions à contrôleur unique	278
4.4.2 Fonctionnement des interruptions distribuées	280
4.4.2.1 Systèmes à interruptions distribuées avec sept CONTROLEURS D'INTERRUPTION ...	280
4.4.2.2 Systèmes à interruptions distribuées avec deux à six CONTROLEURS D'INTERRUPTION	280

Section	Page
2.3.10 ADDRESS-ONLY capability	113
2.3.11 Interaction between DTB functional modules	113
2.4 Typical operation	117
2.4.1 Typical data transfer cycles	117
2.4.2 Address pipelining	127
2.5 Data Transfer Bus acquisition	129
2.6 DTB timing rules and observations	133

CHAPTER 3: IEC 821 BUS DATA TRANSFER BUS ARBITRATION

3.1 Bus arbitration philosophy	193
3.1.1 Types of arbitration	193
3.2 Arbitration bus lines	197
3.2.1 Bus request and bus grant lines	201
3.2.2 Bus busy line (BBSY*)	201
3.2.3 Bus clear line (BCLR*)	201
3.3 Functional modules	203
3.3.1 ARBITER	203
3.3.2 REQUESTER	209
3.3.3 Data Transfer Bus MASTER	215
3.3.3.1 Release of the DTB	215
3.3.3.2 Acquisition of the DTB	217
3.3.3.3 Other information	217
3.4 Typical operation	217
3.4.1 Arbitration of two different levels of bus request	217
3.4.2 Arbitration of two bus requests on the same bus request line	227
3.5 Race conditions between MASTER requests and ARBITER grants	235

CHAPTER 4: IEC 821 BUS PRIORITY INTERRUPT BUS

4.1 Introduction	237
4.1.1 Single handler systems	237
4.1.2 Distributed systems	237
4.2 Priority Interrupt Bus lines	245
4.2.1 Interrupt request lines	245
4.2.2 Interrupt acknowledge line	245
4.2.3 Interrupt acknowledge daisy-chain - IACKIN*/IACKOUT*	245
4.3 Priority Interrupt Bus modules - Basic description	247
4.3.1 INTERRUPT HANDLERS	249
4.3.2 INTERRUPTER	253
4.3.3 IACK DAISY-CHAIN DRIVER	261
4.3.4 Interrupt request handling capabilities	263
4.3.5 Interrupt request generation capabilities	265
4.3.6 STATUS/ID transfer capabilities	265
4.3.7 Interrupt request release capabilities	267
4.3.8 Interaction between Priority Interrupt Bus modules	271
4.4 Typical operation	279
4.4.1 Single handler interrupt operation	279
4.4.2 Distributed interrupt operation	281
4.4.2.1 Distributed interrupt systems with seven INTERRUPT HANDLERS	281
4.4.2.2 Distributed interrupt systems with two to six INTERRUPT HANDLERS	281

Sections	Pages
4.4.3 Exemple: fonctionnement typique d'un système d'interruption à contrôleur unique	284
4.4.4 Exemple: priorité de deux interruptions dans un système à interruptions distribuées	292
4.5 REGLES et OBSERVATIONS concernant le séquençement du bus d'interruption prioritaire	296

CHAPITRE 5: BUS UTILITAIRE DU BUS CEI 821

5.1 Introduction	336
5.2 Signaux du bus utilitaire	336
5.3 Modules du bus utilitaire	336
5.3.1 L'EMETTEUR DE L'HORLOGE DU SYSTEME	336
5.3.2 L'EMETTEUR DE L'HORLOGE DU BUS SERIE	336
5.3.3 Le CONTROLEUR D'ALIMENTATION	336
5.4 Initialisation et diagnostic du système	346
5.5 Broches d'alimentation	352
5.6 Ligne RESERVEE	352

CHAPITRE 6: SPECIFICATIONS ELECTRIQUES DU BUS CEI 821

6.1 Introduction	356
6.2 Distribution du courant d'alimentation	356
6.2.1 Spécifications des tensions courant continu	358
6.2.2 Caractéristiques électriques des broches et supports des connecteurs	360
6.3 Caractéristiques électriques des signaux	360
6.4 Spécifications de commande et de réception du bus	362
6.4.1 Définitions des circuits de commande du bus	362
6.4.2 REGLES pour commander et charger toutes les lignes de signaux du BUS CEI 821	364
6.4.2.1 REGLES pour commander et charger les lignes de signaux trois états à courant élevé (AS*, DS0*, DS1*)	364
6.4.2.2 REGLES pour commander et charger les lignes de signaux trois états standards (A01-A31, D00-D31, AM0-AM5, IACK*, LWORD*, WRITE*)	366
6.4.2.3 REGLES pour commander et charger les lignes "totem-pole" à courant élevé (SERCLK, SYSCLK, BCLR*)	368
6.4.2.4 REGLES pour commander et charger les lignes "totem-pole" standards (BG0OUT*-BG3OUT*/BG0IN*-BG3IN*, IACKOUT*/IACKIN*)	370
6.4.2.5 REGLES pour commander et charger les lignes à collecteur ouvert (BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL*, IACK*)	372
6.5 Interconnexions des lignes de signaux du fond de panier	372
6.5.1 Réseaux d'adaptation d'impédance	374
6.5.2 Impédance caractéristique	376
6.5.3 Informations complémentaires	382
6.6 Signaux définis par l'utilisateur	384
6.7 Emetteurs des lignes de signaux et adaptations	384

CHAPITRE 7: SPECIFICATIONS MECANQUES DU BUS CEI 821

7.1 Introduction	388
7.2 Cartes du BUS CEI 821	390
7.2.1 Cartes simple hauteur	392
7.2.2 Cartes double hauteur	392
7.2.3 Connecteurs de la carte	394

Section	Page
4.4.3 Example: typical single handler interrupt system operation	285
4.4.4 Example: prioritization of two interrupts in a distributed interrupt system ...	293
4.5 Priority Interrupt Bus timing RULES and OBSERVATIONS	297

CHAPTER 5: IEC 821 BUS UTILITY BUS

5.1 Introduction	337
5.2 Utility Bus signal lines	337
5.3 Utility Bus modules	337
5.3.1 The SYSTEM CLOCK DRIVER	337
5.3.2 The SERIAL CLOCK DRIVER	337
5.3.3 The POWER MONITOR	337
5.4 System initialization and diagnostics	347
5.5 Power pins	353
5.6 RESERVED line	353

CHAPTER 6: IEC 821 BUS ELECTRICAL SPECIFICATIONS

6.1 Introduction	357
6.2 Power distribution	357
6.2.1 D.C. voltage specifications	359
6.2.2 Pin and socket connector electrical ratings	361
6.3 Electrical signal characteristics	361
6.4 Bus driving and receiving requirements	363
6.4.1 Bus driver definitions	363
6.4.2 Driving and loading RULES for all IEC 821 BUS lines	365
6.4.2.1 Driving and loading RULES for high current three-state lines (AS*, DS0*, DS1*)	365
6.4.2.2 Driving and loading RULES for standard three-state lines (A01-A31, D00-D31, AM0-AM5, IACK*, LWORD*, WRITE*)	367
6.4.2.3 Driving and loading RULES for high current totem-pole lines (SERCLK, SYSClk, BCLR*)	369
6.4.2.4 Driving and loading RULES for standard totem-pole lines (BG0OUT*-BG3OUT*/BG0IN*-BG3IN*, IACKOUT*/IACKIN*)	371
6.4.2.5 Driving and loading RULES for open-collector lines (BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL*, IACK*)	373
6.5 Backplane signal line interconnections	373
6.5.1 Termination networks	375
6.5.2 Characteristic impedance	377
6.5.3 Additional information	383
6.6 User defined signals	385
6.7 Signal line drivers and terminations	385

CHAPTER 7: IEC 821 BUS MECHANICAL SPECIFICATIONS

7.1 Introduction	389
7.2 IEC 821 BUS boards	391
7.2.1 Single height boards	393
7.2.2 Double height boards	393
7.2.3 Board connectors	395

Sections	Pages
7.2.4 Cartes équipées	396
7.2.5 Largeurs des cartes	396
7.2.6 Gauchissement des cartes du BUS CEI 821, longueur des pattes et hauteur des composants	396
7.3 Panneaux avant	398
7.3.1 Poignées	400
7.3.2 Montage du panneau avant	402
7.3.3 Dimensions du panneau avant	402
7.3.4 Panneaux de remplissage	404
7.3.5 Ejecteurs/injecteurs de cartes	404
7.4 Fonds de panier	406
7.4.1 Dimensions exigées du fond de panier	408
7.4.2 Réseaux d'adaptation d'impédance des lignes de signaux	408
7.5 Assemblage des châssis du BUS CEI 821.....	410
7.5.1 Châssis et largeurs des emplacements	410
7.5.2 Dimensions du châssis	410
7.6 Connecteurs du fond de panier du BUS CEI 821 et connecteurs de cartes du BUS CEI 821	448
7.6.1 Affectation des broches du connecteur J_1/P_1	448
7.6.2 Affectation des broches du connecteur J_2/P_2	450
ANNEXE A - Glossaire de termes du BUS CEI 821	452
ANNEXE B - Description des broches/connecteurs du BUS CEI 821	462
ANNEXE C - Utilisation des lignes SERCLK et SERDAT*	468
 Figures	
1-1: Eléments du système définis par cette norme	34
1-2: Modules fonctionnels et bus définis par cette norme	36
1-3: Notations utilisées dans les chronogrammes	46
2-1: Schéma-bloc fonctionnel du bus de transfert de données	50
2-2: Schéma-bloc: MAITRE	74
2-3: Schéma-bloc: ESCLAVE	78
2-4: Schéma-bloc: LIMITEUR DE TEMPS D'OCCUPATION DU BUS	82
2-5: Schéma-bloc: DETECTEUR D'ACCES	86
2-6: Quatre moyens de mémoriser des données de 32 bits en mémoire	106
2-7: Quatre moyens de mémoriser des données de 16 bits en mémoire	108
2-8: Un exemple de cycle de lecture d'un octet unique	120
2-9: Un exemple de cycle d'écriture d'un double octet	122
2-10: Un exemple de cycle d'écriture d'un quadruple octet	124
2-11: Séquence d'échange du bus de transfert de données du MAITRE	130
2-12: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de diffusion d'adresse TOUS LES CYCLES	168
2-13: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de diffusion d'adresse Transferts d'octet unique pair; transferts d'octet unique impair; transferts double octet; transferts quadruple octet; transferts non alignés	170
2-14: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de diffusion d'adresse Transferts de blocs par octet; transferts de blocs par double octet; transferts de blocs par quadruple octet	172
2-15: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de diffusion d'adresse Cycles RMW pour octet unique; cycles RMW pour double octet; cycles RMW pour quadruple octet	174

Section	Page
7.2.4 Board assemblies	397
7.2.5 Board widths	397
7.2.6 IEC 821 BUS board warpage, lead length and component height	397
7.3 Front panels	399
7.3.1 Handles	401
7.3.2 Front panel mounting	403
7.3.3 Front panel dimensions	403
7.3.4 Filler panels	405
7.3.5 Board ejectors/injectors	405
7.4 Backplanes	407
7.4.1 Backplane dimensional requirements	409
7.4.2 Signal line termination networks	409
7.5 Assembly of IEC 821 BUS subracks	411
7.5.1 Subracks and slot widths	411
7.5.2 Subrack dimensions	411
7.6 IEC 821 BUS backplane connectors and IEC 821 BUS board connectors	449
7.6.1 Pin assignments for the J ₁ /P ₁ connector	449
7.6.2 Pin assignments for the J ₂ /P ₂ connector	451
APPENDIX A - Glossary of IEC 821 BUS terms	453
APPENDIX B - IEC 821 BUS connector/pin description	463
APPENDIX C - Use of the SERCLK and SERDAT* lines	469
 Figures	
1-1: System elements defined by this standard	35
1-2: Functional modules and buses defined by this standard	37
1-3: Signal timing notation	47
2-1: Data Transfer Bus functional block diagram	51
2-2: Block diagram: MASTER	75
2-3: Block diagram: SLAVE	79
2-4: Block diagram: BUS TIMER	83
2-5: Block diagram: LOCATION MONITOR	87
2-6: Four ways that 32 bits of data might be stored in memory	107
2-7: Four ways that 16 bits of data might be stored in memory	109
2-8: An example of a single byte read cycle	121
2-9: An example of a double byte write cycle	123
2-10: An example of a quad byte write cycle	125
2-11: Data Transfer Bus MASTER exchange sequence	131
2-12: MASTER, SLAVE and LOCATION MONITOR - Address broadcast timing ALL CYCLES	169
2-13: MASTER, SLAVE and LOCATION MONITOR - Address broadcast timing Single even byte transfers; single odd byte transfers; double byte transfers; quad byte transfers; unaligned transfers	171
2-14: MASTER, SLAVE and LOCATION MONITOR - Address broadcast timing Single byte block transfers; double byte block transfers; quad byte block transfers	173
2-15: MASTER, SLAVE and LOCATION MONITOR - Address broadcast timing Single byte RMW cycles; double byte RMW cycles; quad byte RMW cycles	175

Figures	Pages
2-16: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données LECTURE OCTET(0); LECTURE OCTET(1); LECTURE OCTET(2); LECTURE OCTET(3); LECTURE OCTET(0-2); LECTURE OCTET(1-3); LECTURE DE BLOCS PAR OCTET	176
2-17: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données LECTURE OCTET(0-1); LECTURE OCTET(2-3); LECTURE OCTET(0-3); LECTURE OCTET(1-2); LECTURE DE BLOCS PAR DOUBLE OCTET; LECTURE DE BLOCS PAR QUADRUPLE OCTET	178
2-18: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données ECRITURE OCTET(0); ECRITURE OCTET(1); ECRITURE OCTET(2); ECRITURE OCTET(3); ECRITURE OCTET(0-2); ECRITURE OCTET(1-3); ECRITURE DE BLOCS PAR OCTET	180
2-19: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données ECRITURE OCTET(0-1); ECRITURE OCTET(2-3); ECRITURE OCTET(0-3); ECRITURE OCTET(1-2); ECRITURE DE BLOCS PAR DOUBLE OCTET; ECRITURE DE BLOCS PAR QUADRUPLE OCTET	182
2-20: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données Cycle RMW pour octet unique	184
2-21: MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données Cycles RMW pour double octet; Cycles RMW pour quadruple octet	184
2-22: Chronogramme du signal de validation d'adresse entre les cycles	186
2-23: Chronogramme des signaux de validation de donnée entre les cycles Un cycle où les deux signaux de validation de donnée passent au niveau bas suivi par un cycle où un ou les deux signaux de validation de donnée passent au niveau bas	186
2-24: Chronogramme des signaux de validation de donnée entre les cycles Un cycle où un signal de validation de donnée passe au niveau bas suivi par un cycle où un ou les deux signaux de validation de donnée passent au niveau bas	188
2-25: MAITRE, ESCLAVE et LIMITEUR DE TEMPS D'OCCUPATION DU BUS - Chronogramme de transfert de données Cycle de dépassement de temps d'occupation du bus	188
2-26: MAITRE - Chronogramme du transfert du contrôle du DTB.....	190
3-1: Schéma-bloc fonctionnel de l'arbitrage du bus	194
3-2: Illustration des lignes de la chaîne série d'allocation du bus	198
3-3: Schéma-bloc: ARBITRE	206
3-4: Schéma-bloc: DEMANDEUR	212
3-5: Organigramme de l'arbitrage: deux DEMANDEURS, deux niveaux de demande	220
3-6: Séquence d'arbitrage: deux DEMANDEURS, deux niveaux de demande	224
3-7: Organigramme de l'arbitrage: deux DEMANDEURS, même niveau de demande	228
3-8: Séquence d'arbitrage: deux DEMANDEURS, même niveau de demande	232
4-1: Schéma-bloc fonctionnel du système d'interruption prioritaire du BUS CEI 821.	238
4-2: Structure du sous-système d'interruption: système à contrôleur unique	240
4-3: Structure du sous-système d'interruption: système distribué	242
4-4: CHAÎNE SERIE IACKIN*/IACKOUT*	246
4-5: Schéma-bloc: CONTROLEUR D'INTERRUPTION	250
4-6: Schéma-bloc: GENERATEUR D'INTERRUPTION	258
4-7: Schéma-bloc: EMETTEUR DE CHAÎNE SERIE IACK	262

Figures	Page
2-16: MASTER, SLAVE and LOCATION MONITOR - Data transfer timing BYTE(0) READ; BYTE(1) READ; BYTE(2) READ; BYTE(3) READ; BYTE(0-2) READ; BYTE(1-3) READ; SINGLE BYTE BLOCK READ	177
2-17: MASTER, SLAVE and LOCATION MONITOR - Data transfer timing BYTE(0-1) READ; BYTE(2-3) READ; BYTE(0-3) READ; BYTE(1-2) READ; DOUBLE BYTE BLOCK READ; QUAD BYTE BLOCK READ	179
2-18: MASTER, SLAVE and LOCATION MONITOR - Data transfer timing BYTE(0) WRITE; BYTE(1) WRITE; BYTE(2) WRITE; BYTE(3) WRITE; BYTE(0-2) WRITE; BYTE(1-3) WRITE; SINGLE BYTE BLOCK WRITE	181
2-19: MASTER, SLAVE and LOCATION MONITOR - Data transfer timing BYTE(0-1) WRITE; BYTE(2-3) WRITE; BYTE(0-3) WRITE; BYTE(1-2) WRITE; DOUBLE BYTE BLOCK WRITE; QUAD BYTE BLOCK WRITE	183
2-20: MASTER, SLAVE and LOCATION MONITOR - Data transfer timing Single byte RMW cycle	185
2-21: MASTER, SLAVE and LOCATION MONITOR - Data transfer timing Double byte RMW cycles quad byte RMW cycles	185
2-22: Address strobe inter-cycle timing	187
2-23: Data strobe inter-cycle timing A cycle where both data strobes go low followed by a cycle where one or both data strobes go low	187
2-24: Data strobe inter-cycle timing A cycle where one data strobe goes low followed by a cycle where one or both data strobes go low	189
2-25: MASTER, SLAVE and BUS TIMER - Data transfer timing Timed-out cycle	189
2-26: MASTER - DTB control transfer timing	191
3-1: Arbitration bus functional block diagram	195
3-2: Illustration of the daisy-chained bus grant lines	199
3-3: Block diagram: ARBITER	207
3-4: Block diagram: REQUESTER	213
3-5: Arbitration flow diagram: two REQUESTERS, two request levels	221
3-6: Arbitration sequence diagram: two REQUESTERS, two request levels	225
3-7: Arbitration flow diagram: two REQUESTERS, same request level	229
3-8: Arbitration sequence diagram: two REQUESTERS, same request level	233
4-1: Priority Interrupt Bus functional block diagram	239
4-2: Interrupt subsystem structure: single handler system	241
4-3: Interrupt subsystem structure: distributed system	243
4-4: IACKIN*/IACKOUT* DAISY-CHAIN	247
4-5: Block diagram: INTERRUPT HANDLER	251
4-6: Block diagram: INTERRUPTER	259
4-7: Block diagram: IACK DAISY-CHAIN DRIVER	263

Figures	Pages
4-8: Libération des lignes de demande d'interruption par les GENERATEURS D'INTERRUPTION ROAK et RORA	270
4-9: Un EMETTEUR DE CHAÎNE SERIE IACK et un GENERATEUR D'INTERRUPTION sur la même carte	274
4-10: Deux GENERATEURS D'INTERRUPTION sur la même carte	276
4-11: Les trois phases d'une séquence d'interruption	278
4-12: Deux CONTROLEURS D'INTERRUPTION surveillant chacun une ligne de demande d'interruption	282
4-13: Deux CONTROLEURS D'INTERRUPTION surveillant chacun plusieurs lignes de demande d'interruption	284
4-14: Organigramme du fonctionnement typique d'un système d'interruption à contrôleur unique d'interruption	288
4-15: Organigramme du fonctionnement typique d'un système à interruptions distribuées avec deux CONTROLEURS D'INTERRUPTION	294
4-16: CONTROLEUR et GENERATEUR D'INTERRUPTION - Chronologie de la sélection du GENERATEUR D'INTERRUPTION CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE, DOUBLE ou QUADRUPLE	326
4-17: EMETTEUR DE CHAÎNE SERIE IACK - Chronologie de la sélection du GENERATEUR D'INTERRUPTION CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE, DOUBLE ou QUADRUPLE	326
4-18: GENERATEUR D'INTERRUPTION qui participe - Chronologie de la sélection du GENERATEUR D'INTERRUPTION CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE, DOUBLE ou QUADRUPLE	328
4-19: GENERATEUR D'INTERRUPTION qui répond - Chronologie de la sélection du GENERATEUR D'INTERRUPTION CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE, DOUBLE ou QUADRUPLE	330
4-20: CONTROLEUR D'INTERRUPTION - Chronologie du transfert du MOT D'ETAT/Identificateur CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE	330
4-21: CONTROLEUR D'INTERRUPTION - Chronologie du transfert du MOT D'ETAT/Identificateur CYCLE DE RECONNAISSANCE D'INTERRUPTION à DOUBLE OCTET; CYCLE DE RECONNAISSANCE D'INTERRUPTION à QUADRUPLE OCTET	332
4-22: GENERATEUR D'INTERRUPTION qui répond - Chronologie du transfert du MOT D'ETAT/Identificateur CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE	332
4-23: GENERATEUR D'INTERRUPTION qui répond - Chronologie du transfert du MOT D'ETAT/Identificateur CYCLE DE RECONNAISSANCE D'INTERRUPTION à DOUBLE OCTET; CYCLE DE RECONNAISSANCE D'INTERRUPTION à QUADRUPLE OCTET	334
4-24: EMETTEUR DE CHAÎNE SERIE IACK, GENERATEUR D'INTERRUPTION qui répond et GENERATEUR D'INTERRUPTION qui participe - Chronologie de la chaîne série IACK entre les cycles	334
5-1: Schéma-bloc du bus utilitaire	340
5-2: Chronogramme de l'EMETTEUR DE L'HORLOGE SYSTEME	342
5-3: Schéma-bloc du module CONTROLEUR D'ALIMENTATION	342
5-4: Chronogramme du CONTROLEUR D'ALIMENTATION lors d'une défaillance d'alimentation	344
5-5: Chronogramme du CONTROLEUR D'ALIMENTATION lors de la mise sous tension	344
5-6: Chronogramme des signaux SYSRESET* et SYSFAIL*	350
5-7: Courant admissible dans les broches d'alimentation	354
6-1: Niveaux des signaux du BUS CEI 821	360
6-2: Adaptation d'impédance standard du bus	376
6-3: Coupe d'une piste de microruban utilisée comme ligne de signal sur le fond de panier	378
6-4: Z_0 en fonction de la largeur de ligne	380
6-5: C_0 en fonction de la largeur de ligne	380

Figures	Page
4-8: Release of the interrupt request lines by ROAK and RORA INTERRUPTERS	271
4-9: An IACK DAISY-CHAIN DRIVER and an INTERRUPTER on the same board	275
4-10: Two INTERRUPTERS on the same board	277
4-11: The three phases of an interrupt sequence	279
4-12: Two INTERRUPT HANDLERS, each monitoring one interrupt request line	283
4-13: Two INTERRUPT HANDLERS, each monitoring several interrupt request lines	285
4-14: Typical single handler interrupt system operation flow diagram	289
4-15: Typical distributed interrupt system with two INTERRUPT HANDLERS, flow diagram	295
4-16: INTERRUPT HANDLER and INTERRUPTER - INTERRUPTER selection timing SINGLE, DOUBLE and QUAD BYTE INTERRUPT ACKNOWLEDGE CYCLE	327
4-17: IACK DAISY-CHAIN DRIVER - INTERRUPTER selection timing SINGLE, DOUBLE and QUAD BYTE INTERRUPT ACKNOWLEDGE CYCLE	327
4-18: Participating INTERRUPTER - INTERRUPTER selection timing SINGLE, DOUBLE and QUAD BYTE INTERRUPT ACKNOWLEDGE CYCLE	329
4-19: Responding INTERRUPTER - INTERRUPTER selection timing SINGLE, DOUBLE and QUAD BYTE INTERRUPT ACKNOWLEDGE CYCLE	331
4-20: INTERRUPT HANDLER - STATUS/ID transfer timing SINGLE BYTE INTERRUPT ACKNOWLEDGE CYCLE	331
4-21: INTERRUPT HANDLER - STATUS/ID transfer timing DOUBLE BYTE INTERRUPT ACKNOWLEDGE CYCLE; QUAD BYTE INTERRUPT ACKNOWLEDGE CYCLE	333
4-22: Responding INTERRUPTER - STATUS/ID transfer timing SINGLE BYTE INTERRUPT ACKNOWLEDGE CYCLE	333
4-23: Responding INTERRUPTER - STATUS/ID transfer timing DOUBLE BYTE INTERRUPT ACKNOWLEDGE CYCLE; QUAD BYTE INTERRUPT ACKNOWLEDGE CYCLE	335
4-24: IACK DAISY-CHAIN DRIVER, responding INTERRUPTER, and participating INTERRUPTER - IACK daisy-chain inter-cycle timing	335
5-1: Utility Bus block diagram	341
5-2: SYSTEM CLOCK DRIVER timing diagram	343
5-3: Block diagram of POWER MONITOR module	343
5-4: POWER MONITOR power failure timing	345
5-5: POWER MONITOR system restart timing	345
5-6: SYSRESET* and SYSFAIL* timing diagram	351
5-7: Current rating for power pins	355
6-1: IEC 821 BUS signal levels	361
6-2: Standard bus termination	377
6-3: Backplane microstrip signal line cross section	379
6-4: Z_0 versus line width	381
6-5: C_0 versus line width	381

Figures	Pages
7-1: Châssis équipé avec des cartes de tailles différentes	412
7-2: Carte simple hauteur: dimensions de base	414
7-3: Carte double hauteur: dimensions de base	416
7-4: Position des connecteurs sur les cartes simple et double hauteur	418
7-5: Vue en coupe d'une carte, d'un connecteur, d'un fond de panier et d'un panneau avant	420
7-6: Hauteur des composants, longueur des pattes et gauchissement de la carte	422
7-7: Panneau avant, simple hauteur, simple largeur	424
7-8: Panneau avant, double hauteur, simple largeur	426
7-9: Supports de montage du panneau avant et dimensions d'une carte simple hauteur	428
7-10: Supports de montage du panneau avant et dimensions d'une carte double hauteur	430
7-11: Panneau de remplissage simple hauteur	432
7-12: Panneau de remplissage double hauteur	434
7-13: Dimensions hors tout du fond de panier	436
7-14: Dimensions détaillées du fond de panier	438
7-15: Adaptation du fond de panier "Type hors carte" (vue d'en haut du fond de panier)	440
7-16: Adaptation du fond de panier "Type sur carte" (vue d'en haut du fond de panier)	442
7-17: Châssis à 21 emplacements	444
7-18: Détail du guide de carte	446
C-1: Chronogramme de SERCLK	470

Tableaux

2-1: Les quatre catégories d'emplacement d'octet	52
2-2: Niveaux des signaux utilisés pour sélectionner le ou les octets auxquels on accède pendant un transfert de données	56
2-3: Codes modificateurs d'adresse	60
2-4: Utilisation des lignes de données pour transférer des données pendant chacun des 34 types de cycle	64
2-5: REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de MAITRES	76
2-6: REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types d'ESCLAVES	80
2-7: Utilisation du mnémonique BT0() pour spécifier la durée de limitation de temps des LIMITEURS DE TEMPS D'OCCUPATION DU BUS	84
2-8: REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de DETECTEURS D'ACCES	88
2-9: Mnémoniques qui spécifient les possibilités d'adressage	90
2-10: Mnémoniques qui spécifient les possibilités de base de transferts de données fondamentaux	96
2-11: Mnémoniques qui spécifient les possibilités de transferts par bloc	100
2-12: Mnémoniques qui spécifient les possibilités de lecture-modification-écriture.	104
2-13: Transferts de 32 bits de données utilisant des cycles de transfert de multiple octet	108
2-14: Transferts de 16 bits de données utilisant des cycles de transfert de multiple octet	110
2-15: Mnémonique qui spécifie la possibilité de transfert non aligné	110
2-16: Mnémonique qui spécifie la possibilité UNIQUEMENT D'ADRESSAGE	112
2-17: Chronogrammes qui définissent le fonctionnement du MAITRE, de l'ESCLAVE et du DETECTEUR D'ACCES	136
2-18: Chronogrammes qui définissent le fonctionnement du MAITRE, de l'ESCLAVE, du DETECTEUR D'ACCES, du CONTROLEUR D'INTERRUPTION et du GENERATEUR D'INTERRUPTION pendant TOUS les cycles	138

Figures	Page
7-1: Subrack with mixed board sizes	413
7-2: Single height board: basic dimensions	415
7-3: Double height board: basic dimensions	417
7-4: Connector position on single and double height boards.....	419
7-5: Cross sectional view of board, connector, backplane, and front panel	421
7-6: Component height, lead length and board warpage	423
7-7: Single height, single width, front panel	425
7-8: Double height, single width, front panel	427
7-9: Front panel mounting brackets and dimensions of single height board	429
7-10: Front panel mounting brackets and dimensions of double height board	431
7-11: Single height filler panel	433
7-12: Double height filler panel	435
7-13: Backplane overall dimensions	437
7-14: Backplane detailed dimensions	439
7-15: "Off board type" backplane termination (view from top of backplane)	441
7-16: "On board type" backplane termination (view from top of backplane)	443
7-17: 21-slot subrack	445
7-18: Board guide detail	447
C-1: SERCLK timing diagram	471
 Tables	
2-1: The four categories of byte location	53
2-2: Signal levels used to select which byte location(s) are accessed during a data transfer	57
2-3: Address modifier codes	61
2-4: Use of the data lines to move data during each of the 34 cycle types	65
2-5: RULES and PERMISSIONS that specify the use of the dotted lines by the various types of MASTERS	77
2-6: RULES and PERMISSIONS that specify the use of the dotted lines by the various types of SLAVES	81
2-7: Use of the BTG() mnemonic to specify the time-out period of BUS TIMERS	85
2-8: RULES and PERMISSIONS that specify the use of the dotted lines by the various types of LOCATION MONITORS	89
2-9: Mnemonics that specify addressing capabilities	91
2-10: Mnemonics that specify basic data transfer capabilities	97
2-11: Mnemonics that specify block transfer capabilities	101
2-12: Mnemonics that specify read-modify-write capabilities	105
2-13: Transferring 32 bits of data using multiple byte transfer cycles	109
2-14: Transferring 16 bits of data using multiple byte transfer cycles	111
2-15: Mnemonic that specifies unaligned transfer capability	111
2-16: Mnemonic that specifies ADDRESS-ONLY capability	113
2-17: Timing diagrams that define MASTER, SLAVE and LOCATION MONITOR operation	137
2-18: Timing diagrams that define MASTER, SLAVE, LOCATION MONITOR, INTERRUPT HANDLER and INTERRUPTER operation during ALL cycles	139

Tableaux	Pages
2-19: Définition des mnémoniques utilisés dans les tableaux 2-20, 2-21 et 2-22	138
2-20: Utilisation des lignes d'adressage pour sélectionner un groupe de 4 octets ..	140
2-21: Utilisation des lignes DS1*, DS0*, A01 et LWORD* pour sélectionner le ou les octets dans le groupe de 4 octets	142
2-22: Utilisation des lignes de données du bus pour transférer des données	144
2-23: Paramètres de temps des chronogrammes du MAITRE, de l'ESCLAVE et du DETECTEUR D'ACCES	148
2-24: Paramètres de temps du chronogramme du LIMITEUR DE TEMPS D'OCCUPATION DU BUS	148
2-25: MAITRE - REGLES et OBSERVATIONS sur la chronologie	150
2-26: ESCLAVE - REGLES et OBSERVATIONS sur la chronologie	158
2-27: DETECTEUR D'ACCES - OBSERVATIONS sur la chronologie	164
2-28: LIMITEUR DE TEMPS D'OCCUPATION DU BUS - REGLES de la chronologie	166
3-1: REGLES et OBSERVATIONS qui spécifient l'utilisation des lignes pointillées par les différents types d'ARBITRES définis dans cette norme	208
3-2: REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de DEMANDEURS définis dans cette norme ..	212
4-1: REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de CONTROLEURS D'INTERRUPTION	252
4-2: REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de GENERATEURS D'INTERRUPTION	260
4-3: Utilisation du mnémonique IH() pour spécifier les possibilités de traitement d'interruption	262
4-4: Utilisation du mnémonique IL() pour spécifier les possibilités de génération d'une demande d'interruption	264
4-5: Mnémoniques qui spécifient les possibilités de transfert du MOT D'ETAT/Identificateur	264
4-6: Mnémoniques qui spécifient les possibilités de libération des demandes d'interruption	270
4-7: Code de la reconnaissance d'interruption sur 3 bits	292
4-8: Chronogrammes qui définissent le fonctionnement du CONTROLEUR D'INTERRUPTION et du GENERATEUR D'INTERRUPTION	300
4-9: Chronogrammes qui définissent le fonctionnement de l'EMETTEUR DE CHAINE SERIE IACK	300
4-10: Chronogrammes qui définissent le fonctionnement du GENERATEUR D'INTERRUPTION qui participe	302
4-11: Chronogrammes qui définissent le fonctionnement du GENERATEUR D'INTERRUPTION qui répond	302
4-12: Définitions des mnémoniques utilisés dans les tableaux 4-13, 4-14 et 4-15 ...	304
4-13: Utilisation des lignes d'adresse pendant les cycles de reconnaissance d'interruption	304
4-14: Utilisation des lignes DS1*, DS0*, LWORD* et WRITE* pendant les cycles de reconnaissance d'interruption	306
4-15: Utilisation des lignes du bus de données pour le transfert du MOT D'ETAT/Identificateur	306
4-16: Paramètres de temps des chronogrammes du CONTROLEUR D'INTERRUPTION, du GENERATEUR D'INTERRUPTION et de l'EMETTEUR DE CHAINE SERIE IACK	308
4-17: CONTROLEUR D'INTERRUPTION - REGLES et OBSERVATIONS sur la chronologie	310
4-18: GENERATEUR D'INTERRUPTION - REGLES et OBSERVATIONS sur la chronologie	316
4-19: EMETTEUR DE CHAINE SERIE IACK - REGLES et OBSERVATIONS sur la chronologie ...	322
5-1: Commandes émises par les modules pendant le démarrage et l'arrêt de l'alimentation	348
6-1: Spécifications des tensions du bus	358
6-2: Spécifications de commande et de réception du bus	362
6-3: Résumé des émetteurs de bus	386
7-1: Affectation des broches J ₁ /P ₁	448
7-2: Affectation des broches J ₂ /P ₂	450
C-1: Valeurs des temps de SERCLK	470

Tables	Page
2-19: Definitions of mnemonics used in Tables 2-20, 2-21 and 2-22	139
2-20: Use of addressing lines to select a 4-byte group	141
2-21: Use of the DS1*, DS0*, A01, and LWORD* lines to select the byte(s) within a 4-byte group	143
2-22: Use of the data bus lines to transfer data	145
2-23: MASTER, SLAVE, and LOCATION MONITOR timing parameters	149
2-24: BUS TIMER timing parameters	149
2-25: MASTER-Timing RULES and OBSERVATIONS	151
2-26: SLAVE-Timing RULES and OBSERVATIONS	159
2-27: LOCATION MONITOR Timing OBSERVATIONS	165
2-28: BUS TIMER-Timing RULES	167
3-1: RULES and PERMISSIONS that specify the use of the dotted lines by the various types of ARBITERS defined by this standard	209
3-2: RULES and PERMISSIONS that specify the use of the dotted lines by the various types of REQUESTERS defined by this standard	213
4-1: RULES and PERMISSIONS that specify the use of the dotted lines by the various types of INTERRUPT HANDLERS	253
4-2: RULES and PERMISSIONS that specify the use of the dotted lines by the various types of INTERRUPTERS	261
4-3: Use of the IH() mnemonic to specify interrupt handling capabilities	263
4-4: Use of the I() mnemonic to specify interrupt request generation capabilities ..	265
4-5: Mnemonics that specify STATUS/ID transfer capabilities	265
4-6: Mnemonics that specify interrupt request release capabilities	271
4-7: 3-bit interrupt acknowledge code	293
4-8: Timing diagrams that define INTERRUPT HANDLER and INTERRUPTER operation	301
4-9: Timing diagrams that define IACK DAISY-CHAIN DRIVER operation	301
4-10: Timing diagrams that define participating INTERRUPTER operation	303
4-11: Timing diagrams that define responding INTERRUPTER operation	303
4-12: Definitions of mnemonics used in Tables 4-13, 4-14 and 4-15	305
4-13: Use of addressing lines during interrupt acknowledge cycles	305
4-14: Use of the DS1*, DS0*, LWORD*, and WRITE* lines during interrupt acknowledge cycles	307
4-15: Use of the data bus lines to transfer the STATUS/ID	307
4-16: INTERRUPT HANDLER, INTERRUPTER, and IACK DAISY-CHAIN DRIVER timing parameters .	309
4-17: INTERRUPT HANDLER - Timing RULES and OBSERVATIONS	311
4-18: INTERRUPTER - Timing RULES and OBSERVATIONS	317
4-19: IACK DAISY-CHAIN DRIVER - Timing RULES and OBSERVATIONS	323
5-1: Module drive during power-up and power-down sequences	349
6-1: Bus voltage specifications	359
6-2: Bus driving and receiving requirements	363
6-3: Bus driver summary	387
7-1: J ₁ /P ₁ pin assignments	449
7-2: J ₂ /P ₂ pin assignments	451
C-1: SERCLK timing values	471

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

BUS CEI 821

BUS SYSTEME A MICROPROCESSEURS POUR DONNEES
DE 1 A 4 OCTETS

PREAMBULE

- 1) Les décisions ou accords officiels de la CEI en ce qui concerne les questions techniques, préparés par des Comités d'Etudes où sont représentés tous les Comités nationaux s'intéressant à ces questions, expriment dans la plus grande mesure possible un accord international sur les sujets examinés.
- 2) Ces décisions constituent des recommandations internationales et sont agréées comme telles par les Comités nationaux.
- 3) Dans le but d'encourager l'unification internationale, la CEI exprime le vœu que tous les Comités nationaux adoptent dans leurs règles nationales le texte de la recommandation de la CEI, dans la mesure où les conditions nationales le permettent. Toute divergence entre la recommandation de la CEI et la règle nationale correspondante doit, dans la mesure du possible, être indiquée en termes clairs dans cette dernière.
- 4) La CEI n'a fixé aucune procédure concernant le marquage comme indication d'approbation et sa responsabilité n'est pas engagée quand il est déclaré qu'un matériel est conforme à l'une de ses recommandations.

PREFACE

La présente norme a été établie par le Sous-Comité 47B: Systèmes à microprocesseurs, du Comité d'Etudes n° 47 de la CEI: Dispositifs à semi-conducteurs.

Le texte de cette norme est issu des documents suivants:

Règle des Six Mois	Rapport de vote
47B(BC)7	47B(BC)12

Pour de plus amples renseignements, consulter le rapport de vote mentionné dans le tableau ci-dessus.

Les publications suivantes de la CEI sont citées dans la présente norme:

- Publications n^{os} 297-1 (1982): Dimensions des structures mécaniques de la série de 482,6 mm (19 in), Première partie: Panneaux et bâtis.
- 297-3 (1984): Troisième partie: Bacs et blocs enfichables associés.
- 603-2 (1980): Connecteurs pour fréquences inférieures à 3 MHz pour utilisation avec cartes imprimées, Deuxième partie: Connecteurs pour circuits imprimés en deux parties, pour grille de base de 2,54 mm (0.1 in) avec caractéristiques de montage communes.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

IEC 821 BUS

MICROPROCESSOR SYSTEM BUS FOR 1 TO 4 BYTE DATA

FOREWORD

- 1) The formal decisions or agreements of the IEC on technical matters, prepared by Technical Committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.
- 2) They have the form of recommendations for international use and they are accepted by the National Committees in that sense.
- 3) In order to promote international unification, the IEC expresses the wish that all National Committees should adopt the text of the IEC recommendation for their national rules in so far as national conditions will permit. Any divergence between the IEC recommendation and the corresponding national rules should, as far as possible, be clearly indicated in the latter.
- 4) The IEC has not laid down any procedure concerning marking as an indication of approval and has no responsibility when an item of equipment is declared to comply with one of its recommendations.

PREFACE

This standard has been prepared by Sub-Committee 47B: Microprocessor Systems, of IEC Technical Committee No. 47: Semiconductor Devices.

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

Six Months' Rule	Report on Voting
47B(C0)7	47B(C0)12

Further information can be found in the Report on Voting indicated in the table above.

The following IEC publications are quoted in this standard:

Publication Nos. 297-1 (1982): Dimensions of Mechanical Structures of the 482.6 mm (19 in) Series, Part 1: Panels and Racks.

297-3 (1984): Subracks and Associated Plug-in Units.

603-2 (1980): Connectors for Frequencies below 3 MHz for Use with Printed Boards, Part 2: Two-part Connectors for Printed Boards for Basic Grid of 2.54 mm (0.1 in), with Common Mounting Features.

BUS CEI 821

BUS SYSTEME A MICROPROCESSEURS POUR DONNEES
DE 1 A 4 OCTETS

CHAPITRE 0: DOMAINE D'APPLICATION

La présente norme décrit un bus de fond de panier à haute performance utilisable dans les systèmes à microprocesseurs. Ce bus parallèle permet des cycles de transferts, soit uniques, soit par blocs, sur une voie d'adresses et de données de 32 bits non multiplexées. La transmission est gérée par un protocole de dialogue asynchrone. L'allocation du bus permet une architecture multiprocesseur. Ce bus permet également l'utilisation d'interruptions entre modules, facilitant une réponse rapide à des événements internes ou externes. La mécanique des cartes et des châssis est conçue à partir de la Publication 297 de la CEI: Dimensions des panneaux et bâtis.

Note au lecteur

Le BUS CEI 822, cité dans la section 7.4, représente le bus en cours de normalisation par le Sous-Comité 47B en tant que bus sous-système du BUS CEI 821 qui constitue la présente norme.

Le BUS CEI 823, cité aux paragraphes 1.2.1.2 et 5.3.2 et dans les annexes B et C, représente le bus qui est à l'étude dans le cadre du Sous-Comité 47B en tant que bus série du présent BUS CEI 821.

IEC 821 BUS
MICROPROCESSOR SYSTEM BUS FOR 1 TO 4 BYTE DATA

CHAPTER 0: SCOPE

This standard describes a high performance backplane bus for use in microprocessor based systems. This parallel bus supports single and block transfer cycles on a 32-bit non-multiplexed address and data highway. Transmission is governed by an asynchronous handshaken protocol. The bus allocation provides for multiprocessor architectures. This bus also supports inter-module interrupts for facilitating quick response to internal and external events. The mechanics of the boards and chassis are based on IEC Publication 297: Dimensions of Panels and Racks.

Note to the reader

IEC 822 BUS, named in Section 7.4 represents the bus which is in the process of being standardized by Sub-Committee 47B as a sub-system bus of the IEC 821 BUS which constitutes this standard.

IEC 823 BUS, named in Paragraphs 1.2.1.2 and 5.3.2 and Appendices B and C, represents the bus which is under consideration by Sub-Committee 47B as the serial bus of this IEC 821 BUS.

CHAPITRE 1: INTRODUCTION A LA NORME DU BUS CEI 821

1.1 Objectifs de la norme du BUS CEI 821

La présente norme du BUS CEI 821 définit un système d'interface pour interconnecter des unités de traitement de données, de stockage de données et de contrôle des périphériques dans une configuration à couplage étroit. Le système a été conçu avec les objectifs suivants:

- a) Permettre à deux unités de communiquer entre elles sans gêner le fonctionnement interne d'autres unités connectées au BUS CEI 821.
- b) Spécifier les caractéristiques électriques et mécaniques du système nécessaires à la conception d'unités capables de communiquer de manière fiable et sans ambiguïté avec d'autres unités connectées au BUS CEI 821.
- c) Spécifier les protocoles qui définissent avec précision l'interaction entre le BUS CEI 821 et les unités qui lui sont connectées.
- d) Fournir une terminologie et des définitions qui décrivent le protocole du système.
- e) Permettre une grande latitude de conception et permettre ainsi d'optimiser le coût et/ou les performances sans affecter la compatibilité du système.
- f) Créer un système dans lequel l'interfaçage n'introduit pas de limites aux performances des unités.

1.2 Eléments du système d'interface BUS CEI 821

1.2.1 Définitions générales

La structure du BUS CEI 821 peut être décrite sous deux aspects: sa structure mécanique et sa structure fonctionnelle. La spécification mécanique décrit les dimensions physiques des châssis, des fonds de panier, des panneaux avant, des cartes enfichables, etc. La spécification fonctionnelle du BUS CEI 821 décrit le fonctionnement du bus, quels sont les modules fonctionnels engagés dans chaque échange et les règles qui gouvernent leur comportement. Ce paragraphe fournit les définitions informelles de quelques termes généraux utilisés pour décrire à la fois la structure physique et mécanique du BUS CEI 821.

1.2.1.1 Termes utilisés pour décrire la structure mécanique du BUS CEI 821

FOND DE PANIER DU BUS CEI 821

Carte de circuit imprimé comprenant des connecteurs 96 broches et des pistes reliant leurs broches. Certains systèmes de BUS CEI 821 ont un circuit imprimé unique appelé fond de panier J_1 . Il fournit les pistes pour les signaux nécessaires aux opérations de base. D'autres systèmes BUS CEI 821 ont aussi un deuxième fond de panier J_2 optionnel. Ce dernier fournit les connecteurs 96 broches supplémentaires et les pistes de signaux nécessaires pour les transferts étendus de données et d'adresses. D'autres encore ont un circuit imprimé unique qui fournit les pistes et les connecteurs des deux fonds de panier J_1 et J_2 .

CHAPTER 1: INTRODUCTION TO THE IEC 821 BUS STANDARD

1.1 IEC 821 BUS standard objectives

This IEC 821 BUS standard defines an interfacing system used to interconnect data processing, data storage and peripheral control devices in a closely coupled hardware configuration. The system has been conceived with the following objectives:

- a) To allow communication between devices on the IEC 821 BUS without disturbing the internal activities of other devices interfaced to the IEC 821 BUS.
- b) To specify the electrical and mechanical system characteristics required to design devices that will reliably and unambiguously communicate with other devices interfaced to the IEC 821 BUS.
- c) To specify protocols that precisely define the interaction between the IEC 821 BUS and devices interfaced to it.
- d) To provide terminology and definitions that describe the system protocol.
- e) To allow a broad range of design latitude so that the designer can optimize cost and/or performance without affecting system compatibility.
- f) To provide a system where performance is primarily device limited, rather than system interface limited.

1.2 IEC 821 BUS interface system elements

1.2.1 Basic definitions

The IEC 821 BUS structure can be described from two points of view: its mechanical structure and its functional structure. The mechanical specification describes the physical dimensions of subracks, backplanes, front panels, plug-in boards, etc. The IEC 821 BUS functional specification describes how the bus works, what functional modules are involved in each transaction, and the rules which govern their behavior. This paragraph provides informal definitions for some basic terms used to describe both the physical and the mechanical structure of the IEC 821 BUS.

1.2.1.1 Terms used to describe the IEC 821 BUS mechanical structure

IEC 821 BUS BACKPLANE

A printed circuit (PC) board with 96-pin connectors and signal paths that bus the connector pins. Some IEC 821 BUS systems have a single PC board, called the J_1 backplane. It provides the signal paths needed for basic operation. Other IEC 821 BUS systems also have an optional second PC board, called a J_2 backplane. It provides the additional 96-pin connectors and signal paths needed for wider data and address transfers. Still others have a single PC board that provides the signal conductors and connectors of both the J_1 and J_2 backplanes.

CARTE

Carte de circuit imprimé, son ensemble de composants électroniques ainsi qu'un ou deux connecteurs 96 broches enfichables dans les connecteurs du fond de panier du BUS CEI 821.

EMPLACEMENT

Position où une carte peut être enfichée dans un fond de panier du BUS CEI 821. Si le système de BUS CEI 821 a deux fonds de panier J_1 et J_2 (ou une combinaison J_1/J_2 de fond de panier) chaque emplacement correspond à une paire de connecteurs 96 broches. Si le système n'a qu'un fond de panier J_1 , alors chaque emplacement comporte un seul connecteur 96 broches.

CHASSIS

Cadre rigide qui fournit un support mécanique pour les cartes enfichées dans le fond de panier, assurant que les connecteurs mâles et femelles s'assemblent correctement et que les cartes adjacentes ne sont pas en contact. Il guide aussi le flux d'air de refroidissement à travers le système, et assure que les cartes enfichées ne se désolidarisent pas du fond de panier à la suite de vibrations ou de chocs.

1.2.1.2 Termes utilisés pour décrire la structure fonctionnelle du BUS CEI 821

La figure 1-1, page 34, montre un schéma-bloc simplifié de la structure fonctionnelle, incluant les signaux du BUS CEI 821, la logique de l'interface du fond de panier et les modules fonctionnels.

LOGIQUE DE L'INTERFACE DU FOND DE PANIER

Logique spéciale de l'interface qui tient compte des caractéristiques du fond de panier: l'impédance des lignes de signaux, le temps de propagation, les valeurs des adaptations, etc. La norme du BUS CEI 821 prescrit certaines règles pour la réalisation de cette logique fondées sur la longueur maximale du fond de panier et le nombre maximal d'emplacements pour les cartes.

MODULE FONCTIONNEL

Ensemble de circuits électroniques résidant sur une carte de BUS CEI 821 et collaborant à l'accomplissement d'une tâche.

BUS DE TRANSFERT DE DONNEES

Un des quatre bus du fond de panier du BUS CEI 821. Le bus de transfert de données permet aux MAITRES de diriger le transfert de données binaires entre eux-mêmes et des ESCLAVES. (On abrège souvent le bus de transfert de données en DTB.)

CYCLE DU BUS DE TRANSFERT DE DONNEES

Suite de transitions de niveaux sur les lignes de signaux du DTB qui a pour résultat le transfert d'une adresse ou d'une adresse et des données entre un MAITRE et un ESCLAVE. Le cycle du bus de transfert de données est divisé en deux parties: la diffusion d'adresse, suivie ou non de un ou plusieurs transferts de données. Il y a 34 types de cycle du bus de transfert de données. Ils sont définis dans la suite de ce chapitre.

MAITRE

Module fonctionnel qui déclenche les cycles du DTB dans le but de transférer des données entre lui-même et un module ESCLAVE.

BOARD

A printed circuit (PC) board, its collection of electronic components, and either one or two 96-pin connectors that can be plugged into IEC 821 BUS backplane connectors.

SLOT

A position where a board can be inserted into an IEC 821 BUS backplane. If the IEC 821 BUS system has both a J_1 and a J_2 backplane (or a combination J_1/J_2 backplane) each slot provides a pair of 96-pin connectors. If the system has only a J_1 backplane, then each slot provides a single 96-pin connector.

SUBRACK

A rigid framework that provides mechanical support for boards inserted into the backplane, ensuring that the connectors mate properly and that adjacent boards do not contact each other. It also guides the cooling airflow through the system, and ensures that inserted boards do not disengage themselves from the backplane due to vibration or shock.

1.2.1.2 Terms used to describe the IEC 821 BUS functional structure

Figure 1-1, page 35, shows a simplified block diagram of the functional structure, including the IEC 821 BUS signal lines, backplane interface logic, and functional modules.

BACKPLANE INTERFACE LOGIC

Special interface logic that takes into account the characteristics of the backplane: its signal line impedance, propagation time, termination values, etc. The IEC 821 BUS standard prescribes certain rules for the design of this logic based on the maximum length of the backplane and its maximum number of board slots.

FUNCTIONAL MODULE

A collection of electronic circuitry that resides on one IEC 821 BUS board and works together to accomplish a task.

DATA TRANSFER BUS

One of the four buses provided by the IEC 821 BUS backplane. The Data Transfer Bus allows MASTERS to direct the transfer of binary data between themselves and SLAVES. (Data Transfer Bus is often abbreviated DTB.)

DATA TRANSFER BUS CYCLE

A sequence of level transitions on the signal lines of the DTB that result in the transfer of an address or an address and data between a MASTER and a SLAVE. The Data Transfer Bus cycle is divided into two portions, the address broadcast and then one or more data transfers. There are 34 types of Data Transfer Bus cycles. They are defined later in this chapter.

MASTER

A functional module that initiates DTB cycles in order to transfer data between itself and a SLAVE module.

ESCLAVE

Module fonctionnel qui détecte les cycles du DTB déclenchés par un MAITRE et, lorsque ces cycles impliquent sa participation, transfère des données entre lui-même et le MAITRE.

DETECTEUR D'ACCES

Module fonctionnel qui surveille les transferts de données sur le DTB dans le but de détecter des adresses dont la surveillance lui a été prescrite. En cas d'accès à l'une de ces adresses, le DETECTEUR D'ACCES génère un signal sur la carte.

LIMITEUR DE TEMPS D'OCCUPATION DU BUS

Module fonctionnel qui mesure la durée de chaque transfert de données sur le DTB et termine le cycle du DTB si un transfert dure trop longtemps. Sans ce module, si le MAITRE essaie de transférer des données à un emplacement d'ESCLAVE non existant, il risque d'attendre indéfiniment. Le LIMITEUR DE TEMPS D'OCCUPATION DU BUS évite cela en terminant le cycle.

BUS D'INTERRUPTION

Un des quatre bus du fond de panier du BUS CEI 821. Le bus d'interruption permet aux modules GENERATEUR D'INTERRUPTION d'envoyer des demandes d'interruption aux modules CONTROLEUR D'INTERRUPTION.

GENERATEUR D'INTERRUPTION

Module fonctionnel qui génère une demande d'interruption sur le bus d'interruption puis fournit les informations du MOT D'ETAT/IDentificateur quand le CONTROLEUR D'INTERRUPTION les demande.

CONTROLEUR D'INTERRUPTION

Module fonctionnel qui détecte les demandes d'interruption produites par des GENERATEURS D'INTERRUPTION et répond à ces requêtes en demandant l'information MOT D'ETAT/IDentificateur.

CHAINE SERIE

Ligne de signal du BUS CEI 821 d'un type spécial utilisée pour propager un niveau de signal de carte en carte, partant du premier emplacement et se terminant au dernier emplacement. Le BUS CEI 821 comporte quatre lignes en chaîne série pour l'allocation du bus et une ligne en chaîne série pour la reconnaissance d'interruption.

EMETTEUR DE CHAINE SERIE IACK

Module fonctionnel qui active la chaîne série de reconnaissance d'interruption chaque fois qu'un CONTROLEUR D'INTERRUPTION reconnaît une demande d'interruption. Cette chaîne garantit qu'un seul GENERATEUR D'INTERRUPTION répondra avec son MOT D'ETAT/IDentificateur dans le cas où plusieurs d'entre eux auront généré des demandes d'interruption.

BUS D'ARBITRAGE

Un des quatre bus du fond de panier du BUS CEI 821. Ce bus permet à un module ARBITRE et à plusieurs modules DEMANDEURS de coordonner l'utilisation du DTB.

DEMANDEUR

Module fonctionnel qui réside sur la carte d'un MAITRE ou d'un CONTROLEUR D'INTERRUPTION et qui demande l'utilisation du DTB chaque fois que le MAITRE ou le CONTROLEUR D'INTERRUPTION en a besoin.

SLAVE

A functional module that detects DTB cycles initiated by a MASTER and, when those cycles specify its participation, transfers data between itself and the MASTER.

LOCATION MONITOR

A functional module that monitors data transfers over the DTB in order to detect accesses to the locations it has been assigned to watch. When an access to one of these assigned locations occurs, the LOCATION MONITOR generates an on-board signal.

BUS TIMER

A functional module that measures how long each data transfer takes on the DTB and terminates the DTB cycle if a transfer takes too long. Without this module, if the MASTER tries to transfer data to or from a non-existent SLAVE location it might wait forever. The BUS TIMER prevents this by terminating the cycle.

PRIORITY INTERRUPT BUS

One of the four buses provided by the IEC 821 BUS backplane. The Priority Interrupt Bus allows INTERRUPTER modules to send interrupt requests to INTERRUPT HANDLER modules.

INTERRUPTER

A functional module that generates an interrupt request on the Priority Interrupt Bus and then provides STATUS/ID information when the INTERRUPT HANDLER requests it.

INTERRUPT HANDLER

A functional module that detects interrupt requests generated by INTERRUPTERS and responds to those requests by asking for STATUS/ID information.

DAISY-CHAIN

A special type of IEC 821 BUS signal line that is used to propagate a signal level from board to board, starting with the first slot and ending with the last slot. There are four bus grant daisy-chains and one interrupt acknowledge daisy-chain on the IEC 821 BUS.

IACK DAISY-CHAIN DRIVER

A functional module which activates the interrupt acknowledge daisy-chain whenever an INTERRUPT HANDLER acknowledges an interrupt request. This daisy-chain ensures that only one INTERRUPTER will respond with its STATUS/ID when more than one has generated an interrupt request.

ARBITRATION BUS

One of the four buses provided by the IEC 821 BUS backplane. This bus allows an ARBITER module and several REQUESTER modules to coordinate use of the DTB.

REQUESTER

A functional module that resides on the same board as a MASTER or INTERRUPT HANDLER and requests use of the DTB whenever its MASTER or INTERRUPT HANDLER needs it.

ARBITRE

Module fonctionnel qui reçoit les demandes du bus provenant des modules DEMANDEURS et accorde le contrôle du DTB à un seul DEMANDEUR à la fois.

BUS UTILITAIRE

Un des quatre bus du fond de panier du BUS CEI 821. Ce bus comprend des signaux d'horloge et des signaux qui coordonnent la mise sous tension et les coupures d'alimentation des systèmes de BUS CEI 821.

EMETTEUR D'HORLOGE DU SYSTEME

Module fonctionnel qui fournit un signal d'horloge 16 MHz sur le bus utilitaire.

EMETTEUR D'HORLOGE DU BUS SERIE

Module fonctionnel qui fournit un signal d'horloge chargé de synchroniser le fonctionnement du BUS CEI 823*. (Bien que la norme du BUS CEI 821 définisse un EMETTEUR D'HORLOGE DU BUS SERIE à utiliser avec le BUS CEI 823, et bien qu'elle réserve deux lignes de signaux du fond de panier pour le BUS CEI 823, le protocole du BUS CEI 823 est complètement indépendant du BUS CEI 821.) La spécification du chronogramme du signal généré par l'EMETTEUR D'HORLOGE DU BUS SERIE est donné dans l'annexe C.

MODULE DE CONTROLE D'ALIMENTATION

Module fonctionnel qui surveille l'état de la source d'alimentation principale du système de BUS CEI 821 et signale quand cette source d'alimentation est hors des limites exigées pour un fonctionnement fiable du système. Puisque la plupart des systèmes sont alimentés par une source alternative, le MODULE DE CONTROLE D'ALIMENTATION est généralement conçu pour détecter des conditions de coupure d'alimentation et de sous-tension de l'alimentation alternative du secteur.

CARTE CONTROLEUR DU SYSTEME

Carte qui réside dans l'emplacement 1 du fond de panier du BUS CEI 821 et qui possède un EMETTEUR D'HORLOGE DU SYSTEME, un ARBITRE, un EMETTEUR DE CHAINE SERIE IACK et un LIMITEUR DE TEMPS D'OCCUPATION DU BUS. Parfois cette carte possède aussi un EMETTEUR D'HORLOGE DU BUS SERIE, un MODULE DE CONTROLE D'ALIMENTATION ou les deux.

1.2.1.3 Types de cycles du BUS CEI 821**CYCLE LECTURE**

Cycle du DTB utilisé pour transférer 1, 2, 3 ou 4 octets d'un ESCLAVE vers un MAITRE. Le cycle commence lorsque le MAITRE diffuse une adresse et un modificateur d'adresse. Chaque ESCLAVE acquiert l'adresse et le modificateur d'adresse, et vérifie s'il doit répondre. Si c'est le cas, il extrait les données de sa mémoire interne, les place sur le bus de données et acquitte le transfert. Le MAITRE termine alors le cycle.

CYCLE ECRITURE

Cycle du DTB utilisé pour transférer 1, 2, 3 ou 4 octets d'un MAITRE vers un ESCLAVE. Le cycle commence quand le MAITRE diffuse une adresse et un modificateur d'adresse et place des données sur le DTB. Chaque ESCLAVE acquiert l'adresse et le modificateur d'adresse, et vérifie s'il doit répondre. Si c'est le cas, il enregistre les données, puis acquitte le transfert. Le MAITRE termine alors le cycle.

* A l'étude.

ARBITER

A functional module that accepts bus requests from REQUESTER modules and grants control of the DTB to one REQUESTER at a time.

UTILITY BUS

One of the four buses provided by the IEC 821 BUS backplane. This bus includes signals that provide periodic timing and coordinate the power-up and power-down of IEC 821 BUS systems.

SYSTEM CLOCK DRIVER

A functional module that provides a 16 MHz timing signal on the Utility Bus.

SERIAL CLOCK DRIVER

A functional module that provides a periodic timing signal that synchronizes operation of the IEC 823 BUS*. (Although the IEC 821 BUS standard defines a SERIAL CLOCK DRIVER for use with the IEC 823 BUS, and although it reserves two backplane signal lines for use by that bus, the IEC 823 BUS protocol is completely independent of the IEC 821 BUS.) Timing specifications for the SERIAL CLOCK DRIVER are given in Appendix C.

POWER MONITOR MODULE

A functional module that monitors the status of the primary power source to the IEC 821 BUS system, and signals when that power has strayed outside the limits required for reliable system operation. Since most systems are powered by an a.c. source, the POWER MONITOR is typically designed to detect drop-out or brown-out conditions on a.c. lines.

SYSTEM CONTROLLER BOARD

A board which resides in slot 1 of an IEC 821 BUS backplane and has a SYSTEM CLOCK DRIVER, an ARBITER, an IACK DAISY-CHAIN DRIVER and a BUS TIMER. Some also have a SERIAL CLOCK DRIVER, a POWER MONITOR or both.

1.2.1.3 Types of cycles on the IEC 821 BUS**READ CYCLE**

A DTB cycle used to transfer 1, 2, 3, or 4 bytes from a SLAVE to a MASTER. The cycle begins when the MASTER broadcasts an address and an address modifier. Each SLAVE captures the address modifier and address, and checks to see if it is to respond to the cycle. If so, it retrieves the data from its internal storage, places it on the data bus and acknowledges the transfer. The MASTER then terminates the cycle.

WRITE CYCLE

A DTB cycle used to transfer 1, 2, 3, or 4 bytes from a MASTER to a SLAVE. The cycle begins when the MASTER broadcasts an address and address modifier and places data on the DTB. Each SLAVE captures the address and address modifier and checks to see if it is to respond to the cycle. If so, it stores the data and then acknowledges the transfer. The MASTER then terminates the cycle.

* Under consideration.

CYCLE DE LECTURE PAR BLOC

Cycle du DTB utilisé pour transférer un bloc de 1 à 256 octets d'un ESCLAVE vers un MAITRE. Ce transfert est fait en utilisant une chaîne de données de 1, 2 ou 4 octets. Une fois que le transfert du bloc est commencé, le MAITRE ne libère le DTB que lorsque tous les octets sont transférés. La différence avec une suite de cycles de lecture est que le MAITRE ne transmet qu'une adresse et un modificateur d'adresse (au début du cycle). Ensuite, l'ESCLAVE incrémente cette adresse à chaque transfert afin que les données du transfert suivant soient extraites de l'emplacement contigu d'adresse supérieure.

CYCLE D'ECRITURE PAR BLOC

Cycle du DTB utilisé pour transférer un bloc de 1 à 256 octets d'un MAITRE vers un ESCLAVE. Le cycle d'écriture par bloc ressemble beaucoup au cycle de lecture par bloc. Ce transfert est fait en utilisant une chaîne de données de 1, 2 ou 4 octets et le MAITRE ne libère le DTB que lorsque tous les octets sont transférés. La différence avec une suite de cycles d'écriture est que le MAITRE ne transmet qu'une adresse et un modificateur d'adresse (au début du cycle). Ensuite, l'ESCLAVE incrémente cette adresse à chaque transfert afin que les données du transfert suivant soient enregistrées à l'emplacement contigu d'adresse supérieure.

CYCLE DE LECTURE-MODIFICATION-ECRITURE

Cycle du DTB utilisé pour lire ou écrire à une adresse d'ESCLAVE sans permettre à un autre MAITRE d'avoir accès à cette adresse. Ce cycle est très utile dans les systèmes de multitraitement où certains emplacements de mémoire sont utilisés pour assurer des fonctions liées aux sémaphores.

CYCLE UNIQUEMENT D'ADRESSAGE

Cycle du DTB qui consiste en une émission d'adresse, sans transfert de données. Les ESCLAVES ne doivent pas acquitter les cycles UNIQUEMENT D'ADRESSAGE et les MAITRES terminent le cycle sans attendre l'acquiescement.

CYCLE DE RECONNAISSANCE DE L'INTERRUPTION

Cycle du DTB, déclenché par un CONTROLEUR D'INTERRUPTION, pour lire un "MOT D'ETAT/IDentificateur" d'un GENERATEUR D'INTERRUPTION. Un CONTROLEUR D'INTERRUPTION déclenche ce cycle lorsqu'il détecte une demande d'interruption d'un GENERATEUR D'INTERRUPTION et qu'il obtient le contrôle du DTB.

1.2.2 Structure générale du BUS CEI 821

Le système d'interface du BUS CEI 821 comprend une logique d'interface de fond de panier, quatre groupes de lignes de signaux appelés "bus" et un ensemble de "modules fonctionnels" qui peuvent être configurés suivant les besoins. Les modules fonctionnels communiquent entre eux par l'intermédiaire des lignes de signaux du bus du fond de panier.

Les "modules fonctionnels" définis dans cette norme pour faciliter la présentation du protocole du bus ne doivent pas être considérés comme une contrainte pour la conception. Par exemple, le concepteur peut concevoir des circuits logiques qui dialoguent avec le BUS CEI 821 conformément à la description, mais qui utilisent des signaux différents sur la carte ou contrôlent d'autres signaux du BUS CEI 821. Les cartes de BUS CEI 821 peuvent être conçues pour inclure n'importe quelle combinaison des modules fonctionnels définis dans cette norme.

BLOCK READ CYCLE

A DTB cycle used to transfer a block of 1 to 256 bytes from a SLAVE to a MASTER. This transfer is done using a string of 1, 2, or 4 byte data transfers. Once the block transfer is started, the MASTER does not release the DTB until all of the bytes have been transferred. It differs from a string of read cycles in that the MASTER broadcasts only one address and address modifier (at the beginning of the cycle). Then the SLAVE increments this address on each transfer so that the data for the next transfer is retrieved from the next higher location.

BLOCK WRITE CYCLE

A DTB cycle used to transfer a block of 1 to 256 bytes from a MASTER to a SLAVE. The block write cycle is very similar to the block read cycle. It uses a string of 1, 2, or 4 byte data transfers and the MASTER does not release the DTB until all of the bytes have been transferred. It differs from a string of write cycles in that the MASTER broadcasts only one address and address modifier (at the beginning of the cycle). Then the SLAVE increments this address on each transfer so that the data from the next transfer is stored in the next higher location.

READ-MODIFY-WRITE CYCLE

A DTB cycle that is used to both read from, and write to, a SLAVE location without permitting any other MASTER to access that location. This cycle is most useful in multiprocessing systems where certain memory locations are used to provide semaphore functions.

ADDRESS-ONLY CYCLE

A DTB cycle that consists of an address broadcast, but no data transfer. SLAVES do not acknowledge ADDRESS-ONLY cycles and MASTERS terminate the cycle without waiting for an acknowledgement.

INTERRUPT ACKNOWLEDGE CYCLE

A DTB cycle, initiated by an INTERRUPT HANDLER, that reads a STATUS/ID from an INTERRUPTER. An INTERRUPT HANDLER generates this cycle whenever it detects an interrupt request from an INTERRUPTER and it has control of the DTB.

1.2.2 Basic IEC 821 BUS structure

The IEC 821 BUS interface system consists of backplane interface logic, four groups of signal lines called "buses", and a collection of "functional modules" which can be configured as required. The functional modules communicate with each other using the backplane signal lines.

The "functional modules" defined in this standard are used as vehicles for discussion of the bus protocol, and need not be considered a constraint to logic design. For example, the designer might choose to design logic which interacts with the IEC 821 BUS in the manner described, but uses different on board signals, or monitors other IEC 821 BUS signals. IEC 821 BUS boards might be designed to include any combination of the functional modules defined by this standard.

La structure fonctionnelle du BUS CEI 821 peut être divisée en quatre catégories. Chaque catégorie comprend un bus et les modules fonctionnels associés qui oeuvrent ensemble pour remplir des tâches spécifiques. La figure 1-2, page 36, représente les modules fonctionnels et les bus du BUS CEI 821. Chaque catégorie est brièvement résumée ci-dessous.

a) Transfert de données

Les unités transfèrent des données sur le bus de transfert de données (DTB) qui contient des lignes de données et d'adresses, et les signaux de contrôle associés. Des modules fonctionnels appelés MAITRES, ESCLAVES, GENERATEURS D'INTERRUPTION et CONTROLEURS D'INTERRUPTION utilisent le DTB pour transférer des données entre eux. Deux autres modules, appelés LIMITEUR DE TEMPS D'OCCUPATION DU BUS et EMETTEUR DE CHAINE SERIE IACK les aident aussi dans ce processus.

b) Arbitrage du DTB

Comme le système BUS CEI 821 peut être configuré avec plusieurs MAITRES ou CONTROLEURS D'INTERRUPTION, un moyen est prévu pour transmettre le contrôle du DTB à chacun d'eux et faire en sorte qu'un seul ait le contrôle du DTB à un instant donné. Les modules d'arbitrage du bus (DEMANDEURS et ARBITRE) coordonnent le transfert du contrôle.

c) Interruption prioritaire

La capacité d'interruption prioritaire du BUS CEI 821 permet aux unités d'effectuer des demandes à un CONTROLEUR D'INTERRUPTION. Ces demandes d'interruption peuvent avoir jusqu'à sept niveaux de priorité. Les GENERATEURS D'INTERRUPTION et CONTROLEURS D'INTERRUPTION utilisent les lignes de signaux du bus d'interruption.

d) Utilitaires

Les horloges, l'initialisation et la détection des erreurs sont fournis par le bus utilitaire. Celui-ci comprend deux lignes d'horloge, une ligne d'initialisation du système, une ligne de défaillance système, une ligne de défaillance de l'alimentation du secteur en courant alternatif et une ligne de données série.

The IEC 821 BUS functional structure can be divided into four categories. Each consists of a bus and its associated functional modules which work together to perform specific duties. Figure 1-2, page 37, shows the IEC 821 BUS functional modules and buses. Each category is briefly summarized below.

a) Data Transfer

Devices transfer data over the Data Transfer Bus (DTB), which contains data and address pathways and associated control signals. Functional modules called MASTERS, SLAVES, INTERRUPTERS, and INTERRUPT HANDLERS use the DTB to transfer data between each other. Two other modules, called a BUS TIMER and an IACK DAISY-CHAIN DRIVER also assist them in this process.

b) DTB Arbitration

Since an IEC 821 BUS system can be configured with more than one MASTER or INTERRUPT HANDLER, a means is provided to transfer control of the DTB between them in an orderly manner and to guarantee that only one controls the DTB at a given time. The Arbitration Bus modules (REQUESTERS and ARBITER) coordinate the control transfer.

c) Priority Interrupt

The priority interrupt capability of the IEC 821 BUS provides a means by which devices can request service from an INTERRUPT HANDLER. These interrupt requests can be prioritized into a maximum of seven levels. INTERRUPTERS and INTERRUPT HANDLERS use the Priority Interrupt Bus signal lines.

d) Utilities

Periodic clocks, initialization, and failure detection are provided by the Utility Bus. It includes two clock lines, a system reset line, a system fail line, an a.c. fail line, and a serial data line.

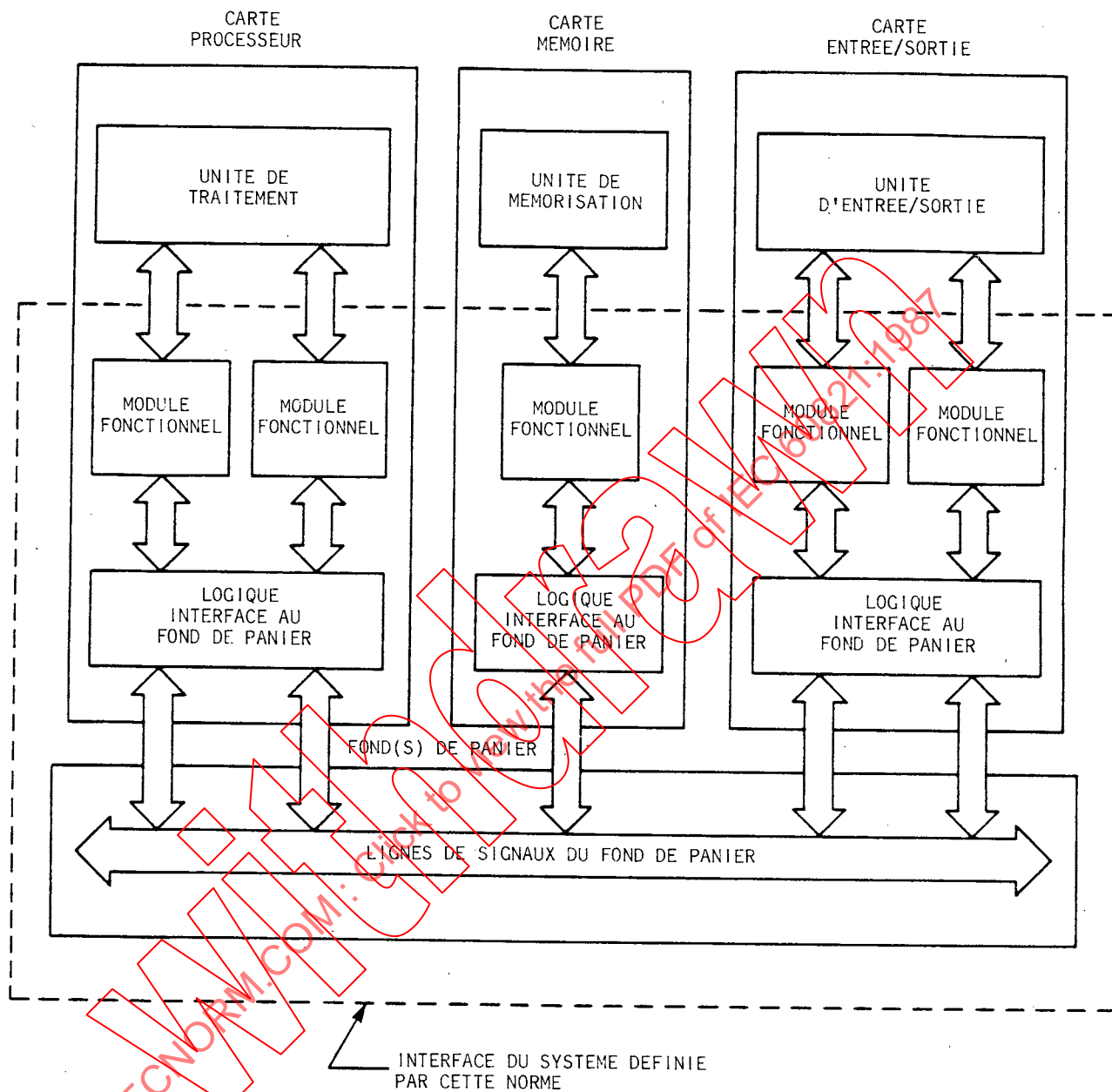


FIG. 1-1. - Eléments du système définis par cette norme.

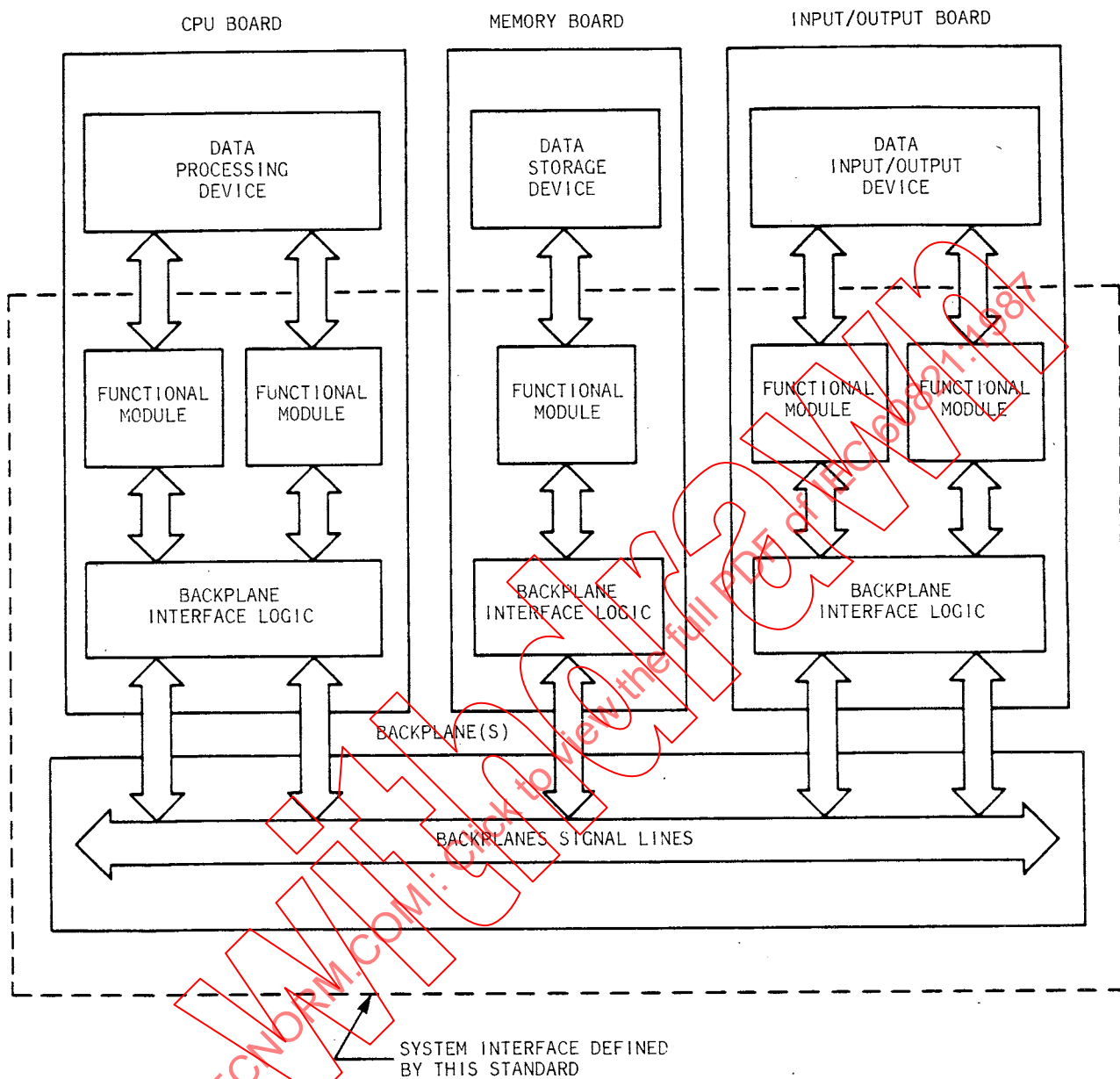


FIG. 1-1. - System elements defined by this standard.

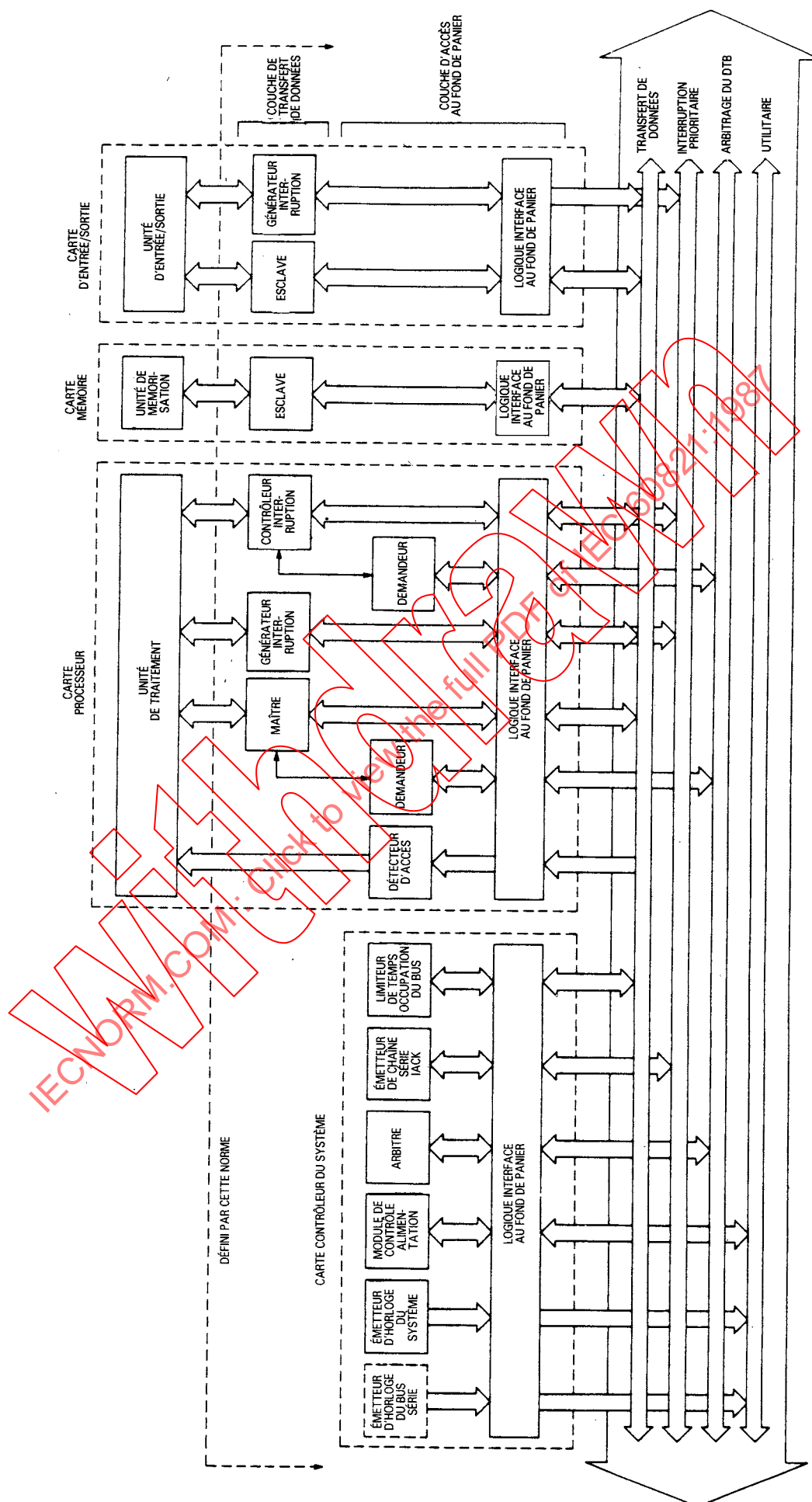


FIG. 1-2. - Modules fonctionnels et bus définis par cette norme.

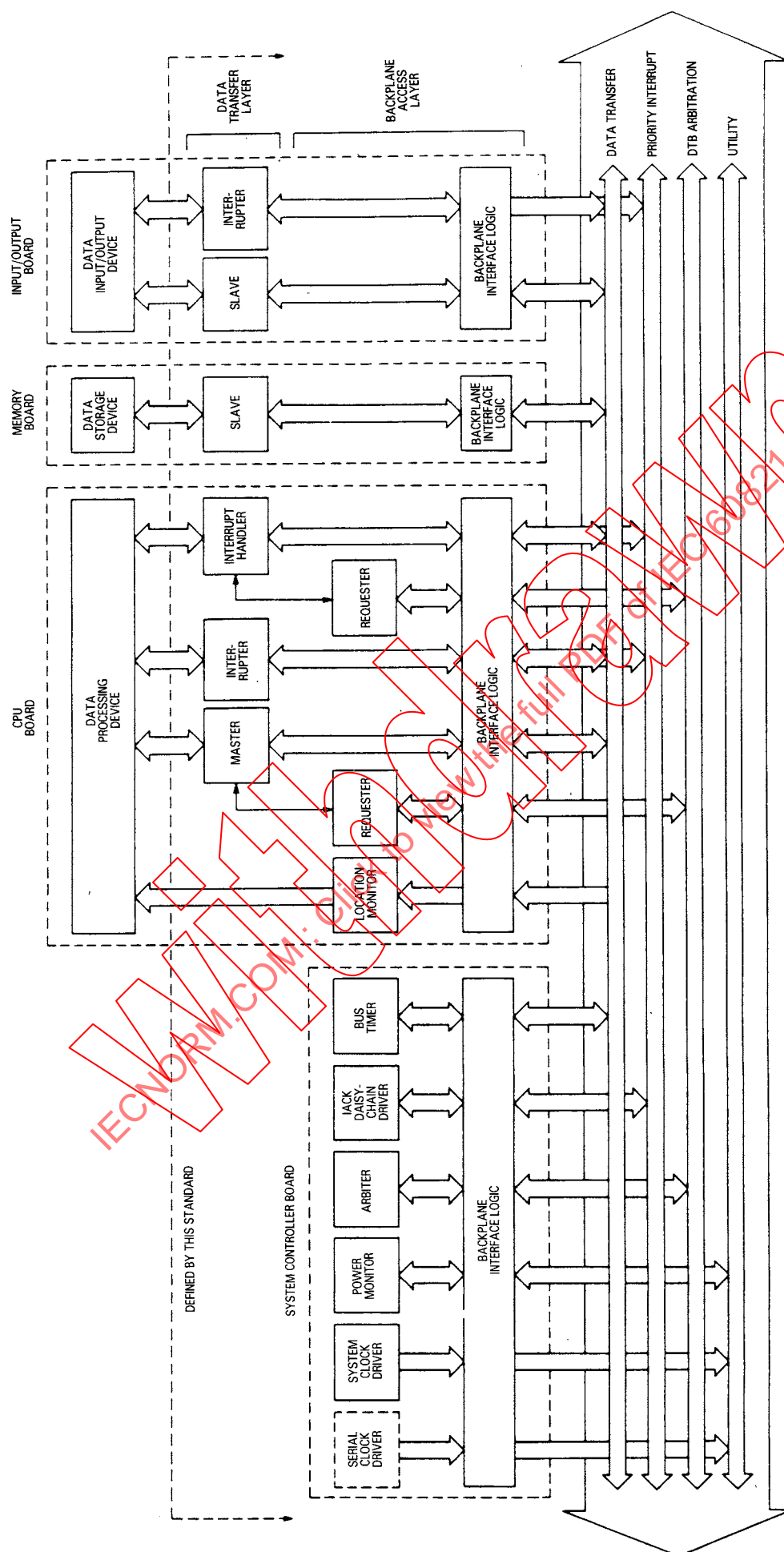


FIG. 1-2. - Functional modules and buses defined by this standard.

1.3 Diagrammes de la norme du BUS CEI 821

Pour définir ou décrire le fonctionnement du BUS CEI 821, différents types de diagrammes sont utilisés:

- a) Les chronogrammes qui montrent les relations de temps entre les transitions des signaux. Les temps considérés doivent avoir des limites inférieures et/ou supérieures associées. Certains des temps indiqués dans ces diagrammes spécifient le comportement de la logique d'interface du fond de panier tandis que d'autres spécifient le comportement d'interverrouillage des modules fonctionnels.
- b) Les schémas des séquences, semblables aux chronogrammes mais montrant seulement les relations temporelles d'interverrouillage des modules fonctionnels. Ces diagrammes montrent la suite des événements plutôt que les temps impliqués. Par exemple, un schéma des séquences peut montrer que le module A ne peut générer le signal de transition B avant d'avoir détecté la génération du signal de transition D du module C.
- c) Les organigrammes qui montrent une chaîne d'événements telle qu'elle se présente lors d'une opération du BUS CEI 821. Les événements résultent de l'interaction entre deux ou plusieurs modules fonctionnels et sont décrits par un texte. L'organigramme décrit les opérations du BUS CEI 821 de façon séquentielle et montre, en même temps, l'interaction entre les modules fonctionnels.

1.4 Terminologie utilisée dans la norme

Pour éviter toute confusion et pour clarifier les exigences de conformité, la plupart des paragraphes de cette norme sont repérés par des mots clefs qui indiquent le type d'informations qu'ils contiennent. La liste des mots clefs est donnée ci-dessous:

REGLE
RECOMMANDATION
SUGGESTION
AUTORISATION
OBSERVATION

Un texte sans l'un de ces mots clefs décrit la structure ou le fonctionnement du BUS CEI 821. Il est écrit dans un style descriptif ou narratif. L'utilisation de ces mots clefs est la suivante:

REGLE chapitre.numéro:

Les règles forment le cadre fondamental de la norme du BUS CEI 821. Elles sont exprimées parfois sous forme de texte et parfois sous forme de figures, tableaux ou dessins. Toutes les règles du BUS CEI 821 DOIVENT être observées pour assurer la compatibilité entre les produits conçus pour le BUS CEI 821. Les règles sont caractérisées par un style impératif. Les mots DOIT et NE DOIT PAS, en lettres majuscules, sont réservés exclusivement pour déclarer les règles de cette norme et ne sont pas utilisés pour d'autres usages.

RECOMMANDATION chapitre.numéro:

Une recommandation indique au concepteur qu'il serait judicieux de suivre les conseils donnés. Faire autrement peut entraîner des ennuis graves ou une mauvaise performance. Bien que le BUS CEI 821 ait été conçu pour des systèmes de haute performance il est possible de

1.3 IEC 821 BUS standard diagrams

As aids to defining or describing IEC 821 BUS operation, several types of diagrams are used, including:

- a) Timing diagrams that show the timing relationships between signal transitions. The times involved will have minimum and/or maximum limits associated with them. Some of the times specified on these diagrams specify the behavior of the backplane interface logic, while other times specify the interlocked behavior of the functional modules.
- b) Sequence diagrams that are similar to a timing diagram but show only the interlocked timing relationships of the functional modules. This diagram is intended to show a sequence of events, rather than to specify the times involved. For example, a sequence diagram might indicate that module A cannot generate signal transition B until it detects module C's generation of signal transition D.
- c) Flow diagrams that show a stream of events as they would occur during an IEC 821 BUS operation. The events are stated in words and result from interaction of two or more functional modules. The flow diagram describes IEC 821 BUS operations in a sequential manner and, at the same time, shows interaction of the functional modules.

1.4 Standard terminology

To avoid confusion, and to make very clear what the requirements for compliance are, many of the paragraphs in this standard are labeled with keywords that indicate the type of information it contains. The keywords are listed below:

RULE
RECOMMENDATION
SUGGESTION
PERMISSION
OBSERVATION

Any text not labeled with one of these keywords describes the IEC 821 BUS structure or operation. It is written in either a descriptive or a narrative style. These keywords are used as follows:

RULE chapter.number:

Rules form the basic framework of the IEC 821 BUS standard. They are sometimes expressed in text form and sometimes in the form of figures, tables, or drawings. All IEC 821 BUS rules **MUST** be followed to ensure compatibility between IEC 821 BUS designs. Rules are characterized by an imperative style. The upper-case words **MUST** and **MUST NOT** are reserved exclusively for stating rules in this standard and are not used for any other purpose.

RECOMMENDATION chapter.number:

Wherever a recommendation appears, designers would be wise to take the advice given. Doing otherwise might result in some awkward problems or poor performance. While IEC 821 BUS has been designed to support high performance systems, it is possible to design an IEC

concevoir un système de BUS CEI 821 conforme à toutes les règles mais ayant des performances médiocres. Souvent le concepteur doit avoir un certain niveau d'expérience du BUS CEI 821 pour concevoir des cartes qui donnent la meilleure performance. Les recommandations contenues dans cette norme sont fondées sur l'expérience et sont fournies aux concepteurs pour accélérer leur apprentissage.

SUGGESTION chapitre.numéro:

Dans la norme du BUS CEI 821, une suggestion est un conseil utile mais non vital. Le lecteur est encouragé à ne pas l'écarter sans y réfléchir. Certains choix de conception des cartes de BUS CEI 821 sont difficiles tant que l'on n'a pas acquis suffisamment d'expérience. Les suggestions sont faites pour aider le concepteur encore inexpérimenté. Plusieurs suggestions se rapportent soit à la conception de cartes facilement reconfigurables pour être compatibles avec d'autres cartes, soit à la conception de cartes facilitant la mise au point du système.

AUTORISATION chapitre.numéro:

Dans certains cas une règle du BUS CEI 821 n'interdit pas spécifiquement une certaine méthode de conception, mais le lecteur peut se demander si celle-ci transgresse la règle ou si elle risque de conduire à un ennui. Les autorisations confirment au lecteur qu'une certaine méthode de conception est acceptable et n'entraînera pas de difficultés. Le mot PEUT, en lettres majuscules, est réservé exclusivement pour énoncer les autorisations de cette norme et n'est pas utilisé pour d'autres usages.

OBSERVATION chapitre.numéro:

Les observations ne donnent pas de conseils spécifiques. Elles découlent généralement de la discussion. Elles soulignent les implications de certaines règles du BUS CEI 821 et attirent l'attention sur certains points qui autrement pourraient être négligés. Elles donnent aussi la raison de certaines règles afin que le lecteur comprenne la nécessité de les suivre.

1.4.1 *Etats des lignes de signaux*

La norme du BUS CEI 821 décrit le protocole en termes de niveaux et de changements d'états sur les lignes du bus.

Une ligne de signal est toujours supposée être à l'un des deux niveaux ou en transition entre ces niveaux. Quand le terme "haut" est utilisé, il se rapporte à un niveau haut de tension. Le terme "bas" se rapporte à un niveau bas de tension. Une ligne de signal est "en transition" quand son niveau de tension se déplace entre ces niveaux. (Voir chapitre 6 pour les seuils de tension utilisés sur le BUS CEI 821.)

Deux types de transitions sont possibles sur une ligne de signal et sont appelés "fronts". Un front montant est le temps de passage de la ligne de signal entre le niveau bas et le niveau haut. Un front descendant est le temps de passage de la ligne de signal du niveau haut au niveau bas.

Certaines spécifications de bus indiquent les temps de montée et descente minimale et maximale pour ces fronts. Une difficulté vient de ce que les concepteurs de la carte ont peu de moyens de contrôle sur ces temps. Si le fond de panier est très chargé, les temps de montée et de descente seront longs. S'il est peu chargé, ces temps peuvent être courts. Même si les concepteurs savent ce que sera la charge minimale

821 BUS system that complies with all the rules, but has abysmal performance. In many cases, a designer needs a certain level of experience with IEC 821 BUS in order to design boards that deliver top performance. Recommendations found in this standard are based on this kind of experience, and are provided to designers to speed their traversal of the learning curve.

SUGGESTION chapter.number:

In the IEC 821 BUS specification, a suggestion contains advice which is helpful but not vital. The reader is encouraged to consider the advice before discarding it. Some design decisions that need to be made in designing IEC 821 BUS boards are difficult until experience has been gained with the IEC 821 BUS. Suggestions are included to help a designer who has not yet gained this experience. Some suggestions have to do with designing boards that can be easily reconfigured for compatibility with other boards, or with designing the board to make the job of system debugging easier.

PERMISSION chapter.number:

In some cases an IEC 821 BUS rule does not specifically prohibit a certain design approach, but the reader might be left wondering whether that approach might violate the spirit of the rule, or whether it might lead to some subtle problem. Permissions reassure the reader that a certain approach is acceptable and will cause no problems. The upper-case word MAY is reserved exclusively for stating permissions in this standard and is not used for any other purpose.

OBSERVATION chapter.number:

Observations do not offer any specific advice. They usually follow naturally from what has just been discussed. They spell out the implications of certain IEC 821 BUS rules and bring attention to things that might otherwise be overlooked. They also give the rationale behind certain rules, so that the reader understands why the rule must be followed.

1.4.1 *Signal line states*

The IEC 821 BUS standard describes its protocol in terms of levels and transitions on bus lines.

A signal line is always assumed to be in one of two levels or in transition between these levels. Whenever the term "high" is used, it refers to a high TTL voltage level. The term "low" refers to a low TTL voltage level. A signal line is "in transition" when its voltage is moving between these levels. (See Chapter 6 for voltage thresholds used on the IEC 821 BUS.)

There are two possible transitions which can appear on a signal line, and these are called "edges". A rising edge is the time during which a signal line makes its transition from a low level to a high level. The falling edge is the time during which a signal line makes its transition from a high level to a low level.

Some bus specifications prescribe maximum or minimum rise and fall times for these edges. The problem with doing this is that board designers have very little control over these times. If the backplane is heavily loaded, the rise and fall times will be long. If it is lightly loaded, these times might be short. Even if designers know what the maximum and minimum loading will be, they still need to spend time in

et maximale, ils doivent effectuer des essais en laboratoire pour trouver les émetteurs qui fourniront les temps spécifiés de montée et de descente.

En fait, les temps de montée et de descente sont le résultat d'un ensemble complexe d'interactions concernant les impédances des pistes du fond de panier, ses adaptations, l'impédance de source des émetteurs et la charge capacitive des pistes. Pour obtenir un compromis entre tous ces facteurs, le concepteur de la carte devrait connaître la théorie des lignes de transmission, et certains paramètres spécifiques des émetteurs et récepteurs qui ne sont pas normalement inclus dans la plupart des feuilles de spécifications des fabricants.

Compte tenu de tout cela, la norme du BUS CEI 821 ne spécifie pas les temps de montée et de descente. En lieu et place, elle spécifie les caractéristiques électriques des émetteurs et récepteurs et la manière de concevoir le fond de panier. Elle indique aussi, aux concepteurs, comment le délai de propagation des émetteurs est affecté dans le plus mauvais cas de charge du bus, si bien qu'ils peuvent s'assurer que la chronologie du BUS CEI 821 est respectée avant de construire une carte. Si les concepteurs du BUS CEI 821 suivent ces spécifications de délai de propagation, leurs cartes fonctionneront correctement avec les autres cartes compatibles de BUS CEI 821 même dans les plus mauvaises conditions.

1.4.2 Utilisation de l'astérisque (*)

Pour mieux définir l'utilisation, les mnémoniques des signaux ont un astérisque en suffixe quand c'est nécessaire:

- a) Un astérisque (*) qui suit le nom d'un signal actif sur un niveau, signifie que le signal est vrai ou valide quand il est au niveau bas.
- b) Un astérisque (*) qui suit le nom d'un signal actif sur un front, signifie que les actions déclenchées par ce signal ont lieu sur une transition du signal du niveau haut vers le niveau bas.

OBSERVATION 1.1:

L'astérisque n'a pas de sens pour les lignes d'horloge asynchrone SYSCLK et SERCLK. Il n'y a pas de relation de phase fixe entre ces deux lignes d'horloge et les autres temps du BUS CEI 821.

1.5 Spécification du protocole

Il y a deux couches de protocole du BUS CEI 821. La couche inférieure du BUS CEI 821, appelée la couche d'accès au fond de panier, est composée de la logique d'interface du fond de panier, des modules du bus utilitaire et des modules du bus d'arbitrage. La couche de transfert de données du BUS CEI 821 est composée des modules du bus de transfert de données et du bus d'interruption. La figure 1-2, page 36, indique ces couches.

OBSERVATION 1.2:

Les lignes de signaux utilisées par les modules des couches de transfert de données forment une classe spéciale parce qu'elles sont commandées par des modules différents à des instants différents. Elles sont commandées par des émetteurs de lignes qui peuvent être activés et désactivés par chaque carte selon les signaux générés dans la

the lab, experimenting to find out which drivers will provide the needed rise and fall times.

In fact, rise and fall times are the result of a complex set of interactions involving the signal line impedances of the backplane, its terminations, the source impedance of the drivers, and the capacitive loading of the signal line. In order to trade off all of these factors, the board designer would have to study transmission line theory, as well as certain specific parameters of drivers and receivers which are not normally found in most manufacturers' data sheets.

Recognizing all of this, the IEC 821 BUS standard doesn't specify rise and fall times. Instead, it specifies the electrical characteristics for drivers and receivers and suggests how to design the backplane. It also tells designers how the worst case bus loading will affect the propagation delay of these drivers so that they can ensure that the IEC 821 BUS timing is met before building a board. If IEC 821 BUS designers follow these propagation delay guidelines, their boards will operate reliably with other IEC 821 BUS compatible boards under worst case conditions.

1.4.2 Use of the asterisk (*)

To help define usage, signal mnemonics have an asterisk suffix where required:

- a) An asterisk (*) following the signal name of signals which are level significant denotes that the signal is true or valid when the signal is low.
- b) An asterisk (*) following the signal name of signals which are edge significant denotes that the actions initiated by that signal occur on a high to low transition.

OBSERVATION 1.1:

The asterisk is inappropriate for the asynchronously running clock lines SYSClk and SERCLK. There is no fixed phase relationship between these clock lines and other IEC 821 BUS timing.

1.5 Protocol specification

There are two layers of IEC 821 BUS protocol. The lowest IEC 821 BUS layer, called the backplane access layer, is composed of the backplane interface logic, the Utility Bus modules, and the Arbitration Bus modules. The IEC 821 BUS data transfer layer is composed of the Data Transfer Bus and Priority Interrupt Bus modules. Figure 1-2, page 37, shows this layering.

OBSERVATION 1.2:

The signal lines used by the data transfer layer modules form a special class because they are driven by different modules at different times. They are driven with line drivers that can be turned on and off at each board, based upon signals generated in the backplane access layer. It is very important that their turn-on and turn-off times be

couche d'accès au fond de panier. Il est très important que leurs temps d'activation et de désactivation soient soigneusement vérifiés afin d'éviter que deux émetteurs ne tentent de porter les mêmes lignes de signaux à des niveaux différents. Une notation spéciale est utilisée dans les chronogrammes de cette norme pour spécifier les temps de commutation. Elle est représentée dans la figure 1-3, page 46.

Il y a deux types fondamentaux de protocoles utilisés sur le BUS CEI 821: les protocoles en boucle fermée et les protocoles en boucle ouverte. Les protocoles en boucle fermée utilisent des signaux d'interverrouillage du bus tandis que les protocoles en boucle ouverte utilisent des signaux de diffusion du bus.

1.5.1 *Signaux d'interverrouillage du bus*

Un signal d'interverrouillage du bus est envoyé d'un module donné à un autre. Ce signal est acquitté par le module de réception. Une relation d'interverrouillage existe entre les deux modules jusqu'à ce que le signal soit acquitté. Par exemple, un GÉNÉRATEUR D'INTERRUPTION peut envoyer une demande d'interruption qui reçoit ensuite comme acquittement un signal de reconnaissance de l'interruption (aucune limite de temps n'est prescrite par la norme du BUS CEI 821). Le GÉNÉRATEUR D'INTERRUPTION maintient la demande d'interruption jusqu'à ce que le CONTRÔLEUR D'INTERRUPTION la reconnaisse.

Des signaux d'interverrouillage du bus règlent le fonctionnement interne du système de BUS CEI 821, par opposition à l'interaction avec des stimuli externes. Chaque signal d'interverrouillage agit d'un module source vers un module de destination dans le système de BUS CEI 821.

Les signaux de validation d'adresse et de donnée sont des signaux d'interverrouillage très importants. Ils sont interverrouillés avec les signaux d'accusé de réception de données et d'erreurs du bus et ils coordonnent le transfert d'adresses et de données qui constituent la base de tous les flux d'informations entre les modules de la couche de transfert de données.

1.5.2 *Signal de diffusion du bus*

Un module génère un signal de diffusion pour répondre à un événement. Il n'y a pas de protocole pour l'acquiescement d'un signal diffusé. La diffusion est maintenue pendant un temps minimal spécifié, assez long pour assurer que tous les modules appropriés détectent le signal. Les signaux de diffusion peuvent être activés à n'importe quel moment, indépendamment d'autres activités qui se passent sur le bus. Ils sont envoyés sur une ligne de signal spécifique. Citons en exemple la ligne de signal d'initialisation du système et celle de défaillance de l'alimentation alternative. Ces lignes de signaux ne sont pas destinées à un module spécifique mais annoncent des conditions d'exceptions à tous les modules.

carefully controlled to prevent two drivers from attempting to drive the same signal line to different levels. Special timing diagram notation is used in this standard to specify their turn-on and turn-off times. It is shown in Figure 1-3, page 47.

There are two basic kinds of protocol used on the IEC 821 BUS: closed loop protocols and open loop protocols. Closed loop protocols use interlocked bus signals while open loop protocols use broadcast bus signals.

1.5.1 *Interlocked bus signals*

An interlocked bus signal is sent from a specific module to another specific module. The signal is acknowledged by the receiving module. An interlocked relationship exists between the two modules until the signal is acknowledged. For example, an INTERRUPTER can send an interrupt request which is answered later with an interrupt acknowledge signal (no time limit is prescribed by the IEC 821 BUS standard). The INTERRUPTER doesn't remove the interrupt request until the INTERRUPT HANDLER acknowledges it.

Interlocked bus signals coordinate internal functions of the IEC 821 BUS system, as opposed to interacting with external stimuli. Each interlocked signal has a source module and a destination module within the IEC 821 BUS system.

The address strobe and data strobes are especially important interlocking signals. They are interlocked with the data transfer acknowledge and bus error signals and coordinate the transfer of addresses and data which are the basis for all information flow between modules in the data transfer layer.

1.5.2 *Broadcast bus signal*

A module generates a broadcast signal in response to an event. There is no protocol for acknowledging a broadcast signal. Instead, the broadcast is maintained for a minimum specified time, long enough to assure that all appropriate modules detect the signal. Broadcast signals might be activated at any time, irrespective of any other activity taking place on the bus. They are each sent over a dedicated signal line. Some examples are the system reset and a.c. failure lines. These signal lines are not sent to any specific module, but announce special conditions to all modules.

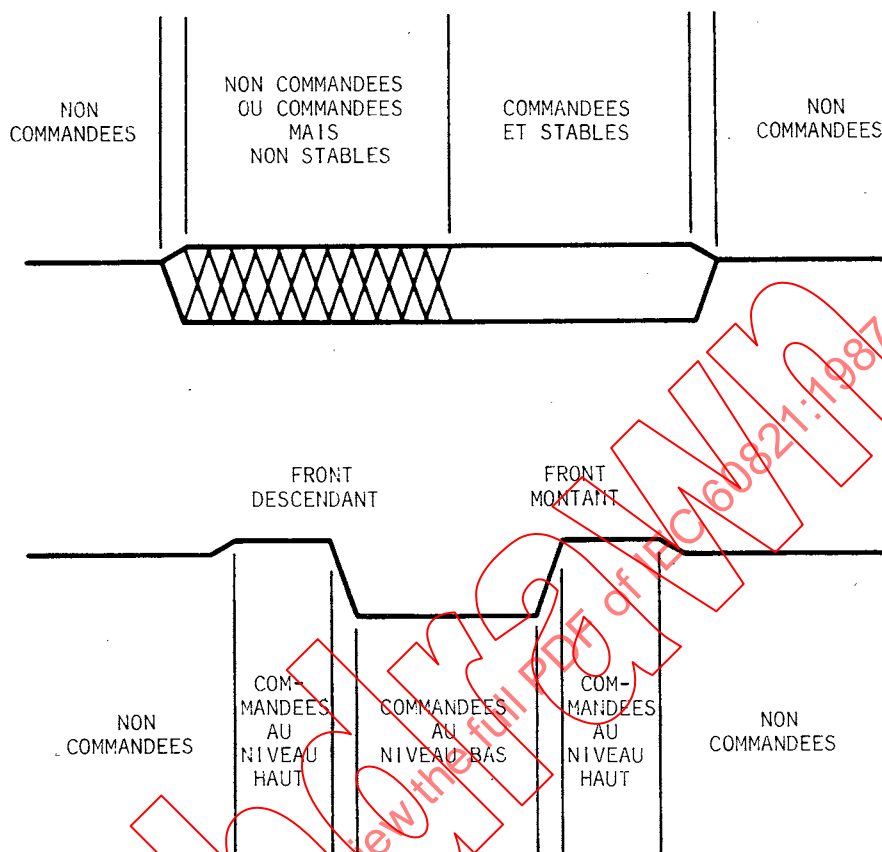


FIG. 1-3. - Notations utilisées dans les chronogrammes.

1.6 Exemples et explications relatifs au système

La spécification du protocole décrit, en détail, le comportement des différents modules fonctionnels. Elle indique comment un module répond à un signal sans indiquer l'origine de ce dernier. Par conséquent, la spécification du protocole ne donne pas au lecteur une image complète de ce qui se passe sur le bus. Pour aider le lecteur, la norme du BUS CEI 821 fournit des exemples de fonctionnement typique. Chaque exemple montre une séquence possible d'événements, mais d'autres séquences sont possibles. En fournissant ces exemples, on risque de voir les lecteurs supposer que la séquence montrée dans l'exemple est la seule correcte. Pour aider le lecteur à éviter ce piège, tous les exemples sont donnés dans un style narratif, utilisant le temps présent. Cela contraste avec le style impératif utilisé pour donner les règles de conformité de la norme du BUS CEI 821.

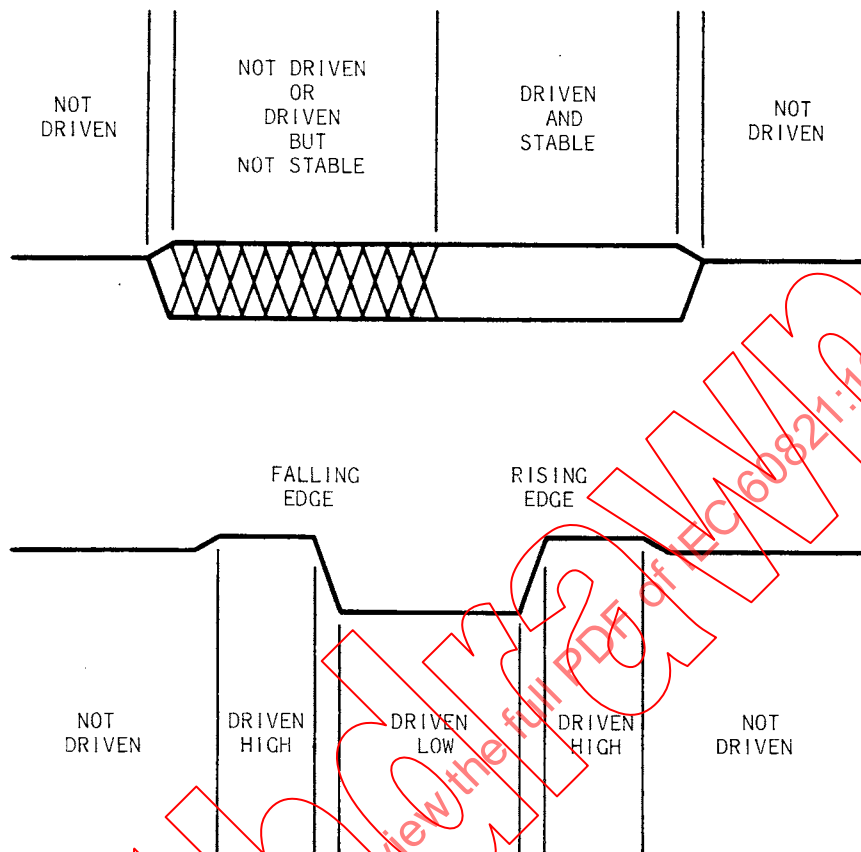


FIG. 1-3. - Signal timing notation.

1.6 System examples and explanations

A protocol specification describes, in detail, the behavior of the various functional modules. It discusses how a module responds to a signal without saying where the signal came from. Because of this, a protocol specification does not give the reader a complete picture of what is going on over the bus. To help the reader, the IEC 821 BUS standard provides examples of typical IEC 821 BUS operations. Each example shows one possible sequence of events; other sequences are also possible. In providing these examples, there is the danger that readers will assume that the sequence shown in the example is the only legal one. To help readers avoid this trap, all examples are given in a narrative style, using the present tense. This is in contrast to the imperative style used when giving rules for compliance with the IEC 821 BUS standard.

CHAPITRE 2: BUS DE TRANSFERT DE DONNEES DU BUS CEI 821

2.1 Introduction

Le BUS CEI 821 comprend un bus de transfert de données (DTB) parallèle, asynchrone, à haute vitesse. La figure 2-1, page 50, indique un système typique de BUS CEI 821, incluant tous les types de modules fonctionnels DTB. Le DTB est utilisé par les MAITRES pour sélectionner des adresses des ESCLAVES et pour transférer des données vers ou à partir de cette adresse. Certains MAITRES et ESCLAVES utilisent toutes les lignes de signaux du DTB, tandis que d'autres utilisent seulement une partie.

Des DETECTEURS D'ACCES surveillent les transferts de données entre des MAITRES et des ESCLAVES. Lorsqu'il y a un accès à ou aux adresses d'octets qu'il surveille, un DETECTEUR D'ACCES génère un signal sur la carte. Il pourrait envoyer, par exemple, un signal à son processeur sur la carte au moyen d'une interruption. Dans une telle configuration, si une carte processeur A écrit à une adresse de la mémoire globale du BUS CEI 821 surveillée par le DETECTEUR D'ACCES du processeur B, le processeur B sera interrompu.

Après qu'un MAITRE eut lancé un cycle de transfert de données, il attend la réponse de l'ESCLAVE sélectionné avant de terminer le cycle. Le caractère asynchrone du BUS CEI 821 autorise un ESCLAVE à prendre le temps qu'il nécessite pour répondre. Si un ESCLAVE ne répond pas à cause de quelque malfonction ou si le MAITRE adresse accidentellement un emplacement où il n'y a pas d'ESCLAVE, le LIMITEUR DE TEMPS D'OCCUPATION DU BUS intervient alors, permettant de terminer le cycle.

2.2 Lignes du bus de transfert de données

Les lignes du bus de transfert de données peuvent être groupées en trois catégories:

Lignes d'adresse	Lignes de données	Lignes de commande
A01-A31	D00-D31	AS*
AM0-AM5		DS0*
DS0*		DS1*
DS1*		BERR*
LWORD*		DTACK*
		WRITE*

OBSERVATION 2.1:

Les deux signaux de validation de donnée (DS0* et DS1*) ont une fonction double:

- 1) Les niveaux de ces deux lignes de signaux de validation de donnée sont utilisés pour sélectionner quel ou quels octets sont accédés.
- 2) Le front descendant du premier signal de validation de donnée est aussi utilisé comme un signal de contrôle du séquençement qui coordonne le transfert des données entre le MAITRE et l'ESCLAVE.

CHAPTER 2: IEC 821 BUS DATA TRANSFER BUS

2.1 Introduction

The IEC 821 BUS includes a high speed asynchronous parallel Data Transfer Bus (DTB). Figure 2-1, page 51, shows a typical IEC 821 BUS system, including all of the DTB functional module types. MASTERS use the DTB to select storage locations provided by SLAVES, and to transfer data to or from those locations. Some MASTERS and SLAVES use all of the DTB lines, while others use only a subset.

LOCATION MONITORS monitor data transfers between MASTERS and SLAVES. When an access is done to one of the byte location(s) that it monitors, a LOCATION MONITOR generates an on-board signal. For example, it might signal its on-board processor by means of an interrupt request. In such a configuration, if processor board A writes into a location of the global IEC 821 BUS memory that is monitored by processor B's LOCATION MONITOR, processor B will be interrupted.

After a MASTER initiates a data transfer cycle it waits for the responding SLAVE to respond before finishing the cycle. The asynchronous definition of the IEC 821 BUS allows a SLAVE to take as long as it needs to respond. If a SLAVE fails to respond because of some malfunction, or if the MASTER accidentally addresses a location where there is no SLAVE, the BUS TIMER intervenes, allowing the cycle to be terminated.

2.2 Data Transfer Bus lines

The Data Transfer Bus lines can be grouped into three categories:

Addressing lines	Data lines	Control lines
A01-A31 AM0-AM5 DS0* DS1* LWORD*	D00-D31	AS* DS0* DS1* BERR* DTACK* WRITE*

OBSERVATION 2.1:

The two data strobes (DS0* and DS1*) serve a dual function:

- 1) The levels of these two data strobe lines are used to select which byte(s) are accessed.
- 2) The edges of the data strobes are also used as timing signals which coordinate the transfer of the data between the MASTER and SLAVE.

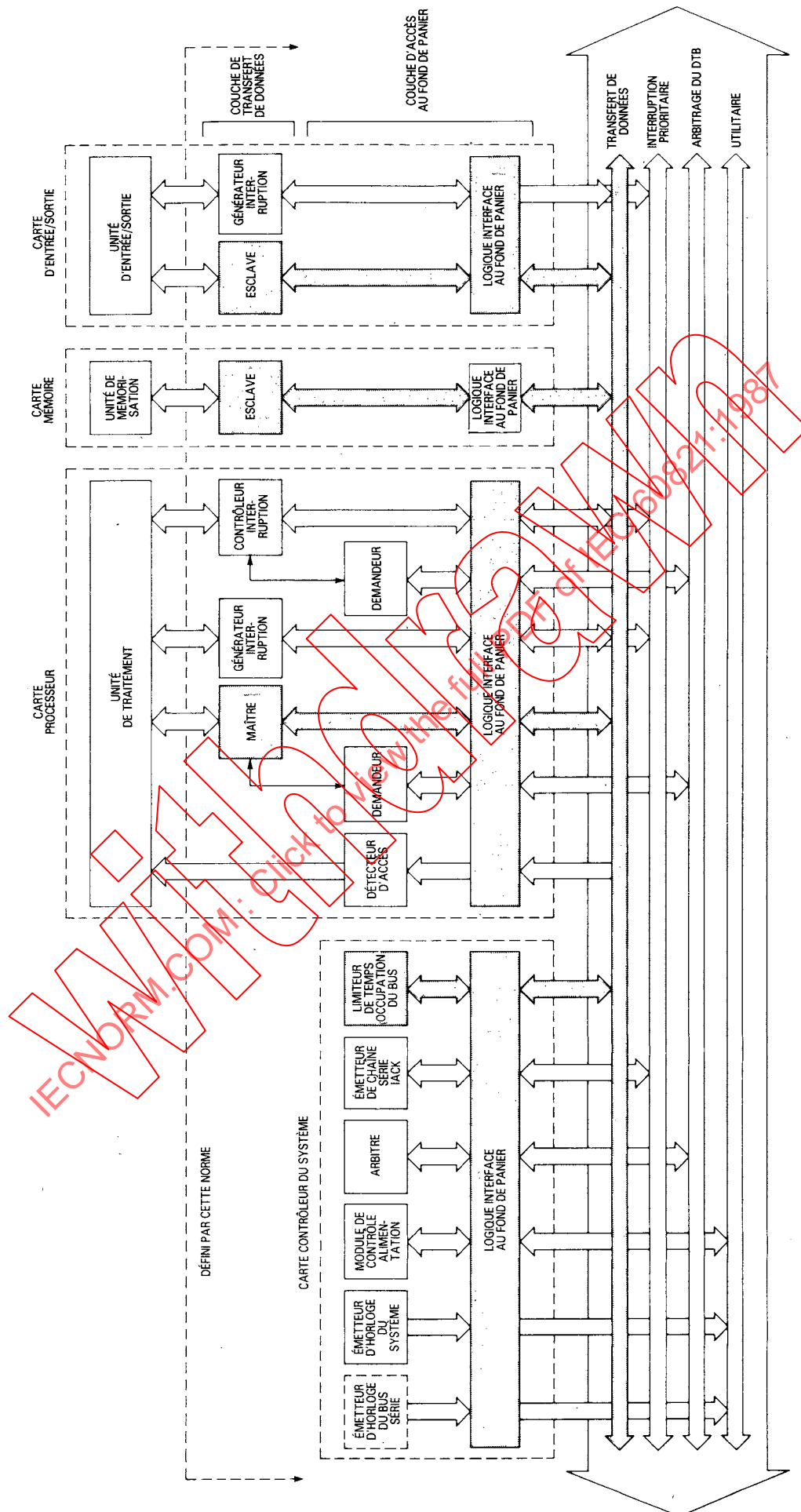


FIG. 2-1. - Schéma-bloc fonctionnel du bus de transfert de données.

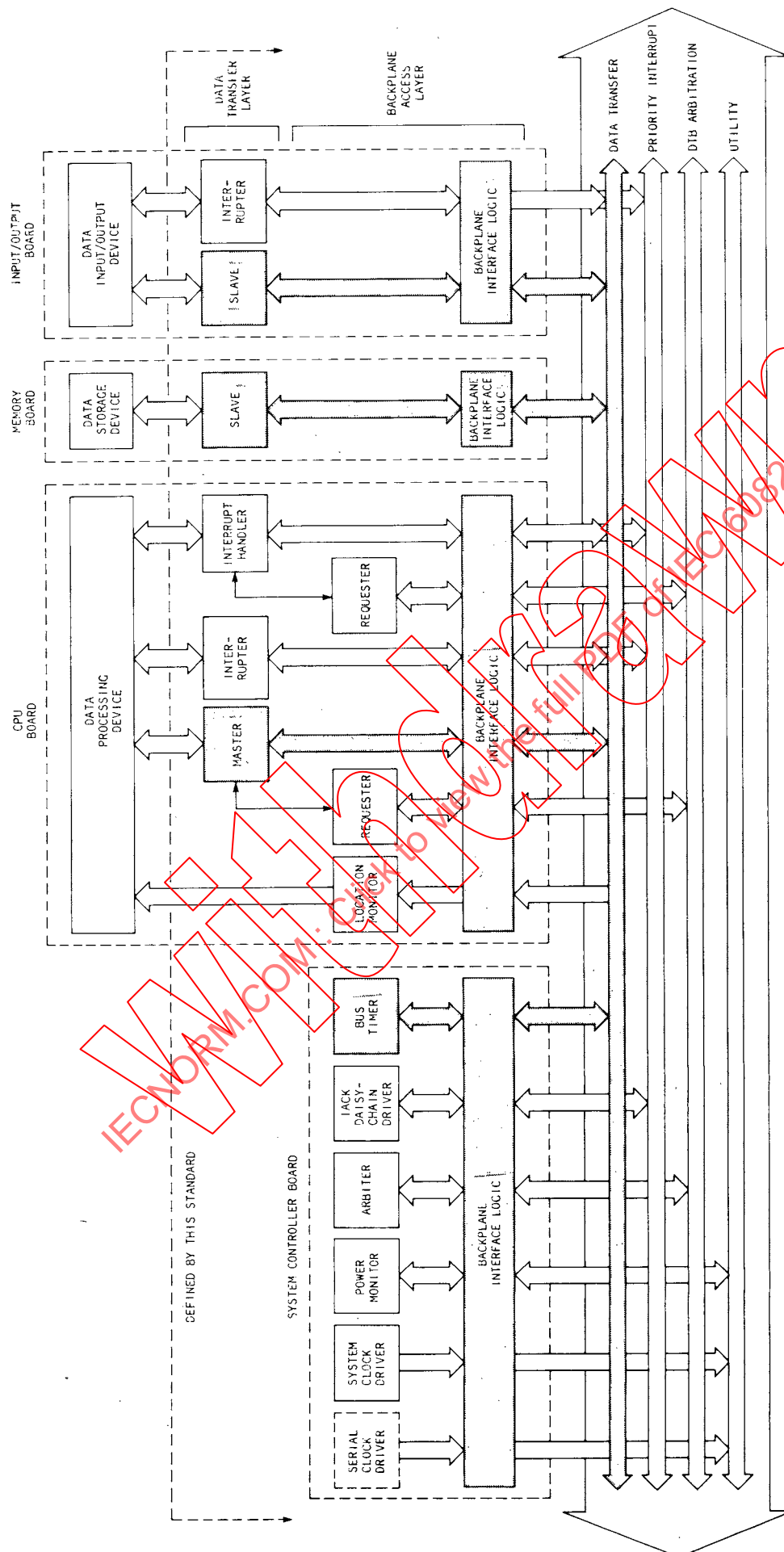


FIG. 2-1. - Data Transfer Bus functional block diagram.

2.2.1 Lignes d'adresse

La plus petite unité de mémoire adressable est l'octet. A chaque octet, est assignée une adresse binaire unique. Chaque octet peut être assigné à l'une des quatre catégories, en fonction des deux bits d'adresses de poids faible (voir tableau 2-1).

Tableau 2-1

Les quatre catégories d'emplacement d'octet

Catégorie	Emplacement d'octet
OCTET(0)	XXXXXX XXXXXX00
OCTET(1)	XXXXXX XXXXXX01
OCTET(2)	XXXXXX XXXXXX10
OCTET(3)	XXXXXX XXXXXX11

Un ensemble d'octets dont l'adresse diffère uniquement par les deux bits de poids faible est appelé groupe de 4 octets ou groupe OCTET(0-3). Certains ou tous les octets d'un groupe de 4 octets peuvent être accédés simultanément par un seul cycle du DTB.

Les MAITRES utilisent des lignes d'adresse A02-A31 pour sélectionner quel groupe de 4 octets sera accédé. Quatre lignes additionnelles (DS1*, DS0*, A01 et LWORD*) sont utilisées ensuite pour sélectionner le ou les octets parmi le groupe de 4 octets accédé pendant le transfert des données. En utilisant ces quatre lignes, un MAITRE peut accéder à 1, 2, 3 ou 4 octets à la fois, selon le type de cycle. Les 34 types de cycles possibles sont énumérés dans le tableau 2-2 avec les niveaux correspondants de ces quatre lignes.

OBSERVATION 2.2:

Dans les cycles où les deux signaux de validation de donnée sont commandés au niveau bas, l'un des signaux de validation peut passer au niveau bas légèrement après l'autre. Dans ce cas, les niveaux des signaux indiqués dans le tableau 2-2 sont les niveaux définitifs.

OBSERVATION 2.3:

Compte tenu des niveaux des quatre lignes de signaux indiqués dans le tableau 2-2, il y a 16 combinaisons possibles de ces niveaux. Parmi ces 16, il y a deux combinaisons illégales qui ne sont pas utilisées (voir tableau dans la REGLE 2.1).

REGLE 2.1:

Les MAITRES NE DOIVENT PAS générer de cycle DTB dans lequel les niveaux atteints par DS0*, DS1*, A01 et LWORD* correspondraient à l'une des combinaisons illégales suivantes:

DS1*	DS0*	A01	LWORD*
haut bas	bas haut	haut haut	bas bas

2.2.1 Addressing lines

The smallest addressable unit of storage is the byte location. Each byte location is assigned a unique binary address. Each byte location can be assigned to one of four categories, according to the least significant two bits of its address (see Table 2-1).

Table 2-1

The four categories of byte location

Category	Byte address
BYTE(0)	XXXXXX XXXXXX00
BYTE(1)	XXXXXX XXXXXX01
BYTE(2)	XXXXXX XXXXXX10
BYTE(3)	XXXXXX XXXXXX11

A set of byte locations, whose addresses differs only in the two least significant bits, is called a 4-byte group or a BYTE(0-3) group. Some, or all, of the bytes in a 4-byte group can be accessed simultaneously by a single DTB cycle.

MASTERS use address lines A02-A31 to select which 4-byte group will be accessed. Four additional lines (DS1*, DS0*, A01 and LWORD*) are then used to select which byte location(s) within the 4-byte group are accessed during the data transfer. Using these four lines, a MASTER can access 1, 2, 3, or 4 byte locations simultaneously, depending upon the type of cycle. The 34 possible cycle types are listed in Table 2-2 with the corresponding levels of these four lines.

OBSERVATION 2.2:

In cycles where both data strobes are driven low, one data strobe might go low slightly after the other. In this case, the signal levels indicated in Table 2-2 are the final levels.

OBSERVATION 2.3:

Given the four signal line levels shown in Table 2-2, there are 16 possible combinations of levels. Of these 16, there are two illegal combinations that are not used (see table in RULE 2.1).

RULE 2.1:

MASTERS MUST NOT generate DTB cycles where the final levels of DS0*, DS1*, A01, and LWORD* are either of the following illegal combinations:

DS1*	DS0*	A01	LWORD*
high low	low high	high high	low low

AUTORISATION 2.1:

Les MAITRES qui génèrent les cycles de LECTURE OCTET(1-2) ou ECRITURE OCTET(1-2) PEUVENT générer brièvement, comme état de transition, l'une des deux combinaisons décrites dans la REGLE 2.1 (c'est-à-dire lorsqu'un des signaux de validation de donnée est descendu et pas l'autre).

OBSERVATION 2.4:

Chaque fois qu'un MAITRE commande LWORD* au niveau bas et A01 au niveau haut, il commande les deux signaux de validation de donnée au niveau bas. (Toute autre combinaison est illégale.) Les concepteurs des cartes de BUS CEI 821 peuvent en profiter pour simplifier la logique des ESCLAVES.

AUTORISATION 2.2:

Pour simplifier la logique demandée, les ESCLAVES qui répondent aux cycles LECTURE OCTET(1-2) et ECRITURE OCTET(1-2) PEUVENT être conçus sans logique pour distinguer ces cycles des deux cycles illégaux décrits dans la REGLE 2.1.

Withdrawing
IEC NORM.COM: Click to view the full PDF of IEC 60821-87

PERMISSION 2.1:

MASTERS that generate BYTE(1-2) READ or BYTE(1-2) WRITE cycles MAY generate either of the two combinations described in RULE 2.1 briefly as transition states (i.e. while one data strobe has fallen, but the other has not).

OBSERVATION 2.4:

Whenever a MASTER drives LWORD* low and A01 high it drives both data strobes low. (Any other combination is illegal.) IEC 821 BUS board designers can take advantage of this to simplify the logic on SLAVES.

PERMISSION 2.2:

To simplify the required logic, SLAVES which respond to BYTE(1-2) READ and BYTE(1-2) WRITE cycles MAY be designed without logic to distinguish between these cycles and the two illegal cycles described in RULE 2.1.

Withdwn
IECNORM.COM: Click to view the full PDF of IEC 60821-1987

Tableau 2-2

Niveaux des signaux utilisés pour sélectionner le ou les octets auxquels on accède pendant un transfert de données

Type de cycle	DS1*	DS0*	A01	LWORD*
UNIQUEMENT D'ADRESSAGE	haut	haut	Note 1	
Transferts octet unique pair LECTURE ou ECRITURE OCTET(0) LECTURE ou ECRITURE OCTET(2)	bas bas	haut haut	bas haut	haut haut
Transferts octet unique impair LECTURE ou ECRITURE OCTET(1) LECTURE ou ECRITURE OCTET(3)	haut haut	bas bas	bas haut	haut haut
Transferts double octet LECTURE ou ECRITURE OCTET(0-1) LECTURE ou ECRITURE OCTET(2-3)	bas bas	bas bas	bas haut	haut haut
Transferts quadruple octet LECTURE ou ECRITURE OCTET(0-3)	bas	bas	bas	bas
Transferts de blocs par octet LECTURE ou ECRITURE DE BLOCS PAR OCTET	Note 2			haut
Transferts de blocs par double octet LECTURE ou ECRITURE DE BLOCS PAR DOUBLE OCTET	bas	bas	Note 3	haut
Transferts de blocs par quadruple octet LECTURE ou ECRITURE DE BLOCS PAR QUADRUPLE OCTET	bas	bas	bas	bas
Transferts octet unique RMW LECTURE-MODIFICATION-ECRITURE OCTET(0) LECTURE-MODIFICATION-ECRITURE OCTET(1) LECTURE-MODIFICATION-ECRITURE OCTET(2) LECTURE-MODIFICATION-ECRITURE OCTET(3)	bas haut bas haut	haut bas haut bas	bas bas haut haut	haut haut haut haut
Transferts double octet RMW LECTURE-MODIFICATION-ECRITURE OCTET(0-1) LECTURE-MODIFICATION-ECRITURE OCTET(2-3)	bas bas	bas bas	bas haut	haut haut
Transferts quadruple octet RMW LECTURE-MODIFICATION-ECRITURE OCTET(0-3)	bas	bas	bas	bas
Transferts non alignés LECTURE ou ECRITURE OCTET(0-2) LECTURE ou ECRITURE OCTET(1-3) LECTURE ou ECRITURE OCTET(1-2)	bas haut bas	haut bas bas	bas bas haut	bas bas bas

Notes:

- 1.- Pendant les cycles UNIQUEMENT D'ADRESSAGE, les deux signaux de validation de donnée sont maintenus au niveau haut, mais les lignes A01 et LWORD* peuvent être soit au niveau haut, soit au niveau bas.

Table 2-2

*Signal levels used to select which byte location(s)
are accessed during a data transfer*

Type of cycle	DS1*	DS0*	A01	LWORD*
ADDRESS-ONLY	high	high	Note 1	
Single even byte transfers BYTE(0) READ or WRITE BYTE(2) READ or WRITE	low low	high high	low high	high high
Single odd byte transfers BYTE(1) READ or WRITE BYTE(3) READ or WRITE	high high	low low	low high	high high
Double byte transfers BYTE(0-1) READ or WRITE BYTE(2-3) READ or WRITE	low low	low low	low high	high high
Quad byte transfers BYTE(0-3) READ or WRITE	low	low	low	low
Single byte block transfers SINGLE BYTE BLOCK READ or WRITE	Note 2			high
Double byte block transfers DOUBLE BYTE BLOCK READ or WRITE	low	low	Note 3	high
Quad byte block transfers QUAD BYTE BLOCK READ or WRITE	low	low	low	low
Single byte RMW transfers BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE	low high low high	high low high low	low low high high	high high high high
Double byte RMW transfers BYTE(0-1) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE	low low	low low	low high	high high
Quad byte RMW transfers BYTE(0-3) READ-MODIFY-WRITE	low	low	low	low
Unaligned transfers BYTE(0-2) READ or WRITE BYTE(1-3) READ or WRITE BYTE(1-2) READ or WRITE	low high low	high low low	low low high	low low low

Notes:

- 1.- During ADDRESS-ONLY cycles, both data strobes are maintained high, but the A01 and LWORD* lines might be either high or low.

Tableau 2-2 (suite)

- 2.- Pendant les transferts de blocs par octet, les deux signaux de validation de donnée sont activés alternativement au niveau bas. Chaque signal de validation de donnée peut être activé au niveau bas lors du premier transfert. Si le premier octet accédé est 1'OCTET(0) ou 1'OCTET(2), alors DS1* est activé en premier au niveau bas. Si le premier octet accédé est 1'OCTET(1) ou 1'OCTET(3), alors DS0* est activé en premier au niveau bas. A01 est valide seulement lors du premier transfert de données (c'est-à-dire jusqu'à ce que l'ESCLAVE active la première fois DTACK* ou BERR* au niveau bas) et peut être soit au niveau haut ou bas selon l'octet par lequel commence le transfert de blocs par octet. Si le premier octet est 1'OCTET(0) ou 1'OCTET(1), alors A01 est au niveau bas. Si le premier octet est 1'OCTET(2) ou 1'OCTET(3), alors A01 est au niveau haut.

Un exemple d'un cycle de transfert de blocs par octet commençant par 1'OCTET(2) est indiqué ci-dessous:

	DS1*	DS0*	A01	LWORD*
Premier transfert de données OCTET(2)	bas	haut	haut	haut
OCTET(3)	haut	bas	X	X
OCTET(0)	bas	haut	X	X
OCTET(1)	haut	bas	X	X
Dernier transfert de données OCTET(2)	bas	haut	X	X

X = haut ou bas.

- 3.- Pendant un transfert de blocs par double octet, les deux signaux de validation de donnée sont activés ensemble au niveau de chaque transfert de données. A01 est valide seulement lors du premier transfert de données (c'est-à-dire jusqu'à ce que l'ESCLAVE active la première fois DTACK* ou BERR* au niveau bas) et peut être soit au niveau haut ou bas selon le groupe double octet par lequel commence le transfert par bloc. Si le premier groupe double octet est 1'OCTET(0-1), alors A01 est au niveau bas. Si le premier groupe double octet est 1'OCTET(2-3), alors A01 est au niveau haut.

Un exemple d'un cycle de transfert de blocs par double octet commençant par 1'OCTET(2-3) est indiqué ci-dessous:

	DS1*	DS0*	A01	LWORD*
Premier transfert de données OCTET(2-3)	bas	bas	haut	haut
OCTET(0-1)	bas	bas	X	X
OCTET(2-3)	bas	bas	X	X
Dernier transfert de données OCTET(0-1)	bas	bas	X	X

X = haut ou bas.

2.2.2 Lignes de modification d'adresse

Il y a six lignes de modification d'adresse. Elles permettent au MAITRE de donner des informations binaires supplémentaires à l'ESCLAVE pendant les transferts de données. Le tableau 2-3 donne une liste des 64 codes modificateurs d'adresse possibles (AM) et classe chacun d'eux dans l'une des trois catégories:

Défini

Réservé

Défini par l'utilisateur

Les codes modificateurs d'adresse définis peuvent être classés en trois catégories:

Les codes AM d'adressage court indiquent que les lignes d'adresse A02-A15 sont utilisées pour sélectionner un groupe d'OCTET(0-3).

Les codes AM d'adressage standard indiquent que les lignes d'adresse A02-A23 sont utilisées pour sélectionner un groupe d'OCTET(0-3).

Les codes AM d'adressage étendu indiquent que les lignes d'adresse A02-A31 sont utilisées pour sélectionner un groupe d'OCTET(0-3).

Table 2-2 (continued)

- 2.- During single byte block transfers, the two data strobes are alternately driven low. Either data strobe might be driven low on the first transfer. If the first accessed byte location is BYTE(0) or BYTE(2), then DS1* is driven low first. If the first accessed byte location is BYTE(1) or BYTE(3), then DS0* is driven low first. A01 is valid only on the first data transfer (i.e. until the SLAVE drives DTACK* or BERR* low the first time) and might be either high or low depending upon which byte the single byte block transfer begins with. If the first byte location is BYTE(0) or BYTE(1), then A01 is low. If the first byte location is BYTE(2) or BYTE(3), then A01 is high.

An example of a single byte block transfer cycle which starts with BYTE(2) is given below:

	DS1*	DS0*	A01	LWORD*
First data transfer BYTE(2)	low	high	high	high
BYTE(3)	high	low	X	X
BYTE(0)	low	high	X	X
BYTE(1)	high	low	X	X
Last data transfer BYTE(2)	low	high	X	X

X = high or low.

- 3.- During a double byte block transfer, the two data strobes are both driven low on each data transfer. A01 is valid only on the first data transfer (i.e. until the SLAVE drives DTACK* or BERR* low the first time) and might be either high or low depending upon what double byte group the double byte block transfer begins with. If the first double byte group is BYTE(0-1), then A01 is low. If the first double byte group is BYTE(2-3), then A01 is high.

An example of a double byte block transfer cycle which starts with BYTE(2-3) is given below:

	DS1*	DS0*	A01	LWORD*
First data transfer BYTE(2-3)	low	low	high	high
BYTE(0-1)	low	low	X	X
BYTE(2-3)	low	low	X	X
Last data transfer BYTE(0-1)	low	low	X	X

X = high or low.

2.2.2 Address modifier lines

There are six address modifier lines. They allow the MASTER to pass additional binary information to the SLAVE during data transfers. Table 2-3 lists all of the 64 possible address modifier (AM) codes and classifies each into one of three categories:

Defined
Reserved
User defined

The defined address modifier codes can be further classified into three categories:

Short addressing AM codes indicate that address lines A02-A15 are being used to select a BYTE(0-3) group.

Standard addressing AM codes indicate that address lines A02-A23 are being used to select a BYTE(0-3) group.

Extended addressing AM codes indicate that address lines A02-A31 are being used to select a BYTE(0-3) group.

Tableau 2-3
Codes modificateurs d'adresse

HEX CODE	MODIFICATEUR D'ADRESSE						FONCTION
	5	4	3	2	1	0	
3F	H	H	H	H	H	H	Transfert standard en mode superviseur par bloc
3E	H	H	H	H	H	L	Accès standard en mode superviseur au programme
3D	H	H	H	H	L	H	Accès standard en mode superviseur aux données
3C	H	H	H	H	L	L	Réservé
3B	H	H	H	L	H	H	Transfert standard en mode non privilégié par bloc
3A	H	H	H	L	H	L	Transfert standard en mode non privilégié au programme
39	H	H	H	L	L	H	Accès standard en mode non privilégié aux données
38	H	H	H	L	L	L	Réservé
37	H	H	L	H	H	H	Réservé
36	H	H	L	H	H	L	Réservé
35	H	H	L	H	L	H	Réservé
34	H	H	L	H	L	L	Réservé
33	H	H	L	L	H	H	Réservé
32	H	H	L	L	H	L	Réservé
31	H	H	L	L	L	H	Réservé
30	H	H	L	L	L	L	Réservé
2F	H	L	H	H	H	H	Réservé
2E	H	L	H	H	H	L	Réservé
2D	H	L	H	H	L	H	Accès court en mode superviseur
2C	H	L	H	H	L	L	Réservé
2B	H	L	H	L	H	H	Réservé
2A	H	L	H	L	H	L	Réservé
29	H	L	H	L	L	H	Accès court en mode non privilégié
28	H	L	H	L	L	L	Réservé
27	H	L	L	H	H	H	Réservé
26	H	L	L	H	H	L	Réservé
25	H	L	L	H	L	H	Réservé
24	H	L	L	H	L	L	Réservé
23	H	L	L	L	H	H	Réservé
22	H	L	L	L	H	L	Réservé
21	H	L	L	L	L	H	Réservé
20	H	L	L	L	L	L	Réservé
1F	L	H	H	H	H	H	Défini par l'utilisateur
1E	L	H	H	H	H	L	Défini par l'utilisateur
1D	L	H	H	H	L	H	Défini par l'utilisateur
1C	L	H	H	H	L	L	Défini par l'utilisateur
1B	L	H	H	L	H	H	Défini par l'utilisateur
1A	L	H	H	L	H	L	Défini par l'utilisateur
19	L	H	H	L	L	H	Défini par l'utilisateur
18	L	H	H	L	L	L	Défini par l'utilisateur
17	L	H	L	H	H	H	Défini par l'utilisateur
16	L	H	L	H	H	L	Défini par l'utilisateur
15	L	H	L	H	L	H	Défini par l'utilisateur
14	L	H	L	H	L	L	Défini par l'utilisateur
13	L	H	L	L	H	H	Défini par l'utilisateur
12	L	H	L	L	H	L	Défini par l'utilisateur
11	L	H	L	L	L	H	Défini par l'utilisateur
10	L	H	L	L	L	L	Défini par l'utilisateur
0F	L	L	H	H	H	H	Transfert étendu en mode superviseur par bloc
0E	L	L	H	H	H	L	Accès étendu en mode superviseur au programme
0D	L	L	H	H	L	H	Accès étendu en mode superviseur aux données
0C	L	L	H	H	L	L	Réservé
0B	L	L	H	L	H	H	Transfert étendu en mode non privilégié par bloc
0A	L	L	H	L	H	L	Accès étendu en mode non privilégié au programme
09	L	L	H	L	L	H	Accès étendu en mode non privilégié aux données
08	L	L	H	L	L	L	Réservé
07	L	L	L	H	H	H	Réservé
06	L	L	L	H	H	L	Réservé
05	L	L	L	H	L	H	Réservé
04	L	L	L	H	L	L	Réservé
03	L	L	L	L	H	H	Réservé
02	L	L	L	L	H	L	Réservé
01	L	L	L	L	L	H	Réservé
00	L	L	L	L	L	L	Réservé

L = niveau bas du signal.
H = niveau haut du signal.

Table 2-3
Address modifier codes

HEX CODE	ADDRESS MODIFIER						FUNCTION
	5	4	3	2	1	0	
3F	H	H	H	H	H	H	Standard supervisory block transfer
3E	H	H	H	H	H	L	Standard supervisory program access
3D	H	H	H	H	L	H	Standard supervisory data access
3C	H	H	H	H	L	L	Reserved
3B	H	H	H	L	H	H	Standard non-privileged block transfer
3A	H	H	H	L	H	L	Standard non-privileged program access
39	H	H	H	L	L	H	Standard non-privileged data access
38	H	H	H	L	L	L	Reserved
37	H	H	L	H	H	H	Reserved
36	H	H	L	H	H	L	Reserved
35	H	H	L	H	L	H	Reserved
34	H	H	L	H	L	L	Reserved
33	H	H	L	L	H	H	Reserved
32	H	H	L	L	H	L	Reserved
31	H	H	L	L	L	H	Reserved
30	H	H	L	L	L	L	Reserved
2F	H	L	H	H	H	H	Reserved
2E	H	L	H	H	H	L	Reserved
2D	H	L	H	H	L	H	Short supervisory access
2C	H	L	H	H	L	L	Reserved
2B	H	L	H	L	H	H	Reserved
2A	H	L	H	L	H	L	Reserved
29	H	L	H	L	L	H	Short non-privileged access
28	H	L	H	L	L	L	Reserved
27	H	L	L	H	H	H	Reserved
26	H	L	L	H	H	L	Reserved
25	H	L	L	H	L	H	Reserved
24	H	L	L	H	L	L	Reserved
23	H	L	L	L	H	H	Reserved
22	H	L	L	L	H	L	Reserved
21	H	L	L	L	L	H	Reserved
20	H	L	L	L	L	L	Reserved
1F	L	H	H	H	H	H	User defined
1E	L	H	H	H	H	L	User defined
1D	L	H	H	H	L	H	User defined
1C	L	H	H	H	L	L	User defined
1B	L	H	H	L	H	H	User defined
1A	L	H	H	L	H	L	User defined
19	L	H	H	L	L	H	User defined
18	L	H	H	L	L	L	User defined
17	L	H	L	H	H	H	User defined
16	L	H	L	H	H	L	User defined
15	L	H	L	H	L	H	User defined
14	L	H	L	H	L	L	User defined
13	L	H	L	L	H	H	User defined
12	L	H	L	L	H	L	User defined
11	L	H	L	L	L	H	User defined
10	L	H	L	L	L	L	User defined
0F	L	L	H	H	H	H	Extended supervisory block transfer
0E	L	L	H	H	H	L	Extended supervisory program access
0D	L	L	H	H	L	H	Extended supervisory data access
0C	L	L	H	H	L	L	Reserved
0B	L	L	H	L	H	H	Extended non-privileged block transfer
0A	L	L	H	L	H	L	Extended non-privileged program access
09	L	L	H	L	L	H	Extended non-privileged data access
08	L	L	H	L	L	L	Reserved
07	L	L	L	H	H	H	Reserved
06	L	L	L	H	H	L	Reserved
05	L	L	L	H	L	H	Reserved
04	L	L	L	H	L	L	Reserved
03	L	L	L	L	H	H	Reserved
02	L	L	L	L	H	L	Reserved
01	L	L	L	L	L	H	Reserved
00	L	L	L	L	L	L	Reserved

L = Low signal level.
H = High signal level.

REGLE 2.2:

Excepté les codes "définis par l'utilisateur", les codes définis dans le tableau 2-3 NE DOIVENT PAS être utilisés pour d'autres buts que ceux spécifiés.

REGLE 2.3:

Les cartes ESCLAVES de BUS CEI 821 ne DOIVENT PAS répondre aux codes réservés modificateurs d'adresse.

OBSERVATION 2.5:

Les codes modificateurs d'adresse "réservés" le sont pour des améliorations futures. Si des cartes ESCLAVES répondent à ces codes, il peut résulter des incompatibilités futures lorsque l'usage de ces codes sera défini.

AUTORISATION 2.3:

Les codes "définis par l'utilisateur" PEUVENT être utilisés dans n'importe quel but par les fournisseurs ou les utilisateurs de cartes (changement de page, protection de la mémoire, identification du MAITRE ou de la tâche, accès privilégié aux ressources, etc.).

RECOMMANDATION 2.1:

Permettre aux utilisateurs du BUS CEI 821 de prévoir l'utilisation des codes modificateurs d'adresse "définis par l'utilisateur" selon leurs propres besoins, les décoder d'une manière flexible sur les cartes ESCLAVES. Les utilisateurs peuvent ensuite configurer la carte pour n'importe quel décodage exigé par le système.

OBSERVATION 2.6:

Les PROM-s et FPLA-s fournissent une flexibilité pour le décodage des codes modificateurs d'adresse.

SUGGESTION 2.1:

Quand les ESCLAVES ont un dispositif programmé (par exemple une PROM ou un FPLA) installé sur un support, programmer le dispositif pour que l'ESCLAVE réponde aux codes AM suivants:

ESCLAVES A16 avec possibilité

D08(O), D08(E0), D16 ou D32: 29, 2D

ESCLAVES A24 avec possibilité

D08(O), D08(E0), D16 ou D32: 39, 3A, 3D et 3E

ESCLAVES A32 avec possibilité

D08(O), D08(E0), D16 ou D32: 09, 0A, 0D et 0E

ESCLAVES A24 avec possibilité BLT:

3B, 3F

ESCLAVES A32 avec possibilité BLT:

0B, 0F

Les mnémoniques A16, A24 et A32 sont définis dans le tableau 2-9.

Les mnémoniques D08(O), D08(E0), D16, D32 et BLT sont définis dans les tableaux 2-10 et 2-11.

2.2.3 Lignes de données

Les systèmes de BUS CEI 821 peuvent être construits avec une configuration de fond de panier de 16 lignes de données (D00-D15) ou 32 lignes de données (D00-D31). Les configurations de fond de panier ayant 16 lignes de données permettent à un MAITRE d'accéder seulement à deux octets simultanément, alors que celles ayant 32 lignes de données permettent un accès jusqu'à quatre octets. Lorsque le MAITRE a adressé des emplacements de 1, 2, 3 ou 4 octets en utilisant la

RULE 2.2:

Except for the user defined codes, the codes defined in Table 2-3 **MUST NOT** be used for purposes other than those specified.

RULE 2.3:

IEC 821 BUS SLAVE boards **MUST NOT** respond to reserved address modifier codes.

OBSERVATION 2.5:

Reserved address modifier codes are for future enhancements. If SLAVE boards respond to these codes, incompatibilities might result at some future date, when usage of these codes is defined.

PERMISSION 2.3:

User defined codes **MAY** be used for any purpose which board suppliers or board users deem appropriate (page switching, memory protection, MASTER or task identification, privileged access to resources, etc).

RECOMMENDATION 2.1:

To allow IEC 821 BUS users to tailor the use of the user defined address modifier codes to their own needs, decode them in a flexible way on SLAVE boards. Users can then configure the board to give any decoding required by their system.

OBSERVATION 2.6:

Socketed PROMS and FPLAS provide a flexible way for decoding the address modifier codes.

SUGGESTION 2.1:

Where SLAVES are manufactured with a programmed device (e.g. a PROM or FPLA) installed in the socket, program the device so that the SLAVE responds to the following AM codes:

A16 SLAVES with D08(O), D08(E0), D16, or D32 capability:	29, 2D
A24 SLAVES with D08(O), D08(E0), D16, or D32 capability:	39, 3A, 3D, and 3E
A32 SLAVES with D08(O), D08(E0), D16, or D32 capability:	09, 0A, 0D, and 0E
A24 SLAVES with BLT capability:	3B, 3F
A32 SLAVES with BLT capability:	0B, 0F

The mnemonics A16, A24 and A32 are defined in Table 2-9.

The mnemonics D08(O), D08(E0), D16, D32, and BLT are defined in Tables 2-10 and 2-11.

2.2.3 Data lines

IEC 821 BUS systems can be built with a backplane configuration that provides either 16 data lines (D00-D15), or 32 data lines (D00-D31). Backplane configurations that provide 16 data lines allow a MASTER to access only two byte locations simultaneously, while those with 32 data lines allow up to four byte locations to be accessed. When the MASTER has selected 1, 2, 3, or 4 byte locations, using the method described in

méthode décrite au paragraphe 2.2.1, il peut transférer des données binaires sur le bus de données entre lui-même et ces adresses. Le tableau 2-4 montre comment les lignes de données sont utilisées pour transférer des données pendant chacun des 34 types de cycle.

AUTORISATION 2.4:

L'émetteur de données (le MAITRE au cours d'un cycle écriture, l'ESCLAVE au cours d'un cycle lecture) PEUT commander les lignes de données qui ne sont pas utilisées pour transférer les données.

Tableau 2-4

Utilisation des lignes de données pour transférer des données pendant chacun des 34 types de cycle

Pendant les types de cycles suivants	Les lignes de données sont utilisées pour transférer les données comme indiqué ci-dessous			
	D24-D31	D16-D23	D08-D15	D00-D07
UNIQUEMENT D'ADRESSAGE	Pas d'octets transférés			
Transferts octet unique pair LECTURE ou ECRITURE OCTET(0) LECTURE ou ECRITURE OCTET(2)			OCTET(0) OCTET(2)	
Transferts octet unique impair LECTURE ou ECRITURE OCTET(1) LECTURE ou ECRITURE OCTET(3)				OCTET(1) OCTET(3)
Transferts double octet LECTURE ou ECRITURE OCTET(0-1) LECTURE ou ECRITURE OCTET(2-3)			OCTET(0) OCTET(2)	OCTET(1) OCTET(3)
Transferts quadruple octet LECTURE ou ECRITURE OCTET(0-3)	OCTET(0)	OCTET(1)	OCTET(2)	OCTET(3)
Transferts de blocs par octet LECTURE ou ECRITURE DE BLOCS PAR OCTET			Note 1	
Transferts de blocs par double octet LECTURE ou ECRITURE DE BLOCS PAR DOUBLE OCTET			Note 2	
Transferts de blocs par quadruple octet LECTURE ou ECRITURE DE BLOCS PAR QUADRUPLE OCTET	OCTET(0)	OCTET(1)	OCTET(2)	OCTET(3)

Paragraph 2.2.1, it can transfer binary data between itself and those locations over the data bus. Table 2-4 shows how the data lines are used to move data during each of the 34 cycle types.

PERMISSION 2.4:

The data sender (MASTER for a write cycle; SLAVE for a read cycle) MAY drive data lines which are not used to transfer data.

Table 2-4

Use of the data lines to move data during each of the 34 cycle types

During the following types of cycles	The data lines are used to transfer data as shown below:			
	D24-D31	D16-D23	D08-D15	D00-D07
ADDRESS ONLY	No bytes transferred			
Single even byte transfers BYTE(0) READ or WRITE BYTE(2) READ or WRITE			BYTE(0) BYTE(2)	
Single odd byte transfers BYTE(1) READ or WRITE BYTE(3) READ or WRITE				BYTE(1) BYTE(3)
Double byte transfers BYTE(0-1) READ or WRITE BYTE(2-3) READ or WRITE			BYTE(0) BYTE(2)	BYTE(1) BYTE(3)
Quad byte transfers BYTE(0-3) READ or WRITE	BYTE(0)	BYTE(1)	BYTE(2)	BYTE(3)
Single byte block transfers SINGLE BYTE BLOCK READ or WRITE			Note 1	
Double byte block transfers DOUBLE BYTE BLOCK READ or WRITE			Note 2	
Quad byte block transfers QUAD BYTE BLOCK READ or WRITE	BYTE(0)	BYTE(1)	BYTE(2)	BYTE(3)

Tableau 2-4 (suite)

Pendant les types de cycles suivants	Les lignes de données sont utilisées pour transférer les données comme indiqué ci-dessous			
	D24-D31	D16-D23	D08-D15	D00-D07
Transferts octet unique RMW LECTURE-MODIFICATION-ECRITURE OCTET(0) LECTURE-MODIFICATION-ECRITURE OCTET(1) LECTURE-MODIFICATION-ECRITURE OCTET(2) LECTURE-MODIFICATION-ECRITURE OCTET(3)			OCTET(0) OCTET(2)	OCTET(1) OCTET(3)
Transferts double octet RMW LECTURE-MODIFICATION-ECRITURE OCTET(0-1) LECTURE-MODIFICATION-ECRITURE OCTET(2-3)			OCTET(0) OCTET(2)	OCTET(1) OCTET(3)
Transferts quadruple octet RMW LECTURE-MODIFICATION-ECRITURE OCTET(0-3)	OCTET(0)	OCTET(1)	OCTET(2)	OCTET(3)
Transferts non alignés LECTURE ou ECRITURE OCTET (0-2) LECTURE ou ECRITURE OCTET (1-3) LECTURE ou ECRITURE OCTET (1-2)	OCTET(0)	OCTET(1) OCTET(1) OCTET(1)	OCTET(2) OCTET(2) OCTET(2)	OCTET(3)

Notes:

- 1.- Pendant les transferts de blocs par octet, les données sont transférées sur D00-D07 ou D08-D15, 8 bits à la fois. Un tel exemple est indiqué ci-dessous.

	D24-D31	D16-D23	D08-D15	D00-D07
Premier transfert de données			OCTET(2) OCTET(0) OCTET(2)	OCTET(1) OCTET(3) OCTET(1)
Dernier transfert de données				OCTET(3)

- 2.- Pendant un transfert de blocs par double octet, les données sont transférées sur D00-D15, 16 bits à la fois. Un tel exemple est indiqué ci-dessous.

	D24-D31	D16-D23	D08-D15	D00-D07
Premier transfert de données			OCTET(2) OCTET(0) OCTET(2) OCTET(0) OCTET(2) OCTET(0)	OCTET(3) OCTET(1) OCTET(3) OCTET(1) OCTET(3) OCTET(1)
Dernier transfert de données				OCTET(3) OCTET(1)

Table 2-4 (continued)

During the following types of cycles	The data lines are used to transfer data as shown below:			
	D24-D31	D16-D23	D08-D15	D00-D07
Single byte RMW transfers BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE			BYTE(0) BYTE(2)	BYTE(1) BYTE(3)
Double byte RMW transfers BYTE(0-1) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE			BYTE(0) BYTE(2)	BYTE(1) BYTE(3)
Quad byte RMW transfers BYTE(0-3) READ-MODIFY-WRITE	BYTE(0)	BYTE(1)	BYTE(2)	BYTE(3)
Unaligned transfers BYTE(0-2) READ or WRITE BYTE(1-3) READ or WRITE BYTE(1-2) READ or WRITE	BYTE(0)	BYTE(1) BYTE(1) BYTE(1)	BYTE(2) BYTE(2) BYTE(2)	BYTE(3)

Notes:

- 1.- During single byte block transfers, data is transferred 8 bits at a time over D00-D07 or D08-D15. One example of this is given below:

	D24-D31	D16-D23	D08-D15	D00-D07
First data transfer			BYTE(2)	BYTE(1)
			BYTE(0)	BYTE(3)
			BYTE(2)	BYTE(1)
Last data transfer				BYTE(3)

- 2.- During a double byte block transfer, data is transferred 16 bits at a time over D00-D15. One example of this is given below:

	D24-D31	D16-D23	D08-D15	D00-D07
First data transfer			BYTE(2) BYTE(0)	BYTE(3) BYTE(1)
			BYTE(2) BYTE(0)	BYTE(3) BYTE(1)
			BYTE(2) BYTE(0)	BYTE(3) BYTE(1)
Last data transfer			BYTE(2) BYTE(0)	BYTE(3) BYTE(1)

2.2.4 Lignes de commande du bus de transfert de données

Les lignes de signaux suivantes sont utilisées pour contrôler le déplacement des données sur les lignes de transfert de données:

AS*	Validation d'adresse
DS0*	Validation de donnée Zéro
DS1*	Validation de donnée Un
BERR*	Erreur du bus
DTACK*	Accusé de réception du transfert de données
WRITE*	Lecture/Ecriture

2.2.4.1 AS*

Un front descendant sur cette ligne avertit tous les modules ESCLAVES que l'adresse est stable et peut être mémorisée.

2.2.4.2 DS0* et DS1*

En plus de leur fonction de sélection des emplacements d'octets pour le transfert de données comme décrit au paragraphe 2.2.1, les signaux de validation de donnée ont aussi des fonctions additionnelles. Pendant les cycles d'écriture, le premier front descendant du signal de validation de donnée indique que le MAÎTRE a placé une donnée valide sur le bus de données. Pendant les cycles de lecture, le premier front montant avertit l'ESCLAVE qu'il peut retirer la donnée valide du bus de données.

OBSERVATION 2.7:

Il n'est pas permis aux MAÎTRES du BUS CEI 821 de commander l'un des signaux de validation de donnée au niveau bas avant d'avoir commandé AS* au niveau bas. Cependant, dû au fait que AS* peut être plus chargé que les signaux de validation de donnée sur le fond de panier, les ESCLAVES et les DETECTEURS D'ACCES peuvent détecter un front descendant d'un signal de validation de donnée avant de détecter le front descendant de AS*.

AUTORISATION 2.5:

Les ESCLAVES et les DETECTEURS D'ACCES du BUS CEI 821 PEUVENT être conçus pour mémoriser l'adresse quand ils détectent un front descendant d'un signal de validation de donnée au lieu du front descendant de AS*.

OBSERVATION 2.8:

Les ESCLAVES et les DETECTEURS D'ACCES du BUS CEI 821 qui mémorisent l'adresse sur le front descendant des signaux de validation de donnée n'ont pas besoin de surveiller AS*.

OBSERVATION 2.9:

Afin de bénéficier des avantages de l'anticipation des adresses comme décrit dans le paragraphe 2.4.3 ou d'exécuter des cycles de lecture ou d'écriture par bloc, il est nécessaire qu'un ESCLAVE mémorise l'adresse sur le front descendant de AS*.

2.2.4.3 DTACK*

DTACK* est commandé au niveau bas par l'ESCLAVE pour indiquer que les données ont été reçues avec succès sur un cycle écriture. Sur un cycle lecture, l'ESCLAVE commande DTACK* au niveau bas pour indiquer qu'il a placé les données sur le bus de données.

2.2.4 Data Transfer Bus control lines

The following signal lines are used to control the movement of data over the data transfer lines:

AS*	Address Strobe
DS0*	Data Strobe Zero
DS1*	Data Strobe One
BERR*	Bus Error
DTACK*	Data Transfer Acknowledge
WRITE*	Read/Write

2.2.4.1 AS*

A falling edge on this line informs all SLAVE modules that the address is stable and can be captured.

2.2.4.2 DS0* and DS1*

In addition to their function in selecting byte locations for data transfer, as described in Paragraph 2.2.1, the data strobes also serve additional functions. On write cycles, the first data strobe falling edge indicates when the MASTER has placed valid data on the data bus. On read cycles, the first rising edge tells the SLAVE when it can remove valid data from the data bus.

OBSERVATION 2.7:

IEC 821 BUS MASTERS are not permitted to drive either of the data strobes low before driving AS* low. However, due to the fact that AS* might be more heavily loaded on the backplane than the data strobes, SLAVES and LOCATION MONITORS might detect a falling edge on a data strobe, before they detect the falling edge on AS*.

PERMISSION 2.5:

IEC 821 BUS SLAVES and LOCATION MONITORS MAY be designed to capture the address when they detect a falling edge on a data strobe instead of on the falling edge of AS*.

OBSERVATION 2.8:

IEC 821 BUS SLAVES and LOCATION MONITORS that capture the address on the falling edge of the data strobe(s) need not monitor AS*.

OBSERVATION 2.9:

In order to take full advantage of address pipelining as described in Paragraph 2.4.2, or to perform block read and write cycles, a SLAVE should capture the address on the falling edge of AS*.

2.2.4.3 DTACK*

The SLAVE drives DTACK* low to indicate that it has successfully received the data on a write cycle. On a read cycle, the SLAVE drives DTACK* low to indicate that it has placed data on the data bus.

2.2.4.4 BERR*

BERR* est commandé au niveau bas par l'ESCLAVE ou le LIMITEUR DE TEMPS D'OCCUPATION DU BUS pour indiquer au MAÎTRE que le transfert de données n'a pas été réussi. Par exemple, si un MAÎTRE essaie d'écrire dans une zone de mémoire qui contient une mémoire morte, l'ESCLAVE qui répond doit activer BERR*. Si un MAÎTRE essaie d'accéder à une zone de la partition mémoire qui n'est pas occupée par un ESCLAVE, le LIMITEUR DE TEMPS D'OCCUPATION DU BUS devra activer BERR* à l'état bas après avoir attendu durant le temps spécifié.

OBSERVATION 2.10:

La ligne BERR* est un moyen utile pour la mise au point des systèmes de BUS CEI 821. Elle permet aussi la détection rapide d'une défaillance du système au cours d'une opération normale. Cette possibilité n'est pas nécessaire pour tous les systèmes de BUS CEI 821.

AUTORISATION 2.6:

Des ESCLAVES du BUS CEI 821 PEUVENT être conçus sans un émetteur sur BERR*.

SUGGESTION 2.2:

Concevoir des ESCLAVES répondant avec un front descendant sur BERR* dans les situations suivantes:

Quand on demande à un ESCLAVE D08(O), D08(E0) ou D16 d'effectuer un cycle quadruple octet.

Quand on demande à un ESCLAVE D08(O), D08(E0) d'effectuer un cycle double octet.

Quand on demande à un ESCLAVE non UAT d'effectuer un transfert non aligné (c'est-à-dire un transfert trois octets ou deux octets (1-2)).

Quand un ESCLAVE détecte une erreur non récupérable dans les données qu'il extrait de sa mémoire interne pendant un cycle de lecture.

Les mnémoniques D08(O), D08(E0), D16 et UAT sont définis dans les tableaux 2-10 et 2-15.

REGLE 2.4:

Les ESCLAVES D08(O), D08(E0) et D16 NE DOIVENT PAS générer une réponse avec un front descendant sur DTACK* au cours d'un cycle quadruple octet s'ils n'ont pas la possibilité quadruple octet.

REGLE 2.5:

Les ESCLAVES D08(O), D08(E0) NE DOIVENT PAS générer une réponse avec un front descendant sur DTACK* au cours d'un cycle double octet s'ils n'ont pas la possibilité double octet.

REGLE 2.6:

Un ESCLAVE NE DOIT PAS générer une réponse avec un front descendant sur DTACK* au cours d'un cycle de transfert non aligné s'il n'a pas la possibilité UAT.

2.2.4.4 BERR*

BERR* is driven low by the SLAVE or by the BUS TIMER to indicate to the MASTER that the data transfer was unsuccessful. For example, if a MASTER tries to write to a location which contains Read-Only Memory, the responding SLAVE might drive BERR* low. If the MASTER tries to access a location that is not provided by any SLAVE, the BUS TIMER would drive BERR* low after waiting a specified period.

OBSERVATION 2.10:

The BERR* line is a convenience which is useful when debugging IEC 821 BUS systems. It also allows system failures to be detected quickly during normal operation. Not all IEC 821 BUS systems will need this capability.

PERMISSION 2.6:

IEC 821 BUS SLAVES MAY be designed without a driver for BERR*.

SUGGESTION 2.2:

Design SLAVES to respond with a falling edge on BERR* in the following situations:

When a D08(O), D08(E0), or D16 SLAVE is requested to do a quad byte cycle.

When a D08(O) or D08(E0) SLAVE is requested to do a double byte cycle.

When a non-UAT SLAVE is requested to do an unaligned transfer (i.e. a triple byte transfer or a double byte BYTE(1-2) transfer).

When a SLAVE detects an uncorrectable error in the data it retrieves from its internal storage during a read cycle.

The mnemonics D08(O), D08(E0), D16, and UAT are defined in Tables 2-10 and 2-15.

RULE 2.4:

D08(O), D08(E0), and D16 SLAVES MUST NOT respond with a falling edge on DTACK* during a quad byte cycle if they do not have quad byte capability.

RULE 2.5:

D08(O) and D08(E0) SLAVES MUST NOT respond with a falling edge on DTACK* during a double byte cycle if they do not have double byte capability.

RULE 2.6:

A SLAVE MUST NOT respond with a falling edge on DTACK* during an unaligned transfer cycle, if it does not have UAT capability.

2.2.4.5 WRITE*

La ligne WRITE* est une ligne à niveau significatif échantillonné par le front descendant du premier signal de validation de donnée. Elle est utilisée par le MAITRE pour indiquer le sens des opérations de transfert de données. Lorsque WRITE* est commandé au niveau bas, le sens du transfert de données est du MAITRE vers l'ESCLAVE.

2.3 Modules DTB - Description générale

Le protocole DTB définit 33 types de cycles différents en plus du CYCLE UNIQUEMENT D'ADRESSAGE qui peuvent être utilisés pour transférer des données. Chacun de ces 34 types de cycles peuvent être utilisés dans n'importe lequel des trois modes d'adressage: court (16 bits), standard (24 bits) et étendu (32 bits). Les possibilités des MAITRES, ESCLAVES et DETECTEURS D'ACCES sont décrites par une liste de mnémoniques montrant les types de cycles qu'ils peuvent respectivement générer, accepter ou contrôler (cela est décrit avec plus de détails par la suite dans ce chapitre).

Les paragraphes 2.3.1 à 2.3.4 fournissent les schémas-blocs des quatre types de modules fonctionnels du DTB: MAITRE, ESCLAVE, DETECTEUR D'ACCES et LIMITEUR DE TEMPS D'OCCUPATION DU BUS.

REGLE 2.7:

Les lignes de signaux de sortie indiquées en lignes continues dans les figures 2-2 à 2-5, pages 74 à 86, DOIVENT être contrôlées par le module, sinon il les commandera toujours au niveau haut.

OBSERVATION 2.11:

SI une ligne de signal de sortie n'est pas contrôlée,
ALORS des adaptateurs sur le fond de panier assurent qu'elle soit au niveau haut.

REGLE 2.8:

Les lignes de signaux d'entrées indiquées en lignes continues dans les figures 2-2 à 2-5 DOIVENT être surveillées et recevoir des réponses de façon appropriée.

OBSERVATION 2.12:

Des REGLES et des AUTORISATIONS pour commander et surveiller les lignes de signaux indiquées en pointillé dans les figures 2-2 à 2-5 sont données dans les tableaux 2-5, 2-6 et 2-8.

2.3.1 MAITRE

Le schéma-bloc du MAITRE est indiqué dans la figure 2-2. Les lignes pointillées montrent les signaux dont l'utilisation varie selon les différents types de MAITRES. Le tableau 2-5 définit de quelle manière les différents types de MAITRES utilisent ces lignes. D'autres informations sur la manière avec laquelle les MAITRES commandent les lignes d'adresse, les lignes de données, LWORD*, DS0* et DS1* sont indiquées dans les tableaux 2-20, 2-21 et 2-22.

2.2.4.5 WRITE*

WRITE* is a level significant signal line that is strobed by the falling edge of the first data strobe driven low. It is used by the MASTER to indicate the direction of the data transfer operations. When WRITE* is driven low the data transfer direction is from the MASTER to the SLAVE.

2.3 DTB modules - Basic description

In addition to the ADDRESS-ONLY CYCLE, the DTB protocol defines 33 different cycle types that can be used to transfer data. Each of these 34 cycle types can be used in any of three addressing modes: short (16 bits), standard (24 bits), and extended (32 bits). The capabilities of MASTERS, SLAVES and LOCATION MONITORS are described by a list of mnemonics that show what cycle types they can generate, accept, or monitor respectively. (This will be described in more detail later in this chapter.)

Paragraphs 2.3.1 through 2.3.4 provide block diagrams for the four types of DTB functional modules: MASTER, SLAVE, LOCATION MONITOR, and BUS TIMER.

RULE 2.7:

Output signal lines shown with solid lines in Figures 2-2 through 2-5, pages 75 to 87, MUST be driven by the module, unless it would always drive them high.

OBSERVATION 2.11:

IF an output signal line is not driven,
THEN terminators on the backplane ensure that it is high.

RULE 2.8:

Input signal lines shown with solid lines in Figures 2-2 through 2-5 MUST be monitored and responded to in the appropriate fashion.

OBSERVATION 2.12:

RULES and PERMISSIONS for driving and monitoring signal lines shown with dotted lines in Figures 2-2 through 2-5 are given in Tables 2-5, 2-6, and 2-8.

2.3.1 MASTER

The block diagram of the MASTER is shown in Figure 2-2. The dotted lines in the diagram show signals whose use varies among the various types of MASTERS. Table 2-5 specifies how the various types of MASTERS use these lines. Further information about how the various types of MASTERS drive the address lines, the data lines, LWORD*, DS0*, and DS1* is given in Tables 2-20, 2-21, 2-22.

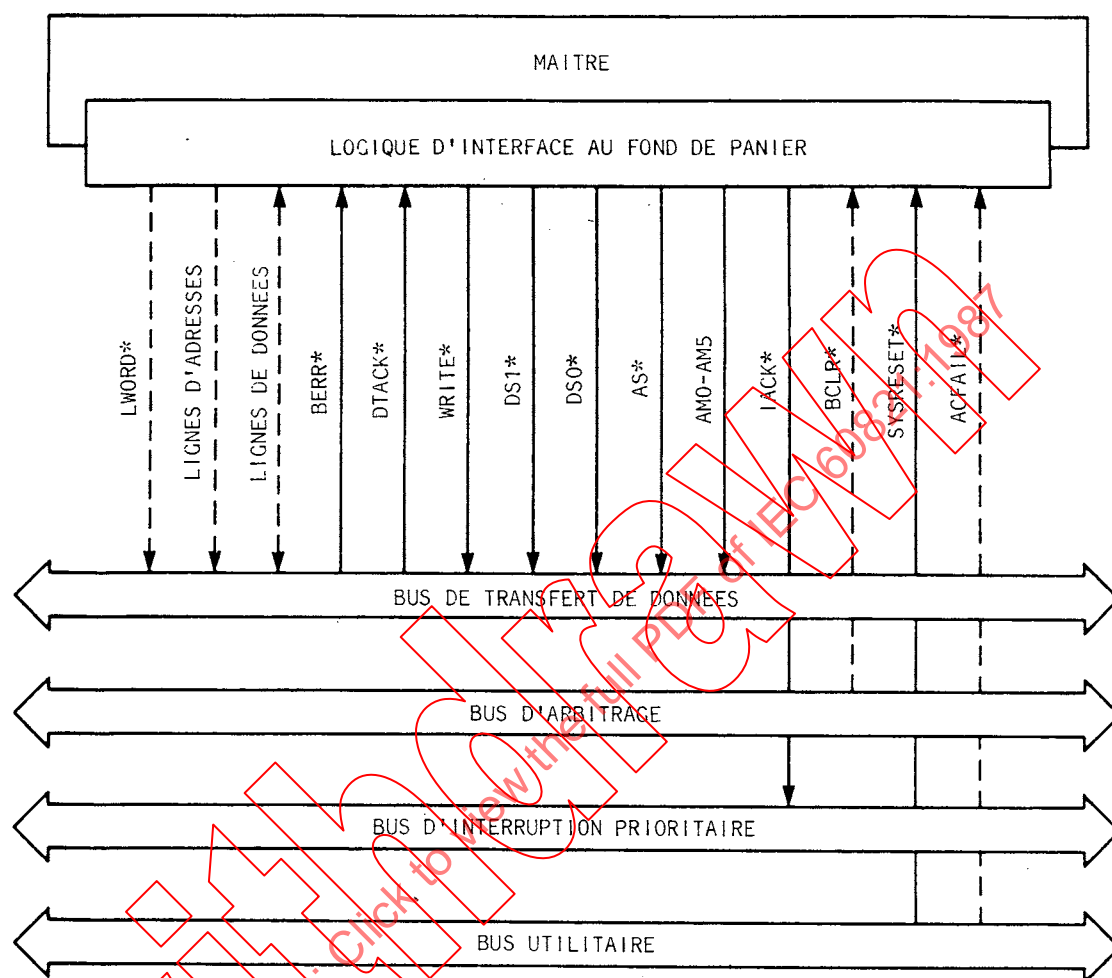


FIG. 2-2. - Schéma-bloc: MAITRE.

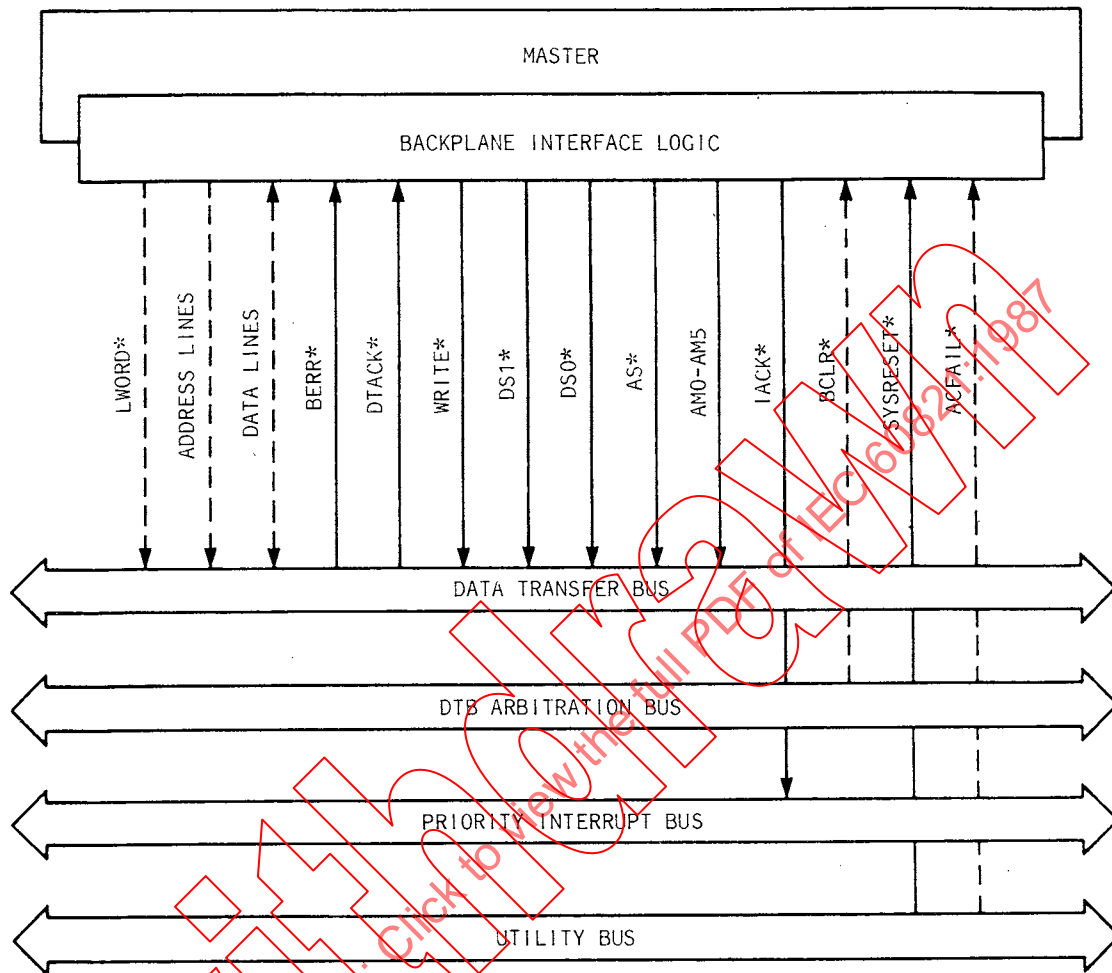


FIG. 2-2. - Block diagram: MASTER.

Tableau 2-5

REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de MAITRES

Type de MAITRE	Utilisation des lignes pointillées
D08(E0)	DOIT surveiller et commander D00-D15. PEUT ou NON commander LWORD*. PEUT ou NON commander ou surveiller D16-D31
D16	DOIT surveiller et commander D00-D15. PEUT ou NON commander LWORD*. PEUT ou NON commander ou surveiller D16-D31
D32	DOIT commander LWORD*. DOIT surveiller et commander D00-D31. NE DOIT PAS commander IACK* au niveau bas
A16	DOIT commander A01-A15. PEUT ou NON commander A16-A31
A24	DOIT commander A01-A23. PEUT ou NON commander A24-A31
A32	DOIT commander A01-A31
TOUS	PEUT ou NON surveiller BCLR* ou ACFAIL* (voir chapitres 3 et 5)

Notes:

- 1.- Les mnémoniques D08(E0), D16 et D32 sont définis dans le tableau 2-10.
- 2.- Les mnémoniques A16, A24 et A32 sont définis dans le tableau 2-9.

Table 2-5

RULES and PERMISSIONS that specify the use of the dotted lines by the various types of MASTERS

Type of MASTER	Use of dotted lines
D08(E0)	MUST monitor and drive D00-D15. MAY or MAY NOT drive LWORD*. MAY or MAY NOT drive or monitor D16-D31
D16	MUST monitor and drive D00-D15. MAY or MAY NOT drive LWORD*. MAY or MAY NOT drive or monitor D16-D31
D32	MUST drive LWORD*. MUST monitor and drive D00-D31. MUST NOT drive IACK* low
A16	MUST drive A01-A15. MAY or MAY NOT drive A16-A31
A24	MUST drive A01-A23 MAY or MAY NOT drive A24-A31
A32	MUST drive A01-A31
ALL	MAY or MAY NOT monitor BCLR*, or ACFAIL* (see Chapters 3 and 5)

Notes:

- 1.- The mnemonics D08(E0), D16, and D32 are defined in Table 2-10.
- 2.- The mnemonics A16, A24, and A32 are defined in Table 2-9.

2.3.2 ESCLAVE

Le schéma-bloc de l'ESCLAVE est indiqué dans la figure 2-3. Les lignes pointillées montrent les signaux dont l'utilisation varie selon les différents types d'ESCLAVES. Le tableau 2-6 définit comment les différents types d'ESCLAVES utilisent ces lignes. D'autres informations sur la manière dont les différents types d'ESCLAVES contrôlent les lignes de données sont indiquées dans le tableau 2-22.

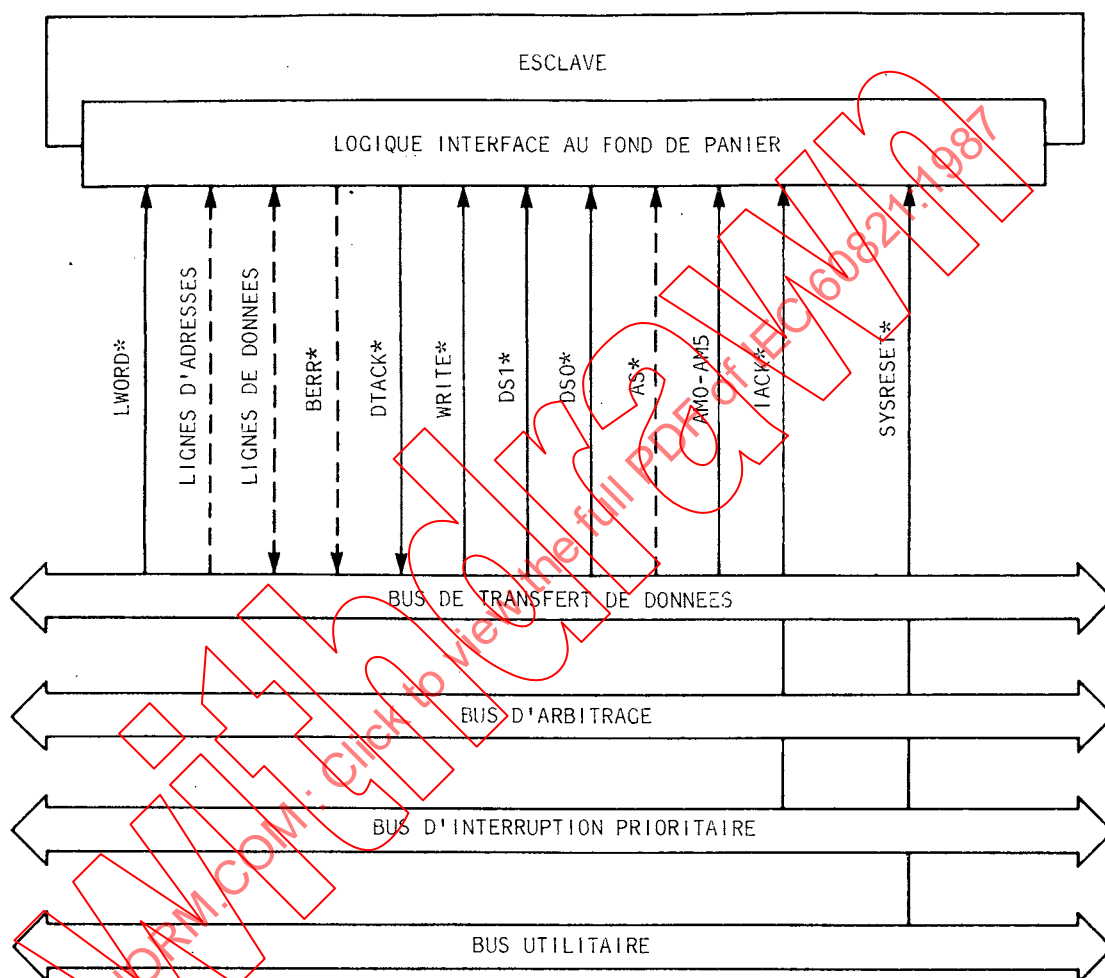


FIG. 2-3. - Schéma-bloc: ESCLAVE.

2.3.2 SLAVE

The block diagram of the SLAVE is shown in Figure 2-3. The dotted lines in the diagram show signals whose use varies among the various types of SLAVES. Table 2-6 shows how the various types of SLAVES use these lines. Further information about how the various types of SLAVES drive the data lines is given in Table 2-22.

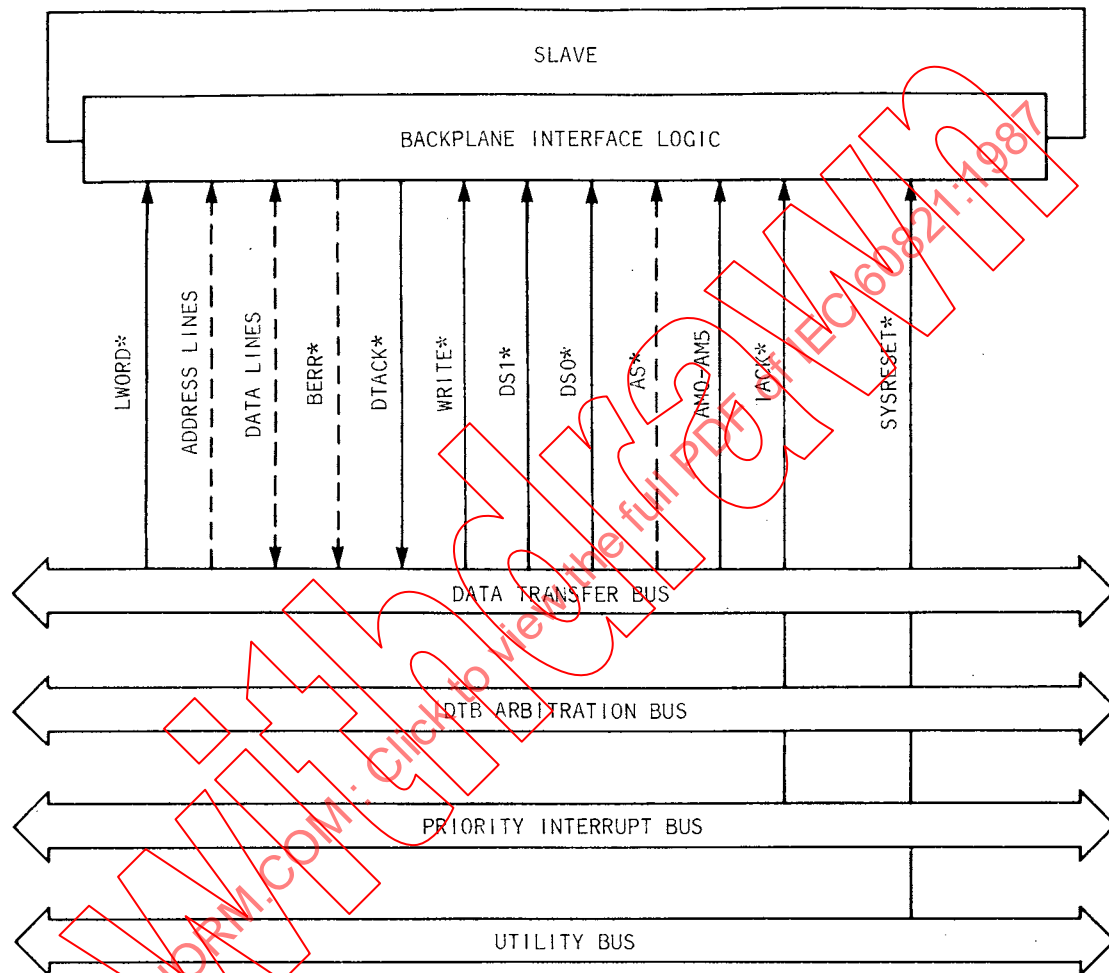


FIG. 2-3. - Block diagram: SLAVE.

Tableau 2-6

REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types d'ESCLAVES

Type d'ESCLAVE	Utilisation des lignes pointillées
D08(0)	DOIT surveiller et commander D00-D07. PEUT ou NON surveiller AS*. PEUT ou NON surveiller ou commander D08-D31
D08(E0)	DOIT surveiller et commander D00-D15. PEUT ou NON surveiller AS*. PEUT ou NON surveiller et commander D16-D31
D16	DOIT surveiller et commander D00-D15. PEUT ou NON surveiller AS*. PEUT ou NON surveiller ou commander D16-D31
D32	DOIT surveiller et commander D00-D31. PEUT ou NON surveiller AS*
A16	DOIT surveiller A01-A15. PEUT ou NON surveiller A16-A31
A24	DOIT surveiller A01-A23. PEUT ou NON surveiller A16-A31
A32	DOIT surveiller A01-A31
TOUS	PEUT ou NON commander BERR*

Notes:

- 1.- Les mnémoniques D08(0), D08(E0), D16 et D32 sont définis dans le tableau 2-10.
- 2.- Les mnémoniques A16, A24 et A32 sont définis dans le tableau 2-9.

Table 2-6

*RULES and PERMISSIONS that specify the use of the dotted lines
by the various types of SLAVES*

Type of SLAVE	Use of dotted lines
D08(0)	MUST monitor and drive D00-D07. MAY or MAY NOT monitor AS*. MAY or MAY NOT monitor or drive D08-D31
D08(E0)	MUST monitor and drive D00-D15. MAY or MAY NOT monitor AS*. MAY or MAY NOT monitor or drive D16-D31
D16	MUST monitor and drive D00-D15. MAY or MAY NOT monitor AS*. MAY or MAY NOT monitor or drive D16-D31
D32	MUST monitor and drive D00-D31. MAY or MAY NOT monitor AS*
A16	MUST monitor A01-A15. MAY or MAY NOT monitor A16-A31
A24	MUST monitor A01-A23. MAY or MAY NOT monitor A24-A31
A32	MUST monitor A01-A31
ALL	MAY or MAY NOT drive BERR*

Notes:

- 1.- The mnemonics D08(0), D08(E0), D16, and D32 are defined in Table 2-10.
- 2.- The mnemonics A16, A24, and A32 are defined in Table 2-9.

2.3.3 LIMITEUR DE TEMPS D'OCCUPATION DU BUS

Le schéma-bloc du LIMITEUR DE TEMPS D'OCCUPATION DU BUS est indiqué dans la figure 2-4. Les LIMITEURS DE TEMPS D'OCCUPATION DU BUS peuvent être conçus pour commander BERR* au niveau bas après des périodes déterminées de temps. Le tableau 2-7 montre comment le mnémonique BT0() est utilisé pour décrire les différents types de LIMITEURS DE TEMPS D'OCCUPATION DU BUS.

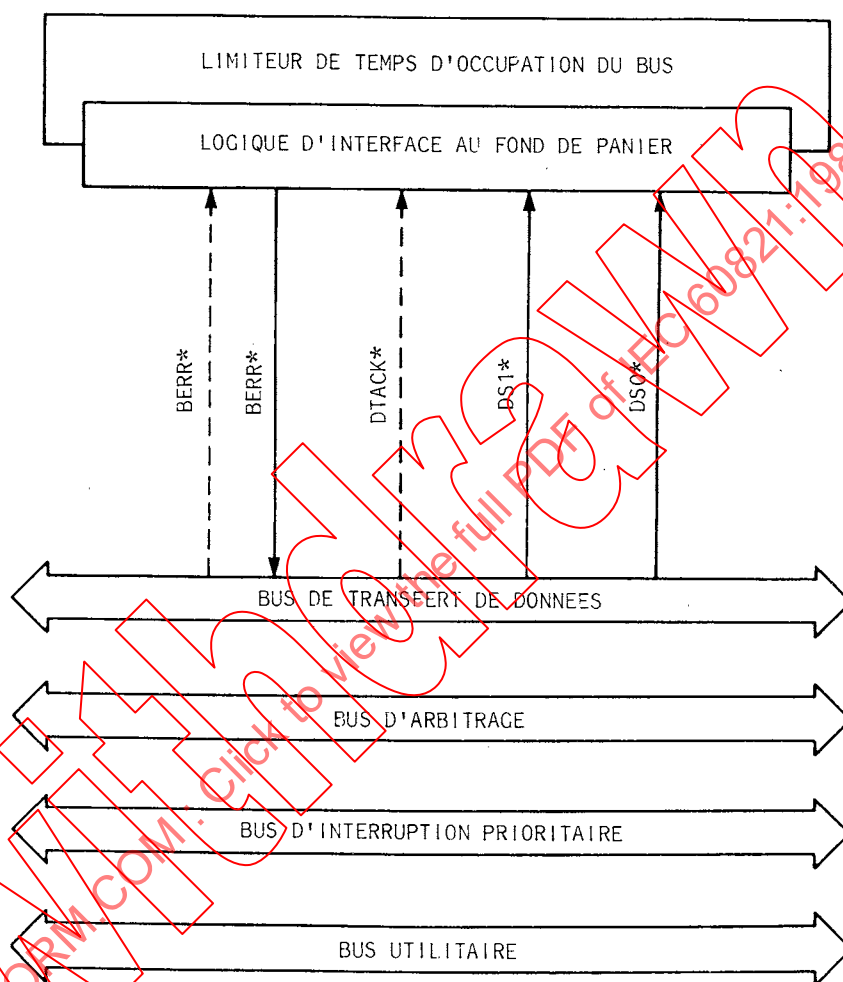


FIG. 2-4. - Schéma-bloc: LIMITEUR DE TEMPS D'OCCUPATION DU BUS.

2.3.3 BUS TIMER

The block diagram of the BUS TIMER is shown in Figure 2-4. BUS TIMERS can be designed to drive BERR* low after various periods of time. Table 2-7 shows how the BT0() mnemonic is used to describe the various types of BUS TIMERS.

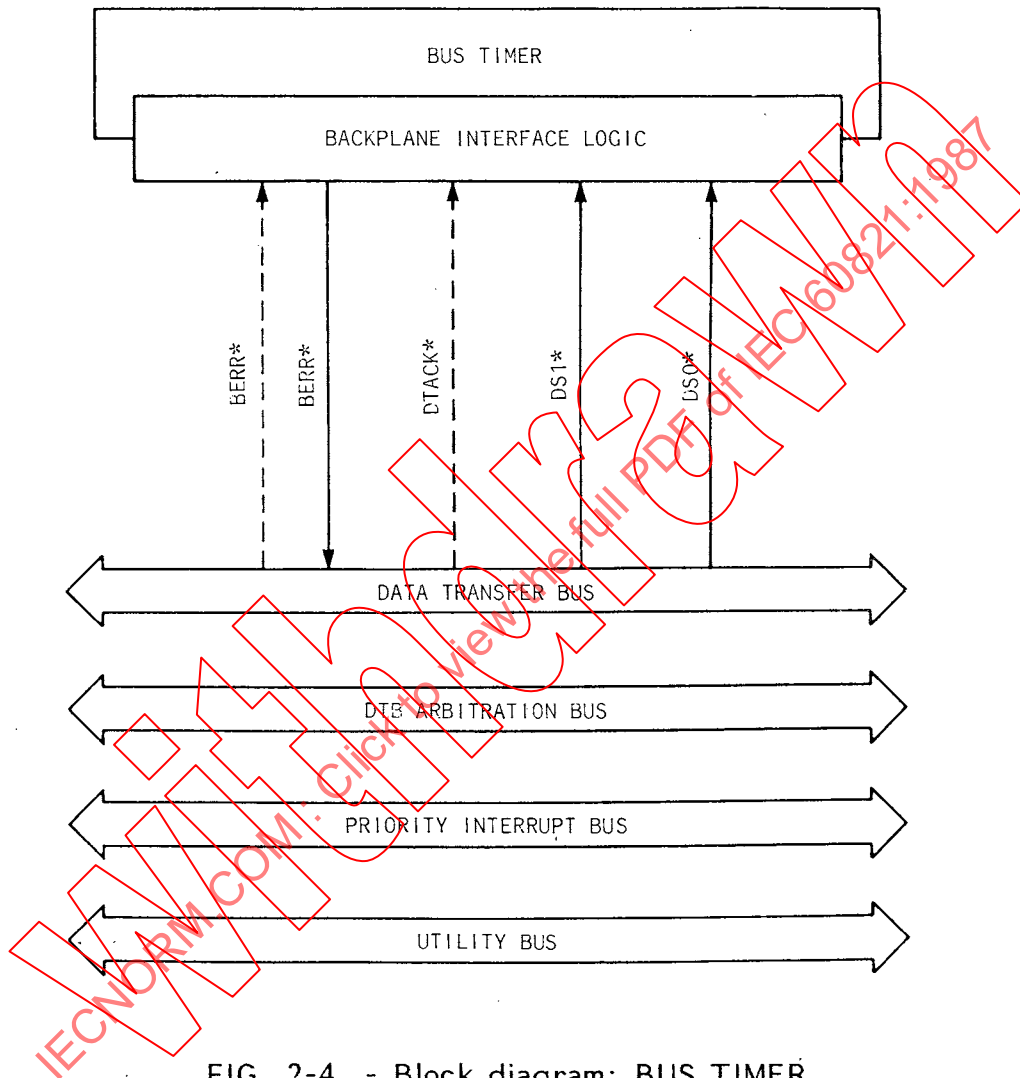


Tableau 2-7

Utilisation du mnémonique BT0() pour spécifier la durée de limitation de temps des LIMITEURS DE TEMPS D'OCCUPATION DU BUS

Le mnémonique suivant	Lorsqu'il est appliqué au	Signifie qu'il
BT0(x)	LIMITEUR DE TEMPS D'OCCUPATION DU BUS	Commande BERR* au niveau bas si le premier signal de validation de donnée reste au niveau bas plus de x microsecondes (µs)

OBSERVATION 2.13:

Les lignes pointillées DTACK* et BERR* de la figure 2-4, page 82, permettent de réaliser le LIMITEUR DE TEMPS D'OCCUPATION DU BUS selon l'une des deux manières suivantes:

- commander BERR* au niveau bas lorsque le premier signal de validation de donnée reste au niveau bas plus longtemps que la limitation de temps d'occupation du bus, sans tenir compte des niveaux des lignes DTACK* et BERR*;
- commander BERR* au niveau bas lorsque le premier signal de validation de donnée reste au niveau bas plus longtemps que la limitation de temps d'occupation du bus, seulement si les deux signaux DTACK* et BERR* sont au niveau haut lors du dépassement de la limitation du temps d'occupation du bus.

Table 2-7

*Use of the BT0() mnemonic to specify
the time-out period of BUS TIMERS*

The following mnemonic	When applied to a	Means that it
BT0(x)	BUS TIMER	Drives BERR* low if the first data strobe stays low for longer than x microseconds (μs)

OBSERVATION 2.13:

The dotted DTACK* and BERR* lines shown in Figure 2-4, page 83, allow to implement a BUS TIMER in one of two ways:

- a) to drive BERR* low when the first data strobe stays low for longer than the bus time-out period, regardless of the levels on the DTACK* and BERR* lines.
- b) to drive BERR* low when the first data strobe stays low for longer than the bus time-out period, but only if both DTACK* and BERR* are high at the point of time-out.

2.3.4 DETECTEUR D'ACCES

Le schéma-bloc du DETECTEUR D'ACCES est indiqué dans la figure 2-5. Les lignes pointillées dans le schéma montrent les signaux dont l'utilisation varie selon les différents types de DETECTEURS D'ACCES. Le tableau 2-8 montre comment les différents types de DETECTEURS D'ACCES utilisent ces lignes.

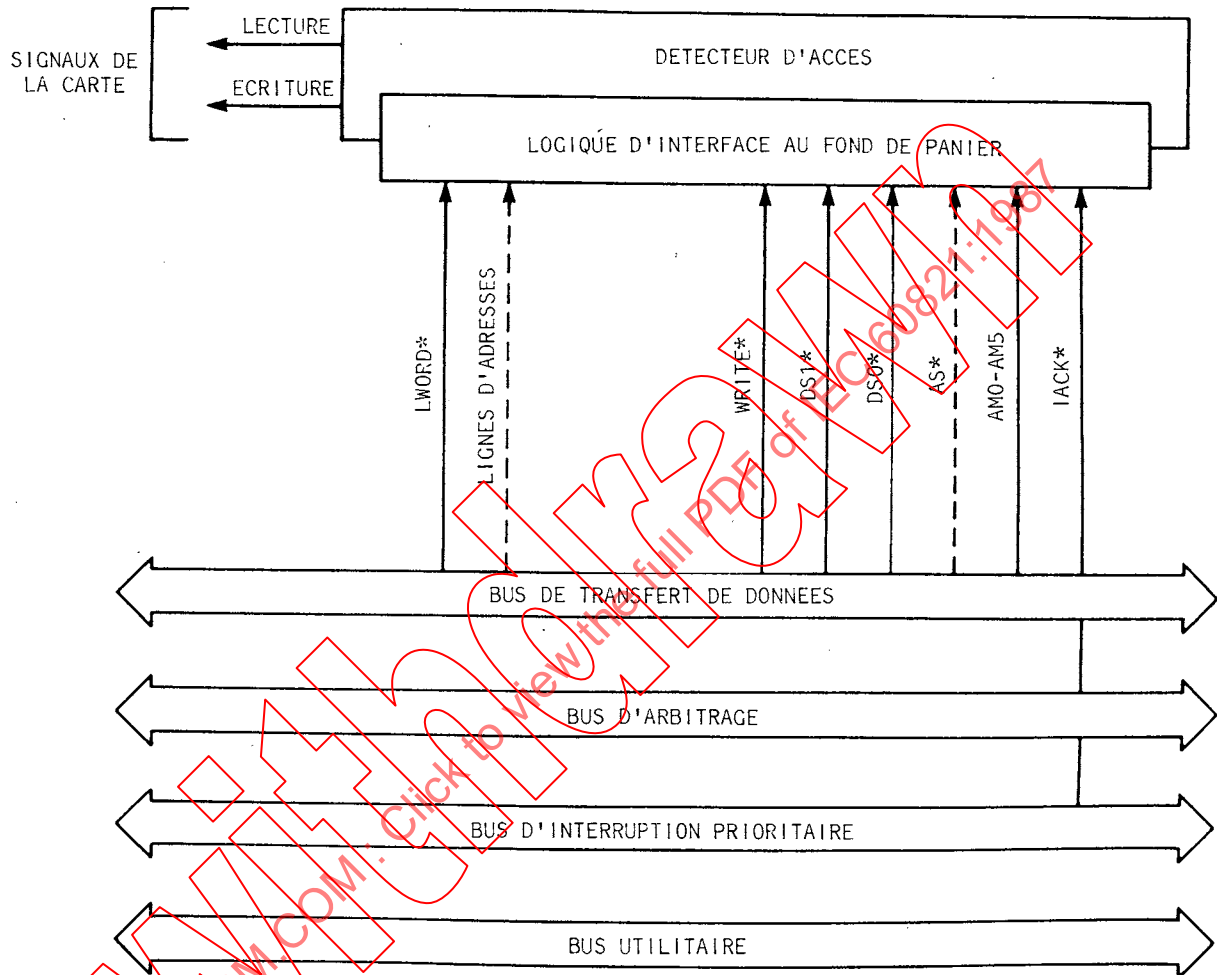


FIG. 2-5. - Schéma-bloc: DETECTEUR D'ACCES.

2.3.4 LOCATION MONITOR

The block diagram of the LOCATION MONITOR is shown in Figure 2-5. The dotted lines in the diagram show signals whose use varies among the various types of LOCATION MONITORS. Table 2-8 shows how the various types of LOCATION MONITORS use these lines.

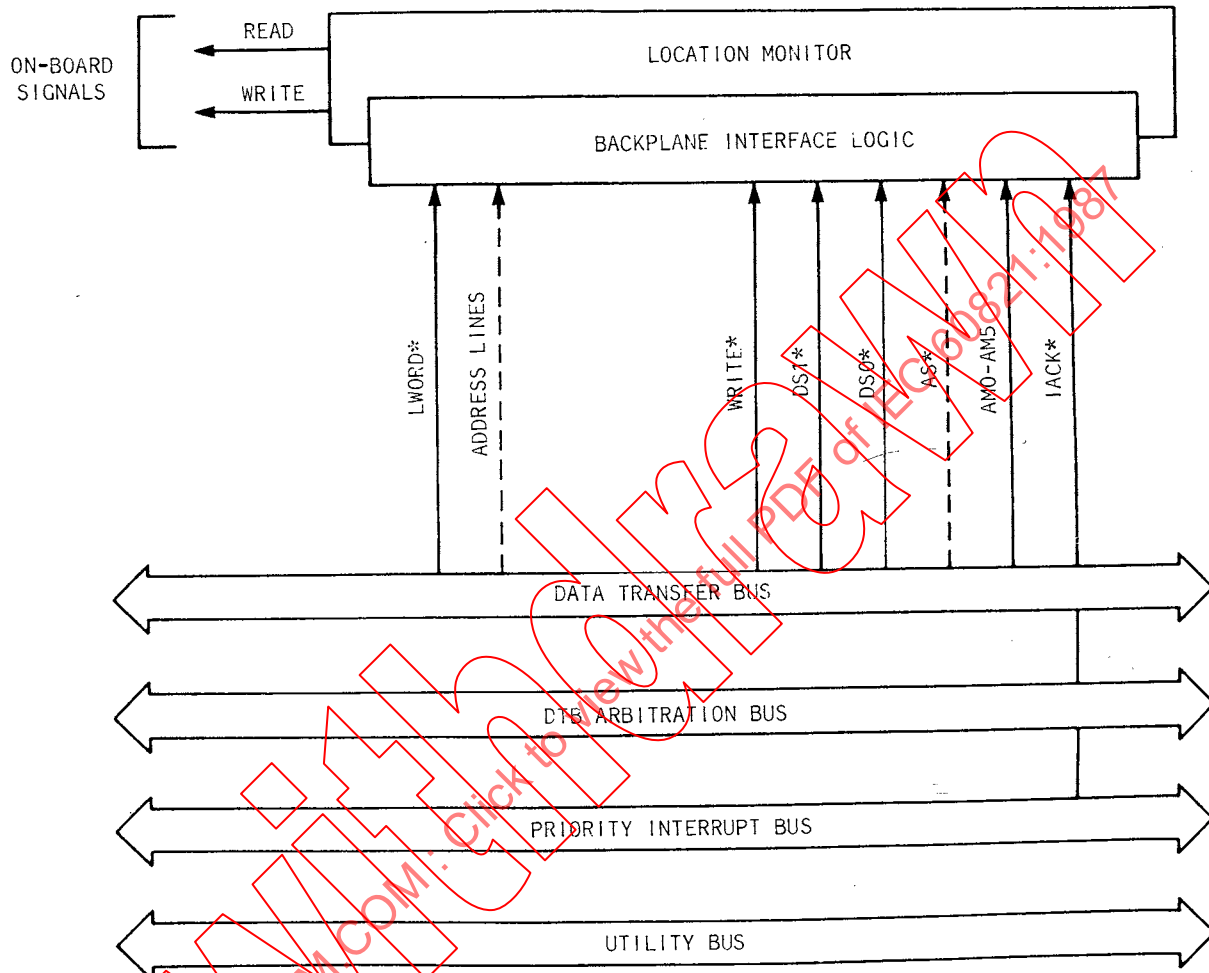


FIG. 2-5. - Block diagram: LOCATION MONITOR.

Tableau 2-8

REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de DETECTEURS D'ACCES

Type de DETECTEURS D'ACCES	Utilisation des lignes pointillées
A16	DOIT surveiller A01-A15. PEUT ou NON surveiller A16-A31
A24	DOIT surveiller A01-A23. PEUT ou NON surveiller A24-A31
A32	DOIT surveiller A01-A31
TOUS	PEUT ou NON surveiller AS*

Note:

Les mnémoniques A16, A24 et A32 sont définies dans le tableau 2-9.

2.3.5 Modes d'adressage

Les MAITRES diffusent une adresse sur le DTB à chaque début de cycle. Cette adresse diffusée peut être une adresse de 16, 24 ou 32 bits, cela dépendant des possibilités qu'a le MAITRE pour la diffuser. On définit les adresses 16 bits comme "adresses courtes", les adresses 24 bits comme "adresses standards" et les adresses 32 bits comme "adresses étendues".

Le tableau 2-9 montre les différents mnémoniques utilisés pour la description des possibilités d'adressage et de quelle manière on utilise chacun pour la description des MAITRES, ESCLAVES et DETECTEURS D'ACCES.

Table 2-8

*RULES and PERMISSIONS that specify the use of the dotted lines
by the various types of LOCATION MONITORS*

Type of LOCATION MONITOR	Use of dotted lines
A16	MUST monitor A01-A15. MAY or MAY NOT monitor A16-A31
A24	MUST monitor A01-A23. MAY or MAY NOT monitor A24-A31
A32	MUST monitor A01-A31
ALL	MAY or MAY NOT monitor AS*

Note:

The mnemonics A16, A24 and A32 are defined in Table 2-9.

2.3.5 Addressing modes

MASTERS broadcast an address over the DTB at the beginning of each cycle. This broadcasted address might be a 16, 24 or 32-bit address, depending upon the capabilities of the MASTER broadcasting it. The 16-bit addresses are "short addresses", the 24-bit addresses are "standard addresses", and the 32-bit addresses are "extended addresses".

Table 2-9 shows the various mnemonics used to describe the addressing capabilities, and how each is used to describe MASTERS, SLAVES, and LOCATION MONITORS.

Tableau 2-9

Mnémoniques qui spécifient les possibilités d'adressage

Le mnémonique suivant	Lorsqu'il est appliqué à (au)	Signifie qu'il
A16	MAITRE	Peut générer des cycles avec des adresses courtes (16 bits).
	ESCLAVE	Peut accepter des cycles avec des adresses courtes (16 bits).
	DETECTEUR D'ACCES	Peut surveiller des cycles avec des adresses courtes (16 bits)
A24	MAITRE	Peut générer des cycles avec des adresses standards (24 bits).
	ESCLAVE	Peut accepter des cycles avec des adresses standards (24 bits).
	DETECTEUR D'ACCES	Peut surveiller des cycles avec des adresses standards (24 bits)
A32	MAITRE	Peut générer des cycles avec des adresses étendues (32 bits).
	ESCLAVE	Peut accepter des cycles avec des adresses étendues (32 bits).
	DETECTEUR D'ACCES	Peut surveiller des cycles avec des adresses étendues (32 bits)

Le MAITRE diffuse un code modificateur d'adresse (AM) accompagnant chaque adresse afin de prévenir des ESCLAVES si l'adresse est courte, standard ou étendue.

Les adresses courtes sont générées par les MAITRES A16 et acceptées par les ESCLAVES A16. Les adresses standards sont générées par les MAITRES A24 et acceptées par les ESCLAVES A24. Les adresses étendues sont générées par les MAITRES A32 et acceptées par les ESCLAVES A32.

L'adressage court est d'abord destiné à l'adressage des entrées/sorties (E/S). Il permet aux ESCLAVES A16 d'être conçus avec moins de circuits logiques, puisqu'ils n'ont pas à décoder autant de lignes d'adresse. Bien que les cartes d'E/S puissent être conçues pour décoder des adresses standards et étendues, l'adressage court les rend souvent inutiles.

Les modes d'adressage standards et étendus sont d'abord destinés à l'adressage de la mémoire, bien qu'il n'y ait aucune règle s'opposant à la conception de cartes d'E/S répondant aussi à ces modes d'adressage. Les modes d'adressage standards et étendus permettent des capacités d'adressage plus grandes.

REGLE 2.9:

Les cartes ESCLAVES DOIVENT décoder toutes les lignes de modification d'adresse.

Table 2-9

Mnemonics that specify addressing capabilities

The following mnemonic	When applied to a	Means that it
A16	MASTER	Can generate cycles with short (16-bit) addresses.
	SLAVE	Can accept cycles with short (16-bit) addresses.
	LOCATION MONITOR	Can monitor cycles with short (16-bit) addresses
A24	MASTER	Can generate cycles with standard (24-bit) addresses.
	SLAVE	Can accept cycles with standard (24-bit) addresses.
	LOCATION MONITOR	Can monitor cycles with standard (24-bit) addresses
A32	MASTER	Can generate cycles with extended (32-bit) addresses.
	SLAVE	Can accept cycles with extended (32-bit) addresses.
	LOCATION MONITOR	Can monitor cycles with extended (32-bit) addresses

The MASTER broadcasts an Address Modifier (AM) code along with each address to tell SLAVES whether the address is short, standard, or extended.

Short addresses are generated by A16 MASTERS, and accepted by A16 SLAVES. Standard addresses are generated by A24 MASTERS, and accepted by A24 SLAVES. Extended addresses are generated by A32 MASTERS, and accepted by A32 SLAVES.

Short addressing is intended primarily for addressing input/output (I/O) devices. It allows A16 SLAVES to be designed with less logic, since they do not have to decode as many address lines. While I/O boards can be designed to decode standard addresses and extended addresses, short addressing usually makes this unnecessary.

Standard and extended addressing modes are intended primarily for addressing memory, although there is no rule against designing I/O boards that also respond to these addressing modes. Standard and extended addressing modes allow much larger addressing ranges.

RULE 2.9:

SLAVE boards MUST decode all of the address modifier lines.

OBSERVATION 2.14:

La REGLE 2.9 permet à un ESCLAVE de différencier les adresses courtes, standards et étendues.

OBSERVATION 2.15:

En plus des trois modes d'adressage décrits ici, un quatrième mode d'adressage est utilisé dans les cycles d'acquiescement d'interruption (voir chapitre 4). Ces cycles d'acquiescement d'interruption peuvent être distingués des cycles de transfert de données par le fait que le signal IACK* est au niveau bas au lieu du niveau haut.

REGLE 2.10:

Chaque fois qu'un MAITRE diffuse une adresse sur le bus d'adresse, il DOIT s'assurer que IACK* est au niveau haut.

AUTORISATION 2.7:

Un MAITRE PEUT commander IACK* au niveau haut au cours d'une diffusion d'adresse ou il PEUT NE PAS commander IACK*. (Les adaptateurs de bus le maintiendront au niveau haut.)

REGLE 2.11:

Les ESCLAVES NE DOIVENT PAS répondre aux cycles du DTB quand IACK* est au niveau bas.

RECOMMANDATION 2.2:

Puisque de nombreux systèmes comprendront des ESCLAVES A16, A24 et A32, inclure les possibilités A16 et A24 sur les cartes CPU en plus de la possibilité A32, et inclure la possibilité A16 sur les cartes CPU avec possibilité A24.

2.3.6 Possibilités de base de transferts de données

Il y a quatre possibilités de base de transferts de données associées au DTB: D08(E0) (octet pair et impair), D08(O) (octet impair seulement), D16 et D32. Ces possibilités autorisent une flexibilité pour l'interfaçage de différents types de processeurs et périphériques sur le bus.

Les processeurs 8 bits peuvent avoir une interface au bus en tant que MAITRES D08(E0). Les processeurs 16 bits peuvent avoir une interface au bus en tant que MAITRES D16. Le module ESCLAVE D16 est utile pour l'interface au DTB des dispositifs à mémoire 16 bits ou les ESCLAVES d'E/S 16 bits.

Beaucoup de circuits intégrés contrôleurs de périphériques ont seulement des registres 8 bits. Bien que ces circuits aient souvent plusieurs registres, ils ne peuvent pas fournir simultanément le contenu de deux registres quand un MAITRE D16 essaye d'accéder à deux emplacements contigus de mémoire par un cycle de lecture double octet. Ces circuits intégrés périphériques 8 bits peuvent avoir une interface au DTB par un ESCLAVE D08(O). Les ESCLAVES D08(O) n'occupent seulement que les emplacements d'OCTET(1) ou d'OCTET(3) et ne répondent qu'aux accès octet unique. (Bien que les accès octet unique à ces emplacements d'octets impairs soient toujours situés sur D00-D07, cela simplifie la logique d'interface des ESCLAVES D08(O).)

OBSERVATION 2.14:

RULE 2.9 allows a SLAVE to differentiate short addresses, standard addresses, and extended addresses.

OBSERVATION 2.15:

In addition to the three modes of addressing described here, there is a fourth mode which is used on interrupt acknowledge cycles (see Chapter 4). These interrupt acknowledge cycles can be distinguished from data transfer cycles by the fact the the IACK* signal line is low instead of high.

RULE 2.10:

Whenever a MASTER broadcasts an address over the address bus, it MUST ensure that IACK* is high.

PERMISSION 2.7:

A MASTER MAY either drive IACK* high during the address broadcast or it MAY leave IACK* undriven. (The bus terminators will then hold it high.)

RULE 2.11:

SLAVES MUST NOT respond to DTB cycles when IACK* is low.

RECOMMENDATION 2.2:

Since many systems will include a mixture of A16, A24, and A32 SLAVES, include A16 and A24 capability on 32-bit CPU cards in addition to the A32 capability and include A16 capability on CPU cards with A24 capability.

2.3.6 Basic data transfer capabilities

There are four basic data transfer capabilities associated with the DTB: D08(E0) (Even and Odd byte), D08(O) (Odd byte only), D16, and D32. These capabilities allow flexibility when interfacing different types of processors and peripherals to the bus.

Eight-bit processors can be interfaced to the bus as D08(E0) MASTERS. Sixteen-bit processors can be interfaced to the bus as D16 MASTERS. The D16 SLAVE module is useful for interfacing 16-bit memory devices or 16-bit I/O SLAVES to the DTB.

Many existing peripheral chips have registers that are only 8 bits wide. While these chips often have several of these registers, they cannot provide the contents of two registers simultaneously when a D16 MASTER attempts to access two adjacent locations with a double byte read cycle. These 8-bit peripheral ICs can be interfaced to the DTB as a D08(O) SLAVE. D08(O) SLAVES provide only BYTE(1) or BYTE(3) locations and respond only to single byte accesses. (Since single byte accesses to these odd byte locations always take place over D00-D07, this simplifies the D08(O) SLAVE'S interface logic.)

RECOMMANDATION 2.3:

Puisque la plupart des microprocesseurs 16 bits peuvent accéder aussi à la mémoire par 8 bits à la fois, inclure la possibilité MAITRE D08(E0) sur les cartes CPU 16 bits en plus de leur possibilité D16. Cela lui permet aussi d'accéder aux ESCLAVES D08(O), en plus des transferts de données 8 bits de et vers la mémoire.

RECOMMANDATION 2.4:

Puisque la plupart des microprocesseurs 32 bits peuvent accéder aussi à la mémoire par 8 bits ou 16 bits à la fois, inclure la possibilité D08(E0) et D16 sur les cartes CPU 32 bits en plus de leur possibilité D32. La possibilité D08(E0) permet non seulement des transferts de données 8 bits de et vers la mémoire, mais permet aussi au MAITRE d'accéder aux ESCLAVES D08(O).

OBSERVATION 2.16:

Il paraît logique de définir des ESCLAVES "d'octet pair seulement" qui correspondent aux emplacements mémoire d'octet pair adjacents aux ESCLAVES D08(O). Mais cela ne peut pas être réalisé car il n'y a qu'une seule ligne d'acquiescement de données. Si un MAITRE allait sélectionner à la fois des emplacements mémoire d'octet pair et d'octet impair en effectuant un transfert double octet, les deux ESCLAVES iraient commander à la fois la même ligne de signal d'acquiescement de données (DTACK*) et le MAITRE ne saurait pas si les deux cartes ont acquiescé l'accès.

RECOMMANDATION 2.5:

Puisque de nombreux systèmes comprendront des MAITRES avec des possibilités différentes de transferts de données, inclure les possibilités D08(E0) et D16 sur les cartes ESCLAVE 32 bits (mémoire) en plus de la possibilité D32, et inclure la possibilité D08(E0) sur les cartes ESCLAVE 16 bits en plus de la possibilité D16.

OBSERVATION 2.17:

Les ESCLAVES D08(O) répondent seulement aux adresses d'octets impaires, il ne peuvent pas accéder à une mémoire contiguë. Les ESCLAVES D08(O) sont utiles uniquement pour les E/S, les registres d'état ou de contrôle, alors que les ESCLAVES D08(E0), D16 et D32 le sont aussi pour la mémoire.

Le tableau 2-10 montre les différents mnémoniques utilisés pour décrire les possibilités de base de transferts de données et comment chacune est utilisée pour décrire des MAITRES, ESCLAVES et DETECTEURS D'ACCES.

RECOMMENDATION 2.3:

Since most 16-bit microprocessors can also access memory 8 bits at a time, include D08(E0) MASTER capability on 16-bit CPU boards in addition to the D16 capability. In addition to allowing 8-bit data transfers to and from memory, this also allows them to access D08(O) SLAVES.

RECOMMENDATION 2.4:

Since most 32-bit microprocessors can also transfer data to and from memory 8 and 16 bits at a time, include D08(E0) and D16 capability on 32-bit CPU boards in addition to D32 capability. The D08(E0) capability not only allows 8-bit data transfers to and from memory, but it also allows the MASTER to access D08(O) SLAVES.

OBSERVATION 2.16:

It might seem logical to define "Even byte only" SLAVES which respond to the even byte memory locations adjacent to the D08(O) SLAVES. But, this cannot be done because there is only one data transfer acknowledge line. If a MASTER were to select both an even byte and an odd byte location simultaneously, by doing a double byte transfer, both SLAVES would drive the same data acknowledge (DTACK*) line and the MASTER would not know whether both boards had acknowledged the access.

RECOMMENDATION 2.5:

Since many systems will include MASTERS with various data transfer capabilities, include D08(E0) and D16 capability on 32-bit SLAVE (memory) boards in addition to the D32 capability, and include D08(E0) capability on 16-bit SLAVE boards in addition to the D16 capability.

OBSERVATION 2.17:

Since D08(O) SLAVES respond only to odd byte addresses, they cannot provide contiguous memory. D08(O) SLAVES are useful only for I/O, status, or control registers, while D08(E0), D16 and D32 SLAVES are also useful for memory.

Table 2-10 shows the various mnemonics used to describe the basic data transfer capabilities, and how each is used to describe MASTERS, SLAVES, and LOCATION MONITORS.

Tableau 2-10

Mnémoniques qui spécifient les possibilités de base de transferts de données

Le mnémonique suivant	Lorsqu'il est appliqué au	Signifie qu'il
D08(E0)	MAITRE ESCLAVE DETECTEUR D'ACCES	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycles de lecture pour octet unique: LECTURE OCTET(0) LECTURE OCTET(1) LECTURE OCTET(2) LECTURE OCTET(3) Cycles d'écriture pour octet unique: ECRITURE OCTET(0) ECRITURE OCTET(1) ECRITURE OCTET(2) ECRITURE OCTET(3)
D08(0)	ESCLAVE	Peut accepter les cycles suivants: Cycles de lecture pour octet unique: LECTURE OCTET(1) LECTURE OCTET(3) Cycles d'écriture pour octet unique: ECRITURE OCTET(1) ECRITURE OCTET(3)
D16	MAITRE ESCLAVE DETECTEUR D'ACCES	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycles de lecture pour double octet: LECTURE OCTET(0-1) LECTURE OCTET(2-3) Cycles d'écriture pour double octet: ECRITURE OCTET(0-1) ECRITURE OCTET(2-3)
D32	MAITRE ESCLAVE DETECTEUR D'ACCES	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycle de lecture pour quadruple octet: LECTURE OCTET(0-3) Cycle d'écriture pour quadruple octet: ECRITURE OCTET(0-3)

Note:

(E0) signifie pair (Even) ou impair (Odd), et (0) signifie seulement impair (Odd).

2.3.7 Possibilités de transferts par bloc

Les MAITRES accèdent souvent à plusieurs emplacements de mémoire dans un ordre ascendant. Dans ce cas, les cycles de transfert par bloc sont très utiles. Ils permettent au MAITRE de fournir une adresse

Table 2-10

Mnemonics that specify basic data transfer capabilities

The following mnemonic	When applied to a	Means that it
D08(E0)	MASTER SLAVE LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Single byte read cycles: BYTE(0) READ BYTE(1) READ BYTE(2) READ BYTE(3) READ Single byte write cycles: BYTE(0) WRITE BYTE(1) WRITE BYTE(2) WRITE BYTE(3) WRITE
D08(0)	SLAVE	Can accept the following cycles: Single byte read cycles: BYTE(1) READ BYTE(3) READ Single byte write cycles: BYTE(1) WRITE BYTE(3) WRITE
D16	MASTER SLAVE LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Double byte read cycles: BYTE(0-1) READ BYTE(2-3) READ Double byte write cycles: BYTE(0-1) WRITE BYTE(2-3) WRITE
D32	MASTER SLAVE LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Quad byte read cycle: BYTE(0-3) READ Quad byte write cycle: BYTE(0-3) WRITE

Note:

(E0) is Even Odd, and (0) is Odd only.

2.3.7 Block transfer capabilities

MASTERS often access several memory locations in ascending order. When this is the case, block transfer cycles are very useful. They allow the MASTER to provide a single address, and then access data in

unique, et d'accéder à la donnée de cet emplacement et à celles des adresses supérieures, sans fournir d'adresses additionnelles.

Lorsqu'un MAITRE déclenche un cycle de transfert par bloc, l'ESCLAVE qui répond mémorise l'adresse dans un compteur d'adresses sur la carte. Le MAITRE, à la fin du premier transfert de données (c'est-à-dire qu'il commande les signaux de validation de donnée au niveau haut) ne permet pas au signal de validation d'adresse de monter au niveau haut. Par contre, il commande de façon répétitive le ou les signaux de validation de donnée au niveau bas en réponse aux signaux de reconnaissances de données de l'ESCLAVE et transfère les données vers ou à partir des emplacements séquentiels de mémoire en ordre ascendant.

Pour accéder à ou aux adresses suivantes, l'ESCLAVE incrémente le compteur d'adresses sur la carte pour chaque cycle d'un ou des signaux de validation de donnée.

OBSERVATION 2.18:

Les cycles de transfert par bloc de longueur non définie ne sont pas permis parce qu'ils compliqueraient la conception des cartes mémoire. En particulier tous les ESCLAVES à transfert par bloc (celui qui répond et ceux qui ne répondent pas) devraient mémoriser l'adresse initiale et augmenter le compteur d'adresse sur chaque cycle du bus. La sortie du compteur incrémentée devrait être décodée par tous les ESCLAVES pour vérifier si elle est dans les frontières des adresses de la carte. Bien que cela soit certainement possible, décoder l'adresse limite généralement les temps d'accès de l'ESCLAVE. Pour simplifier la conception de ces ESCLAVES et pour permettre des temps d'accès plus courts, la REGLE 2.12 a été formulée:

REGLE 2.12:

Les cycles lecture par bloc et écriture par bloc NE DOIVENT PAS dépasser une frontière de 256 octets.

OBSERVATION 2.19:

La REGLE 2.12 limite la longueur maximale des transferts par bloc à 256 octets. Pourtant, sachant que seulement A01 à A07 changeront pendant la durée du cycle de transfert par bloc, elle simplifie la conception des ESCLAVES. Les lignes d'adresse supérieures doivent être décodées une seule fois au début du cycle de transfert par bloc, permettant des temps d'accès plus courts sur tous les transferts de données suivants.

OBSERVATION 2.20:

Dans certains cas, il peut être nécessaire de transférer un bloc important de données qui chevauche une ou plusieurs frontières de 256 octets. Dans ce cas, si la circuiterie des cartes effectuant le transfert par bloc est conçue pour détecter l'arrivée à une frontière de 256 octets, elle peut commander momentanément AS* au niveau haut, puis lancer un autre transfert par bloc sans intervention du logiciel système.

Le cycle de lecture par bloc est similaire à une chaîne de cycles lecture. Le cycle écriture par bloc est aussi similaire à une chaîne de cycles d'écriture. La différence est que l'adresse initiale seulement est diffusée par le MAITRE et le signal de validation d'adresse est maintenu au niveau bas pendant tous les transferts de données.

that location and those at higher addresses, without providing additional addresses.

When a MASTER initiates a block transfer cycle, the responding SLAVE latches the address into an on-board address counter. The MASTER, upon completing the first data transfer (i.e. driving the data strobes high) does not allow the address strobe to go high. Instead, it repeatedly drives the data strobe(s) low in response to data transfer acknowledgements from the SLAVE, and transfers data to or from sequential memory locations in ascending order.

To access the next location(s), the SLAVE increments an on-board counter that generates the address for each transition of the data strobe(s).

OBSERVATION 2.18:

Block transfer cycles of indefinite length are not allowed, because this complicates the design of memory boards. Specifically, all block transfer SLAVES (the one that responds, as well as those that do not) would need to latch the initial address and then increment the address counter on each bus transfer. All SLAVES would then have to decode the incremented address to see if the block transfer has crossed a board boundary into their address range. While this is certainly possible, such address decoding typically limits access times of the SLAVE. To simplify the design of these SLAVES, and to permit faster access times, RULE 2.12 has been formulated.

RULE 2.12:

Block transfer cycles MUST NOT cross any 256-byte boundary.

OBSERVATION 2.19:

RULE 2.12 limits the maximum length of block transfers to 256 bytes. However, knowing that only A01 through A07 will change during the course of the block transfer simplifies the design of block transfer SLAVES. The upper address lines only have to be decoded once, at the beginning of the block transfer cycle, allowing much faster access times on all subsequent data transfers.

OBSERVATION 2.20:

In some cases it might be necessary to transfer a large block of data which crosses one or more 256-byte boundaries; in such a case if the hardware on boards which does the block transfers is designed to recognize the arrival at a 256-byte boundary, it can momentarily drive AS* high and then initiate another block transfer without the intervention of system software.

The block read cycle is very similar to a string of read cycles. Likewise, the block write cycle is very similar to a string of write cycles. The difference is that only the initial address is broadcast by the MASTER and the address strobe is held low during all of the data transfers.

OBSERVATION 2.21:

Le contrôle du DTB ne peut être transféré pendant le transfert par bloc parce que le signal de validation d'adresse est maintenu au niveau bas pendant tous les transferts de données, et le contrôle du DTB ne peut être transféré que lorsque la validation d'adresse est au niveau haut.

Le tableau 2-11 liste les différents mnémoniques utilisés pour décrire les possibilités de transferts par bloc et comment chacun est utilisé pour décrire les MAITRES, ESCLAVES et DETECTEURS D'ACCES.

Tableau 2-11

Mnémoniques qui définissent les possibilités de transferts par bloc

Le mnémonique suivant	Lorsqu'il est appliqué à (au)	Signifie qu'il
BLT	MAITRE D08(E0) ESCLAVE D08(E0) DETECTEUR D'ACCES D08(E0)	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycles lecture par bloc: LECTURE DE BLOCS PAR OCTET Cycles écriture par bloc: ECRITURE DE BLOCS PAR OCTET
	MAITRE D16 ESCLAVE D16 DETECTEUR D'ACCES D16	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycles lecture par bloc: LECTURE DE BLOCS PAR DOUBLE OCTET Cycles écriture par bloc: ECRITURE DE BLOCS PAR DOUBLE OCTET
	MAITRE D32 ESCLAVE D32 DETECTEUR D'ACCES D32	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycles lecture par bloc: LECTURE DE BLOCS PAR QUADRUPLE OCTET Cycles écriture par bloc: ECRITURE DE BLOCS PAR QUADRUPLE OCTET

2.3.8 Possibilités de lecture-modification-écriture

Dans les systèmes multiprocesseurs qui partagent des ressources comme la mémoire et les E/S, une méthode est nécessaire pour allouer ces ressources. Une tâche très importante de cet algorithme d'allocation est d'assurer qu'une ressource utilisée par une tâche ne peut pas être utilisée simultanément par une autre. Le problème est décrit par un exemple:

OBSERVATION 2.21:

Control of the DTB cannot be transferred during block transfers because the address strobe is held low through all of the data transfers, and control of the DTB can only be transferred while the address strobe is high.

Table 2-11 lists the various mnemonics used to describe block transfer capabilities and how each is used to describe MASTERS, SLAVES, and LOCATION MONITORS.

Table 2-11

Mnemonics that specify block transfer capabilities

The following mnemonic	When applied to a	Means that it
BLT	D08(E0) MASTER D08(E0) SLAVE D08(E0) LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Block read cycles: SINGLE BYTE BLOCK READ Block write cycles: SINGLE BYTE BLOCK WRITE
	D16 MASTER D16 SLAVE D16 LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Block read cycles: DOUBLE BYTE BLOCK READ Block write cycles: DOUBLE BYTE BLOCK WRITE
	D32 MASTER D32 SLAVE D32 LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Block read cycles: QUAD BYTE BLOCK READ Block write cycles: QUAD BYTE BLOCK WRITE

2.3.8 Read-modify-write capabilities

In multiprocessor systems which share resources such as memory and I/O, a method is needed to allocate these resources. One very important goal of this allocation algorithm is to ensure that a resource being used by one task cannot be used by another at the same time. The problem is best described by an example:

Deux processeurs dans un système distribué de traitement partagent une ressource commune (par exemple une imprimante). Un seul processeur peut utiliser la ressource à la fois. La ressource est allouée par un bit en mémoire, c'est-à-dire si le bit est positionné, la ressource est occupée; s'il est libre, la ressource est disponible. Pour obtenir une ressource, le processeur A lit le bit et le teste pour déterminer s'il a été mis à zéro. Si le bit est à zéro, le processeur A positionne le bit pour empêcher l'accès par le processeur B. Cette opération comprend deux transferts de données: une lecture pour effectuer le test du bit, et une écriture pour positionner le bit. Cependant, une difficulté peut survenir si le bus est donné au processeur B entre les deux cycles de transfert du bus. Le processeur B peut aussi trouver le bit à zéro et supposer que la ressource est disponible. Les deux processeurs positionneraient alors le bit dans le prochain cycle disponible pour tenter d'utiliser la ressource.

Ce conflit est évité en définissant un cycle lecture-modification-écriture qui empêche l'arbitrage d'avoir lieu entre la lecture et l'écriture. Ce cycle est très semblable au cycle lecture suivi immédiatement par un cycle d'écriture. La différence est que le signal de validation d'adresse est commandé continuellement au niveau bas pendant les deux cycles de transfert. Cela assure que à la différence du cycle lecture suivi d'un cycle d'écriture, le contrôle du DTB ne peut pas être transféré pendant un cycle lecture-modification-écriture, comme c'est possible lorsque le signal de validation d'adresse est au niveau haut.

Le tableau 2-12 liste les différents mnémoniques utilisés pour décrire les possibilités de lecture-modification-écriture et comment chacun est utilisé pour décrire les MAÎTRES, ESCLAVES et DETECTEURS D'ACCES.

Two processors in a distributed processing system share a common resource (e.g. a printer). Only one processor can use the resource at a time. The resource is allocated by a bit in memory i.e., if the bit is set, the resource is busy; if it is cleared, the resource is available. To gain use of the resource, processor A reads the bit and tests it to determine whether it is cleared. If the bit is cleared, processor A sets the bit to lock out processor B. This operation takes two data transfers: a read to test the bit, and a write to set the bit. However, a difficulty might arise if the bus is given to processor B between these two transfers. Processor B might then also find the bit cleared and assume the resource is available. Both processors will then set the bit in the next available cycle and attempt to use the resource.

This conflict is avoided by defining a read-modify-write cycle which prevents transferring control of the DTB between the read portion and the write portion of the cycle. This cycle is very similar to a read cycle immediately followed by a write cycle. The difference is that the address strobe is held low during both transfers. This ensures that, unlike a read cycle followed by a write cycle, control of the DTB cannot be transferred during a read-modify-write cycle, as this is only possible while the address strobe is high.

Table 2-12 lists the various mnemonics used to describe read-modify-write capabilities and how each is used to describe MASTERS, SLAVES, and LOCATION MONITORS.

Tableau 2-12

*Mnémoniques qui spécifient les possibilités
de lecture-modification-écriture*

Le mnémonique suivant	Lorsqu'il est appliqué à (au)	Signifie qu'il
RMW	MAITRE D08(E0) ESCLAVE D08(E0) DETECTEUR D'ACCES D08(E0)	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycles lecture-modification-écriture pour octet unique: LECTURE-MODIFICATION-ECRITURE OCTET(0) LECTURE-MODIFICATION-ECRITURE OCTET(1) LECTURE-MODIFICATION-ECRITURE OCTET(2) LECTURE-MODIFICATION-ECRITURE OCTET(3)
	ESCLAVE D08(0)	Peut accepter les cycles suivants: Cycles lecture-modification-écriture pour octet unique: LECTURE-MODIFICATION-ECRITURE OCTET(1) LECTURE-MODIFICATION-ECRITURE OCTET(3)
	MAITRE D16 ESCLAVE D16 DETECTEUR D'ACCES D16	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycles lecture-modification-écriture pour double octet: LECTURE-MODIFICATION-ECRITURE OCTET(0-1) LECTURE-MODIFICATION-ECRITURE OCTET(2-3)
	MAITRE D32 ESCLAVE D32 DETECTEUR D'ACCES D32	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycles lecture-modification-écriture pour quadruple octet: LECTURE-MODIFICATION-ECRITURE OCTET(0-3)

Table 2-12

Mnemonics that specify read-modify-write capabilities

The following mnemonic	When applied to a	Means that it
RMW	D08(E0) MASTER D08(E0) SLAVE D08(E0) LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Single byte read-modify-write cycles: BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE
	D08(0) SLAVE	Can accept the following cycles: Single byte read-modify-write cycles: BYTE(1) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE
	D16 MASTER D16 SLAVE D16 LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Double byte read-modify-write cycles: BYTE(0-1) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE
	D32 MASTER D32 SLAVE D32 LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Quad byte read-modify-write cycles: BYTE(0-3) READ-MODIFY-WRITE

2.3.9 Possibilités de transferts non alignés

Plusieurs microprocesseurs 32 bits mémorisent et extraient des données d'une manière non alignée. Par exemple, une valeur 32 bits peut être mémorisée de quatre façons différentes comme indiqué sur la figure 2-6:

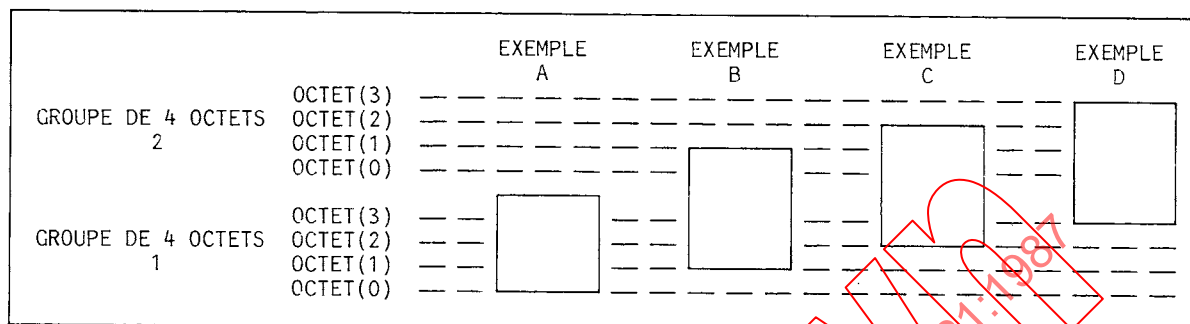


FIG. 2-6. - Quatre moyens de mémoriser des données de 32 bits en mémoire.

Le MAITRE peut transférer les 32 bits de données en utilisant plusieurs séquences différentes de cycles DTB. Par exemple, il peut transférer les données un octet à la fois, en utilisant quatre transferts de données octet unique. Cependant, un MAITRE peut effectuer le transfert beaucoup plus vite en utilisant une des séquences de cycles indiqués dans le tableau 2-13.

OBSERVATION 2.22:

Les séquences indiquées dans le tableau 2-13 seraient caractéristiques d'un MAITRE qui accède aux emplacements d'octets en ordre séquentiel ascendant. Le protocole du BUS CEI 821 ne le nécessite pas.

Comme indiqué dans le tableau 2-13, chacun de ces transferts 32 bits peut être accompli avec une combinaison de transferts octet unique ou double octet. Cependant, les exemples B et D nécessitent trois cycles de bus quand ils sont effectués de cette manière. C'est pourquoi le protocole du DTB inclut aussi deux cycles de transfert trois octets. Quand ils sont utilisés en combinaison avec un cycle octet unique, ces cycles triple octet permettent de mémoriser des données en utilisant seulement deux cycles bus comme indiqué dans les exemples B et D.

2.3.9 Unaligned transfer capabilities

Some 32-bit microprocessors store and retrieve data in an unaligned fashion. For example, a 32-bit value might be stored in four different ways, as shown in Figure 2-6.

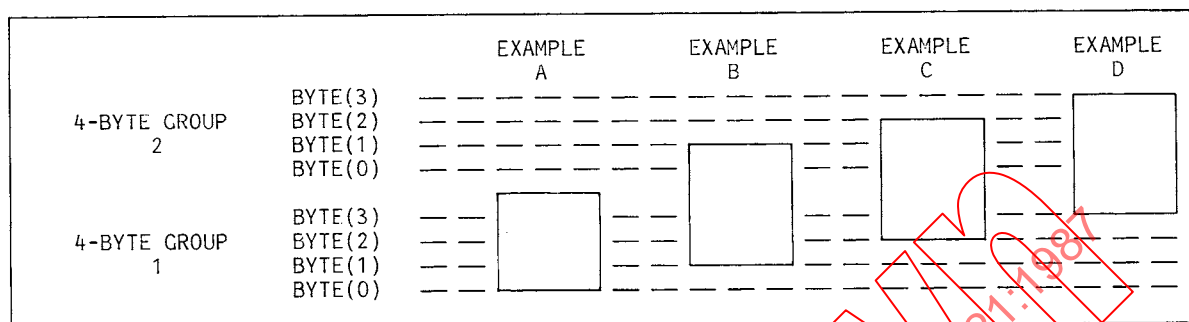


FIG. 2-6. - Four ways that 32 bits of data might be stored in memory.

The MASTER can transfer the 32 bits of data using several different sequences of DTB cycles. For example, it can transfer the data one byte at a time, using four single byte data transfers. However, a MASTER can accomplish the transfer much quicker by using one of the cycle sequences shown in Table 2-13.

OBSERVATION 2.22:

The sequences shown in Table 2-13 would be typical of a MASTER that accesses the byte locations in ascending order. The IEC 821 BUS protocol does not require this.

As shown in Table 2-13, each of these 32-bit transfers can be accomplished with a combination of single byte and double byte transfers. However, examples B and D require three bus cycles when done this way. Because of this, the DTB protocol also includes two triple byte transfer cycles. When used in combination with a single byte cycle, these triple byte cycles allow data to be stored as shown in examples B and D using only two bus cycles.

Tableau 2-13

*Transferts de 32 bits de données utilisant des cycles
de transfert de multiple octet*

Exemple	Séquences de cycles utilisées pour effectuer le transfert	Lignes du bus de données utilisées	Emplacements d'octets accédés (voir figure 2-6)
A	Transfert quadruple octet	D00-D31	Groupe 1, OCTET(0-3)
B	Transfert octet unique Transfert double octet Transfert octet unique ou Transfert triple octet Transfert octet unique	D00-D07 D00-D15 D08-D15 D00-D23 D08-D31	Groupe 1, OCTET(1) Groupe 1, OCTET(2-3) Groupe 1, OCTET(0) Groupe 1, OCTET(1-3) Groupe 1, OCTET(0)
C	Transfert double octet Transfert double octet	D00-D15 D00-D15	Groupe 1, OCTET(2-3) Groupe 2, OCTET(0-1)
D	Transfert octet unique Transfert double octet Transfert octet unique ou Transfert octet unique Transfert triple octet	D00-D07 D00-D15 D08-D15 D00-D07 D00-D23	Groupe 1, OCTET(3) Groupe 2, OCTET(0-1) Groupe 2, OCTET(2) Groupe 1, OCTET(3) Groupe 2, OCTET(0-2)

Certains microprocesseurs 32 bits mémorisent et extraient aussi des données 16 bits à la fois, d'une façon non alignée, comme indiqué sur la figure 2-7.

		EXEMPLE E	EXEMPLE F	EXEMPLE G	EXEMPLE H
GROUPE DE 4 OCTETS 2	OCTET(3)	---	---	---	---
	OCTET(2)	---	---	---	---
	OCTET(1)	---	---	---	---
	OCTET(0)	---	---	---	---
GROUPE DE 4 OCTETS 1	OCTET(3)	---	---	---	---
	OCTET(2)	---	---	---	---
	OCTET(1)	---	---	---	---
	OCTET(0)	---	---	---	---

FIG. 2-7. - Quatre moyens de mémoriser des données de 16 bits en mémoire.

Le MAITRE peut transférer les 16 bits de données en utilisant plusieurs séquences différentes de cycles DTB listées dans le tableau 2-14.

OBSERVATION 2.23:

Les séquences listées dans le tableau 2-14 seraient caractéristiques d'un MAITRE qui accède aux emplacements d'octets en ordre croissant. Le protocole du BUS CEI 821 ne l'impose pas.

Table 2-13

*Transferring 32 bits of data using multiple
byte transfer cycles*

Example	Cycle sequences used to accomplish the transfer	Data bus lines used	Byte locations accessed (see Figure 2-6)
A	Quad byte transfer	D00-D31	Group 1, BYTE(0-3)
B	Single byte transfer Double byte transfer Single byte transfer or Triple byte transfer Single byte transfer	D00-D07 D00-D15 D08-D15 D00-D23 D08-D31	Group 1, BYTE(1) Group 1, BYTE(2-3) Group 2, BYTE(0) Group 1, BYTE(1-3) Group 2, BYTE(0)
C	Double byte transfer Double byte transfer	D00-D15 D00-D15	Group 1, BYTE(2-3) Group 2, BYTE(0-1)
D	Single byte transfer Double byte transfer Single byte transfer or Single byte transfer Triple byte transfer	D00-D07 D00-D15 D08-D15 D00-D07 D00-D23	Group 1, BYTE(3) Group 2, BYTE(0-1) Group 2, BYTE(2) Group 1, BYTE(3) Group 2, BYTE(0-2)

Some 32-bit microprocessors also store and retrieve data 16 bits at a time, in an unaligned fashion, as shown in Figure 2-7.

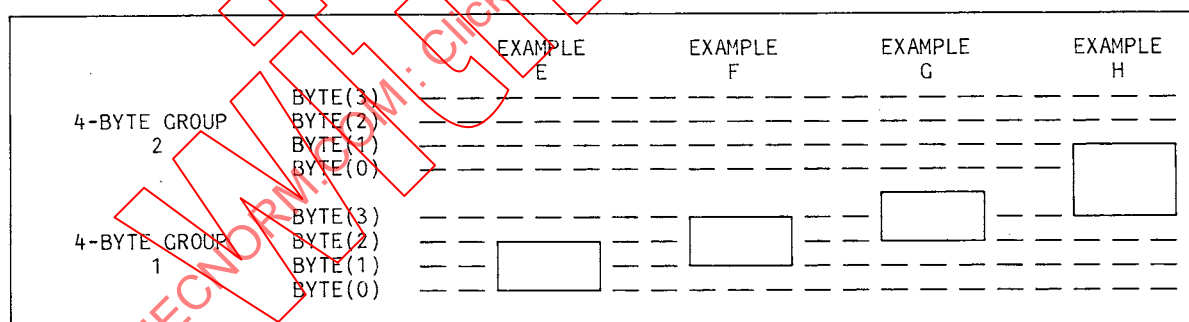


FIG. 2-7. - Four ways that 16 bits of data might be stored in memory.

The MASTER can transfer the 16 bits of data using several different sequences of DTB cycles as listed in Table 2-14.

OBSERVATION 2.23:

The sequences listed in Table 2-14 would be typical of a MASTER that accesses the byte locations in ascending order. The IEC 821 BUS protocol does not require this.

Comme indiqué dans la figure 2-13, page 170, le transfert de 16 bits de l'exemple F peut être effectué avec deux transferts d'octet unique. Cependant, cela demande deux cycles de bus. C'est pourquoi, le protocole DTB inclut aussi un cycle de transfert double octet qui permet la mémorisation des données utilisant un seul cycle de bus comme indiqué dans l'exemple F.

OBSERVATION 2.24:

Puisque les transferts non alignés utilisent toutes les 32 lignes de données, seuls les MAITRES D32 et les ESCLAVES D32 peuvent effectuer des transferts non alignés.

Tableau 2-14

Transferts de 16 bits de données utilisant des cycles de transfert de multiple octet

Exemple	Séquences de cycles utilisées pour effectuer le transfert	Lignes de données du bus utilisées	Emplacements d'octets accédés (voir figure 2-7)
E	Transfert double octet	D00-D15	Groupe 1, OCTET(0-1)
F	Transfert octet unique Transfert octet unique ou Transfert double octet	D08-D07 D08-D15 D08-D23	Groupe 1, OCTET(1) Groupe 1, OCTET(2) Groupe 1, OCTET(1-2)
G	Transfert double octet	D00-D15	Groupe 1, OCTET(2-3)
H	Transfert octet unique Transfert octet unique	D00-D07 D08-D15	Groupe 1, OCTET(3) Groupe 2, OCTET(0)

Le tableau 2-15 liste comment le mnémonique (UAT) de transfert non aligné est utilisé pour décrire des MAITRES, ESCLAVES et DETECTEURS D'ACCES.

Tableau 2-15

Mnémonique qui spécifie la possibilité de transfert non aligné

Le mnémonique suivant	Lorsqu'il est appliqué à (au)	Signifie qu'il
UAT	MAITRE D32 ESCLAVE D32 DETECTEUR D'ACCES D32	Peut générer les cycles suivants: Peut accepter les cycles suivants: Peut surveiller les cycles suivants: Cycles de lecture pour triple octet: LECTURE OCTET(0-2) LECTURE OCTET(1-3) Cycles d'écriture pour triple octet: ECRITURE OCTET(0-2) ECRITURE OCTET(1-3) Cycles de lecture pour double octet: LECTURE OCTET(1-2) Cycles d'écriture pour double octet: ECRITURE OCTET(1-2)

As shown in Figure 2-13, page 171, the 16-bit transfer in example F can be accomplished with two single byte transfers. However, this requires two bus cycles. Because of this, the DTB protocol also includes a double byte transfer cycle which allows data to be stored as shown in example F using only one bus cycle.

OBSERVATION 2.24:

Since unaligned transfers make use of all 32 data lines, only D32 MASTERS and SLAVES can do unaligned transfers.

Table 2-14

Transferring 16 bits of data using multiple byte transfer cycles

Example	Cycle sequences used to accomplish the transfer	Data bus lines used	Byte locations accessed (see Figure 2-7)
E	Double byte transfer	D00-D15	Group 1, BYTE(0-1)
F	Single byte transfer Single byte transfer or Double byte transfer	D00-D07 D08-D15 D08-D23	Group 1, BYTE(1) Group 1, BYTE(2) Group 1, BYTE(1-2)
G	Double byte transfer	D00-D15	Group 1, BYTE(2-3)
H	Single byte transfer Single byte transfer	D00-D07 D08-D15	Group 1, BYTE(3) Group 2, BYTE(0)

Table 2-15 lists how the unaligned transfer (UAT) mnemonic is used to describe MASTERS, SLAVES, and LOCATION MONITORS.

Table 2-15

Mnemonic that specifies unaligned transfer capability

The following mnemonic	When applied to a	Means that it
UAT	D32 MASTER D32 SLAVE D32 LOCATION MONITOR	Can generate the following cycles: Can accept the following cycles: Can monitor the following cycles: Triple byte read cycles: BYTE(0-2) READ BYTE(1-3) READ Triple byte write cycles: BYTE(0-2) WRITE BYTE(1-3) WRITE Double byte read cycle: BYTE(1-2) READ Double byte write cycle: BYTE(1-2) WRITE

2.3.10 Possibilité UNIQUEMENT D'ADRESSAGE

Le cycle UNIQUEMENT D'ADRESSAGE est le seul cycle du DTB qui ne soit pas utilisé pour transférer des données. Il commence comme un cycle typique du DTB avec l'adresse, le code de modification d'adresse, les lignes IACK* et LWORD* devenant valides et le signal de validation d'adresse passant au niveau bas après un temps d'établissement. Pourtant, les signaux de validation de donnée ne sont jamais commandés au niveau bas. Après avoir maintenu stables les lignes qui doivent être validées par le signal de validation d'adresse pendant une durée minimale prescrite, le MAITRE termine le cycle sans attendre le passage au niveau bas de DTACK* ou BERR*. (Le cycle UNIQUEMENT D'ADRESSAGE est aussi le seul type de cycle du DTB qui ne nécessite pas de réponse pour se terminer.)

Le tableau 2-16 liste comment le mnémonique (ADO) UNIQUEMENT D'ADRESSAGE est utilisé pour décrire des MAITRES et des ESCLAVES.

Tableau 2-16

Mnémonique qui spécifie la possibilité UNIQUEMENT D'ADRESSAGE

Le mnémonique suivant	Lorsqu'il est appliqué à (au)	Signifie qu'il
ADO	MAITRE	Peut générer des cycles UNIQUEMENT D'ADRESSAGE

OBSERVATION 2.25:

Les cycles UNIQUEMENT D'ADRESSAGE peuvent être utilisés pour améliorer la performance des cartes en permettant à une carte CPU de diffuser une adresse avant d'avoir déterminé si cette adresse sélectionne ou non un ESCLAVE sur le BUS CEI 821. Ce mode de diffusion d'adresse permet aux ESCLAVES du BUS CEI 821 de décoder l'adresse en même temps que la carte CPU.

REGLE 2.12A:

Tous les ESCLAVES DOIVENT inclure la possibilité de cycle ADO sans perte de données ou mauvais fonctionnement.

2.3.11 Interaction entre les modules fonctionnels du DTB

Des transferts de données ont lieu entre des MAITRES et des ESCLAVES. Le MAITRE est le module qui contrôle le transfert. L'ESCLAVE qui reconnaît l'adresse comme la sienne est l'ESCLAVE qui répond et tous les autres ESCLAVES sont des ESCLAVES qui ne répondent pas.

Après avoir lancé le cycle de transfert de données, le MAITRE attend une réponse de l'ESCLAVE sélectionné. Quand le MAITRE détecte la réponse de l'ESCLAVE, il commande les signaux de validation de donnée et d'adresse au niveau haut, terminant le cycle. L'ESCLAVE répond en libérant sa ligne de réponse.

2.3.10 ADDRESS-ONLY *capability*

The ADDRESS-ONLY cycle is the only cycle on the DTB that is not used to transfer data. It begins as a typical DTB cycle, with the address, address modifier code, IACK* and LWORD* lines becoming valid and the address strobe falling after a set-up time. However, the data strobes are never driven low. After holding the various lines strobed by the address strobe stable for a prescribed minimum period, the MASTER finishes the cycle without waiting for DTACK* or BERR* to go low. (The ADDRESS-ONLY cycle is also the only type of DTB cycle that does not require a response in order to complete.)

Table 2-16 lists how the ADDRESS-ONLY mnemonic (ADO) is used to describe MASTERS and SLAVES.

Table 2-16

Mnemonic that specifies ADDRESS-ONLY capability

The following mnemonic	When applied to a	Means that it
ADO	MASTER	Can generate ADDRESS-ONLY cycles

OBSERVATION 2.25:

ADDRESS-ONLY cycles can be used to enhance board performance by allowing a CPU board to broadcast an address before it determined whether or not the address selects a SLAVE on the IEC 821 BUS. Broadcasting the address in this fashion allows IEC 821 BUS SLAVES to decode the address concurrently with the CPU board.

RULE 2.12A:

ALL SLAVES MUST be designed to accommodate ADO cycles without loss of data or erroneous operation.

2.3.11 *Interaction between DTB functional modules*

Data transfers take place between MASTERS and SLAVES. The MASTER is the module controlling the transfer. The SLAVE which recognizes the address as its own is the responding SLAVE, and all other SLAVES are non-responding SLAVES.

After initiating a data transfer cycle, the MASTER waits for a response from the responding SLAVE. When the MASTER detects the response from the SLAVE it drives its data strobes and address strobe high, terminating the cycle. The SLAVE responds by releasing its response line.

OBSERVATION 2.26:

Bien que les chronogrammes d'adresses et de données soient largement indépendants, il y a deux exceptions. Premièrement, le MAITRE attend d'avoir commandé AS* au niveau bas avant de commander l'un des deux signaux de validation de donnée au niveau bas. Deuxièmement, l'ESCLAVE acquitte à la fois les signaux de validation d'adresse et de donnée avec soit DTACK* ou BERR*.

REGLE 2.13:

SI un ESCLAVE répond à un cycle de transfert de données,
ALORS il DOIT soit commander DTACK* au niveau bas, soit il DOIT commander BERR* au niveau bas mais pas les deux.

OBSERVATION 2.27:

A cause de la possibilité de déphasage des signaux sur le bus dû à la charge différente des signaux de validation d'adresse et de donnée, le front descendant des signaux de validation de donnée pourrait être détecté par l'ESCLAVE un peu avant le front descendant du signal de validation d'adresse.

OBSERVATION 2.28:

La ligne WRITE* est au niveau haut/bas pour identifier un cycle lecture/écriture avant que le premier signal de validation de donnée soit commandé au niveau bas et reste stable jusqu'à ce que les deux signaux de validation de données soient au niveau haut.

REGLE 2.14:

Avant de commander le bus de données, le MAITRE DOIT s'assurer que le précédent ESCLAVE qui répond a arrêté de commander le bus de données. Il le fait en vérifiant que DTACK* et BERR* sont tous les deux au niveau haut avant de commander le ou les signaux de validation de données au niveau bas sur n'importe quel cycle et avant de commander n'importe laquelle des lignes de données pendant un cycle d'écriture.

REGLE 2.15:

A la fin d'un CYCLE LECTURE, l'ESCLAVE qui répond DOIT relâcher le bus de données avant d'autoriser DTACK* à passer au niveau haut.

REGLE 2.16:

Quand un MAITRE lit des données provenant d'un ESCLAVE, l'ESCLAVE DOIT maintenir des données valides sur le bus de données jusqu'à ce que le MAITRE libère le premier signal de validation de donnée au niveau haut.

OBSERVATION 2.29:

L'information d'adresse sur le BUS CEI 821 peut changer immédiatement après qu'un module eut commandé DTACK* ou BERR* au niveau bas, et avant que le MAITRE ne commande le signal de validation de donnée au niveau haut.

SUGGESTION 2.3:

Afin d'obtenir une performance optimale, concevoir des MAITRES de manière qu'ils commandent dès que possible des signaux de validation de donnée au niveau haut après que DTACK* ou BERR* eut passé au niveau bas. Concevoir aussi des ESCLAVES de manière à ce qu'ils libèrent dès que possible le bus de données et DTACK* après avoir détecté les signaux de validation de donnée au niveau haut. Cela permet d'avoir une vitesse de transfert de données maximale sur le bus.

OBSERVATION 2.26:

Although the address and data timing are largely independent, there are two exceptions. First, the MASTER waits until it has driven AS* low before driving either of the data strobes low. Second, the SLAVE acknowledges both the address strobe and the data strobes with either DTACK* or BERR*.

RULE 2.13:

IF a SLAVE responds to a data transfer cycle,
THEN it MUST either drive DTACK* low or it MUST drive BERR* low,
but not both.

OBSERVATION 2.27:

Because of possible bus skew, due to different loading of the address strobe and the data strobes, the falling edge of the data strobes might be detected by the SLAVE slightly before the address strobe falling edge.

OBSERVATION 2.28:

The WRITE* line is high/low to identify a read/write cycle before the first data strobe is driven low, and remains stable until both data strobes are high.

RULE 2.14:

Before driving the data bus, the MASTER MUST ensure that the previous responding SLAVE has stopped driving the data bus. It does so by verifying that DTACK* and BERR* are both high before it drives the data strobe(s) to low on any cycle, and before it drives any of the data lines during a write cycle.

RULE 2.15:

At the end of a read cycle the responding SLAVE MUST release the data bus before allowing DTACK* to go high.

RULE 2.16:

When the MASTER reads data from the SLAVE, the SLAVE MUST maintain valid data on the data bus until the MASTER returns the first data strobe to high.

OBSERVATION 2.29:

Addressing information on the IEC 821 BUS might change soon after a module drives DTACK* or BERR* low, and before the MASTER drives the data strobe to high.

SUGGESTION 2.3:

For optimum performance, design MASTERS so that they drive the data strobes high as soon as possible after DTACK* or BERR* goes low. Also design SLAVES so that they release the data bus and DTACK* as soon as possible after detecting the data strobes high. This allows the maximum data transfer rate on the bus.

Un troisième type de module appelé DETECTEUR D'ACCES contrôle le transfert des données et active un ou deux des deux signaux sur sa carte dès qu'un accès est effectué à un emplacement d'octet qu'il contrôle. Si l'accès est un cycle écriture, le signal d'ECRITURE sur sa carte est alors activé. Si l'accès est un cycle lecture, le signal de LECTURE sur sa carte est alors activé. Si un cycle lecture-modification-écriture est effectué, les deux signaux sur sa carte sont alors activés.

Si le cycle dure trop longtemps, un quatrième module, appelé DETECTEUR DE TEMPS D'OCCUPATION DU BUS intervient en commandant BERR* au niveau bas terminant le dialogue du transfert de données et permettant la reprise de l'opération sur le bus.

REGLE 2.17:

Il y a un interverrouillage strict entre les fronts montant et descendant des signaux de validation de donnée et DTACK*/BERR*. Dès qu'un MAITRE a commandé son ou ses signaux de validation de donnée au niveau bas, il NE DOIT PAS commander son ou ses signaux de validation de donnée au niveau haut et finir un transfert tant qu'il n'a pas reçu un acquittement du transfert de données ou une réponse d'erreur de bus.

OBSERVATION 2.30:

Une carte contenant un processeur qui doit diriger des transferts de données entre elle-même et d'autres cartes de BUS CEI 821 contient un module MAITRE. Si la même carte contient aussi une mémoire accessible depuis le BUS CEI 821, elle contiendra aussi un module ESCLAVE. Un processeur de virgule flottante ou un contrôleur intelligent de périphériques peut recevoir des commandes à travers l'interface d'un ESCLAVE à partir d'une carte processeur d'usage général. Il peut ensuite agir comme un MAITRE pour accéder à la mémoire globale du BUS CEI 821 et exécuter la commande qui lui a été donnée.

2.4 Fonctionnement typique

Les MAITRES déclenchent les transferts de données sur le DTB. L'ESCLAVE qui est adressé acquitte alors le transfert. Après avoir reçu l'acquiescement du transfert de données, le MAITRE termine le cycle de transfert de données. La caractéristique asynchrone du DTB permet à l'ESCLAVE de contrôler le temps pris pour le transfert.

Avant d'effectuer n'importe quel transfert de données, un MAITRE doit se voir allouer le contrôle exclusif du DTB. Cela assure que plusieurs MAITRES n'essaieront pas d'utiliser le DTB au même moment. Le MAITRE prend le contrôle du DTB en utilisant les modules et les lignes de signaux du bus d'arbitrage. (La section 2.5 explique cela plus loin.) La discussion suivante suppose que le MAITRE s'est déjà vu allouer le contrôle du DTB.

2.4.1 Cycles typiques de transfert de données

La figure 2-8, page 120, montre un cycle typique de lecture d'octet unique. Pour commencer le transfert, le MAITRE commande les lignes d'adresse avec l'adresse souhaitée et le code modificateur d'adresse. Puisque cet exemple est un CYCLE LECTURE OCTET(1), le MAITRE commande LWORD* au niveau haut et A01 au niveau bas. Puisque ce n'est pas un cycle de reconnaissance d'interruption, il ne commande pas IACK* au niveau bas. Le MAITRE attend le temps d'établissement spécifié avant de commander AS* au niveau bas pour permettre aux

A third type of module, called the LOCATION MONITOR, monitors the data transfer and generates either or both of two on-board signals whenever an access is done to a byte location that it monitors. If the access is a write cycle, then the on-board WRITE signal is generated. If the access is a read cycle, then the on-board READ signal is generated. If a read-modify-write cycle is performed, then both on-board signals are generated.

If the cycle takes too long, a fourth module, called a BUS TIMER intervenes by driving BERR* low, completing the data transfer handshake and allowing the bus to resume operation.

RULE 2.17:

There is a strict interlock between the rising and falling edges of the data strobes and DTACK*/BERR*. Once a MASTER has driven its data strobe(s) low it MUST NOT drive its data strobe(s) high and finish a transfer without first receiving a data transfer acknowledge or a bus error response.

OBSERVATION 2.30:

A board containing a processor, which needs to direct data transfers between itself and other IEC 821 BUS boards, would contain a MASTER module. If the same board also contained memory accessible from the IEC 821 BUS, it would also contain a SLAVE module. A floating point processor or intelligent peripheral controller might receive commands through a SLAVE interface from a general purpose processor board. It then might act as a MASTER to access global IEC 821 BUS memory to execute the command it has been given.

2.4 Typical operation

MASTERS initiate data transfers over the DTB. The addressed SLAVE then acknowledges the transfer. After receiving the data transfer acknowledge, the MASTER terminates the data transfer cycle. The asynchronous nature of the DTB allows the SLAVE to control the time taken for the transfer.

Before doing any data transfers, a MASTER has to be granted exclusive control of the DTB. This ensures that multiple MASTERS will not try to use the DTB at the same time. The MASTER gains control of the DTB using the modules and signal lines of the Arbitration Bus. (Section 2.5 explains this further.) The following discussion presumes that the MASTER has already been granted and assumed control of the DTB.

2.4.1 Typical data transfer cycles

Figure 2-8, page 121, shows a typical single byte read cycle. To start the transfer, the MASTER drives the addressing lines with the desired address and address modifier code. Since this example is a BYTE(1) READ CYCLE, the MASTER drives LWORD* high and A01 low. Since it is not doing an interrupt acknowledge cycle, it does not drive IACK* low. The MASTER then waits for a specified set-up time before driving

lignes d'adresse et de modificateur d'adresse de se stabiliser avant que les ESCLAVES les échantillonnent.

Chaque ESCLAVE détermine s'il doit répondre en examinant les niveaux sur les lignes d'adresse, sur les lignes de modificateur d'adresse et sur IACK*. Entre-temps, le MAITRE commande WRITE* au niveau haut pour indiquer une opération de lecture. Le MAITRE vérifie par la suite que DTACK* et BERR* sont au niveau haut pour s'assurer que l'ESCLAVE du cycle précédent ne commande plus le bus de données. Après cette vérification, le MAITRE commande DS0* au niveau bas et maintient DS1* au niveau haut.

L'ESCLAVE qui répond détermine alors quel groupe de 4 octets et quel octet de ce groupe doit être lu et commence le transfert. Après avoir extrait la donnée de sa mémoire interne et l'avoir placée sur les lignes de données du bus D00-D07, l'ESCLAVE le signale au MAITRE en commandant DTACK* au niveau bas. L'ESCLAVE maintient alors DTACK* au niveau bas ainsi que les données valides aussi longtemps que le MAITRE maintient DS0* au niveau bas.

Lorsque le MAITRE reçoit DTACK* commandé au niveau bas, il capte la donnée sur D00-D07, libère les lignes d'adresse et commande DS0* et AS* au niveau haut. L'ESCLAVE répond en libérant D00-D07 et en libérant DTACK* au niveau haut.

OBSERVATION 2.31:

Le MAITRE dans la figure 2-8, page 120, libère toutes les lignes du DTB à la fin du transfert de données. Cela n'est pas demandé sauf si le DEMANDEUR DU MAITRE libère BBSY* pendant le transfert de données comme décrit dans la section 2.5.

Le déroulement du cycle pour des transferts de données de double octet et quadruple octet est similaire au cycle d'octet unique. Les organigrammes de ces cycles sont présentés dans les figures 2-9 et 2-10, pages 122 et 124.

AS* low, to allow the address lines and the address modifier lines to stabilize before the SLAVES sample them.

Each SLAVE determines whether it should respond by examining the levels on the address lines, the address modifier lines, and IACK*. While this is happening, the MASTER drives WRITE* high to indicate a read operation. The MASTER then verifies that DTACK* and BERR* are high to ensure that the SLAVE from the previous cycle is no longer driving the data bus. If this is the case, the MASTER then drives DS0* low, while keeping DS1* high.

The responding SLAVE then determines which 4-byte group and which byte of that group is to be accessed, and starts the transfer. After it has retrieved the data from its internal storage and placed it on data bus lines D00-D07, the SLAVE signals the MASTER by driving DTACK* low. The SLAVE then holds DTACK* low and maintains the data valid for as long as the MASTER holds DS0* low.

When the MASTER receives DTACK* driven to low, it captures the data on D00-D07, releases the address lines, and drives DS0* and AS* to high. The SLAVE responds by releasing D00-D07 and releasing DTACK* to high.

OBSERVATION 2.31:

The MASTER in Figure 2-8, page 121, releases all of the DTB lines at the end of the data transfer. This is not required unless the MASTER'S REQUESTER released BBSY* during the data transfer as described in Section 2.5.

The cycle flows for double byte and quad byte data transfer cycles are very similar to the single byte cycle. Flow diagrams for these cycles are shown in Figures 2-9 and 2-10, pages 123 and 125.

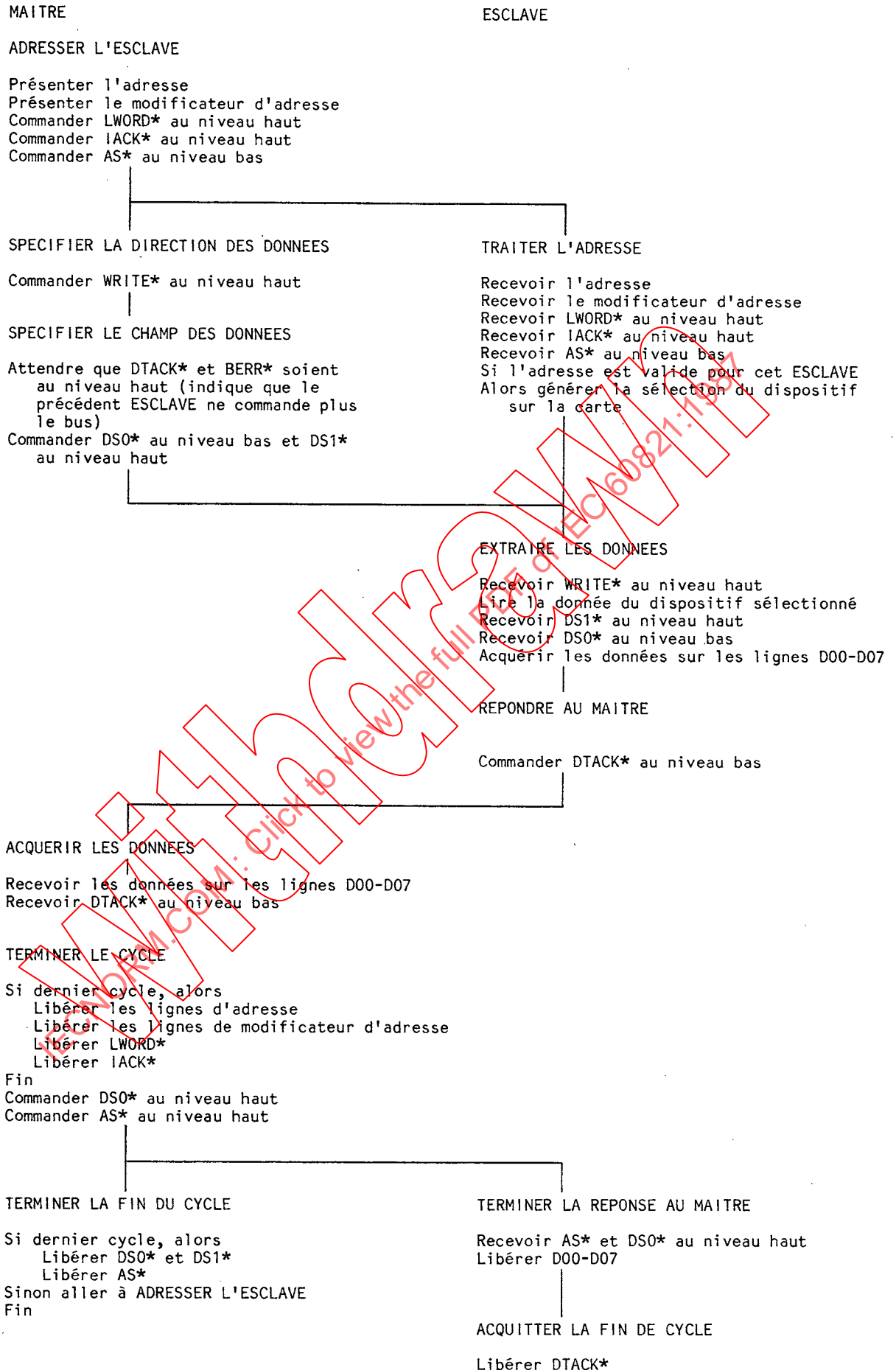


FIG. 2-8. - Un exemple de cycle de lecture d'un octet unique.

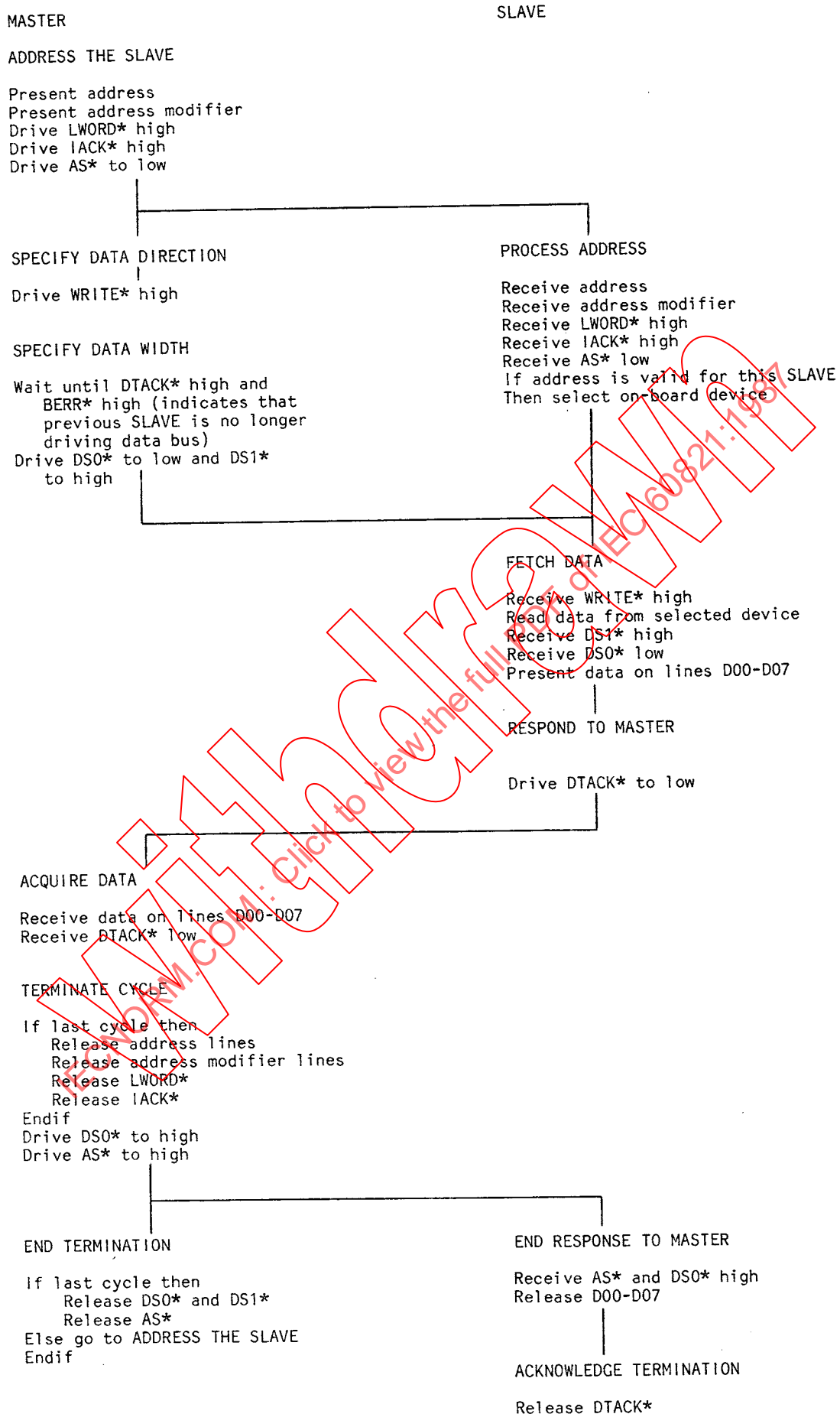


FIG. 2-8. - An example of a single byte read cycle.

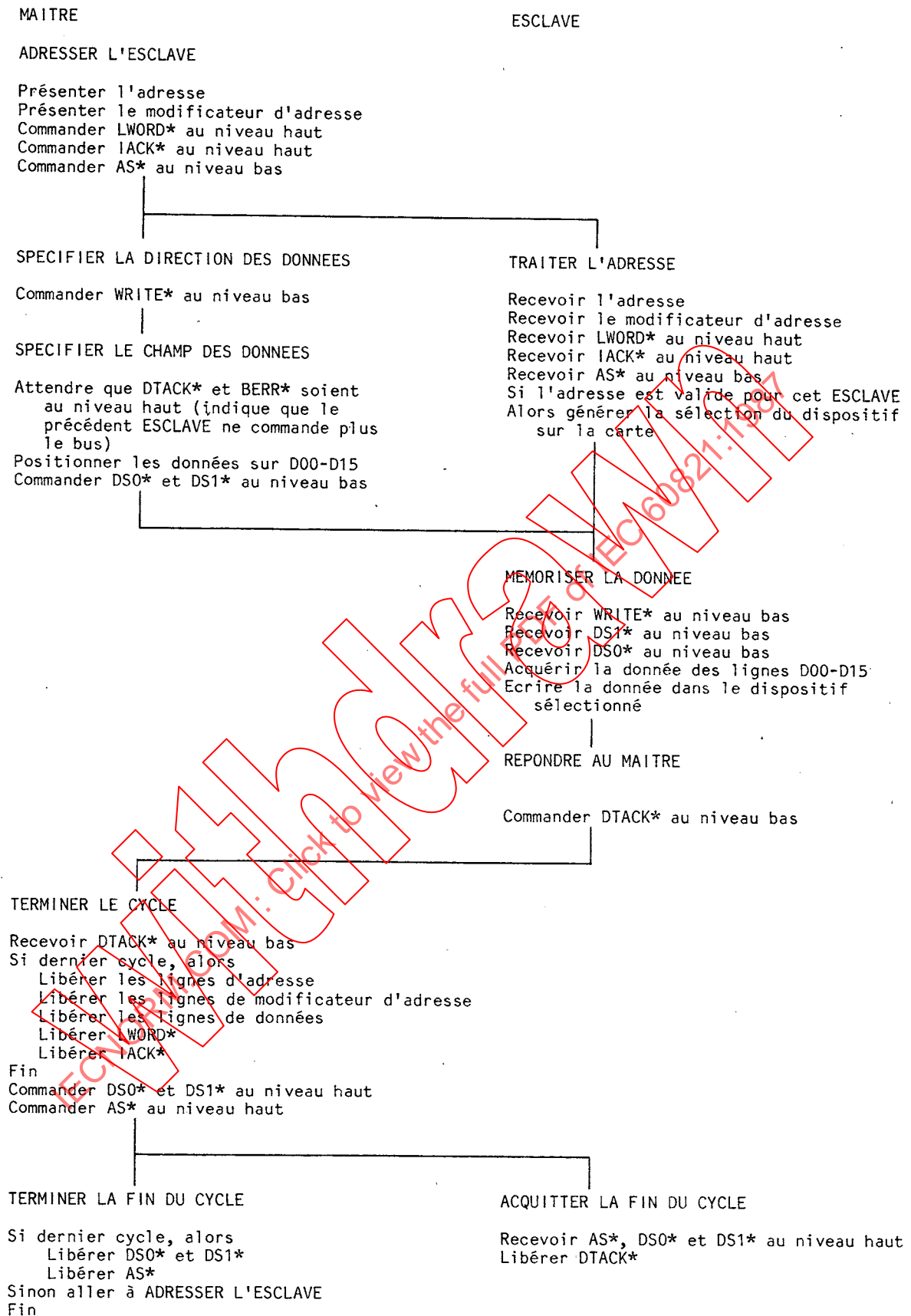


FIG. 2-9. - Un exemple de cycle d'écriture d'un double octet.

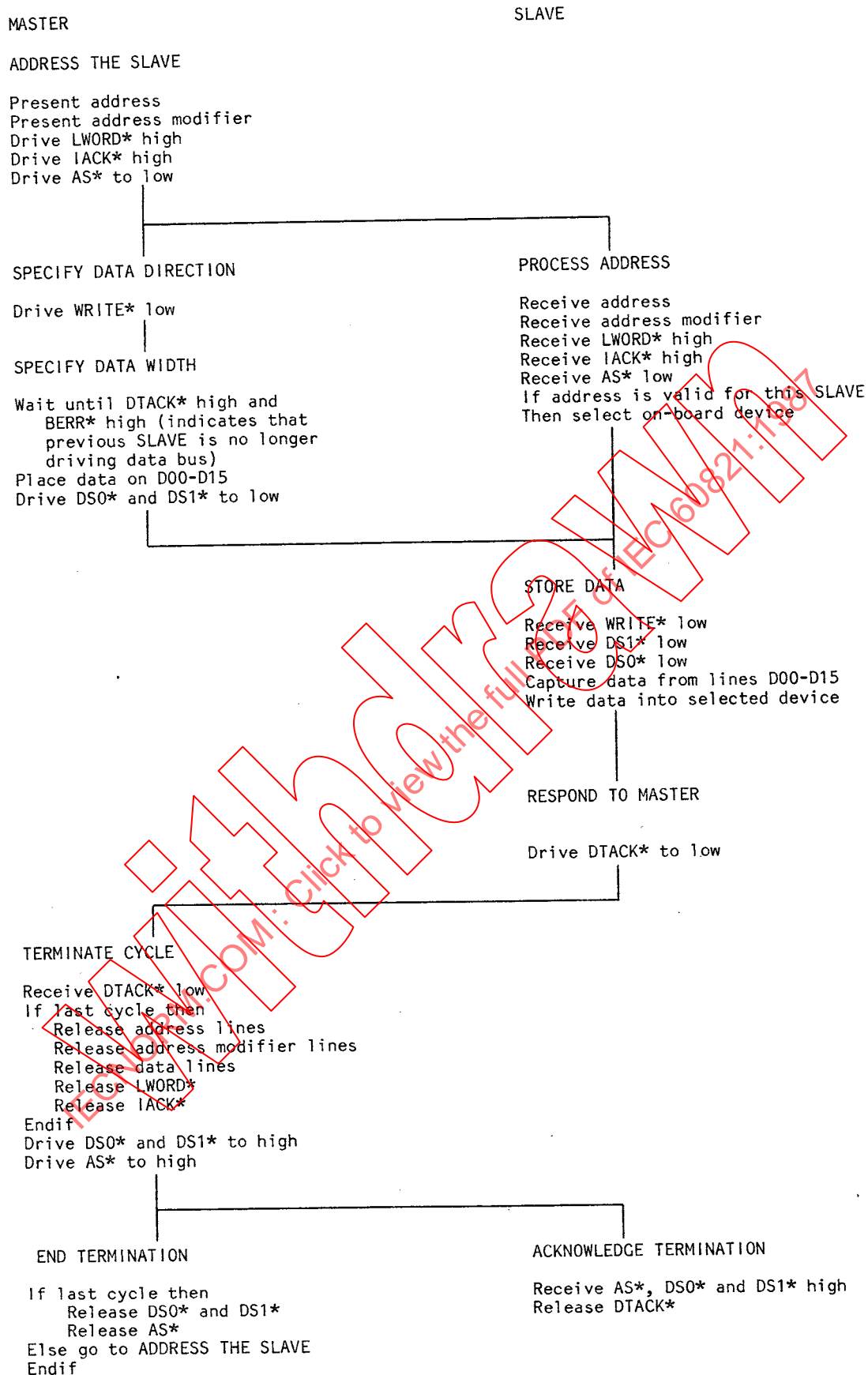


FIG. 2-9. - An example of a double byte write cycle.

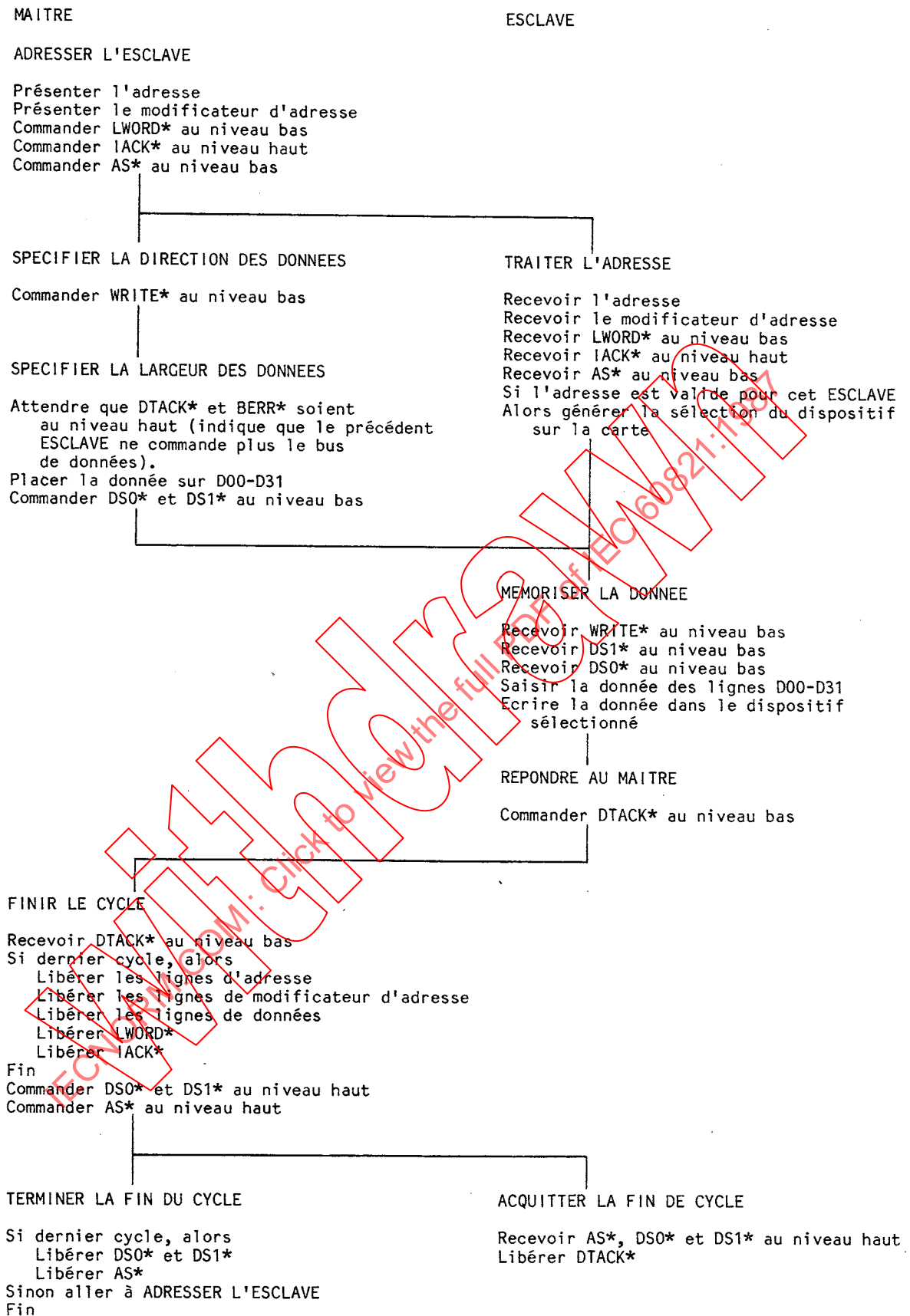


FIG. 2-10. - Un exemple de cycle d'écriture d'un quadruple octet.

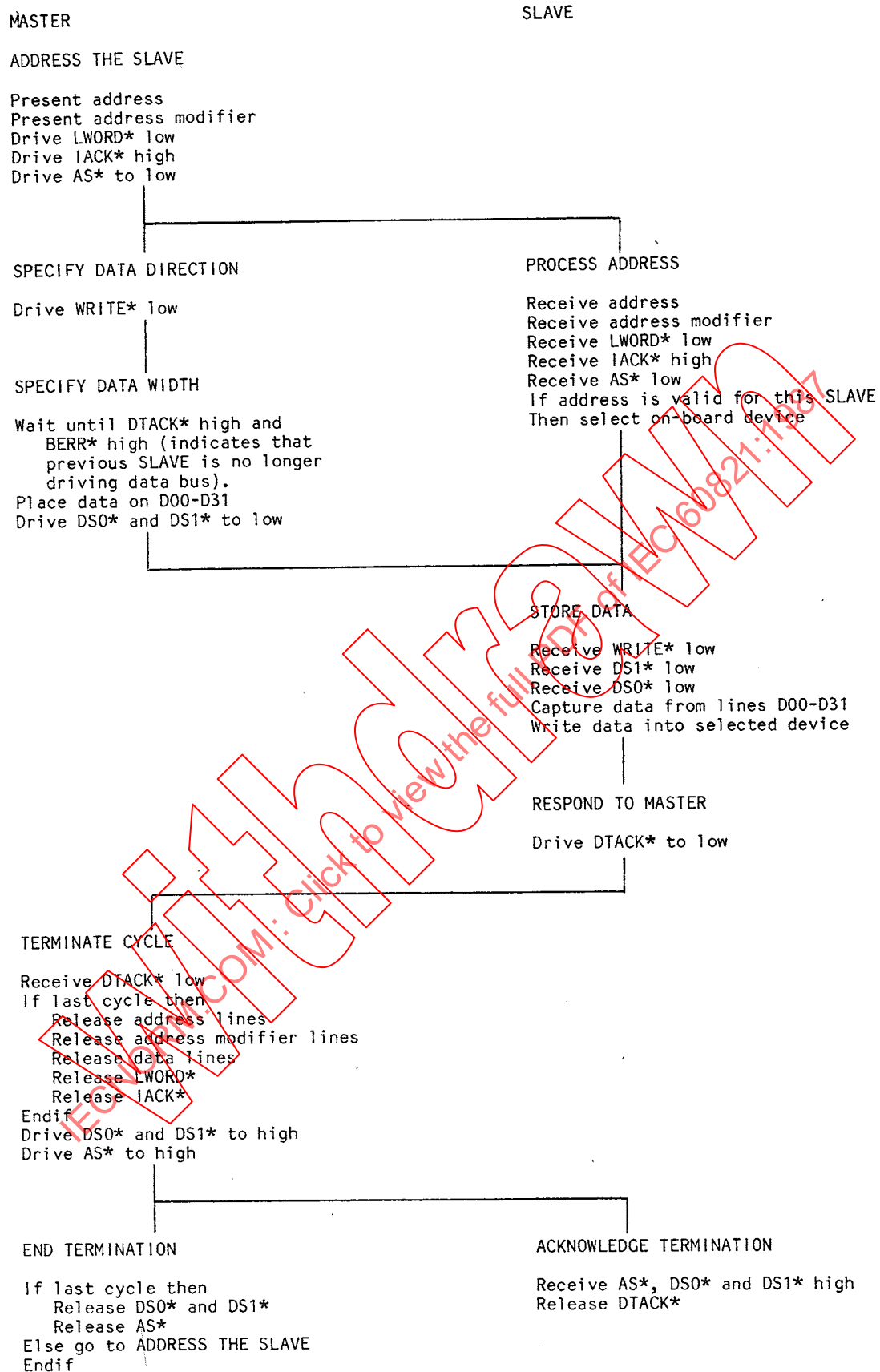


FIG. 2-10. - An example of a quad byte write cycle.

2.4.2 Anticipation d'adresse

Le fait que le BUS CEI 821 ait des signaux de validation d'adresse et de donnée permet au MAITRE de diffuser l'adresse pour le cycle suivant avant que le transfert de données du cycle précédent soit terminé. Cela s'appelle "l'anticipation d'adresse".

AUTORISATION 2.8:

Dès qu'un ESCLAVE commande DTACK* ou BERR* au niveau bas, le MAITRE PEUT changer l'adresse et, après avoir autorisé AS* à passer au niveau haut pendant un temps minimal, commander de nouveau AS* au niveau bas.

Par exemple, lorsqu'un ESCLAVE commande DTACK* ou BERR* au niveau bas dans un cycle de lecture, le MAITRE peut diffuser une nouvelle adresse sur le bus pendant qu'il lit les données du bus. Cela provoque un empiètement d'un cycle sur le suivant, et permet des vitesses de transfert plus élevées sur le BUS CEI 821.

REGLE 2.18:

Tous les ESCLAVES DOIVENT inclure la possibilité d'anticipation d'adresse sans perte de données ou mauvais fonctionnement.

OBSERVATION 2.32:

L'ESCLAVE qui répond peut reconnaître son adresse et répondre très rapidement sur les lignes DTACK* ou BERR*. Puisque le MAITRE a le droit d'enlever l'adresse immédiatement après que l'ESCLAVE qui répond eut commandé DTACK* ou BERR* au niveau bas, tous les ESCLAVES qui ne répondent pas pourraient être incapables de décoder l'information d'adresse avant que le MAITRE ne l'enlève du bus.

SUGGESTION 2.4:

Concevoir des ESCLAVES qui mémorisent l'information d'adresse sur le front descendant de AS*.

OBSERVATION 2.33:

Du fait qu'un MAITRE peut diffuser une nouvelle adresse pendant que le cycle précédent se termine, le concepteur doit s'assurer que l'activation du deuxième signal de validation de données ne rend pas invalide la première adresse, si celle-ci est encore nécessaire pour la logique de la carte afin de maintenir les données sur le bus.

AUTORISATION 2.9:

Des MAITRES PEUVENT être conçus sans anticipation d'adresse (c'est-à-dire qu'ils PEUVENT attendre que l'ESCLAVE qui répond relâche DTACK* et BERR* avant de commander AS* au niveau bas pour le cycle suivant).

OBSERVATION 2.34:

Un MAITRE pourrait commander AS* au niveau bas pour un nouveau cycle avant de commander les validations de données du cycle précédent au niveau haut. A cause de cela, il POURRAIT y avoir une période où la validation d'adresse du nouveau cycle et au moins un des signaux de validation de donnée du cycle précédent soient au niveau bas pendant l'anticipation du cycle.

2.4.2 Address pipelining

The IEC 821 BUS strobes the address and data with separate strobe signals. This allows a MASTER to broadcast the address for the next cycle while the data transfer for the previous cycle is still in progress. This is called "address pipelining".

PERMISSION 2.8:

As soon as a SLAVE drives DTACK* or BERR* low, the MASTER MAY change the address, and; after driving AS* high for a minimum time, drive AS* low again.

For example, when a SLAVE drives DTACK* or BERR* low on a read cycle, the MASTER can broadcast a new address on the address bus while it is reading the data from the bus. This amounts to an overlapping of one cycle with the next one, and permits greater speeds on the IEC 821 BUS.

RULE 2.18:

All SLAVES MUST be designed to accommodate address pipelining without loss of data or erroneous operation.

OBSERVATION 2.32:

The responding SLAVE might recognize its address and respond very quickly on the DTACK* or BERR* lines. Since the MASTER is permitted to remove the address after the responding SLAVE drives DTACK* or BERR* low, non-responding SLAVES might not be able to decode the addressing information before the MASTER removes it from the bus.

SUGGESTION 2.4:

Design SLAVES to capture the addressing information on the falling edge of AS*.

OBSERVATION 2.33:

Because a MASTER might broadcast a new address while a previous cycle is finishing, the designer needs to ensure that the assertion of the second address strobe does not invalidate the first address if it is still needed by on-board logic to maintain the data on the bus.

PERMISSION 2.9:

MASTERS MAY be designed without the ability for address pipelining (e.g. they MAY wait until the responding SLAVE releases DTACK* and BERR* before driving AS* low for the next cycle).

OBSERVATION 2.34:

A MASTER might drive AS* low for a new cycle before it drives data strobes high from the previous cycle. Because of this, there might be a period when the address strobe for the new cycle, as well as at least one data strobe from the previous cycle coincide during the cycle overlap.

SUGGESTION 2.5:

Concevoir des ESCLAVES qui déclenchent les transferts de données vers et à partir du bus lorsque au moins l'un des signaux de validation de donnée est au niveau bas et que DTACK* et BERR* sont au niveau haut au lieu d'un niveau bas simultané sur les validations d'adresse et de donnée.

2.5 Acquisition du bus de transfert de données**REGLÉ 2.19:**

Avant de transférer des données sur le DTB, un MAITRE DOIT obtenir l'autorisation de l'utiliser.

Plusieurs MAITRES peuvent vouloir utiliser le DTB en même temps. Le procédé qui détermine quel MAITRE peut utiliser le DTB s'appelle arbitrage et est discuté dans le chapitre 3. Parce que l'arbitrage est très lié à l'opération du DTB, il est présenté brièvement ici.

La figure 2-11, page 130, présente deux séquences possibles où un MAITRE (appelé MAITRE A) termine l'utilisation du DTB et permet à l'arbitrage d'avoir lieu.

Dans l'exemple 1, le MAITRE A, dans son dernier transfert, indique qu'il n'a plus besoin du DTB. Il procède en faisant libérer la ligne de signal bus occupé (BBSY*) par son DEMANDEUR. Puisque le MAITRE A annonce par anticipation que le DTB sera bientôt disponible, l'arbitrage se fait pendant le dernier transfert de données. L'arbitrage est terminé et le MAITRE B reçoit l'autorisation d'utiliser le DTB avant que le MAITRE A ait terminé son cycle mais il attend jusqu'à ce que le MAITRE A libère AS*. (Cela assure que le MAITRE B ne commencera pas à commander le DTB avant que le MAITRE A ait terminé son dernier transfert de données.)

Dans l'exemple 2, le MAITRE A attend que son dernier transfert soit terminé (c'est-à-dire après que AS* eut été libéré) avant de libérer BBSY*. Dans ce cas, le DTB est au repos pendant que l'arbitrage a lieu. Le bus est alors octroyé au MAITRE B et puisque AS* est déjà au niveau haut, il commence à utiliser le DTB tout de suite.

REGLÉ 2.20:

Lorsqu'un DEMANDEUR d'un MAITRE libère BBSY* au niveau haut, le MAITRE NE DOIT PAS commander AS* du niveau haut au niveau bas (c'est-à-dire il NE DOIT commencer aucun nouveau cycle) jusqu'à ce que son DEMANDEUR ait reçu une nouvelle allocation du bus.

SUGGESTION 2.5:

Design SLAVES to initiate data transfers to and from the bus when one or both data strobes are low, and DTACK* and BERR* are high, instead of a simultaneous low level on the address strobe and data strobes.

2.5 Data Transfer Bus acquisition**RULE 2.19:**

Before transferring any data on the DTB, a MASTER MUST get permission to use it.

Several MASTERS might want to use the DTB at the same time. The process which determines which MASTER can use the DTB is called arbitration and is discussed in Chapter 3. Because arbitration is closely tied to the operation of the DTB, it is briefly described here.

Figure 2-11, page 131, provides two examples that show possible sequences when a MASTER (called MASTER A) finishes using the DTB and allows arbitration to take place.

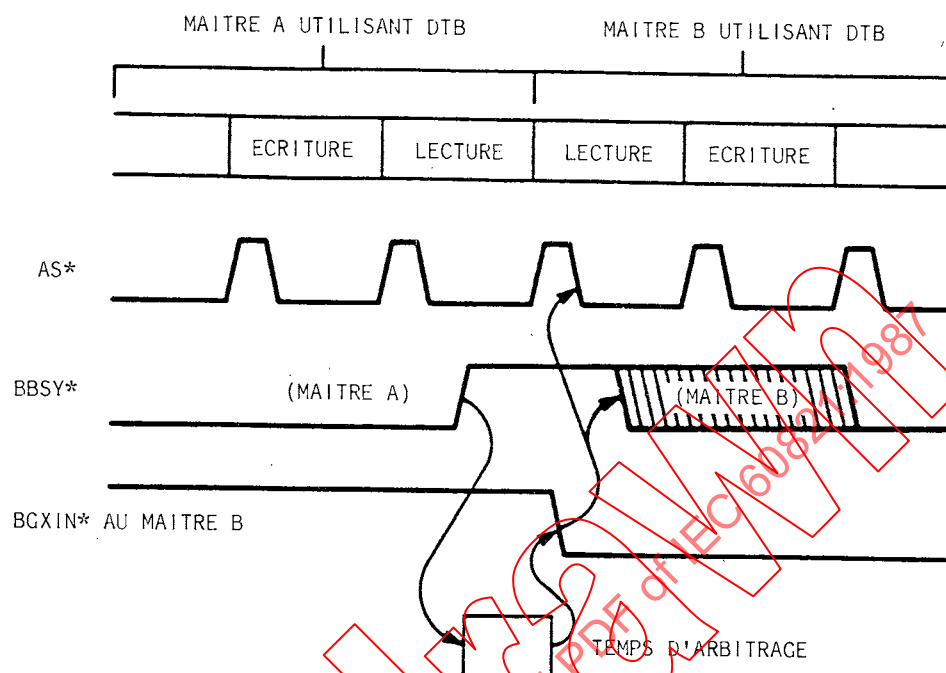
In Example 1, MASTER A, partway into its last transfer, indicates that it no longer needs the DTB. It does this by having its REQUESTER release the bus busy (BBSY*) signal line. Since MASTER A gives this early notice that the DTB will soon be available, the arbitration is done during the last data transfer. The arbitration is completed and MASTER B is granted permission to use the DTB before MASTER A has finished its cycle, but it waits until MASTER A releases AS*. (This assures that MASTER B will not start driving the DTB before MASTER A has finished with its last data transfer.)

In Example 2, MASTER A waits until after its last transfer (i.e., after AS* is released) before releasing BBSY*. In this case the DTB is idle while the arbitration is done. MASTER B is then granted the bus and, since AS* is already high, it begins using the DTB immediately.

RULE 2.20:

Once a MASTER'S REQUESTER releases BBSY* to high, the MASTER MUST NOT drive AS* from high to low (i.e. it MUST NOT begin any new cycle) until its REQUESTER receives a new bus grant.

Exemple 1: arbitrage PENDANT le dernier transfert de données



Exemple 2: Arbitrage APRES le dernier transfert de données

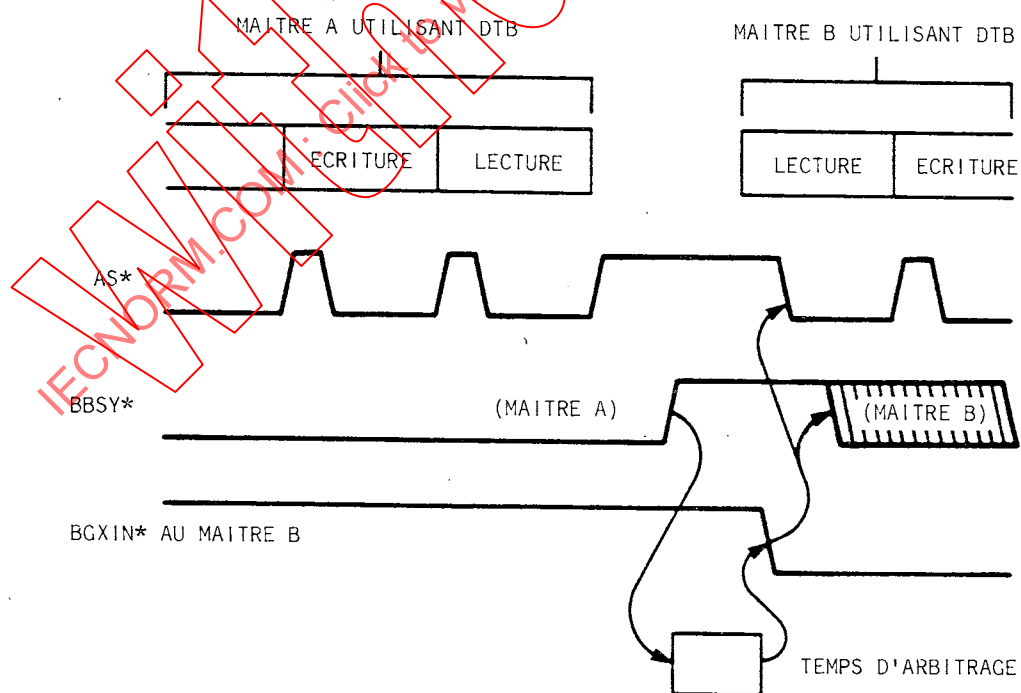
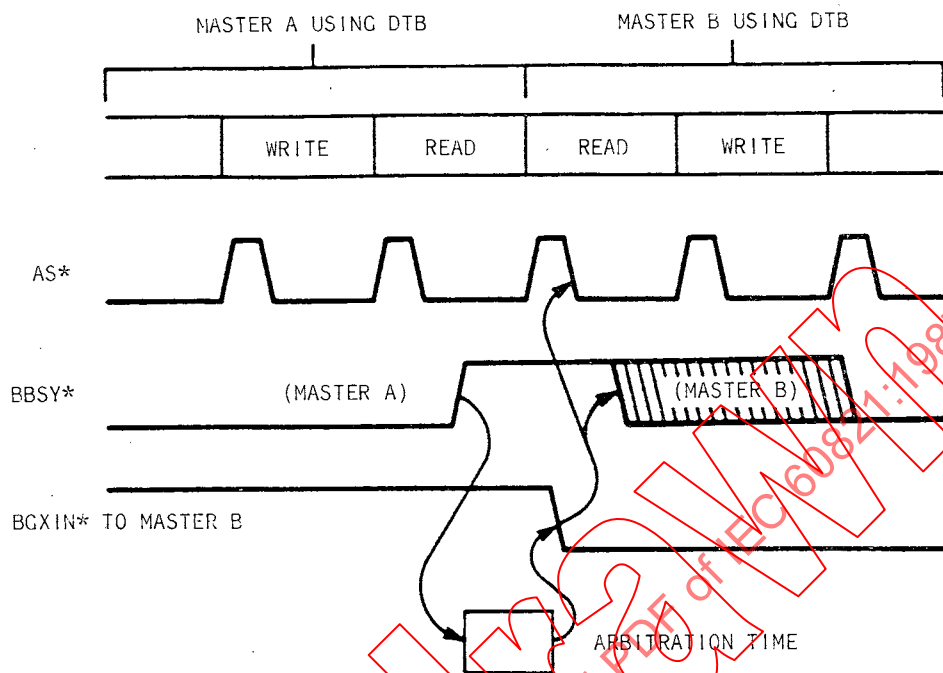


FIG. 2-11. - Séquence d'échange du bus de transfert de données du MAITRE.

Example 1: Arbitration DURING the last data transfer



Example 2: Arbitration AFTER the last data transfer

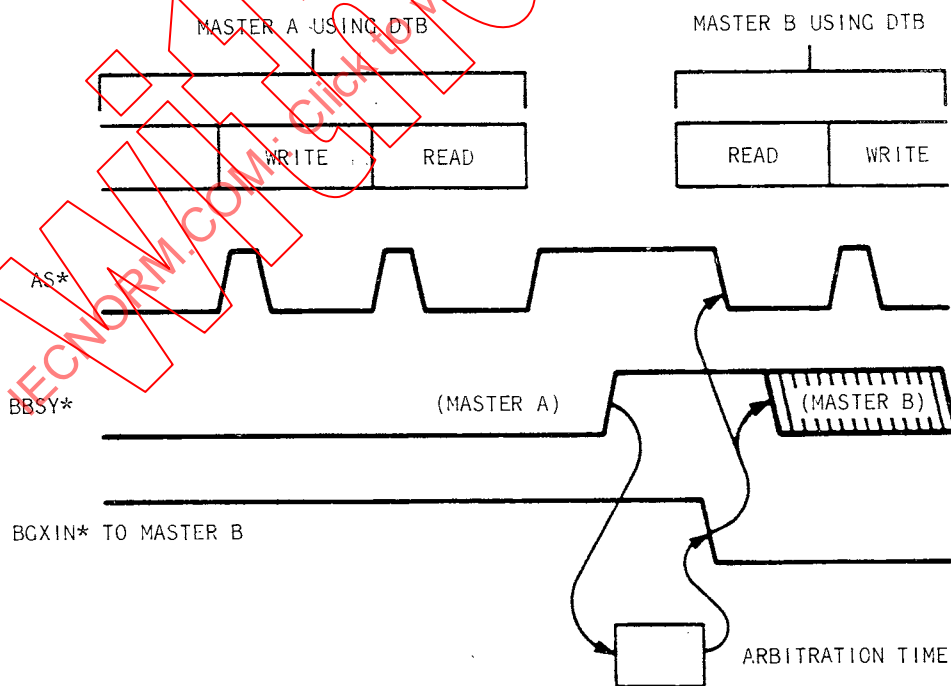


FIG. 2-11. - Data Transfer Bus MASTER exchange sequence.

2.6 Règles et observations de séquençement du DTB

Cette section décrit les REGLES et OBSERVATIONS de séquençement qui gèrent le comportement des MAITRES et ESCLAVES. Ces informations de chronologie sont données sous la forme de figures et de tableaux:

Les tableaux 2-17 et 2-18 font référence à un tableau et à des chronogrammes qui spécifient le fonctionnement du MAITRE, de l'ESCLAVE et du DETECTEUR D'ACCES.

Le tableau 2-19 définit les différents mnémoniques utilisés dans cette norme.

Les tableaux 2-20 à 2-22 spécifient l'utilisation des signaux du DTB.

Les tableaux 2-23 à 2-28 spécifient les paramètres de temps du DTB. (Les numéros de référence utilisés dans les tableaux 2-25 à 2-28 correspondent aux numéros des paramètres de temps dans les tableaux 2-23 et 2-24.)

Les figures 2-12 à 2-15, pages 168 à 174, spécifient les REGLES et les OBSERVATIONS de chronologie durant la diffusion d'adresse.

Les figures 2-16 à 2-21, pages 176 à 184, spécifient les REGLES et les OBSERVATIONS de chronologie des MAITRES, des ESCLAVES et des DETECTEURS D'ACCES pendant le transfert de données.

Les figures 2-22 à 2-24, pages 186 à 188, spécifient les REGLES et les OBSERVATIONS de chronologie des MAITRES et des ESCLAVES entre les cycles DTB.

La figure 2-25, page 188, présente le chronogramme du MAITRE, de l'ESCLAVE et du LIMITEUR DE TEMPS D'OCCUPATION DU BUS pour un cycle de dépassement de temps d'occupation du bus.

La figure 2-26, page 190, présente la chronologie du transfert du contrôle du DTB.

Afin de satisfaire aux REGLES de chronologie spécifiées, les concepteurs doivent tenir compte des cas les plus défavorables de délais de propagation des émetteurs et récepteurs du bus utilisés sur leurs cartes de BUS CEI 821. Le délai de propagation des émetteurs dépend de leur charge de sortie et les spécifications des fabricants ne donnent pas toujours assez d'informations pour calculer les délais de propagation avec différentes charges. Pour aider le concepteur de la carte de BUS CEI 821, quelques suggestions sont données dans le chapitre 6.

Les OBSERVATIONS spécifient la chronologie des transitions des signaux d'entrée. Cette chronologie peut servir de référence tant que les REGLES de charge du fond de panier spécifiées dans le chapitre 6 ne sont pas violées. Les REGLES pour les adaptateurs de bus du chapitre 6 garantissent que les paramètres de temps des lignes de signal qui sont libérées après avoir été commandées sont respectés.

Généralement, pour chaque REGLE de chronologie, il y a une OBSERVATION correspondante. Cependant, le temps qui est garanti par l'OBSERVATION pourrait différer du temps spécifié par la REGLE. Par exemple, une analyse attentive des chronogrammes indique que le

2.6 DTB timing rules and observations

This section describes the timing RULES and OBSERVATIONS that govern the behavior of MASTERS and SLAVES. This timing information is in the form of figures and tables:

Tables 2-17 and 2-18 list a timing table and timing diagrams that specify MASTER, SLAVE and LOCATION MONITOR operation.

Table 2-19 defines the various mnemonics that are used in this section.

Tables 2-20 through 2-22 specify the use of the DTB signals.

Tables 2-23 through 2-28 specify the timing parameters of the DTB. (The reference numbers used in Tables 2-25 through 2-28 correspond to the timing parameter numbers in Tables 2-23 and 2-24.)

Figures 2-12 through 2-15, pages 169 to 175, specify the timing RULES and OBSERVATIONS during address broadcasting time.

Figures 2-16 through 2-21, pages 177 to 185, specify the timing RULES and OBSERVATIONS for MASTERS, SLAVES, and LOCATION MONITORS during data transfer time.

Figures 2-22 through 2-24, pages 187 to 189, specify the timing RULES and OBSERVATIONS for MASTERS and SLAVES between DTB cycles.

Figure 2-25, page 189, is the timing diagram for the MASTER, SLAVE and the BUS TIMER during a timed-out cycle.

Figure 2-26, page 191, shows the timing during the mastership transfer of the DTB.

In order to meet the specified timing RULES, board designers need to take into account the worst case propagation delays of the bus drivers and receivers used on their IEC 821 BUS boards. The propagation delay of the drivers depends on their output loads and manufacturers' specifications do not always give enough information to calculate the propagation delays under various loads. To help the IEC 821 BUS board designer, some suggestions are offered in Chapter 6.

The OBSERVATIONS specify the timing of incoming lines signal transitions. These times can be relied upon as long as the backplane loading RULES in Chapter 6 are not violated. The RULES for the bus terminators in Chapter 6 guarantee that the timing parameters for signal lines that are released after they have been driven, are met.

Typically, for each timing RULE there is a corresponding OBSERVATION. However, the time that is guaranteed in the OBSERVATION might differ from the time specified by the RULE. For example, a careful inspection of the timing diagrams shows that the MASTER is

MAITRE est contraint d'assurer 35 ns de temps d'établissement des adresses et des données, mais l'ESCLAVE n'est garanti que de 10 ns. Cela est dû au fait que les émetteurs d'adresse et de donnée du bus ne peuvent pas toujours commander complètement les lignes de signaux du fond de panier du niveau bas au niveau haut à travers le seuil avant que la transition ne soit réfléchie à l'extrémité du fond de panier. Le front descendant de validation d'adresse et de donnée, cependant, traverse généralement le seuil de 0,8 V sans qu'une réflexion ne le perturbe. Le temps d'établissement qui résulte sur l'ESCLAVE est le temps d'établissement du MAITRE moins deux temps de propagation du bus.

Une notation spéciale a été utilisée pour décrire la chronologie de la validation de donnée. Les deux validations de données (DS0* et DS1*) n'auront pas toujours leurs transitions simultanées. Pour ces chronogrammes, DSA* représente la première validation de donnée à avoir une transition (que ce soit DS0* ou DS1*) et DSB* la seconde. La ligne pointillée montre que lorsque les signaux de validation de donnée sont stables, la première validation de donnée à transition descendante n'est pas forcément la première à exécuter la transition montante, c'est-à-dire DSA* peut représenter DS0* sur le front descendant et DS1* sur le front montant.

required to provide 35 ns of address and data set-up time, but the SLAVE is only guaranteed 10 ns. This is because the address and data bus drivers are not always able to drive the backplane's signal lines completely through the threshold region from low to high until the transition propagates to the end of the backplane and is reflected back. The falling edge of the address and data strobes, however, typically cross the 0.8 V threshold without being disturbed by reflection. The resulting set-up time at the SLAVE is the MASTER'S set-up time less two bus propagation times.

A special notation has been used to describe the data strobe timing. The two data strobes (DS0* and DS1*) will not always make their transitions simultaneously. For purposes of these timing diagrams, DSA* represents the first data strobe to make its transition (whether that is DS0* or DS1*) and DSB* represents the second data strobe to make a transition (whether that is DS1* or DS0*). The broken line shown while the data strobes are stable is to indicate that the first data strobe to make a falling transition might not be the first to make its rising transition, i.e. DSA* can represent DS0* on its falling edge and DS1* on its rising edge.

Tableau 2-17

*Chronogrammes qui définissent le fonctionnement du MAITRE,
de l'ESCLAVE et du DETECTEUR D'ACCES*

(Voir tableau 2-23 pour les valeurs de temps)

Mnémonique	Type de cycles	Chronogrammes de diffusion d'adresse	Chronogrammes de transfert de données
ADO	UNIQUEMENT D'ADRESSAGE	Figure 2-12	N/A
D08(E0)	Transferts d'octet unique pair LECTURE OCTET(0) LECTURE OCTET(2) LECTURE OCTET(0) LECTURE OCTET(2)	Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-16 Figure 2-16 Figure 2-18 Figure 2-18
D08(E0) ou D08(0)	Transferts d'octet unique impair LECTURE OCTET(0) LECTURE OCTET(3) ECRITURE OCTET(1) ECRITURE OCTET(3)	Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-16 Figure 2-16 Figure 2-18 Figure 2-18
D16	Transferts de deux octets LECTURE OCTETS(0-1) LECTURE OCTETS(2-3) ECRITURE OCTETS(0-1) ECRITURE OCTETS(2-3)	Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-17 Figure 2-17 Figure 2-19 Figure 2-19
D32	Transferts de quatre octets LECTURE OCTETS(0-3) ECRITURE OCTETS(0-3)	Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-17 Figure 2-19
D08(E0):BLT	Transferts de blocs par octet LECTURE DE BLOCS PAR OCTET ECRITURE DE BLOCS PAR OCTET	Figures 2-13, 2-14 Figures 2-13, 2-14	Figure 2-16 Figure 2-18
D16:BLT	Transferts de blocs par double octet LECTURE DE BLOCS PAR DOUBLE OCTET ECRITURE DE BLOCS PAR DOUBLE OCTET	Figures 2-13, 2-14 Figures 2-13, 2-14	Figure 2-17 Figure 2-19
D32:BLT	Transferts de blocs par quadruple octet LECTURE DE BLOCS PAR QUADRUPLE OCTET ECRITURE DE BLOCS PAR QUADRUPLE OCTET	Figures 2-13, 2-14 Figures 2-13, 2-14	Figure 2-17 Figure 2-19
D08(E0):RMW	Transferts RMW pour octet unique LECTURE-MODIFICATION-ECRITURE OCTET(0) LECTURE-MODIFICATION-ECRITURE OCTET(1) LECTURE-MODIFICATION-ECRITURE OCTET(2) LECTURE-MODIFICATION-ECRITURE OCTET(3)	Figures 2-13, 2-15 Figures 2-13, 2-15 Figures 2-13, 2-15 Figures 2-13, 2-15	Figure 2-20 Figure 2-20 Figure 2-20 Figure 2-20
D16:RMW	Transferts RMW pour double octet LECTURE-MODIFICATION-ECRITURE OCTET(0-1) LECTURE-MODIFICATION-ECRITURE OCTET(2-3)	Figures 2-13, 2-15 Figures 2-13, 2-15	Figure 2-21 Figure 2-21
D32:RMW	Transferts RMW pour quadruple octet LECTURE-MODIFICATION-ECRITURE OCTET(0-3)	Figures 2-13, 2-15	Figure 2-21
D32:UAT	Transferts non alignés LECTURE OCTET(0-2) LECTURE OCTET(1-3) LECTURE OCTET(1-2) ECRITURE OCTET(0-2) ECRITURE OCTET(1-3) ECRITURE OCTET(1-2)	Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-16 Figure 2-16 Figure 2-17 Figure 2-18 Figure 2-18 Figure 2-19

Table 2-17

*Timing diagrams that define MASTER, SLAVE
and LOCATION MONITOR operation*

(See Table 2-23 for timing values)

Mnemonic	Type of cycles	Address broadcast timing diagrams	Data transfer timing diagrams
ADO	ADDRESS ONLY	Figure 2-12	N/A
D08(E0)	Single even byte transfers BYTE(0) READ BYTE(2) READ BYTE(0) WRITE BYTE(2) WRITE	Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-16 Figure 2-16 Figure 2-18 Figure 2-18
D08(E0) or D08(0)	Single odd byte transfers BYTE(0) READ BYTE(3) READ BYTE(1) WRITE BYTE(3) WRITE	Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-16 Figure 2-16 Figure 2-18 Figure 2-18
D16	Double byte transfers BYTE(0-1) READ BYTE(2-3) READ BYTE(0-1) WRITE BYTE(2-3) WRITE	Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-17 Figure 2-17 Figure 2-19 Figure 2-19
D32	Quad byte transfers BYTE(0-3) READ BYTE(0-3) WRITE	Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-17 Figure 2-19
D08(E0):BLT	Single byte block transfers SINGLE BYTE BLOCK READ SINGLE BYTE BLOCK WRITE	Figures 2-13, 2-14 Figures 2-13, 2-14	Figure 2-16 Figure 2-18
D16:BLT	Double byte block transfers DOUBLE BYTE BLOCK READ DOUBLE BYTE BLOCK WRITE	Figures 2-13, 2-14 Figures 2-13, 2-14	Figure 2-17 Figure 2-19
D32:BLT	Quad byte block transfers QUAD BYTE BLOCK READ QUAD BYTE BLOCK WRITE	Figures 2-13, 2-14 Figures 2-13, 2-14	Figure 2-17 Figure 2-19
D08(E0):RMW	Single byte RMW transfers BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE	Figures 2-13, 2-15 Figures 2-13, 2-15 Figures 2-13, 2-15 Figures 2-13, 2-15	Figure 2-20 Figure 2-20 Figure 2-20 Figure 2-20
D16:RMW	Double byte RMW transfers BYTE(0-1) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE	Figures 2-13, 2-15 Figures 2-13, 2-15	Figure 2-21 Figure 2-21
D32:RMW	Quad byte RMW transfers BYTE(0-3) READ-MODIFY-WRITE	Figures 2-13, 2-15	Figure 2-21
D32:UAT	Unaligned transfers BYTE(0-2) READ BYTE(1-3) READ BYTE(1-2) READ BYTE(0-2) WRITE BYTE(1-3) WRITE BYTE(1-2) WRITE	Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13 Figures 2-12, 2-13	Figure 2-16 Figure 2-16 Figure 2-17 Figure 2-18 Figure 2-18 Figure 2-19

Tableau 2-18

*Chronogrammes qui définissent le fonctionnement du MAITRE,
de l'ESCLAVE, du DETECTEUR D'ACCES,
du CONTROLEUR D'INTERRUPTION
et du GENERATEUR D'INTERRUPTION pendant TOUS les cycles*

(Voir tableau 2-23 pour les valeurs de temps)

Mnémonique	Type de cycles	Chronogrammes de diffusion d'adresse	Chronogrammes de transfert de données
TOUS	TOUS	Figure 2-12	N/A

Les tableaux 2-20, 2-21 et 2-22 montrent comment les différentes lignes de signal du DTB sont utilisées pour diffuser les adresses et pour transférer les données. Ces tableaux sont référencés par les différents chronogrammes qui les suivent. Pour garder ces tableaux compacts, des mnémoniques sont utilisées pour décrire quand et comment les différentes lignes sont commandées. Ces mnémoniques sont définis dans le tableau 2-19.

Tableau 2-19

*Définitions des mnémoniques utilisés
dans les tableaux 2-20, 2-21 et 2-22*

Mnémonique	Description	Commentaires
DVBM	COMMANDEES VALIDES PAR LE MAITRE	REGLE 2.21: Le MAITRE DOIT commander les lignes DVBM à un niveau valide
DLBM	COMMANDEES AU NIVEAU BAS PAR LE MAITRE	REGLE 2.22: Le MAITRE DOIT commander les lignes DLBM à un niveau bas
DHBM	COMMANDEES AU NIVEAU BAS PAR LE MAITRE	REGLE 2.23: Le MAITRE DOIT commander les lignes DHBM à un niveau haut
dxbm?	COMMANDEES AU NIVEAU HAUT PAR LE MAITRE?	AUTORISATION 2.10: Le MAITRE PEUT commander les lignes dxbm? à un niveau haut. REGLE 2.24: Le MAITRE NE DOIT PAS commander les lignes dxbm? au niveau bas
dxbm?	COMMANDEES PAR LE MAITRE?	AUTORISATION 2.11: Le MAITRE PEUT commander les lignes dxbm? ou il PEUT laisser ces lignes non activées. (Lorsque les lignes dxbm? sont commandées, elles ne transportent pas d'information valide.)
DVBS	COMMANDEES VALIDES PAR L'ESCLAVE	REGLE 2.25: L'ESCLAVE DOIT commander les lignes DVBS à un niveau valide

Table 2-18

*Timing diagrams that define MASTER,
SLAVE, LOCATION MONITOR, INTERRUPT HANDLER
and INTERRUPTER operation during ALL cycles*

(See Table 2-23 for timing values)

Mnemonic	Type of cycle	Address broadcast timing diagrams	Data transfer timing diagrams
ALL	ALL	Figure 2-12	N/A

Tables 2-20, 2-21, and 2-22 show how the various signal lines of the DTB are used to broadcast addresses and to transfer data. These tables are referenced by the various timing diagrams that follow. In order to keep these tables compact, mnemonics are used to describe when and how the various lines are driven. These mnemonics are defined in Table 2-19.

Table 2-19

*Definitions of mnemonics
used in Tables 2-20, 2-21, and 2-22*

Mnemonic	Description	Comments
DVBM	DRIVEN VALID BY MASTER	RULE 2.21: The MASTER MUST drive DVBM lines to a valid level
DLBM	DRIVEN LOW BY MASTER	RULE 2.22: The MASTER MUST drive DLBM lines to a low level
DHBM	DRIVEN HIGH BY MASTER	RULE 2.23: The MASTER MUST drive DHBM lines to a high level
dhbm?	DRIVEN HIGH BY MASTER?	PERMISSION 2.10: The MASTER MAY drive dhbm? lines high. RULE 2.24: The MASTER MUST NOT drive dhbm? lines low
dxbm?	DRIVEN BY MASTER?	PERMISSION 2.11: The MASTER MAY drive dxbm? lines, or it MAY leave these lines undriven. (When dxbm? lines are driven, they carry no valid information.)
DVBS	DRIVEN VALID BY SLAVE	RULE 2.25: The SLAVE MUST drive DVBS lines to a valid level

Tableau 2-19 (suite)

Mnémonique	Description	Commentaires
dxbs?	COMMANDEES PAR L'ESCLAVE?	AUTORISATION 2.12: L'ESCLAVE DOIT commander les lignes dxbs? ou il PEUT laisser ces lignes non activées. (Lorsque les lignes dxbs? sont commandées, elles ne transportent pas d'information valide.)
DVBB	COMMANDEES VALIDES PAR L'ESCLAVE ET PAR LE MAITRE	REGLE 2.26: Pendant la partie "lecture" du cycle lecture-modification-écriture, l'ESCLAVE DOIT commander ces lignes DVBB avec des données valides. Pendant la partie "écriture" du cycle lecture-modification-écriture, le MAITRE DOIT commander ces lignes DVBB avec des données valides.
dxbb?	COMMANDEES VALIDES PAR L'ESCLAVE ET PAR LE MAITRE?	AUTORISATION 2.13: Pendant la partie "lecture" du cycle lecture-modification-écriture, l'ESCLAVE PEUT commander ces lignes dxbb? ou il PEUT laisser ces lignes non commandées. Pendant la partie "écriture" du cycle lecture-modification-écriture, le MAITRE PEUT commander les lignes dxbb? ou il PEUT laisser ces lignes non commandées. (Lorsque les lignes dxbb? sont commandées, elles ne transportent pas d'information valide.)

Tableau 2-20

Utilisation des lignes d'adressage pour sélectionner un groupe de 4 octets

Mnémonique	Mode d'adressage	A02-A15 (Voir note ci-dessous)	A16-A23	A24-A31	IACK*
A16	COURT	DVBM	dxbm?	dxbm?	dhbm?
A24	STANDARD	DVBM	DVBM	dxbm?	dhbm?
A32	ETENDU	DVBM	DVBM	DVBM	dhbm?

Note:

A01 est utilisé conjointement avec LWORD* et les deux signaux de validation de donnée pour sélectionner lequel des quatre octets dans le groupe de 4 octets doit être accédé (voir tableau 2-21).

Table 2-19 (continued)

Mnemonic	Description	Comments
dxbs?	DRIVEN BY SLAVE?	PERMISSION 2.12: The SLAVE MAY drive dxbs? lines, or it MAY leave these lines undriven. (When dxbs? lines are driven, they carry no valid information.)
DVBB	DRIVEN VALID BY BOTH SLAVE AND MASTER	RULE 2.26: During the "read" portion of a read-modify-write cycle, the SLAVE MUST drive DVBB lines with valid data. During the "write" portion of a read-modify-write cycle, the MASTER MUST drive DVBB lines with valid data
dxbb?	DRIVEN BY BOTH SLAVE AND MASTER?	PERMISSION 2.13: During the "read" portion of a read-modify-write cycle, the SLAVE MAY drive dxbb? lines, or it MAY leave them undriven. During the "write" portion of a read-modify-write cycle, the MASTER MAY drive dxbb? lines, or it MAY leave them undriven. (When dxbb? lines are driven, they carry no valid information.)

Table 2-20

*Use of addressing lines to select
a 4-byte group*

Mnemonic	Addressing mode	A02-A15 (see note below)	A16-A23	A24-A31	IACK*
A16	SHORT	DVBM	dxbm?	dxbm?	dhbm?
A24	STANDARD	DVBM	DVBM	dxbm?	dhbm?
A32	EXTENDED	DVBM	DVBM	DVBM	dhbm?

Note:

A01 is used in conjunction with LWORD* and the two data strobes to select which of the four bytes within the 4-byte group is accessed (see Table 2-21).

Tableau 2-21

Utilisation des lignes DS1, DS0*, A01 et LWORD*
pour sélectionner le ou les octets dans le groupe de 4 octets*

Mnémonique	Type de cycles	DS1*	DS0*	A01	LWORD*
AD0	UNIQUEMENT D'ADRESSAGE	dhbm?	dhbm?	dxbm?	dxbm?
D08(E0)	Transferts d'octet unique pair LECTURE ou ECRITURE OCTET(0) LECTURE ou ECRITURE OCTET(2)	DLBM DLBM	dhbm? dhbm?	DLBM DHBM	dhbm? dhbm?
D08(E0) ou D08(0)	Transferts d'octet unique impair LECTURE ou ECRITURE OCTET(1) LECTURE ou ECRITURE OCTET(3)	dhbm? dhbm?	DLBM DLBM	DLBM DHBM	dhbm? dhbm?
D16	Transferts double octet LECTURE ou ECRITURE OCTET(0-1) LECTURE ou ECRITURE OCTET(2-3)	DLBM DLBM	DLBM DLBM	DLBM DHBM	dhbm? dhbm?
D32	Transferts quadruple octet LECTURE ou ECRITURE OCTET(0-3)	DLBM	DLBM	DLBM	DLBM
D08(E0):BLT	Transferts de blocs par octet LECTURE ou ECRITURE DE BLOCS PAR OCTET	Note 1			dhbm?
D16:BLT	Transferts de blocs par double octet LECTURE ou ECRITURE DE BLOCS PAR DOUBLE OCTET	DLBM	DLBM	Note 2	dhbm?
D32:BLT	Transferts de blocs par quadruple octet LECTURE ou ECRITURE DE BLOCS PAR QUADRUPLE OCTET	DLBM	DLBM	DLBM	DLBM
D08(E0):RMW	Transferts RMW pour octet unique LECTURE-MODIFICATION-ECRITURE OCTET(0) LECTURE-MODIFICATION-ECRITURE OCTET(1) LECTURE-MODIFICATION-ECRITURE OCTET(2) LECTURE-MODIFICATION-ECRITURE OCTET(3)	DLBM dhbm? DLBM dhbm?	dhbm? DLBM dhbm? DLBM	DLBM DLBM DHBM DHBM	dhbm? dhbm? dhbm? dhbm?
D16:RMW	Transferts RMW pour double octet LECTURE-MODIFICATION-ECRITURE OCTET (0-1) LECTURE-MODIFICATION-ECRITURE OCTET (2-3)	DLBM DLBM	DLBM DLBM	DLBM DHBM	dhbm? dhbm?
D32:RMW	Transferts RMW pour quadruple octet LECTURE-MODIFICATION-ECRITURE OCTET (0-3)	DLBM	DLBM	DLBM	DLBM
D32:UAT	Transfert non alignés LECTURE ou ECRITURE OCTET(0-2) LECTURE ou ECRITURE OCTET(1-3) LECTURE ou ECRITURE OCTET(1-2)	DLBM dhbm? DLBM	dhbm? DLBM DLBM	DLBM DLBM DHBM	DLBM DLBM DLBM

Notes:

- 1.- Pendant les transferts de blocs par octet, les deux signaux de validation de données sont commandés alternativement au niveau bas. N'importe lequel des deux signaux peut être commandé au niveau bas pendant le premier transfert. Si l'emplacement du premier octet accédé est l'OCTET(0) ou l'OCTET(2), alors le MAÎTRE commande au début DS1* au niveau bas. Si l'emplacement du premier octet accédé est l'OCTET(1) ou l'OCTET(3),

Table 2-21

Use of the DS1, DS0*, A01, and LWORD* lines
to select the byte(s) within a 4-byte group*

Mnemonic	Type of cycles	DS1*	DS0*	A01	LWORD*
AD0	ADDRESS ONLY	dhbm?	dhbm?	dxbm?	dxbm?
D08(E0)	Single even byte transfers BYTE(0) READ or WRITE BYTE(2) READ or WRITE	DLBM DLBM	dhbm? dhbm?	DLBM DHBM	dhbm? dhbm?
D08(E0) or D08(0)	Single odd byte transfers BYTE(1) READ or WRITE BYTE(3) READ or WRITE	dhbm? dhbm?	DLBM DLBM	DLBM DHBM	dhbm? dhbm?
D16	Double byte transfers BYTE(0-1) READ or WRITE BYTE(2-3) READ or WRITE	DLBM DLBM	DLBM DLBM	DLBM DHBM	dhbm? dhbm?
D32	Quad byte transfers BYTE(0-3) READ or WRITE	DLBM	DLBM	DLBM	DLBM
D08(E0):BLT	Single byte block transfers SINGLE BYTE BLOCK READ or WRITE	Note 1			dhbm?
D16:BLT	Double byte block transfers DOUBLE BYTE BLOCK READ or WRITE	DLBM	DLBM	Note 2	dhbm?
D32:BLT	Quad byte block transfers QUAD BYTE BLOCK READ or WRITE	DLBM	DLBM	DLBM	DLBM
D08(E0):RMW	Single byte RMW transfers BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE	DLBM dhbm? DLBM dhbm?	dhbm? DLBM dhbm? DLBM	DLBM DLBM DHBM DHBM	dhbm? dhbm? dhbm? dhbm?
D16:RMW	Double byte RMW transfers BYTE(0-3) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE	DLBM DLBM	DLBM DLBM	DLBM DHBM	dhbm? dhbm?
D32:RMW	Quad byte RMW transfers BYTE(0-3) READ-MODIFY-WRITE	DLBM	DLBM	DLBM	DLBM
D32:UAT	Unaligned transfers BYTE(0-2) READ or WRITE BYTE(1-3) READ or WRITE BYTE(1-2) READ or WRITE	DLBM dhbm? DLBM	dhbm? DLBM DLBM	DLBM DLBM DHBM	DLBM DLBM DLBM

Notes:

- During single byte block transfers, the two data strobes are alternately driven low. Either data strobe might be driven low on the first transfer. If the first accessed byte location is BYTE(0) or BYTE(2), then the MASTER drives DS1* to low first. If the first accessed byte location is BYTE(1) or BYTE(3), then it drives DS0* to low first. A01 is valid only on the first data transfer (i.e. until the SLAVE drives DTACK* or

Tableau 2-21 (suite)

alors il commande au début DS0* au niveau bas. A01 est valide seulement pendant le premier transfert de données (c'est-à-dire jusqu'à ce que l'ESCLAVE commande DTACK* ou BERR* au niveau bas pour la première fois) et peut être après au niveau bas ou haut dépendant de l'octet avec lequel a commencé le transfert de blocs par octets. Si l'emplacement du premier octet est l'OCTET(0) ou l'OCTET(1), alors le MAITRE commande A01 au niveau bas. Si l'emplacement du premier octet est l'OCTET(2) ou l'OCTET(3), alors le MAITRE commande A01 au niveau haut.

Un exemple d'utilisation de DS0*, DS1*, A01 et LWORD* pendant le cycle de transfert de bloc par octet commençant avec l'OCTET(2) est donné ci-dessous:

	DS1*	DS0*	A01	LWORD*
Premier transfert de données OCTET(2)	DLBM	DHBM	DHBM	dhbm?
OCTET(3)	DHBM	DLBM	dxbm?	dxbm?
OCTET(0)	DLBM	DHBM	dxbm?	dxbm?
OCTET(1)	DHBM	DLBM	dxbm?	dxbm?
Dernier transfert de données OCTET(2)	DLBM	DHBM	dxbm?	dxbm?

- 2.- Pendant un transfert de blocs par double octet, A01 est valide seulement pendant le premier transfert de données (c'est-à-dire jusqu'à ce que l'ESCLAVE commande DTACK* ou BERR* au niveau bas pour la première fois), ensuite A01 est commandé au niveau haut ou bas selon le groupe de deux octets avec lequel commence le transfert de deux octets par bloc. Si le premier groupe de deux octets est l'OCTET(0-1), alors le MAITRE commande A01 au niveau bas. Si le premier groupe de deux octets est l'OCTET(2-3), alors le MAITRE commande A01 au niveau haut.

Tableau 2-22

Utilisation des lignes de données du bus
pour transférer des données

Mnémonique	Type de cycles	D24-D31	D16-D23	D08-D15	D00-D07
ADO	UNIQUEMENT D'ADRESSAGE	dxbm?	dxbm?	dxbm?	dxbm?
D08(E0)	Transferts d'octet unique pair				
	LECTURE OCTET(0)	dxbs?	dxbs?	DVBS	dxbs?
	LECTURE OCTET(2)	dxbs?	dxbs?	DVBS	dxbs?
	ECRITURE OCTET(0)	dxbm?	dxbm?	DVBM	dxbm?
	ECRITURE OCTET(2)	dxbm?	dxbm?	DVBM	dxbm?
D08(E0) ou D08(0)	Transferts d'octet unique impair				
	LECTURE OCTET(1)	dxbs?	dxbs?	dxbs?	DVBS
	LECTURE OCTET(3)	dxbs?	dxbs?	dxbs?	DVBS
	ECRITURE OCTET(1)	dxbm?	dxbm?	dxbm?	DVBM
	ECRITURE OCTET(3)	dxbm?	dxbm?	dxbm?	DVBM
D16	Transferts double octet				
	LECTURE OCTET(0-1)	dxbs?	dxbs?	DVBS	DVBS
	LECTURE OCTET(2-3)	dxbs?	dxbs?	DVBS	DVBS
	ECRITURE OCTET(0-1)	dxbm?	dxbm?	DVBM	DVBM
	ECRITURE OCTET(2-3)	dxbm?	dxbm?	DVBM	DVBM
D32	Transferts quadruple octet				
	LECTURE OCTET(0-3)	DVBS	DVBS	DVBS	DVBS
	ECRITURE OCTET(0-3)	DVBM	DVBM	DVBM	DVBM
D08(E0):BLT	Transferts de blocs par octet				
	LECTURE DE BLOCS PAR OCTET	dxbs?	dxbs?	Note	
	ECRITURE DE BLOCS PAR OCTET	dxbm?	dxbm?	Note	

Table 2-21 (continued)

BERR* low the first time) and might be either high or low depending upon which byte the single byte block transfer begins with. If the first byte location is BYTE(0) or BYTE(1), then the MASTER drives A01 to low. If the first byte location is BYTE(2) or BYTE(3), then the MASTER drives A01 to high.

An example of the use of DS0*, DS1*, A01, and LWORD* during a single byte block transfer cycle which starts with BYTE(2) is given below:

	DS1*	DS0*	A01	LWORD*
First data transfer	DLBM	DHBM	DHBM	dhbm?
BYTE(2)	DHBM	DLBM	dxbm?	dxbm?
BYTE(3)	DLBM	DHBM	dxbm?	dxbm?
BYTE(0)	DHBM	DLBM	dxbm?	dxbm?
BYTE(1)	DLBM	DHBM	dxbm?	dxbm?
Last data transfer	DLBM	DHBM	dxbm?	dxbm?
BYTE(2)				

- 2.- During a double byte block transfer, A01 is valid only on the first data transfer (i.e. until the SLAVE drives DTACK* or BERR* low the first time) and is driven either high or low depending upon what double byte group the double byte block transfer begins with. If the first double byte group is BYTE(0-1), then the MASTER drives A01 to low. If the first double byte group is BYTE(2-3), then the MASTER drives A01 to high.

Table 2-22

Use of the data bus lines
to transfer data

Mnemonic	Type of cycles	D24-D31	D16-D23	D08-D15	D00-D07
AD0	ADDRESS ONLY	dxbm?	dxbm?	dxbm?	dxbm?
D08(E0)	Single even byte transfers BYTE(0) READ BYTE(2) READ BYTE(0) WRITE BYTE(2) WRITE	dxbs? dxbs? dxbm? dxbm?	dxbs? dxbs? dxbm? dxbm?	DVBS DVBS DVBM DVBM	dxbs? dxbs? dxbm? dxbm?
D08(E0) or D08(0)	Single odd byte transfers BYTE(1) READ BYTE(3) READ BYTE(1) WRITE BYTE(3) WRITE	dxbs? dxbs? dxbm? dxbm?	dxbs? dxbs? dxbm? dxbm?	dxbs? dxbs? dxbm? dxbm?	DVBS DVBS DVBM DVBM
D16	Double byte transfers BYTE(0-1) READ BYTE(2-3) READ BYTE(0-1) WRITE BYTE(2-3) WRITE	dxbs? dxbs? dxbm? dxbm?	dxbs? dxbs? dxbm? dxbm?	DVBS DVBS DVBM DVBM	DVBS DVBS DVBM DVBM
D32	Quad byte transfers BYTE(0-3) READ BYTE(0-3) WRITE	DVBS DVBM	DVBS DVBM	DVBS DVBM	DVBS DVBM
D08(E0):BLT	Single byte block transfers SINGLE BYTE BLOCK READ SINGLE BYTE BLOCK WRITE	dxbs? dxbm?	dxbs? dxbm?	Note Note	

Tableau 2-22 (suite)

Mnémonique	Type de cycles	D24-D31	D16-D23	D08-D15	D00-D07
D16:BLT	Transferts de blocs par double octet LECTURE DE BLOCS PAR DOUBLE OCTET ECRITURE DE BLOCS PAR DOUBLE OCTET	dxbs? dxbm?	dxbs? dxbm?	DVBS DVBM	DVBS DVBM
D32:BLT	Transferts de blocs par quadruple octet LECTURE DE BLOCS PAR QUADRUPLE OCTET ECRITURE DE BLOCS PAR QUADRUPLE OCTET	DVBS DVBM	DVBS DVBM	DVBS DVBM	DVBS DVBM
D08(E0):RMW	Transferts RMW pour octet unique LECTURE-MODIFICATION-ECRITURE OCTET(0) LECTURE-MODIFICATION-ECRITURE OCTET(1) LECTURE-MODIFICATION-ECRITURE OCTET(2) LECTURE-MODIFICATION-ECRITURE OCTET(3)	dxbb? dxbb? dxbb? dxbb?	dxbb? dxbb? dxbb? dxbb?	DVBB dxbb DVBB dxbb	dxbb? DVBB dxbb? DVBB
D16:RMW	Transferts RMW pour double octet LECTURE-MODIFICATION-ECRITURE OCTET (0-1) LECTURE-MODIFICATION-ECRITURE OCTET (2-3)	dxbb? dxbb?	dxbb? dxbb?	DVBB DVBB	DVBB DVBB
D32:RMW	Transferts RMW pour quadruple octet LECTURE-MODIFICATION-ECRITURE OCTET (0-3)	DVBB	DVBB	DVBB	DVBB
D32:UAT	Transferts non alignés LECTURE OCTET(0-2) LECTURE OCTET(1-3) LECTURE OCTET(1-2) ECRITURE OCTET(0-2) ECRITURE OCTET(1-3) ECRITURE OCTET(1-2)	DVBS dxbs? dxbs? DVBM dxbm? dxbm?	DVBS DVBS DVBS DVBM DVBM DVBM	DVBS DVBS DVBS DVBM DVBM DVBM	dxbs? DVBS dxbs? dxbm? DVBM dxbm?

Note:

Pendant les transferts de blocs par octet, les données sont transférées 8 bits à la fois sur les lignes D00-D07 ou D08-D15. Un exemple de lecture de blocs par octet est donné ci-dessous.

	D08-D15	D00-D07
Premier transfert de données	DVBS dxbs? DVBS dxbs? DVBS dxbs? DVBS	dxbs? DVBS dxbs? DVBS dxbs? DVBS dxbs?
Dernier transfert de données	DVBS	dxbs?

Table 2-22 (continued)

Mnemonic	Type of cycles	D24-D31	D16-D23	D08-D15	D00-D07
D16:BLT	Double byte block transfers DOUBLE BYTE BLOCK READ DOUBLE BYTE BLOCK WRITE	dxbs? dxbm?	dxbs? dxbm?	DVBS DVBM	DVBS DVBM
D32:BLT	Quad byte block transfers QUAD BYTE BLOCK READ QUAD BYTE BLOCK WRITE	DVBS DVBM	DVBS DVBM	DVBS DVBM	DVBS DVBM
D08(E0):RMW	Single byte RMW transfers BYTE(0) READ-MODIFY-WRITE BYTE(1) READ-MODIFY-WRITE BYTE(2) READ-MODIFY-WRITE BYTE(3) READ-MODIFY-WRITE	dxbb? dxbb? dxbb? dxbb?	dxbb? dxbb? dxbb? dxbb?	DVBB dxbb DVBB dxbb	dxbb? DVBB dxbb? DVBB
D16:RMW	Double byte RMW transfers BYTE(0-1) READ-MODIFY-WRITE BYTE(2-3) READ-MODIFY-WRITE	dxbb? dxbb?	dxbb? dxbb?	DVBB DVBB	DVBB DVBB
D32:RMW	Quad byte RMW transfers BYTE(0-3) READ-MODIFY-WRITE	DVBB	DVBB	DVBB	DVBB
D32:UAT	Unaligned transfers BYTE(0-2) READ BYTE(1-3) READ BYTE(1-2) READ BYTE(0-2) WRITE BYTE(1-3) WRITE BYTE(1-2) WRITE	DVBS dxbs? dxbs? DVBM dxbm? dxbm?	DVBS DVBS DVBS DVBM DVBM DVBM	DVBS DVBS DVBS DVBM DVBM DVBM	dxbs? DVBS dxbs? dxbm? DVBM dxbm?

Note:

During single byte block transfers, data is transferred 8 bits at a time over D00-D07 or D08-D15. A single byte block read example is given below.

	D08-D15	D00-D07
First data transfer	DVBS dxbs? DVBS dxbs? DVBS dxbs?	dxbs? DVBS dxbs? DVBS dxbs? DVBS
Last data transfer	DVBS	dxbs?

Tableau 2-23

*Paramètres de temps des chronogrammes du MAITRE,
de l'ESCLAVE et du DETECTEUR D'ACCES*

NUMERO DU PARAMETRE	MAITRE Voir aussi tableau 2-25		ESCLAVE Voir aussi tableau 2-26		DETECTEUR D'ACCES Voir aussi tableau 2-27	
	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.
1	0	-	-	-	-	-
2	0	-	-	-	-	-
3	60	-	-	-	-	-
4	35	-	10	-	10	-
5	40	-	30	-	30	-
6	0	-	0	-	-	-
7	0	-	0	-	-	-
8	35	-	10	-	-	-
9	0	-	0	-	-	-
10	0	-	-10	-	-10	-
11	40	-	30	-	30	-
12	35	-	10	-	10	-
13	-	10	-	20	-	20
14	0	-	0	-	-	-
15	0	-	0	-	-	-
16	0	-	0	-	-	-
17	40	-	30	-	30	-
18	0	-	0	-	-	-
19	40	-	30	-	30	-
20	0	-	0	-	-	-
21	0	-	0	-	-	-
22	0	-	0	-	-	-
23	10	-	0	-	0	-
24A	0	-	-	-	-	-
24B	0	-	-	-	-	-
25	-	25	-	-	-	-
26	0	-	0	-	-	-
27	-25	-	0	-	-	-
28	30	2T	30	-	-	-
29	0	-	0	-	-	-
30	0	-	0	-	-	-
31	0	-	0	-	-	-
32	-	-	10	-	10	-
33	-	-	30	-	30	-

Notes:

1.- Tous les temps sont en nanosecondes.

2.- T = valeur du temps maximal d'occupation du bus, en microsecondes.

Tableau 2-24

*Paramètres de temps du chronogramme du LIMITEUR DE TEMPS
D'OCCUPATION DU BUS*

(voir aussi tableau 2-28)

NUMERO DU PARAMETRE	MAITRE	
	MIN.	MAX.
28	T	2T
30	0	-

Note:

T = valeur du temps maximal d'occupation du bus, en microsecondes.

Table 2-23
MASTER, SLAVE, and
LOCATION MONITOR *timing parameters*

PARAMETER NUMBER	MASTER See also Table 2-25		SLAVE See also Table 2-26		LOCATION MONITOR See also Table 2-27	
	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.
1	0	-	-	-	-	-
2	0	-	-	-	-	-
3	60	-	-	-	-	-
4	35	-	10	-	10	-
5	40	-	30	-	30	-
6	0	-	0	-	-	-
7	0	-	0	-	-	-
8	35	-	10	-	-	-
9	0	-	0	-	-	-
10	0	-	-10	-	-10	-
11	40	-	30	-	30	-
12	35	-	10	-	10	-
13	-	10	-	20	-	20
14	0	-	0	-	-	-
15	0	-	0	-	-	-
16	0	-	0	-	-	-
17	40	-	30	-	30	-
18	0	-	0	-	-	-
19	40	-	30	-	30	-
20	0	-	0	-	-	-
21	0	-	0	-	-	-
22	0	-	0	-	-	-
23	10	-	0	-	0	-
24A	0	-	-	-	-	-
24B	0	-	-	-	-	-
25	-	25	-	-	-	-
26	0	-	0	-	-	-
27	-25	-	0	-	-	-
28	30	21	30	-	-	-
29	0	-	0	-	-	-
30	0	-	0	-	-	-
31	0	-	0	-	-	-
32	-	-	10	-	10	-
33	-	-	30	-	30	-

Notes:

1.- All times are in nanoseconds.

2.- T = time-out value, in microseconds.

Table 2-24
BUS TIMER *timing parameters*

(see also Table 2-28)

PARAMETER NUMBER	MASTER	
	MIN.	MAX.
28	T	2T
30	0	-

Note:

T = time-out value, in microseconds.

Tableau 2-25

MAITRE
REGLES et OBSERVATIONS sur la chronologie

Note: La numérotation correspond aux paramètres de temps spécifiés dans le tableau 2-23.

1. REGLE 2.27:

En prenant le contrôle du BUS CEI 821, le MAITRE NE DOIT commander aucun des signaux IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0*, DS1* ou AS*, avant ce délai après que le MAITRE précédent eut permis à AS* de passer au-dessus du niveau bas.

OBSERVATION 2.35:

Le chapitre 3 décrit comment un DEMANDEUR du MAITRE reçoit le contrôle du BUS CEI 821.

2. REGLE 2.28:

En prenant le contrôle du BUS CEI 821, le MAITRE NE DOIT commander aucun des signaux IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0*, DS1* ou AS* avant ce délai après que son DEMANDEUR eut reçu le contrôle du bus.

OBSERVATION 2.36:

Le chapitre 3 décrit comment un DEMANDEUR du MAITRE reçoit le contrôle du BUS CEI 821.

3. REGLE 2.29:

En prenant le contrôle du BUS CEI 821, le MAITRE NE DOIT PAS commander AS* au niveau bas avant ce délai après que le MAITRE précédent eut permis à AS* de passer au-dessus du niveau bas.

OBSERVATION 2.37:

La REGLE 2.29 assure que le paramètre de temps 5 est garanti pour les ESCLAVES quand il y a un changement du MAITRE du DTB.

4. REGLE 2.30:

Le MAITRE NE DOIT PAS commander AS* au niveau bas avant que IACK* ait été au niveau haut et que les lignes requises parmi A01-A31, AM0-AM5 et LWORD* aient été valides au minimum pendant ce temps.

OBSERVATION 2.38:

Les tableaux 2-20 et 2-21 spécifient les lignes parmi A01-A31, AM0-AM5 et LWORD* qu'un MAITRE doit commander.

5. REGLE 2.31:

Lorsqu'on utilise le DTB pour deux cycles consécutifs, le MAITRE NE DOIT PAS commander AS* au niveau bas si ce signal n'est pas resté au niveau haut au minimum pendant ce temps.

6. REGLE 2.32:

Après un cycle de lecture, le MAITRE NE DOIT commander aucun des signaux D00-D31 avant que les deux signaux DTACK* et BERR* soient au niveau haut.

Table 2-25

MASTER
Timing RULES and OBSERVATIONS

Note: The numbers correspond to the timing parameters specified in Table 2-23.

1. RULE 2.27:

When taking control of the IEC 821 BUS, the MASTER MUST NOT drive any of IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0*, DS1*, or AS* until after the previous MASTER allows AS* to rise above the low level.

OBSERVATION 2.35:

Chapter 3 describes how a MASTER'S REQUESTER is granted use of the IEC 821 BUS.

2. RULE 2.28:

When taking control of the IEC 821 BUS, the MASTER MUST NOT drive any of IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0*, DS1*, or AS* until after its REQUESTER is granted the bus.

OBSERVATION 2.36:

Chapter 3 describes how a MASTER'S REQUESTER is granted use of the IEC 821 BUS.

3. RULE 2.29:

When taking control of the IEC 821 BUS, the MASTER MUST NOT drive AS* low until this time after the previous MASTER allows AS* to rise above the low level.

OBSERVATION 2.37:

RULE 2.29 ensures that timing parameter 5 for SLAVES is guaranteed when there is an interchange of the DTB MASTER.

4. RULE 2.30:

The MASTER MUST NOT drive AS* low until IACK* has been high, and the required lines among A01-A31, AM0-AM5, and LWORD* have been valid for this minimum time.

OBSERVATION 2.38:

Tables 2-20 and 2-21 specify the lines among A01-A31, AM0-AM5, and LWORD* that a MASTER is required to drive.

5. RULE 2.31:

When using the DTB for two consecutive cycles, the MASTER MUST NOT drive AS* low until it has been high for this minimum time.

6. RULE 2.32:

After a read cycle, the MASTER MUST NOT drive any of D00-D31 until both DTACK* and BERR* are high.

Tableau 2-25 (suite)

7. REGLE 2.33:

Pendant les cycles de lecture, le MAITRE NE DOIT PAS commander DSA* au niveau bas avant d'avoir libéré toutes les lignes D00-D31.

8. REGLE 2.34:

Pendant les cycles d'écriture, le MAITRE NE DOIT PAS commander DSA* au niveau bas avant que les lignes requises parmi D00-D31 aient été valides au minimum pendant ce temps.

OBSERVATION 2.39:

Le tableau 2-22 spécifie les lignes parmi D00-D31 qu'un MAITRE doit commander.

9. REGLE 2.35:

Le MAITRE NE DOIT PAS commander DSA* au niveau bas avant que les deux signaux DTACK* et BERR* soient au niveau haut.

10. REGLE 2.36:

Le MAITRE NE DOIT PAS commander DSA* au niveau bas avant d'avoir commandé AS* au niveau bas.

11. REGLE 2.37:

Le MAITRE NE DOIT PAS commander DSA* au niveau bas avant que DS0* et DS1* aient été simultanément au niveau haut au minimum pendant ce temps.

12. REGLE 2.38:

Le MAITRE NE DOIT PAS commander DSA* au niveau bas avant que WRITE* soit valide au minimum pendant ce temps.

13. REGLE 2.39:

Dans les cycles où le MAITRE commande les deux signaux de validation de donnée au niveau bas, il DOIT commander DSB* au niveau bas dans ce délai maximal après avoir commandé DSA* au niveau bas.

OBSERVATION 2.40:

Le paramètre de temps 13 n'est pas applicable aux transferts par octet où un seul signal de validation de donnée est commandé au niveau bas.

14. REGLE 2.40:

Pendant tous les cycles de transfert de données, à l'exception des cycles de lecture-modification-écriture, le MAITRE DOIT maintenir l'adresse valide et maintenir le niveau approprié sur LWORD* jusqu'à ce qu'il détecte le premier front descendant sur DTACK* ou BERR*.

OBSERVATION 2.41:

Pendant tous les cycles de transfert de données, à l'exception des cycles de transfert par bloc et de lecture-modification-écriture, il y a un seul front descendant de DTACK* ou BERR*.

15. REGLE 2.41:

Pendant les cycles lecture-modification-écriture, le MAITRE DOIT maintenir l'adresse valide et maintenir le niveau approprié sur LWORD* jusqu'à ce qu'il détecte le deuxième front descendant de DTACK* ou BERR*.

Table 2-25 (continued)

7. RULE 2.33:
During read cycles, the MASTER MUST NOT drive DSA* low until it has released all of D00-D31.

8. RULE 2.34:
During write cycles, the MASTER MUST NOT drive DSA* low until the required lines among D00-D31 have been valid for this minimum time.

OBSERVATION 2.39:

Table 2-22 specifies the lines among D00-D31 that a MASTER is required to drive.

9. RULE 2.35:
The MASTER MUST NOT drive DSA* low until both DTACK* and BERR* are high.
10. RULE 2.36:
The MASTER MUST NOT drive DSA* low until it has driven AS* low.
11. RULE 2.37:
The MASTER MUST NOT drive DSA* low until DS0* and DS1* have both been simultaneously high for this minimum time.
12. RULE 2.38:
The MASTER MUST NOT drive DSA* low until WRITE has been valid for this minimum time.
13. RULE 2.39:
During cycles where the MASTER drives both data strobes low, it MUST drive DSB* low within this maximum time after it drives DSA* low.

OBSERVATION 2.40:

Timing parameter 13 does not apply to transfers where only one data strobe is driven low.

14. RULE 2.40:
During all data transfer cycles, except read-modify-write cycles, the MASTER MUST hold the address valid and maintain the appropriate level on LWORD* until it detects the first falling edge on DTACK* or BERR*.

OBSERVATION 2.41:

During all data transfer cycles, except block transfer and read-modify-write cycles, there will be only one falling edge on DTACK* or BERR*.

15. RULE 2.41:
During read-modify-write cycles, the MASTER MUST hold the address valid and maintain the appropriate level on LWORD* until it detects the second falling edge on DTACK* or BERR*.

Tableau 2-25 (suite)

16. REGLE 2.42:

Pendant tous les cycles de transfert de données, le MAITRE DOIT maintenir un code modificateur d'adresse valide et assurer que IACK* soit maintenu au niveau haut jusqu'à ce qu'il détecte le dernier front descendant de DTACK* ou BERR*.

OBSERVATION 2.42:

Pendant tous les cycles de transfert de données, excepté les transferts par bloc et les cycles lecture-modification-écriture, il y aura un seul front descendant de DTACK* ou BERR*.

17. REGLE 2.43:

Pendant les cycles UNIQUEMENT D'ADRESSAGE, le MAITRE NE DOIT PAS changer les niveaux sur IACK*, A01-A31, AM0-AM5 et LWORD* au minimum pendant ce temps après avoir commandé AS* au niveau bas.

18. REGLE 2.44:

Pendant tous les cycles de transfert de données, le MAITRE DOIT maintenir AS* au niveau bas jusqu'à ce qu'il détecte le dernier front descendant sur DTACK* ou BERR*.

19. REGLE 2.45:

Le MAITRE DOIT maintenir AS* au niveau bas au minimum pendant ce temps.

20. REGLE 2.46:

Lorsqu'un MAITRE a commandé DSA* au niveau bas, il DOIT maintenir cette ligne au niveau bas jusqu'à ce qu'il détecte DTACK* ou BERR* au niveau bas.

21. REGLE 2.47:

Lorsqu'un MAITRE a commandé DSB* au niveau bas, il DOIT maintenir cette ligne au niveau bas jusqu'à ce qu'il détecte DTACK* ou BERR* au niveau bas.

22. REGLE 2.48:

Pendant les cycles d'écriture, lorsqu'un MAITRE a commandé DSA* au niveau bas, il NE DOIT changer aucun des signaux sur les lignes D00-D31 avant d'avoir détecté un niveau bas sur DTACK* ou BERR*.

23. REGLE 2.49:

Lorsqu'un MAITRE a commandé DSA* au niveau bas, il NE DOIT PAS changer le niveau sur la ligne WRITE*, avant ce délai minimal après avoir commandé les deux signaux de validation de donnée au niveau haut.

24A. REGLE 2.50:

SI le MAITRE commande ou libère AS* au niveau haut après la libération de BBSY* par son DEMANDEUR,
ALORS il DOIT libérer IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0* et DS1* avant de permettre à AS* de passer au-dessus du niveau bas.

OBSERVATION 2.43:

Le chapitre 3 décrit comment un DEMANDEUR d'un MAITRE libère la ligne BBSY*.

Table 2-25 (continued)

16. RULE 2.42:

During all data transfer cycles, the MASTER MUST maintain a valid address modifier code, and ensure that IACK* stays high until it detects the last falling edge on DTACK* or BERR*.

OBSERVATION 2.42:

During all data transfer cycles, except block transfer and read-modify-write cycles, there will be only one falling edge on DTACK* or BERR*.

17. RULE 2.43:

During ADDRESS-ONLY cycles, the MASTER MUST NOT change the levels on IACK*, A01-A31, AM0-AM5, and LWORD* for this minimum time after it drives AS* low.

18. RULE 2.44:

During all data transfer cycles, the MASTER MUST hold AS* low until it detects the last falling edge of DTACK* or BERR*.

19. RULE 2.45:

The MASTER MUST hold AS* low for this minimum time.

20. RULE 2.46:

Once a MASTER has driven DSA* low, it MUST maintain that line low until it detects DTACK* or BERR* low.

21. RULE 2.47:

Once a MASTER has driven DSB* low, it MUST maintain that line low until it detects DTACK* or BERR* low.

22. RULE 2.48:

During write cycles, once the MASTER has driven DSA* low, it MUST not change any of D00-D31 until it detects DTACK* or BERR* low.

23. RULE 2.49:

Once a MASTER has driven DSA* low, it MUST NOT change the level on the WRITE* line, until this minimum time after both data strobes are high.

24A. RULE 2.50:

IF the MASTER drives or releases AS* to high after its REQUESTER releases BBSY*,
THEN it MUST release IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0*, and DS1* before allowing AS* to rise above the low level.

OBSERVATION 2.43:

Chapter 3 describes how a MASTER'S REQUESTER releases the BBSY* line.

Tableau 2-25 (suite)

24B. REGLE 2.51:

SI le MAITRE commande ou libère AS* au niveau haut avant la libération de BBSY* par son DEMANDEUR,
ALORS il DOIT libérer AS*, IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0* et DS1* avant de permettre à son DEMANDEUR de libérer BBSY*.

OBSERVATION 2.44:

Le chapitre 3 décrit comment un DEMANDEUR d'un MAITRE libère la ligne BBSY*.

25. REGLE 2.52:

SI le MAITRE commande ou libère AS* au niveau haut après la libération de BBSY* par son DEMANDEUR,
ALORS il DOIT libérer AS* dans ce délai après lui avoir permis de passer au-dessus du niveau bas.

OBSERVATION 2.45:

Le chapitre 3 décrit comment un DEMANDEUR d'un MAITRE libère la ligne BBSY*.

26. OBSERVATION 2.46:

Le paramètre de temps 26 garantit que, pendant les cycles de lecture, le bus de données ne sera pas commandé avant que le MAITRE ne commande DSA* au niveau bas.

27. OBSERVATION 2.47:

Pendant les cycles de lecture, le MAITRE est assuré que le bus de données sera valide dans ce délai après le passage du signal DTACK* au niveau bas. Ce temps ne s'applique pas aux cycles où l'ESCLAVE commande BERR* au niveau bas au lieu de commander DTACK*.

28. OBSERVATION 2.48:

Le MAITRE est assuré que ni le signal DTACK* ni BERR* ne passeront au niveau bas avant ce délai minimal après avoir commandé DSA* au niveau bas. Le LIMITEUR DE TEMPS D'OCCUPATION DU BUS garantit au MAITRE que, si DTACK* n'est pas passé au niveau bas dans un délai compris entre la limite de temps et le double de cette limite, alors le LIMITEUR DE TEMPS D'OCCUPATION DU BUS commande BERR* au niveau bas.

29. OBSERVATION 2.49:

Pendant les cycles de lecture, le MAITRE est assuré que le bus de données restera valide jusqu'à ce qu'il commande DSA* au niveau haut.

30. OBSERVATION 2.50:

Le paramètre de temps 30 garantit que ni DTACK* ni BERR* ne passeront au niveau haut avant que le MAITRE ait commandé DS0* et DS1* au niveau haut.

31. OBSERVATION 2.51

Pendant les cycles de lecture, le MAITRE est assuré que le bus de données a été libéré au moment où les signaux DTACK* et BERR* passent au niveau haut.

Table 2-25 (continued)

24B. RULE 2.51:

IF the MASTER drives or releases AS* to high before its REQUESTER releases BBSY*,
THEN it MUST release AS*, IACK*, AM0-AM5, A01-A31, LWORD*, D00-D31, WRITE*, DS0*, and DS1* before allowing its REQUESTER to release BBSY*.

OBSERVATION 2.44:

Chapter 3 describes how a MASTER'S REQUESTER releases the BBSY* line.

25. RULE 2.52:

IF the MASTER drives or releases AS* to high before its REQUESTER releases BBSY*,
THEN it MUST release AS* within this time after allowing it to rise above the low level.

OBSERVATION 2.45:

Chapter 3 describes how a MASTER'S REQUESTER releases the BBSY* line.

26. RULE 2.46:

Timing parameter 26 guarantees that during read cycles, the data bus will not be driven until the MASTER drives DSA* low.

27. OBSERVATION 2.47:

During read cycles, the MASTER is guaranteed that the data bus will be valid within this time after DTACK* goes low. This time does not apply to cycles where the SLAVE drives BERR* low instead of DTACK*.

28. OBSERVATION 2.48:

The MASTER is guaranteed that neither DTACK* nor BERR* will go low until this minimum time after it drives DSA* low. The BUS TIMER guarantees the MASTER that if DTACK* has not gone low after its time-out period has elapsed, and within twice its time-out period, then the BUS TIMER will drive BERR* low.

29. OBSERVATION 2.49:

During read cycles, the MASTER is guaranteed that the data bus will remain valid until it drives DSA* high.

30. OBSERVATION 2.50:

Timing parameter 30 guarantees that neither DTACK* nor BERR* will go high until the MASTER drives both DS0* and DS1* high.

31. OBSERVATION 2.51:

During read cycles, the MASTER is guaranteed that the data bus has been released by the time DTACK* and BERR* are high.

Tableau 2-26

ESCLAVE
REGLES et OBSERVATIONS sur la chronologie

Note: La numérotation correspond aux paramètres de temps spécifiés dans le tableau 2-23.

4. OBSERVATION 2.52:
Tous les ESCLAVES sont assurés que IACK*, A01-A31, AM0-AM5 et LWORD* ont été valides au minimum pendant ce temps lorsqu'ils détectent le front descendant sur AS*.
5. OBSERVATION 2.53:
Tous les ESCLAVES sont assurés que AS* est au niveau haut au minimum pendant ce temps entre les cycles DTB.
6. OBSERVATION 2.54:
Pendant les cycles de lecture, l'ESCLAVE qui répond est assuré qu'aucune ligne D00-D31 ne sera commandée par aucun autre module avant que l'ESCLAVE qui répond libère DTACK* et BERR* au niveau haut.
7. OBSERVATION 2.55:
Pendant les cycles de lecture, l'ESCLAVE qui répond est assuré que le bus de données sera libéré par tous les autres modules lorsque DSA* passera au niveau bas.
8. OBSERVATION 2.56:
Pendant les cycles d'écriture, l'ESCLAVE qui répond est assuré que le bus de données a été valide au minimum pendant ce temps lorsqu'il détecte un front descendant sur DSA*.
9. OBSERVATION 2.57:
L'ESCLAVE qui répond est assuré que ni DS0* ni DS1* ne passeront au niveau bas avant que DTACK* et BERR* du cycle précédent soient passés au niveau haut.
10. OBSERVATION 2.58:
Suite au déphasage du bus, les ESCLAVES sur le DTB peuvent détecter un front descendant sur DSA* avant le front descendant sur AS*. Malgré cela, les ESCLAVES sont assurés que le front descendant sur DSA* ne précédera par le front descendant sur AS* avec un délai plus grand que celui-ci.
11. OBSERVATION 2.59:
Les ESCLAVES sont assurés de ce délai minimal pendant lequel les deux signaux de validation de donnée sont simultanément au niveau haut entre deux transferts de données consécutifs.
12. OBSERVATION 2.60:
Les ESCLAVES sont assurés que WRITE* est valide au minimum pendant ce temps avant un front descendant sur DSA*.
13. OBSERVATION 2.61:
SI les deux signaux de validation de donnée sont commandés au niveau bas,
ALORS l'ESCLAVE qui répond est assuré que dans ce délai maximal, DSB* passera au niveau bas après que DSA* eut passé au niveau bas.

Table 2-26

SLAVE
Timing RULES and OBSERVATIONS

Note: The numbers correspond to the timing parameters specified in Table 2-23.

4. OBSERVATION 2.52:
All SLAVES are guaranteed that IACK*, A01-A31, AM0-AM5, and LWORD* have been valid for this minimum time when they detect a falling edge on AS*.
5. OBSERVATION 2.53:
All SLAVES are guaranteed this minimum high time on AS* between DTB cycles.
6. OBSERVATION 2.54:
During read cycles, the responding SLAVE is guaranteed that none of D00-D31 will be driven by any other module until the responding SLAVE releases DTACK* and BERR* to high.
7. OBSERVATION 2.55:
During read cycles, the responding SLAVE is guaranteed that the data bus will be released by all other modules by the time DSA* goes low.
8. OBSERVATION 2.56:
During write cycles, the responding SLAVE is guaranteed that the data bus has been valid for this minimum time when it detects a falling edge on DSA*.
9. OBSERVATION 2.57:
The responding SLAVE is guaranteed that neither DS0* nor DS1* will go low until DTACK* and BERR* from the previous cycle have gone high.
10. OBSERVATION 2.58:
Due to bus skew, SLAVES on the DTB might detect a falling edge on DSA* before detecting the falling edge on AS*. However, SLAVES are guaranteed that a falling edge on DSA* will not precede the falling edge on AS* by more than this time.
11. OBSERVATION 2.59:
SLAVES are guaranteed this minimum time during which both data strobes are simultaneously high between consecutive data transfers.
12. OBSERVATION 2.60:
SLAVES are guaranteed that WRITE* has been valid for this minimum time before a falling edge on DSA*.
13. OBSERVATION 2.61:
IF both data strobes are going to be driven low,
THEN the responding SLAVE is guaranteed that DSB* will go low within this maximum time after DSA* has gone low.

Tableau 2-26 (suite)

14. OBSERVATION 2.62:
Pendant tous les cycles de transfert de données, excepté les cycles lecture-modification-écriture, l'ESCLAVE qui répond est assuré que l'adresse et LWORD* restent valides jusqu'à ce qu'il commande DTACK* ou BERR* au niveau bas pour la première fois et à condition de le faire dans la limite de temps d'occupation du bus.
15. OBSERVATION 2.63:
Pendant tous les cycles lecture-modification-écriture, l'ESCLAVE qui répond est assuré que l'adresse et LWORD* restent valides jusqu'à ce qu'il commande DTACK* ou BERR* au niveau bas pour la deuxième fois et à condition de le faire dans la limite de temps d'occupation du bus.
16. OBSERVATION 2.64:
L'ESCLAVE qui répond est assuré que IACK* et AM0-AM5 restent valides jusqu'à ce qu'il commande DTACK* et BERR* au niveau bas pour la dernière fois et à condition de le faire dans la limite de temps d'occupation du bus.
17. OBSERVATION 2.65:
Les ESCLAVES sont assurés que IACK*, A01-A31, AM0-AM5 et LWORD* resteront valides au minimum pendant ce temps après le front descendant de AS*. Pendant les cycles UNIQUEMENT D'ADRESSAGE ce temps est garanti par le MAITRE. Pendant tous les autres types de cycles, ce temps se déduit à partir des paramètres de temps 10, 14, 16 et 28.
18. OBSERVATION 2.66:
L'ESCLAVE qui répond est assuré que AS* sera maintenu au niveau bas jusqu'à ce qu'il ait commandé DTACK* ou BERR* au niveau bas et à condition de le faire dans la limite de temps d'occupation du bus.
19. OBSERVATION 2.67:
Les ESCLAVES sont assurés que AS* sera maintenu au niveau bas au minimum pendant ce temps.
20. OBSERVATION 2.68:
L'ESCLAVE qui répond est assuré que, lorsque DSA* passera au niveau bas, il restera à ce niveau jusqu'à ce que DTACK* ou BERR* soit commandé au niveau bas et à condition que l'ESCLAVE le fasse dans la limite de temps d'occupation du bus.
21. OBSERVATION 2.69:
L'ESCLAVE qui répond est assuré que lorsque DSB* passera au niveau bas, il restera à ce niveau jusqu'à ce que DTACK* ou BERR* soit commandé au niveau bas et à condition que l'ESCLAVE le fasse dans la limite de temps d'occupation du bus.
22. OBSERVATION 2.70:
Pendant les cycles d'écriture, l'ESCLAVE qui répond est assuré que le bus de données restera valide au minimum pendant ce temps et jusqu'à ce qu'il commande DTACK* ou BERR* au niveau bas, à condition de le faire dans la limite de temps d'occupation du bus.
23. OBSERVATION 2.71:
L'ESCLAVE qui répond est assuré que la ligne WRITE* reste valide jusqu'à ce que les deux signaux de validation de données passent au niveau haut.

Table 2-26 (continued)

14. OBSERVATION 2.62:
During all data transfer cycles except read-modify-write cycles, the responding SLAVE is guaranteed that the address and LWORD* remain valid until it drives DTACK* or BERR* low for the first time, provided that it does so within the bus time-out period.
15. OBSERVATION 2.63:
During all read-modify-write cycles, the responding SLAVE is guaranteed that the address and LWORD* remain valid until it drives DTACK* or BERR* low for the second time, provided that it does so within the bus time-out period.
16. OBSERVATION 2.64:
The responding SLAVE is guaranteed that IACK* and AM0-AM5 remain valid until it drives DTACK* or BERR* low for the last time, provided that it does so within the bus time-out period.
17. OBSERVATION 2.65:
SLAVES are guaranteed that IACK*, A01-A31, AM0-AM5, and LWORD* will remain valid for this minimum time after the falling edge of AS*. During ADDRESS-ONLY cycles this time is guaranteed by the MASTER. During all other cycle types, this time is derived from timing parameters 10, 14, 16 and 28.
18. OBSERVATION 2.66:
The responding SLAVE is guaranteed that AS* will remain low until it drives DTACK* or BERR* low, provided that it does so within the bus time-out period.
19. OBSERVATION 2.67:
SLAVES are guaranteed that the AS* will remain low for this minimum time.
20. OBSERVATION 2.68:
The responding SLAVE is guaranteed that once DSA* goes low, it will remain low until it drives DTACK* or BERR* low, provided that the SLAVE does so within the bus time-out period.
21. OBSERVATION 2.69:
The responding SLAVE is guaranteed that once DSB* goes low, it will remain low until it drives DTACK* or BERR* low, provided that the SLAVE does so within the bus time-out period.
22. OBSERVATION 2.70:
During write cycles, the responding SLAVE is guaranteed that the data bus will remain valid until it drives DTACK* or BERR* low, provided that it does so within the bus time-out period.
23. OBSERVATION 2.71:
The responding SLAVE is guaranteed that the WRITE* line remains valid until both data strobes are high.

Tableau 2-26 (suite)

26. REGLE 2.53:
Pendant les cycles de lecture, l'ESCLAVE qui répond NE DOIT PAS commander le bus de données avant que DSA* soit passé au niveau bas.
27. REGLE 2.54:
Pendant les cycles de lecture, l'ESCLAVE qui répond NE DOIT PAS commander DTACK* avant d'avoir commandé les lignes de données avec des données valides.
- OBSERVATION 2.72:
La REGLE 2.55 ne s'applique pas aux cycles où l'ESCLAVE qui répond commande BERR* au niveau bas au lieu de DTACK*.
28. REGLE 2.55:
L'ESCLAVE qui répond DOIT attendre ce délai au minimum après le passage de DSA* au niveau bas, avant de commander DTACK* ou BERR* au niveau bas.
29. REGLE 2.56:
Pendant les cycles de lecture, lorsque l'ESCLAVE qui répond a commandé DTACK* au niveau bas, il NE DOIT PAS changer D00-D31 jusqu'à ce que DSA* soit passé au niveau haut.
30. REGLE 2.57:
Lorsque l'ESCLAVE qui répond a commandé DTACK* ou BERR* au niveau bas, il NE DOIT PAS libérer celui qu'il a commandé avant de détecter que DS0* et DS1* passent au niveau haut.
31. REGLE 2.58:
Pendant les cycles de lecture, l'ESCLAVE qui répond DOIT libérer toutes les lignes D00-D31 avant de libérer DTACK* ou BERR* au niveau haut.
32. OBSERVATION 2.73:
Les ESCLAVES sont assurés que IACK*, LWORD*, A00-A31 et AM0-AM5 sont valides au minimum pendant ce temps lorsqu'ils détectent un front descendant sur DSA*. Ce temps se déduit à partir des paramètres de temps 4 et 10.
33. OBSERVATION 2.74:
Pendant les cycles de transfert de données, les ESCLAVES sont assurés que DS0* et/ou DS1* resteront au niveau bas au minimum pendant ce temps. Ce temps se déduit du paramètre de temps 28 qui demande à l'ESCLAVE qui répond d'attendre un temps minimal avant de commander BERR* ou DTACK* au niveau bas.

Table 2-26 (continued)

26. RULE 2.53:
During read cycles, the responding SLAVE MUST NOT drive the data bus until DSA* goes low.
27. RULE 2.54:
During read cycles, the responding SLAVE MUST NOT drive DTACK* before it drives the data lines with valid data.
- OBSERVATION 2.72:
RULE 2-55 does not apply to cycles where the responding SLAVE drives BERR* low instead of DTACK*.
28. RULE 2.55:
The responding SLAVE MUST wait this minimum time after DSA* goes low before driving DTACK* or BERR* low.
29. RULE 2.56:
During read cycles, once the responding SLAVE has driven DTACK* low, it MUST NOT change D00-D31 until DSA* goes high.
30. RULE 2.57:
Once the responding SLAVE has driven DTACK* or BERR* low, it MUST NOT release it until it detects both DS0* and DS1* high.
31. RULE 2.58:
During read cycles, the responding SLAVE MUST release all of D00-D31 before releasing DTACK* or BERR* to high.
32. OBSERVATION 2.73:
SLAVES are guaranteed that IACK*, LWORD*, A00-A31, and AM0-AM5 have been valid for this minimum time when they detect a falling edge on DSA*. This time is derived from timing parameters 4 and 10.
33. OBSERVATION 2.74:
During data transfer cycles, SLAVES are guaranteed that DS0* and/or DS1* will remain low for at least this minimum time. This time is derived from timing parameter 28, where the responding SLAVE is required to wait a minimum time before driving BERR* or DTACK* to low.

Tableau 2-27

DETECTEUR D'ACCES
OBSERVATIONS sur la chronologie

Note: La numérotation correspond aux paramètres de temps spécifiés dans le tableau 2-23.

4. OBSERVATION 2.75:
Le DETECTEUR D'ACCES est assuré que IACK*, A01-A31, AM0-AM5 et LWORD* ont été valides au minimum pendant ce temps lorsqu'il détecte un front descendant sur AS*.
5. OBSERVATION 2.76:
Le DETECTEUR D'ACCES est assuré que AS* est au niveau haut, au minimum pendant ce temps entre les cycles du DTB.
10. OBSERVATION 2.77:
Suite au déphasage du bus, les DETECTEURS D'ACCES sur le DTB peuvent détecter des fronts descendants sur DSA* avant d'avoir détecté le front descendant sur AS*. Malgré cela, le DETECTEUR D'ACCES est assuré que le front descendant sur DSA* ne va pas précéder le front descendant sur AS* de plus que ce temps.
11. OBSERVATION 2.78:
Le DETECTEUR D'ACCES est assuré de ce temps minimal pendant lequel les deux signaux de validation de donnée du bus sont simultanément au niveau haut entre deux transferts de données consécutifs.
12. OBSERVATION 2.79:
Le DETECTEUR D'ACCES est assuré que WRITE* a été valide au minimum pendant ce temps lorsqu'il détecte un front descendant sur DSA*.
13. OBSERVATION 2.80:
Si les deux signaux de validation de donnée sont commandés au niveau bas,
ALORS le DETECTEUR D'ACCES est assuré que DSB* passera au niveau bas dans ce délai après le passage de DSA* au niveau bas.
17. OBSERVATION 2.81:
Le DETECTEUR D'ACCES est assuré que IACK*, A01-A31, AM0-AM5 et LWORD* resteront valides au minimum pendant ce temps après le front descendant de AS*. Pendant les cycles UNIQUEMENT D'ADRESSAGE, ce temps est garanti par le MAITRE. Pendant tous les autres types de cycles, ce temps se déduit à partir des paramètres de temps 10, 14, 16 et 28.
19. OBSERVATION 2.82:
Le DETECTEUR D'ACCES est assuré que AS* restera au niveau bas au minimum pendant ce temps.
23. OBSERVATION 2.83:
Le DETECTEUR D'ACCES est assuré que la ligne WRITE* reste valide jusqu'à ce que les deux signaux de validation de donnée passent au niveau haut.

Table 2-27

LOCATION MONITOR
Timing OBSERVATIONS

Note: The numbers correspond to the timing parameters specified in Table 2-23.

4. OBSERVATION 2.75:
The LOCATION MONITOR is guaranteed that IACK*, A01-A31, AM0-AM5, and LWORD* have been valid for this minimum time when it detects a falling edge on AS*.
5. OBSERVATION 2.76:
The LOCATION MONITOR is guaranteed this minimum high time on AS* between DTB cycles.
10. OBSERVATION 2.77:
Due to bus skew, LOCATION MONITORS on the DTB might detect a falling edge on DSA* before detecting the falling edge on AS*. However, the LOCATION MONITOR is guaranteed that the falling edge on DSA* will not precede the falling edge on AS* by more than this time.
11. OBSERVATION 2.78:
The LOCATION MONITOR is guaranteed this minimum time during which both data strobes are simultaneously high between consecutive data transfers.
12. OBSERVATION 2.79:
The LOCATION MONITOR is guaranteed that WRITE* has been valid for this minimum time when it detects a falling edge on DSA*.
13. OBSERVATION 2.80:
IF both data strobes are going to be driven low,
THEN the LOCATION MONITOR is guaranteed that DSB* will go low within this maximum time after DSA* has gone low.
17. OBSERVATION 2.81:
The LOCATION MONITOR is guaranteed that IACK*, A01-A31, AM0-AM5, and LWORD* will remain valid for this minimum time after the falling edge of AS*. During ADDRESS-ONLY cycles this time is guaranteed by the MASTER. During all other cycle types, this time is derived from timing parameters 10, 14, 16 and 28.
19. OBSERVATION 2.82:
The LOCATION MONITOR is guaranteed that the AS* will remain low for this minimum time.
23. OBSERVATION 2.83:
The LOCATION MONITOR is guaranteed that the WRITE* line remains valid until both data strobes are high.

Tableau 2-27 (suite)

32. OBSERVATION 2.84:
Le DETECTEUR D'ACCES est assuré que IACK*, LWORD*, A01-A31 et AM0-AM5 ont été valides au minimum pendant ce temps lorsqu'il détecte le front descendant de DSA*.
33. OBSERVATION 2.85:
Pendant les cycles de transfert de données, le DETECTEUR D'ACCES est assuré que DS0* et/ou DS1* resteront au niveau bas au minimum pendant ce temps. Ce temps se déduit du paramètre de temps 28, pendant lequel on demande à l'ESCLAVE qui répond d'attendre un temps minimal avant de commander BERR* ou DTACK* au niveau bas.

Tableau 2-28

LIMITEUR DE TEMPS D'OCCUPATION DU BUS
REGLES de la chronologie

Note: La numérotation correspond aux paramètres de temps spécifiés dans le tableau 2-24.

28. REGLE 2.59:
Le LIMITEUR DE TEMPS D'OCCUPATION DU BUS DOIT attendre pendant une durée au moins égale à la limite de temps d'occupation du bus mais NE DOIT PAS dépasser le double de ce temps, après le passage du premier signal de validation de donnée au niveau bas, avant de commander BERR* au niveau bas.
30. REGLE 2.60:
Le LIMITEUR DE TEMPS D'OCCUPATION DU BUS NE DOIT PAS libérer BERR* avant de détecter que DS0* et DS1* soient au niveau haut, après qu'il eut commandé BERR* au niveau bas.

Table 2-27 (continued)

32. OBSERVATION 2.84:
The LOCATION MONITOR is guaranteed that IACK*, LWORD*, A01-A31, and AM0-AM5 have been valid for this minimum time when it detects a falling edge on DSA*.
33. OBSERVATION 2.85:
During data transfer cycles, the LOCATION MONITOR is guaranteed that DS0* and/or DS1* will remain low for at least this minimum time. This time is derived from timing parameter 28, where the responding SLAVE is required to wait a minimum time before driving BERR* or DTACK* to low.

Table 2-28

BUS TIMER
Timing RULES

Note: The numbers correspond to the timing parameters specified in Table 2-24.

28. RULE 2.59:
The BUS TIMER MUST wait at least its time-out time, but no longer than twice its time-out time, after the first data strobe goes low before driving BERR* low.
30. RULE 2.60:
Once it has driven BERR* low, the BUS TIMER MUST NOT release BERR* until it detects both DS0* and DS1* high.

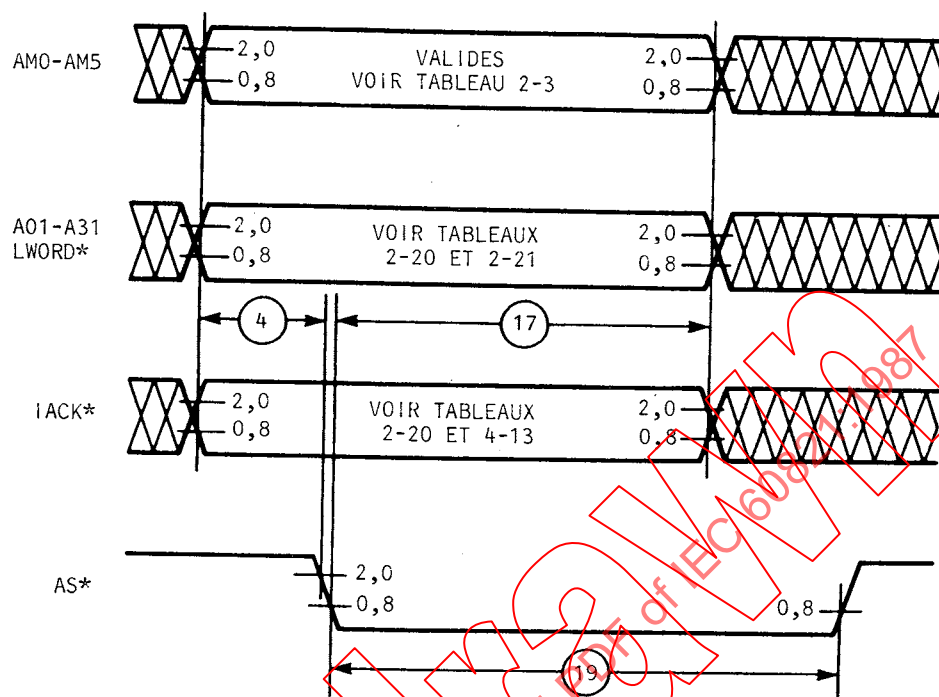


FIG. 2-12. - MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de diffusion d'adresse - TOUS LES CYCLES.

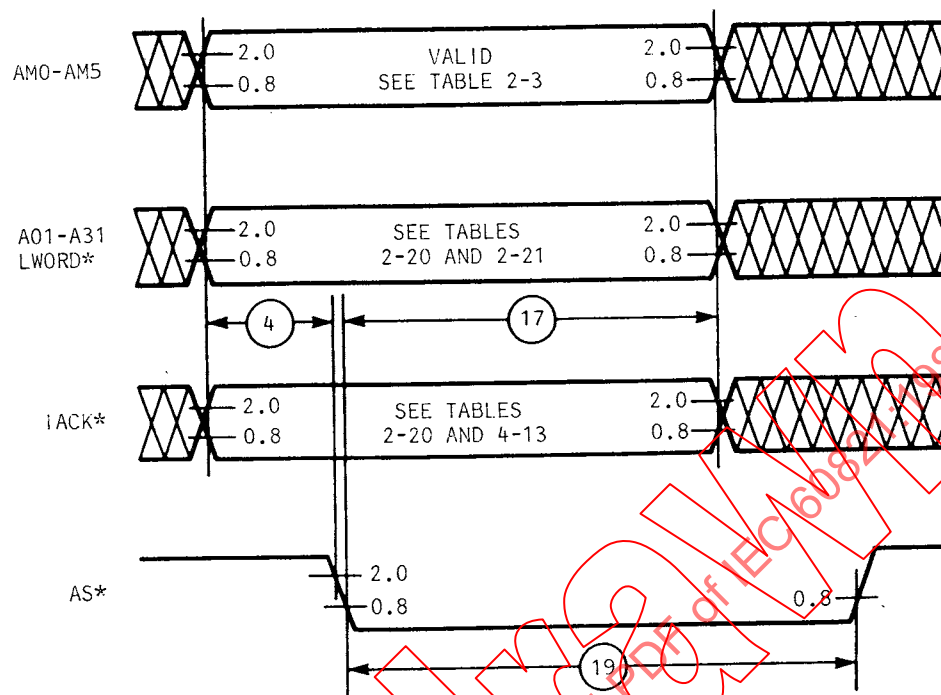


FIG. 2-12. - MASTER, SLAVE and LOCATION MONITOR -
Address broadcast timing - ALL CYCLES.

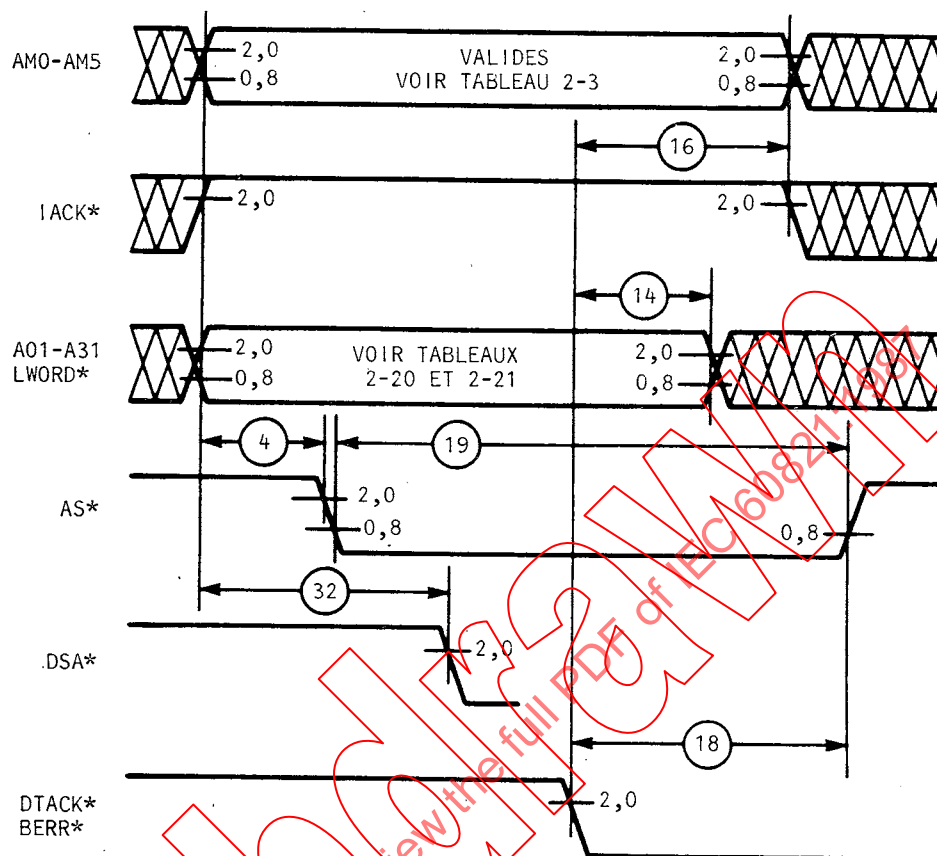


FIG. 2-13. - MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de diffusion d'adresse.

Transferts d'octet unique pair;
 transferts d'octet unique impair;
 transferts double octet;
 transferts quadruple octet;
 transferts non alignés.

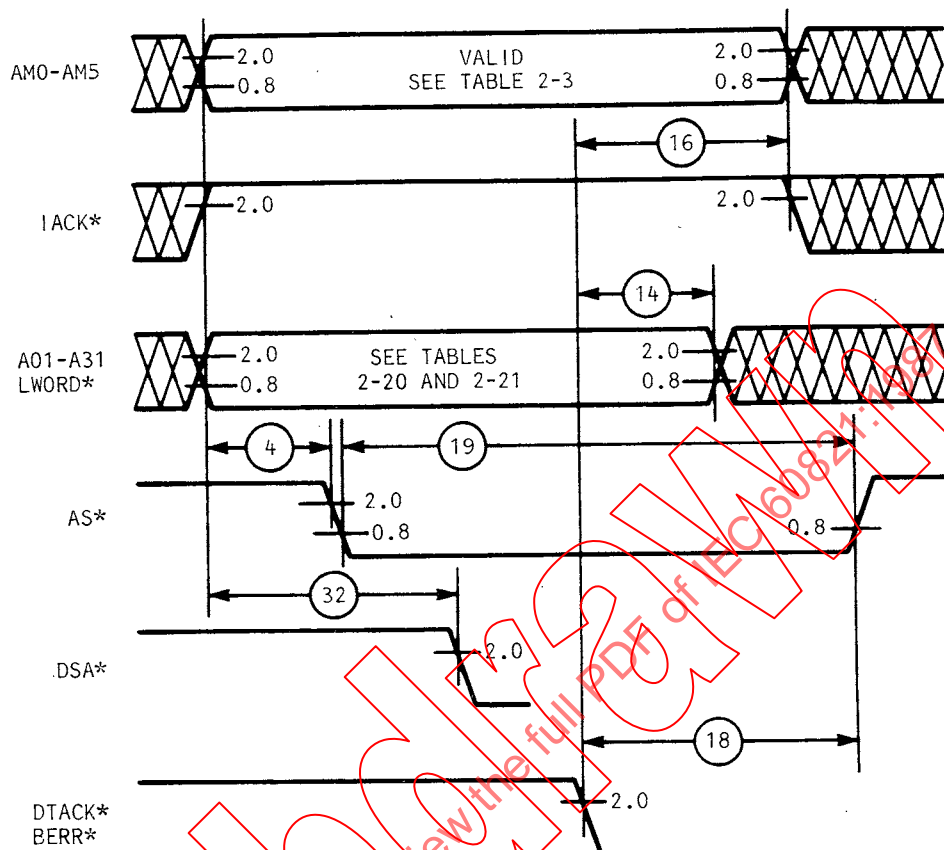


FIG. 2-13. - MASTER, SLAVE and LOCATION MONITOR - Address broadcast timing.

Single even byte transfers;
single odd byte transfers;
double byte transfers;
quad byte transfers;
unaligned transfers.

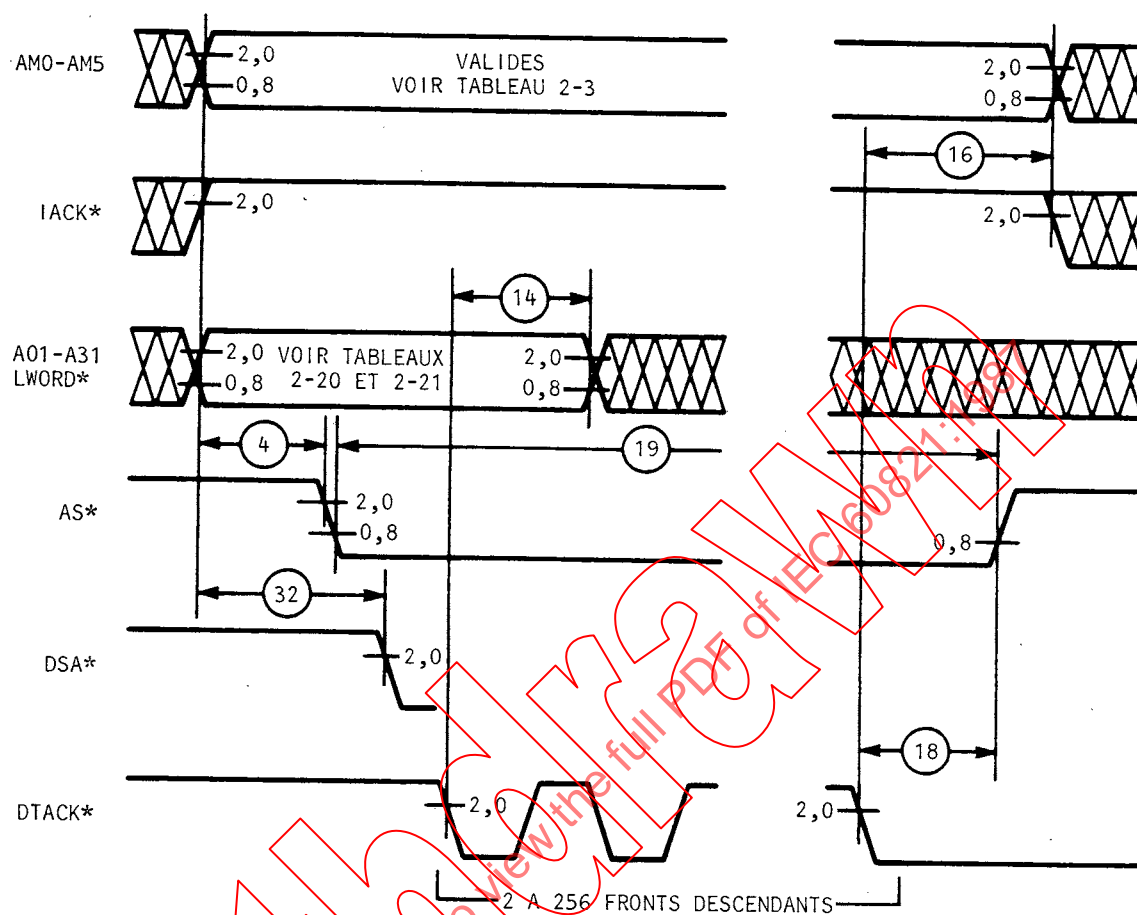


FIG. 2-14. - MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de diffusion d'adresse.

Transferts de blocs par octet;
transferts de blocs par double octet;
transferts de blocs par quadruple octet.

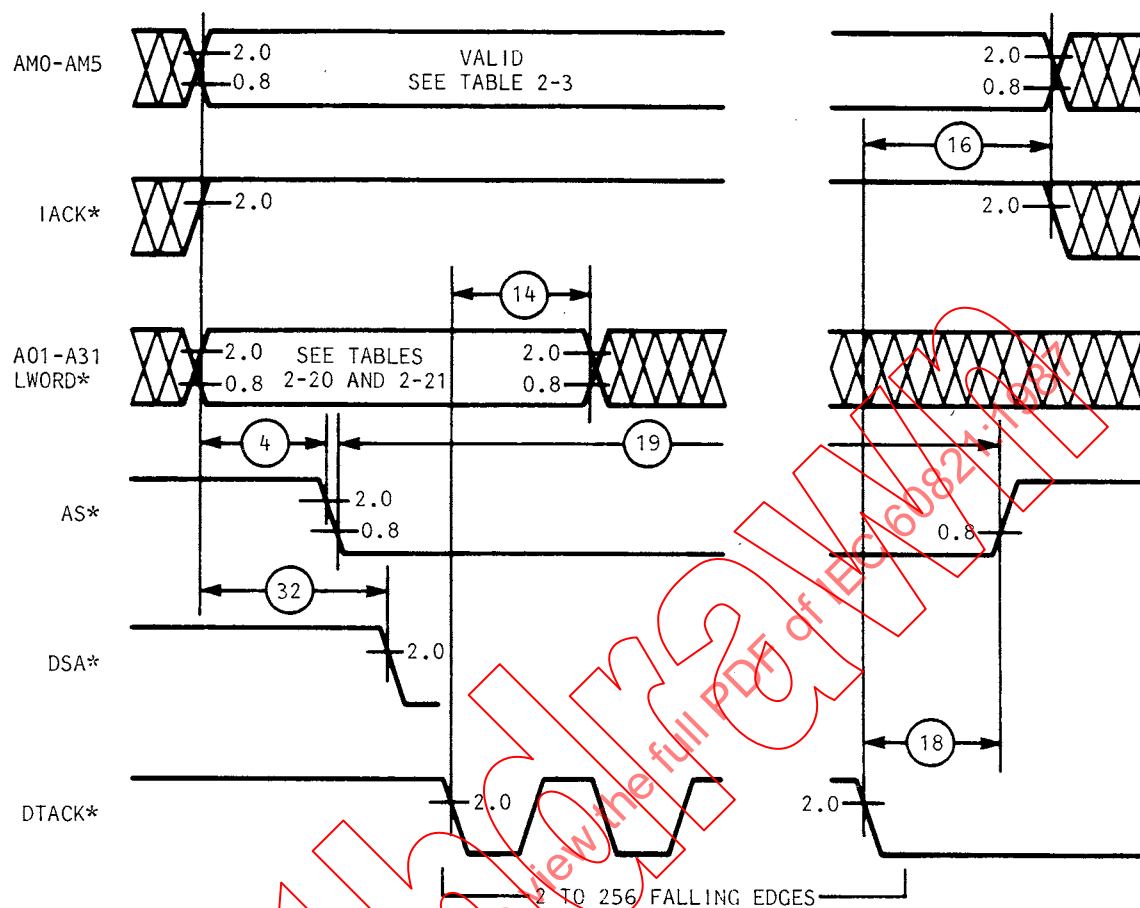


FIG. 2-14. ~ MASTER, SLAVE and LOCATION MONITOR - Address broadcast timing.

Single byte block transfers;
double byte block transfers;
quad byte block transfers.

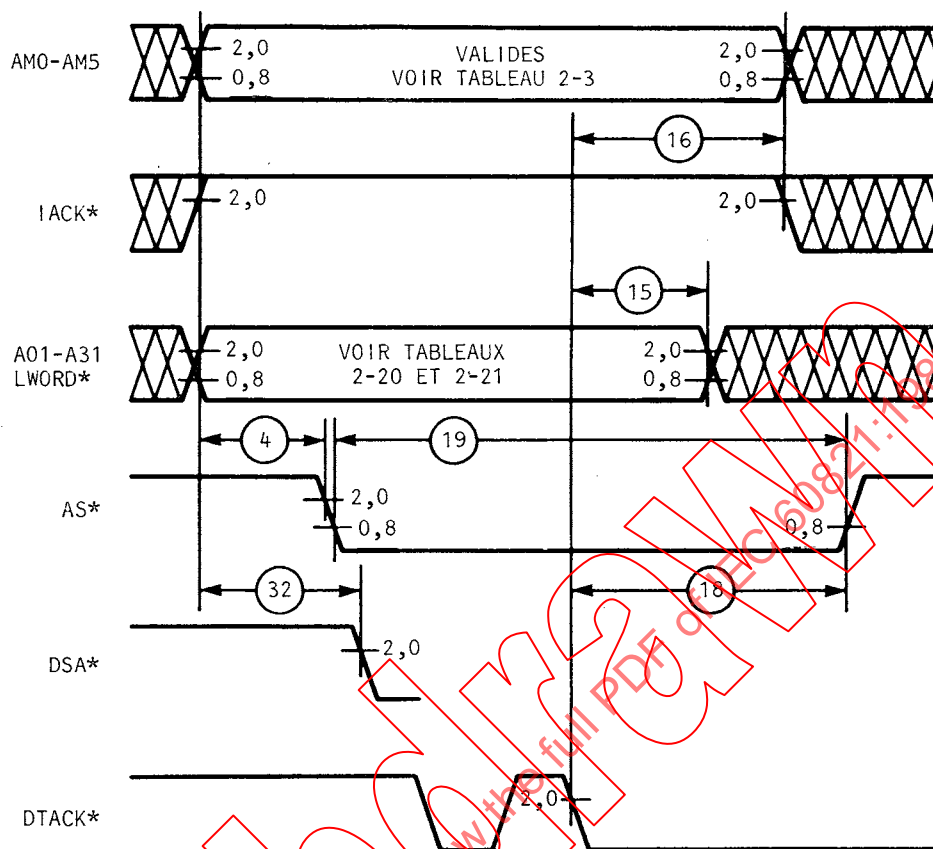


FIG. 2-15. - MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de diffusion d'adresse.

Cycles RMW pour octet unique;
cycles RMW pour double octet;
cycles RMW pour quadruple octet.

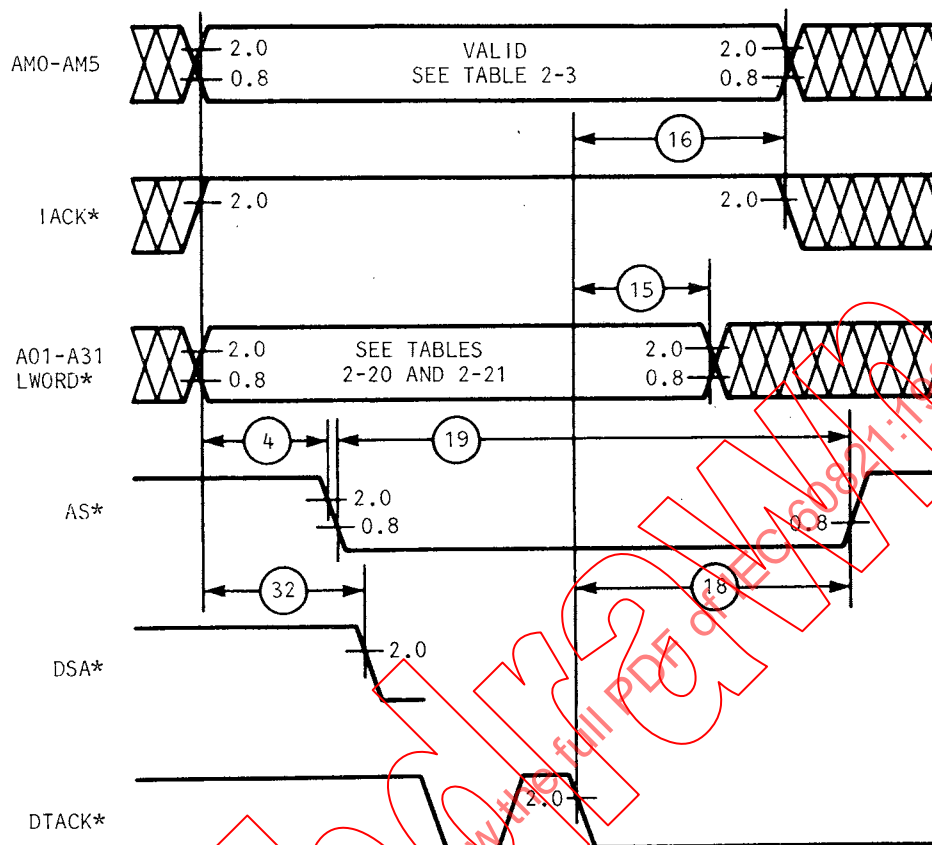


FIG. 2-15. - MASTER, SLAVE and LOCATION MONITOR -
Address broadcast timing.

Single byte RMW cycles;
double byte RMW cycles;
quad byte RMW cycles.

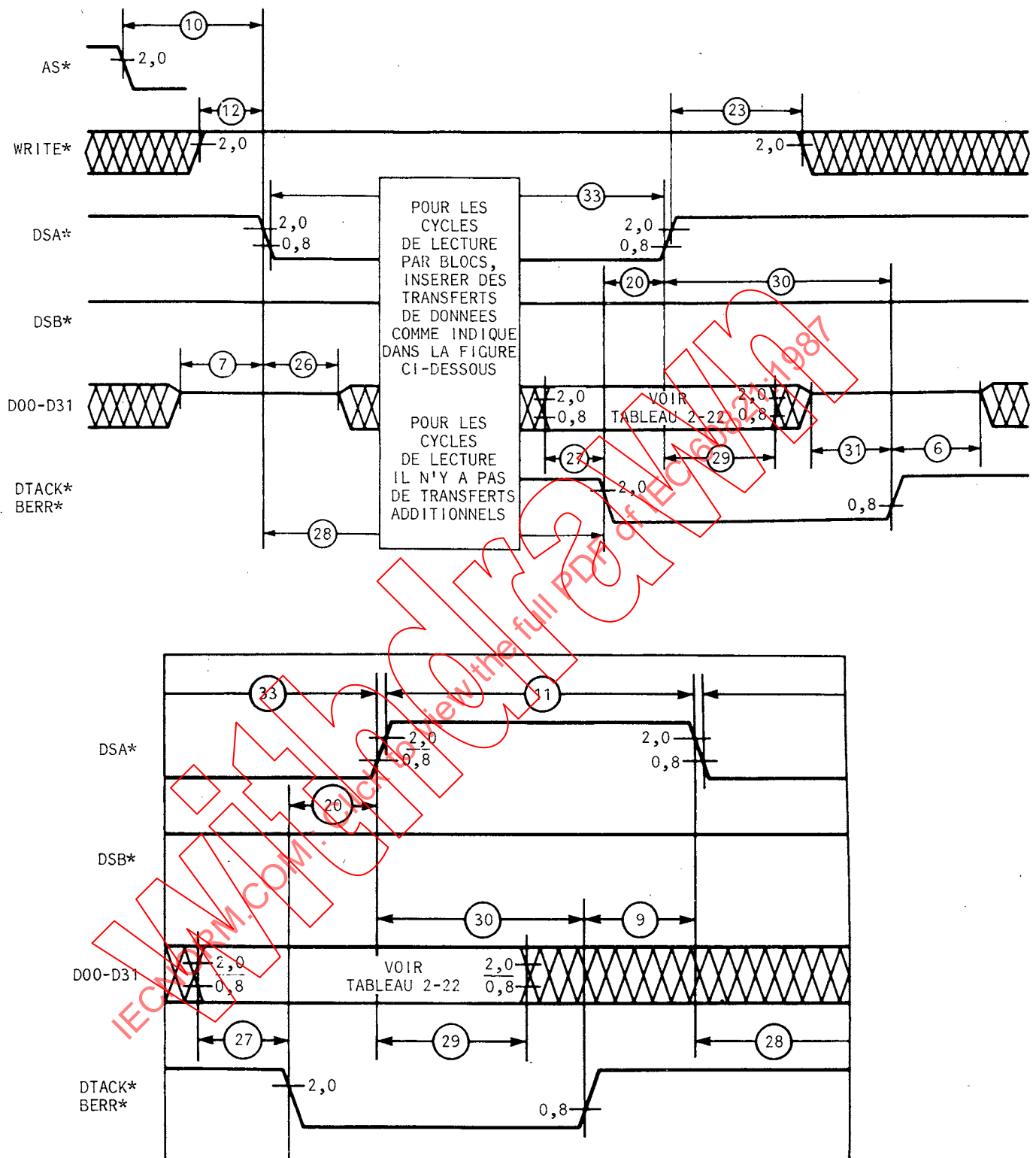


FIG. 2-16. - MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données.

```
LECTURE OCTET(0);    LECTURE OCTET(1);
LECTURE OCTET(2);    LECTURE OCTET(3);
LECTURE OCTET(0-2);  LECTURE OCTET(1-3);
LECTURE DE BLOCS PAR OCTET.
```

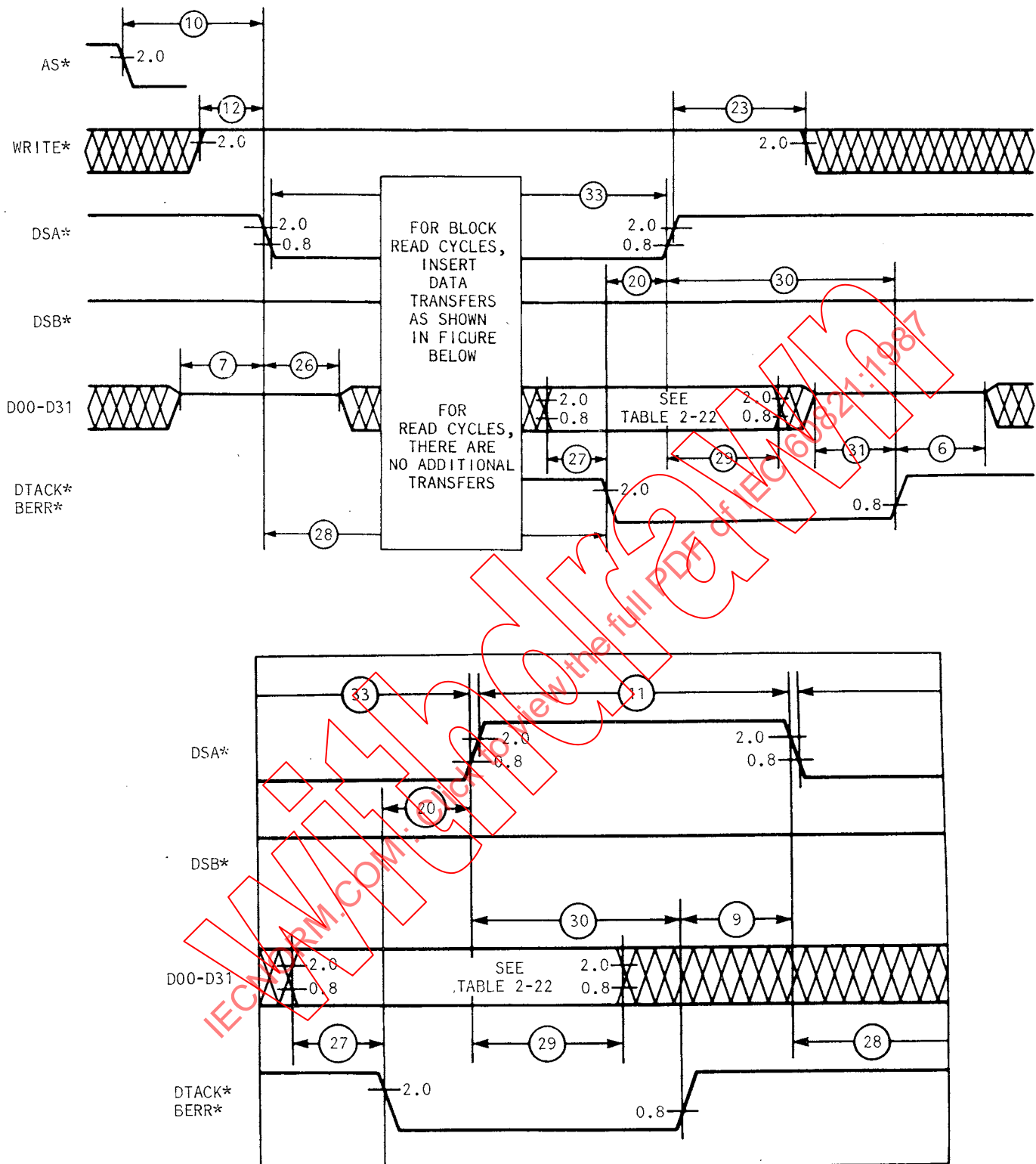


FIG. 2-16. - MASTER, SLAVE and LOCATION MONITOR - Data transfer timing.

BYTE(0) READ; BYTE(1) READ;
 BYTE(2) READ; BYTE(3) READ;
 BYTE(0-2) READ; BYTE(1-3) READ;
 SINGLE BYTE BLOCK READ.

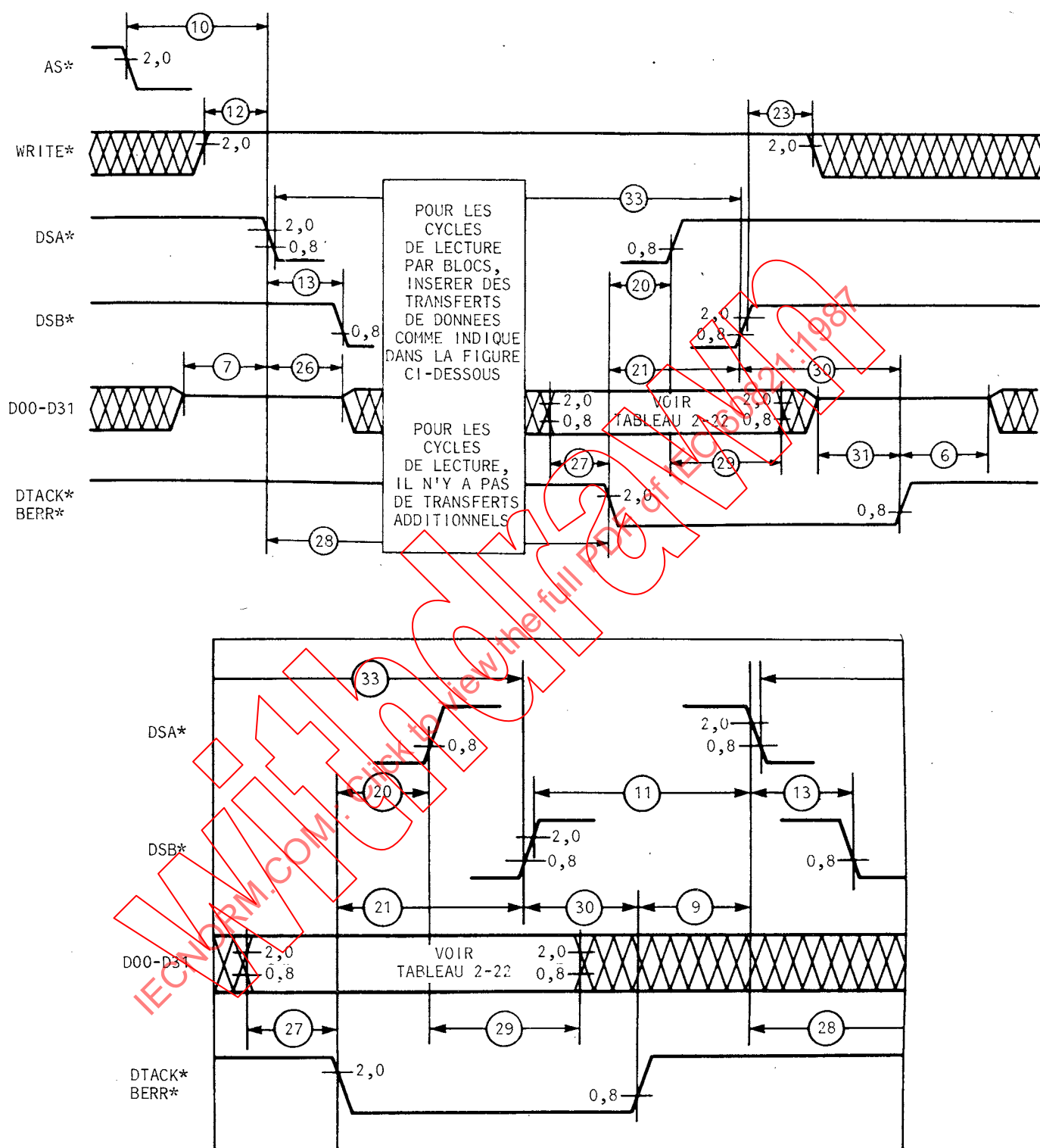


FIG. 2-17. - MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données.

LECTURE OCTET(0-1); LECTURE OCTET(2-3);
LECTURE OCTET(0-3); LECTURE OCTET(1-2);
LECTURE DE BLOCS PAR DOUBLE OCTET;
LECTURE DE BLOCS PAR QUADRUPLE OCTET.

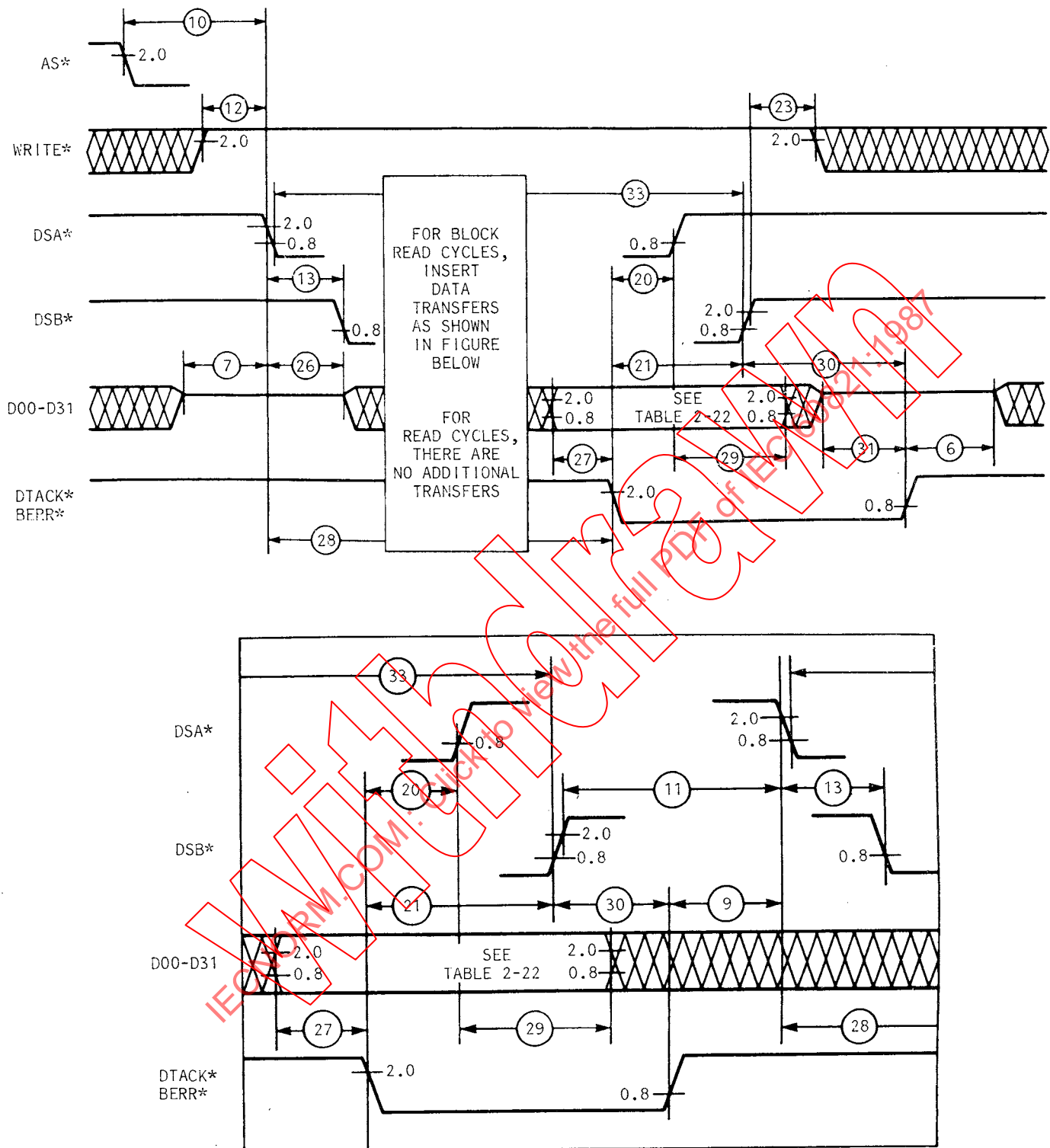
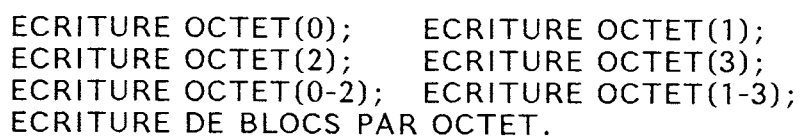


FIG. 2-17. - MASTER SLAVE and LOCATION MONITOR - Data transfer timing.

BYTE(0-1) READ; BYTE(2-3) READ;
 BYTE(0-3) READ; BYTE(1-2) READ;
 DOUBLE BYTE BLOCK READ;
 QUAD BYTE BLOCK READ.



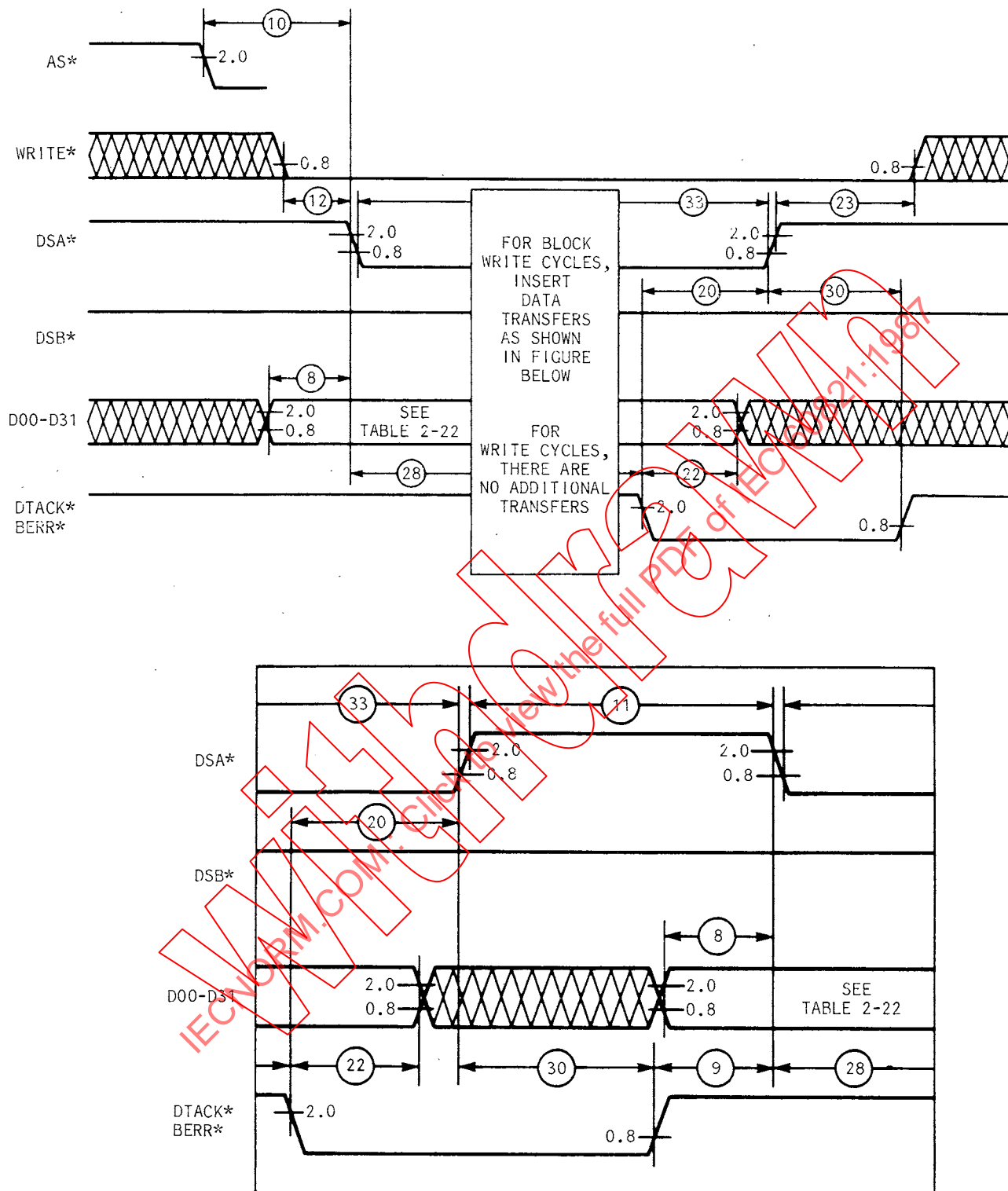


FIG. 2-18. - MASTER, SLAVE and LOCATION MONITOR - Data transfer timing.

BYTE(0) WRITE; BYTE(1) WRITE;
 BYTE(2) WRITE; BYTE(3) WRITE;
 BYTE(0-2) WRITE; BYTE(1-3) WRITE;
 SINGLE BYTE BLOCK WRITE.

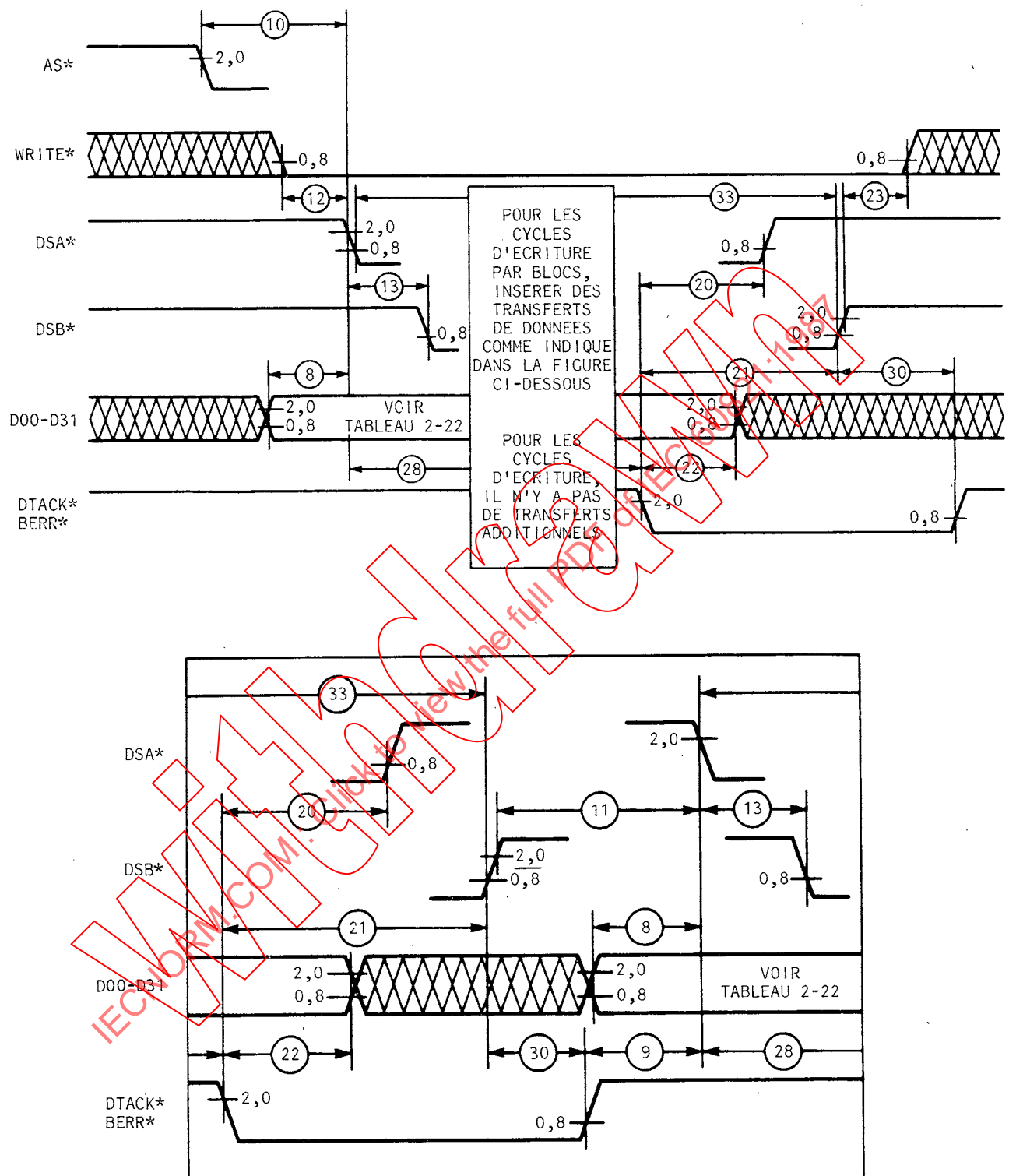


FIG. 2-19. - MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données.

ECRITURE OCTET(0-1); ECRITURE OCTET(2-3);
 ECRITURE OCTET(0-3); ECRITURE OCTET(1-2);
 ECRITURE DE BLOCS PAR DOUBLE OCTET;
 ECRITURE DE BLOCS PAR QUADRUPLE OCTET.

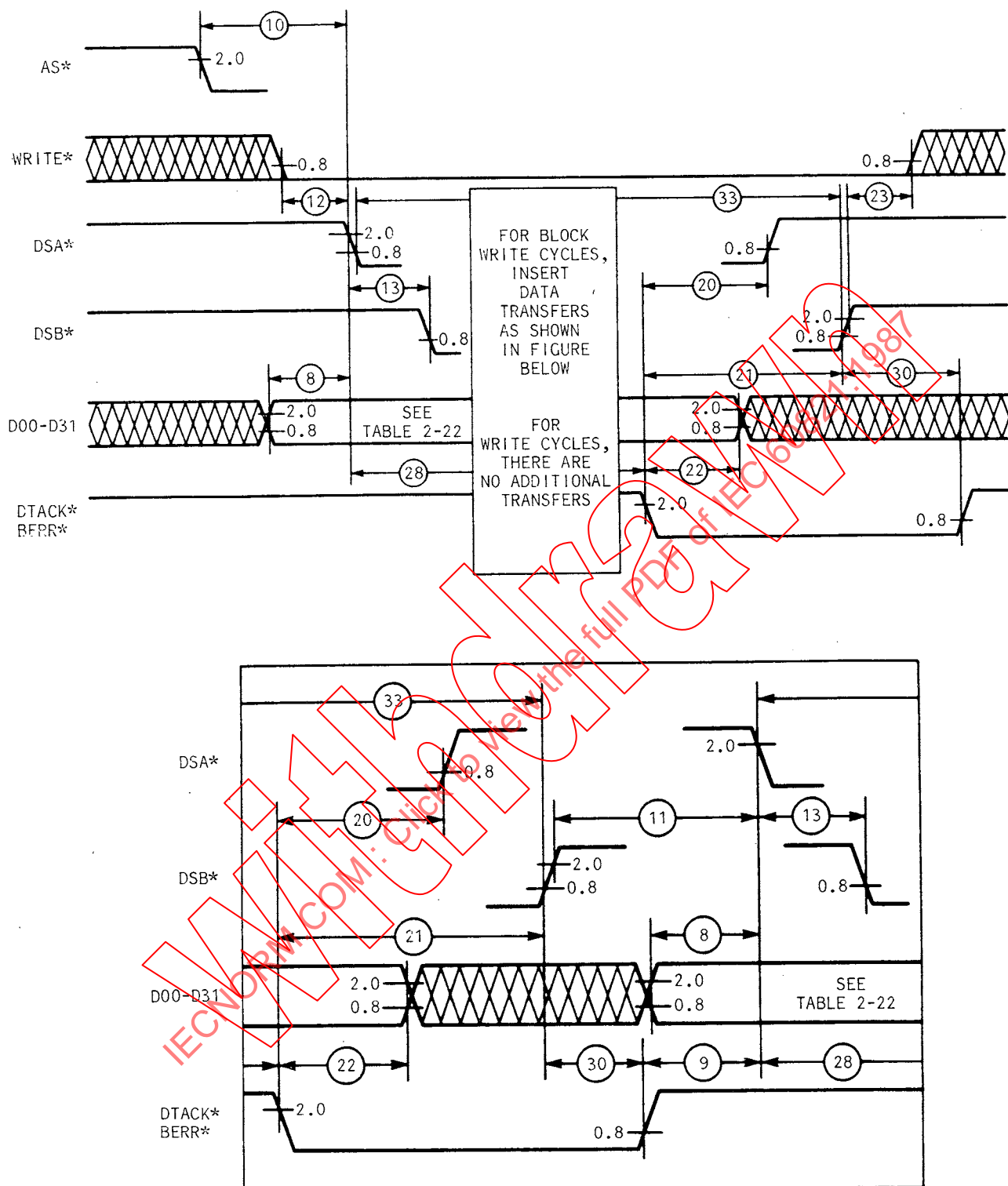


FIG. 2-19. - MASTER, SLAVE and LOCATION MONITOR - Data transfer timing.

BYTE(0-1) WRITE; BYTE(2-3) WRITE;
 BYTE(0-3) WRITE; BYTE(1-2) WRITE;
 DOUBLE BYTE BLOCK WRITE;
 QUAD BYTE BLOCK WRITE.

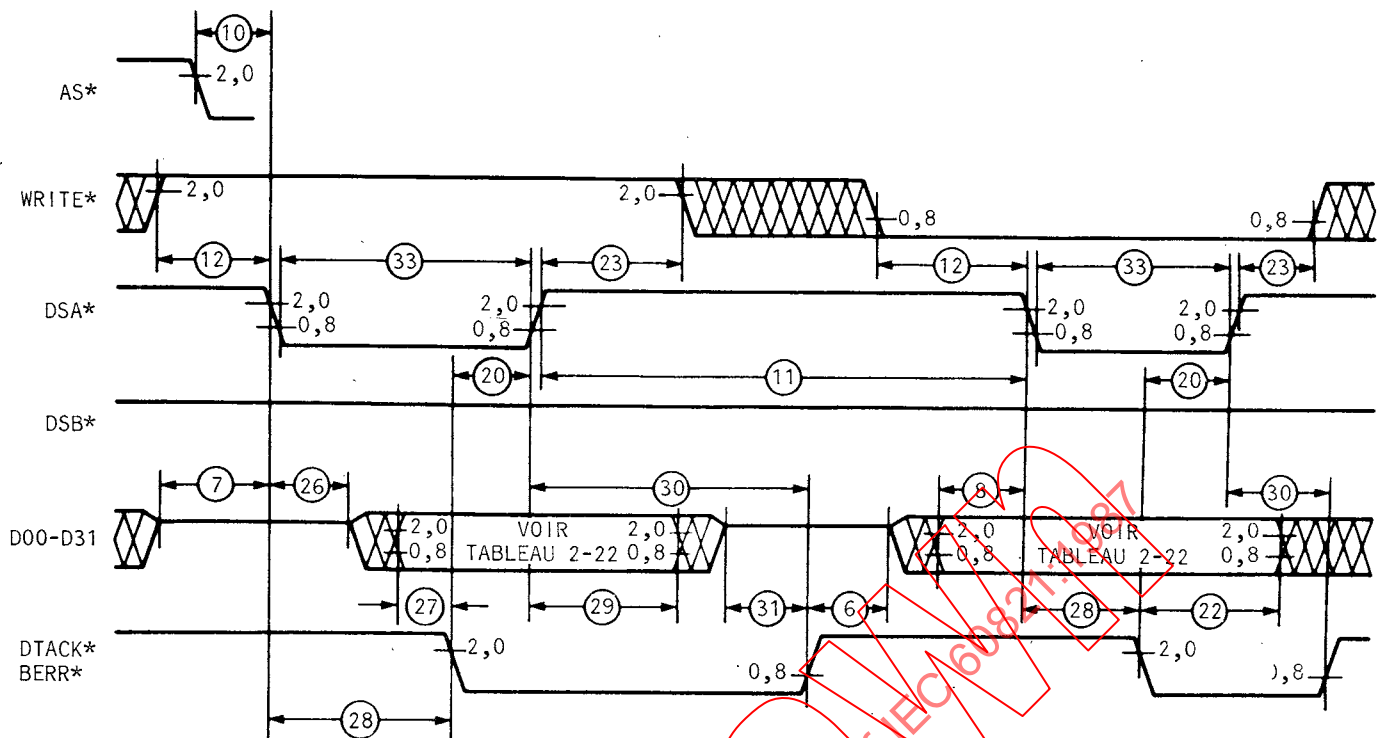


FIG. 2-20. - MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données.

Cycle RMW pour octet unique.

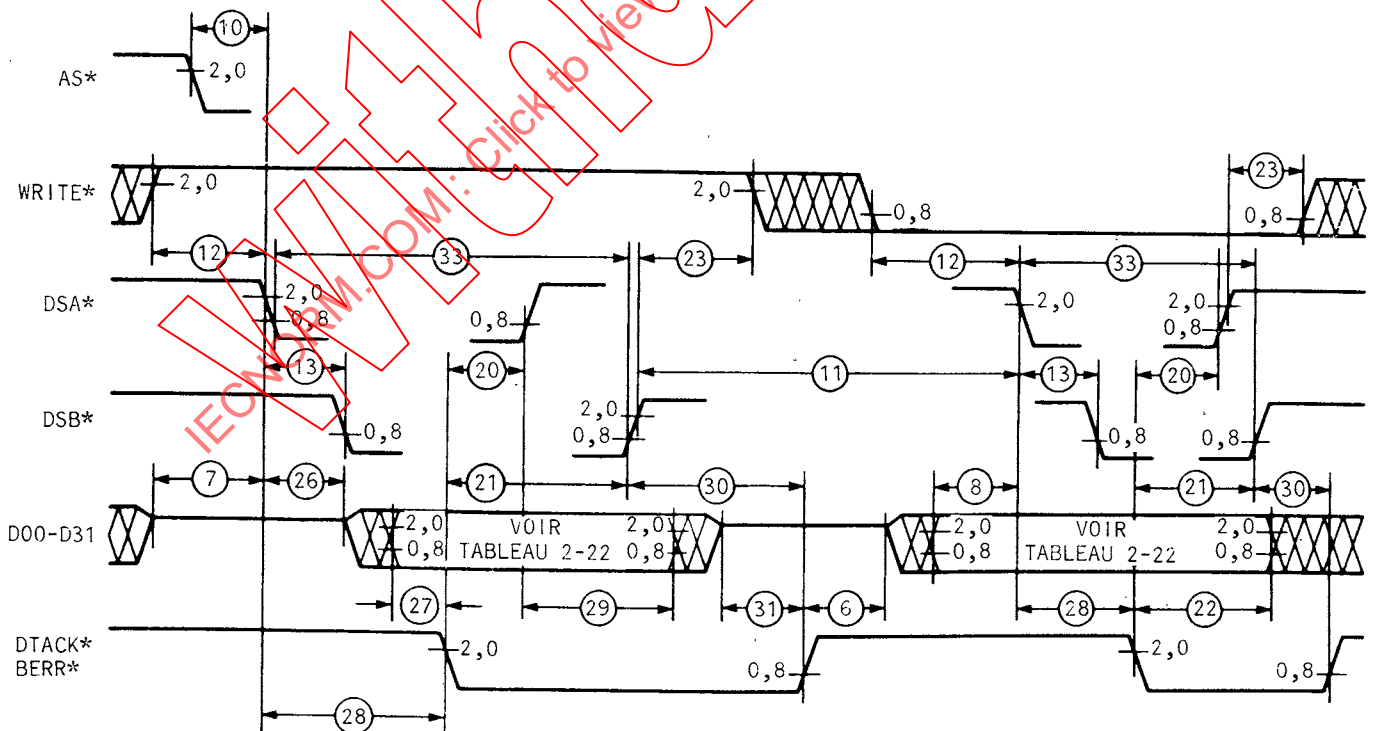


FIG. 2-21. - MAITRE, ESCLAVE et DETECTEUR D'ACCES - Chronogramme de transfert de données.

Cycles RMW pour double octet; cycles RMW pour quadruple octet.

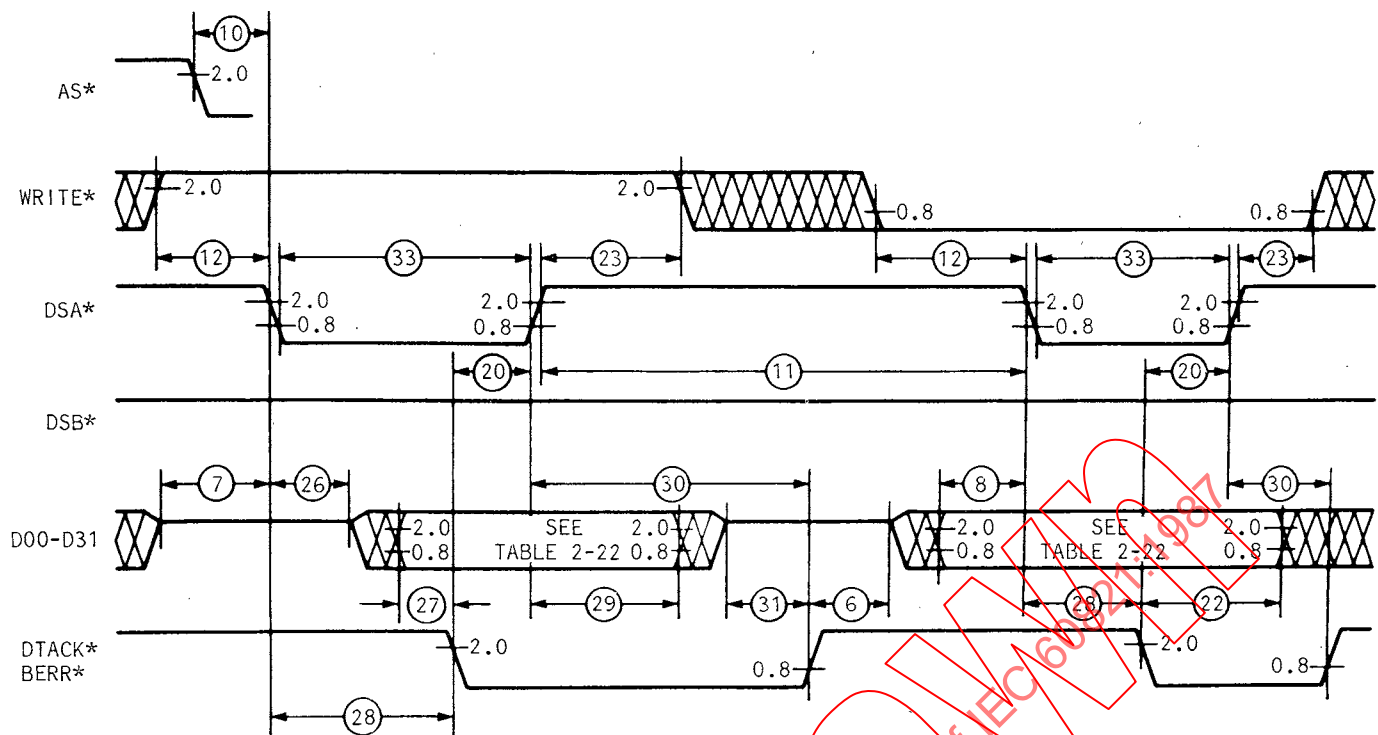


FIG. 2-20. - MASTER, SLAVE and LOCATION MONITOR - Data transfer timing.

Single byte RMW cycle.

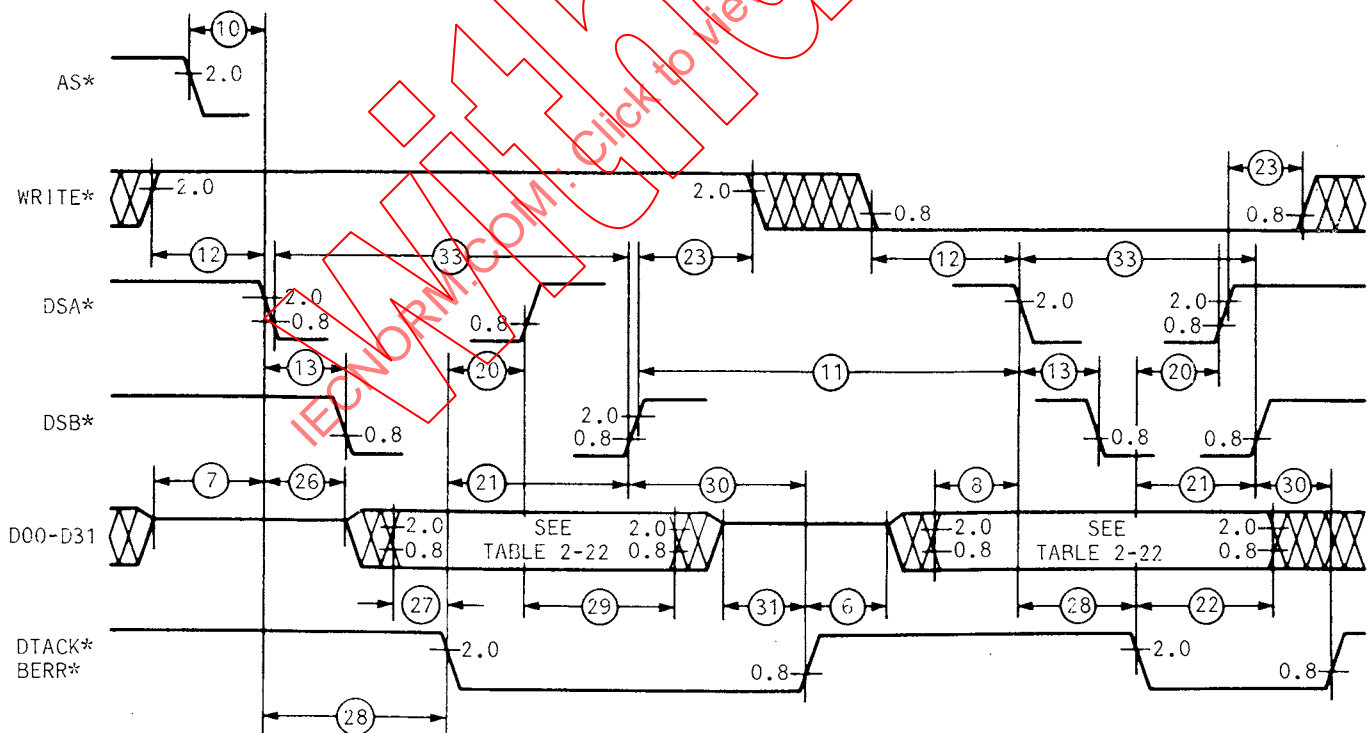


FIG. 2-21. - MASTER, SLAVE and LOCATION MONITOR - Data transfer timing.

Double byte RMW cycles; quad byte RMW cycles.

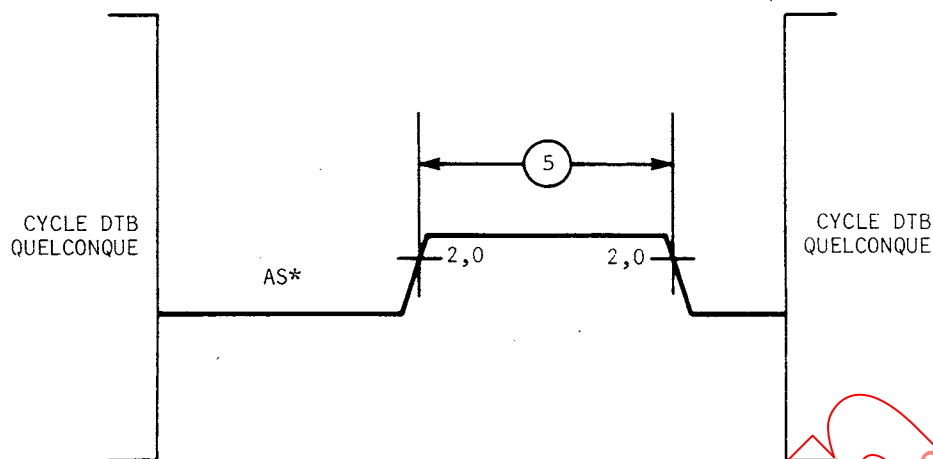


FIG. 2-22. - Chronogramme du signal de validation d'adresse entre les cycles.

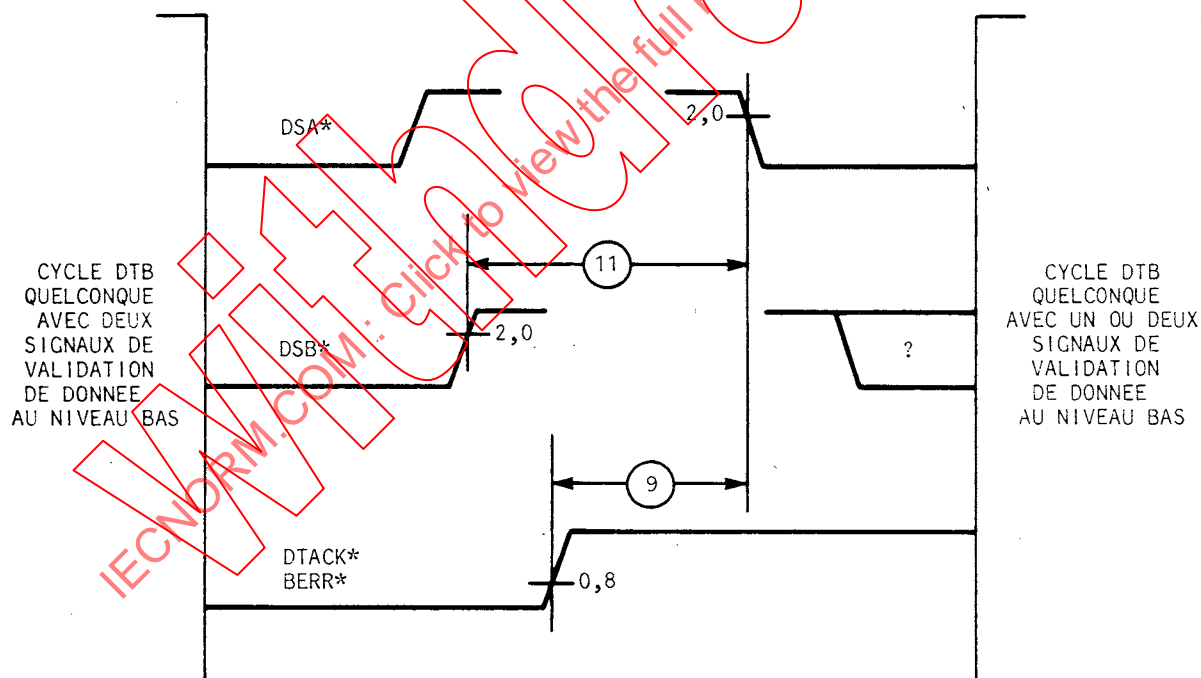


FIG. 2-23. - Chronogramme des signaux de validation de donnée entre les cycles.

Un cycle où les deux signaux de validation de donnée passent au niveau bas suivi par un cycle où un ou les deux signaux de validation de donnée passent au niveau bas.

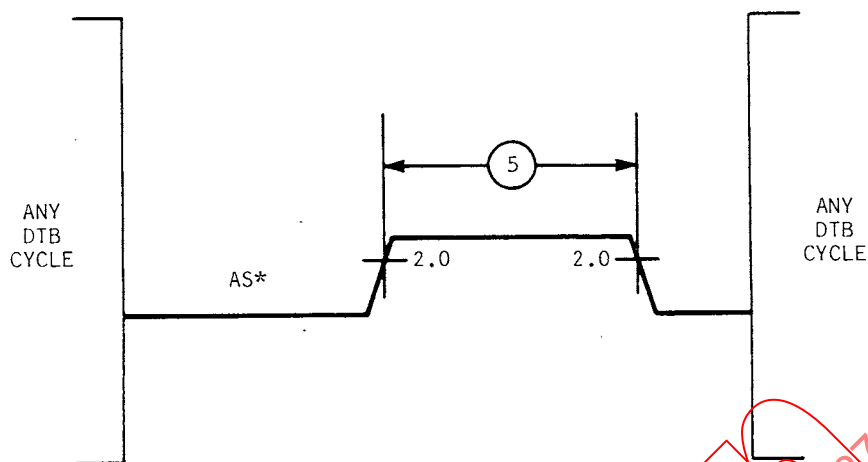


FIG. 2-22. - Address strobe inter-cycle timing.

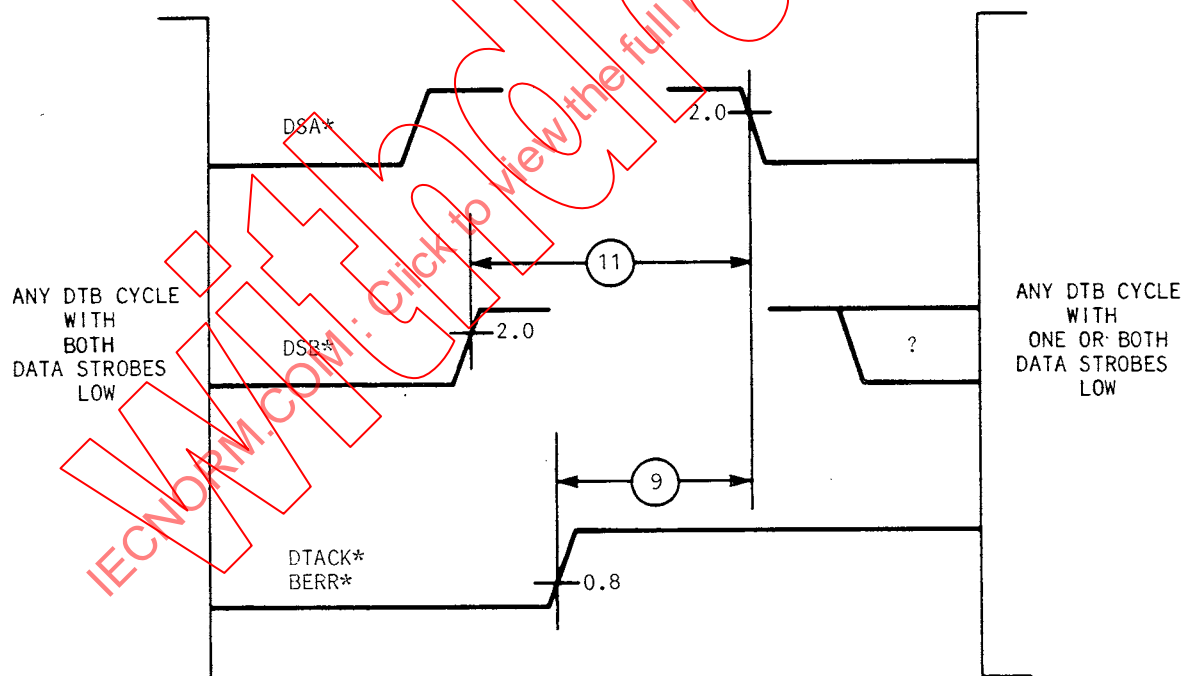


FIG. 2-23. - Data strobe inter-cycle timing.

A cycle where both data strobes go low followed by a cycle where one or both data strobes go low.

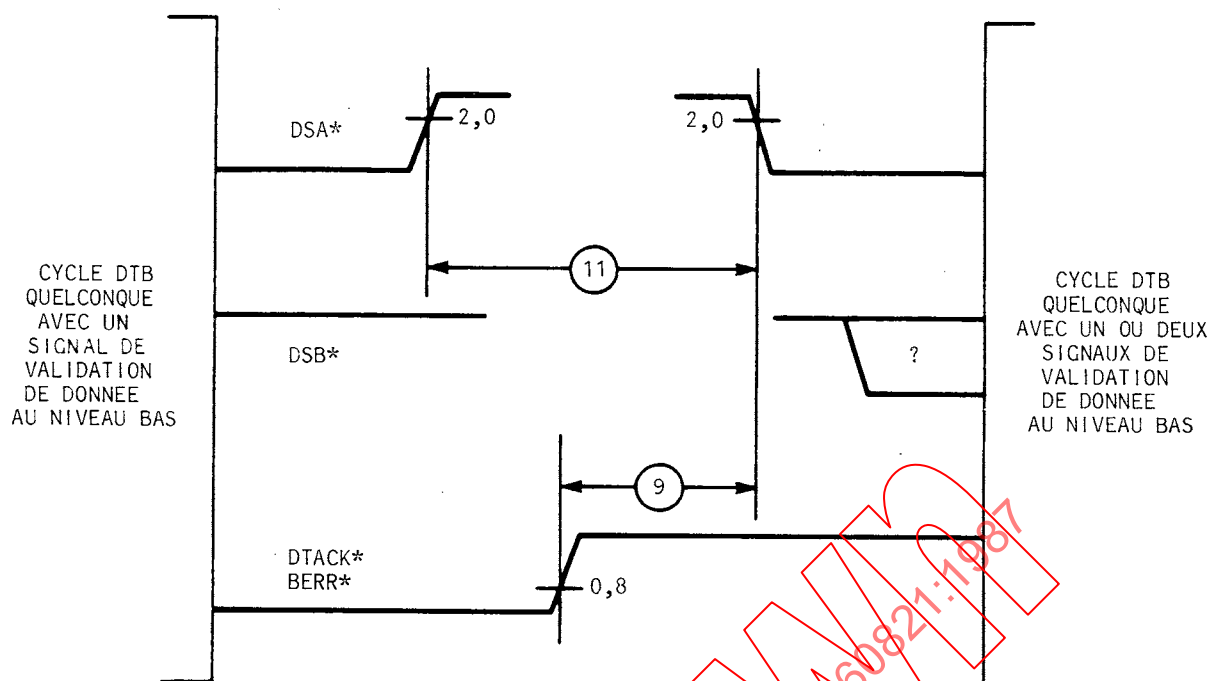


FIG. 2-24. - Chronogramme des signaux de validation de donnée entre les cycles.

Un cycle où un signal de validation de donnée passe au niveau bas suivi par un cycle où un ou les deux signaux de validation de donnée passent au niveau bas.

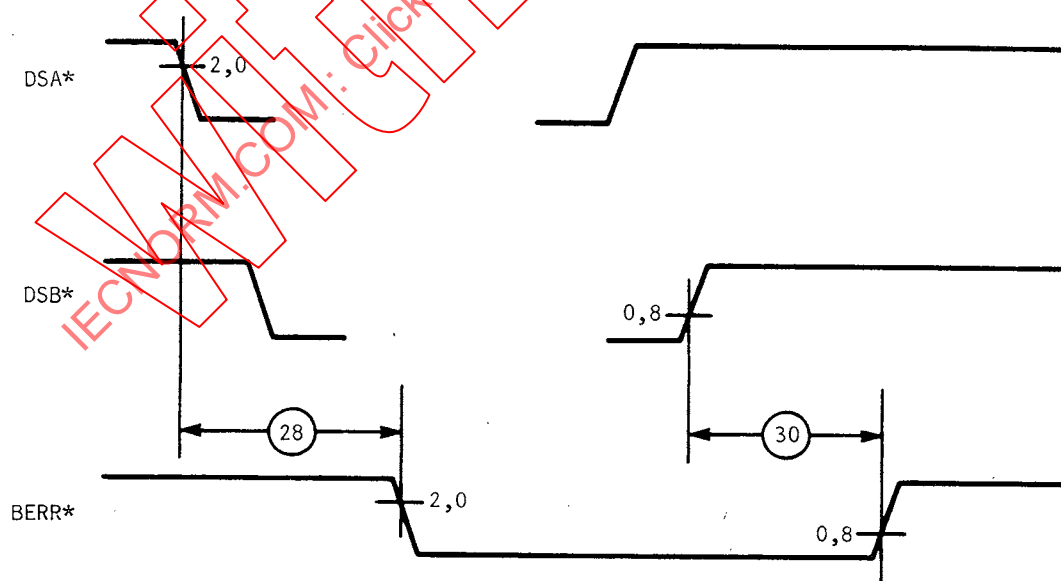


FIG. 2-25. - MAITRE, ESCLAVE et LIMITEUR DE TEMPS D'OCCUPATION DU BUS - Chronogramme de transfert de données.

Cycle de dépassement de temps d'occupation du bus.

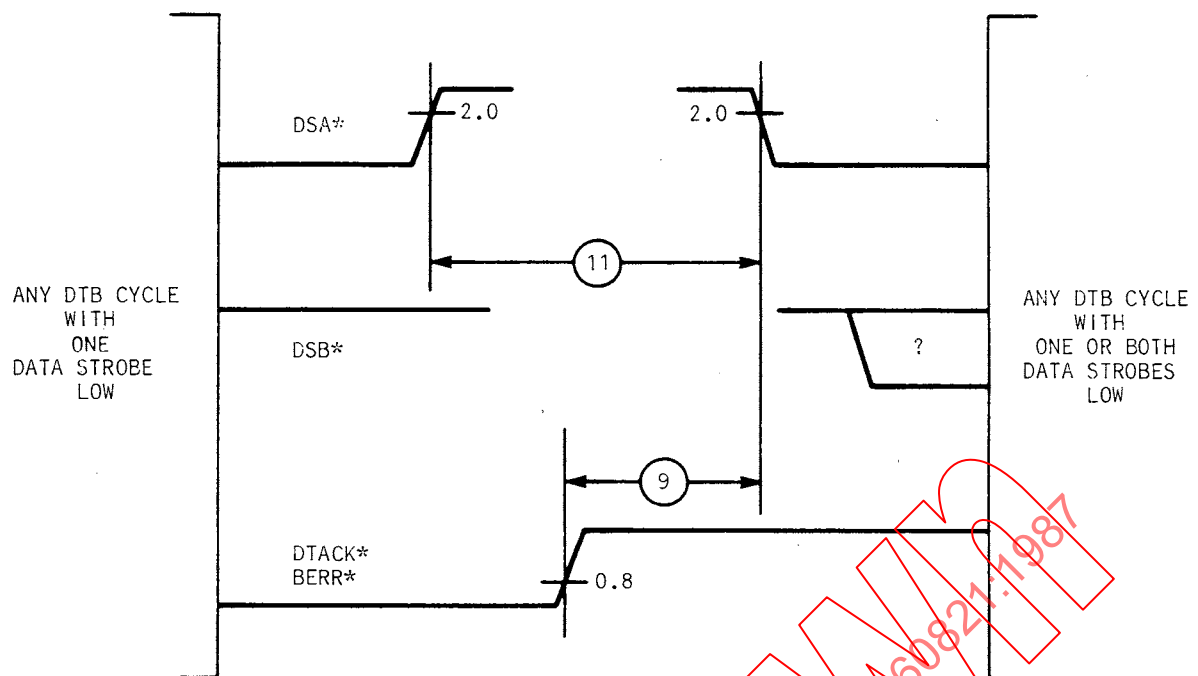


FIG. 2-24. - Data strobe inter-cycle timing.

A cycle where one data strobe goes low followed by a cycle where one or both data strobes go low.

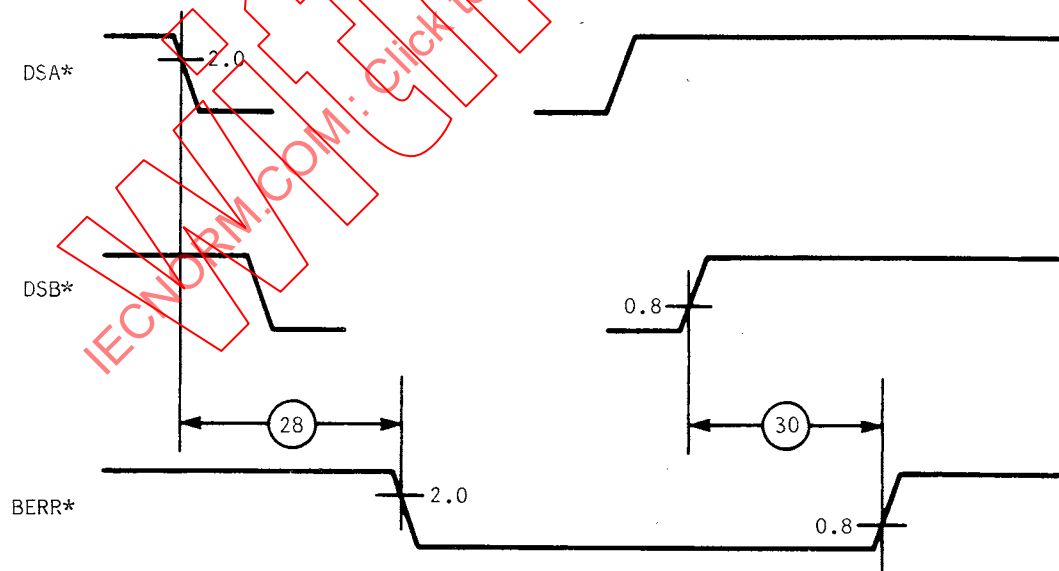


FIG. 2-25. - MASTER, SLAVE and BUS TIMER - Data transfer timing.

Timed-out cycle.

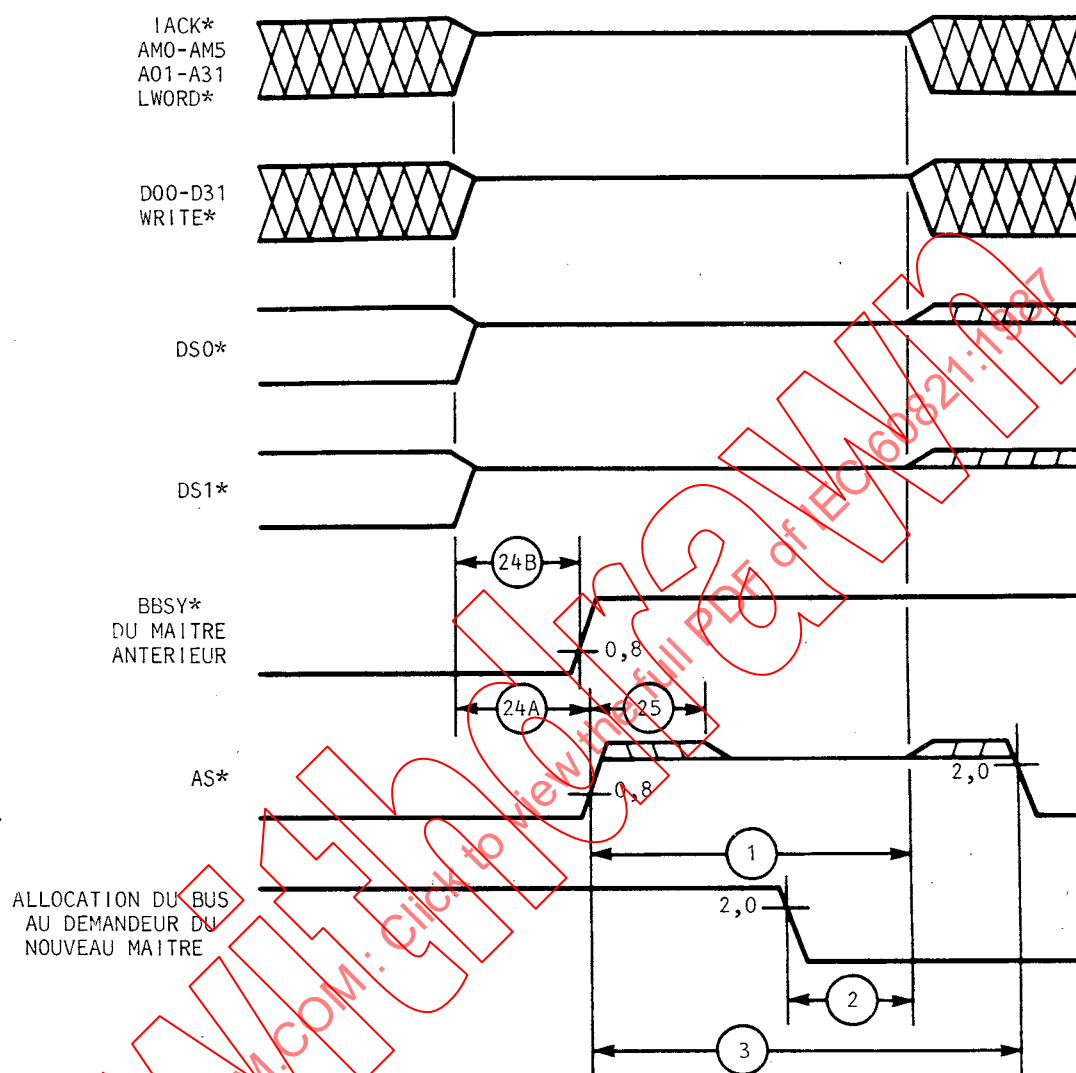


FIG. 2-26. - MAITRE - Chronogramme du transfert du contrôle du DTB.

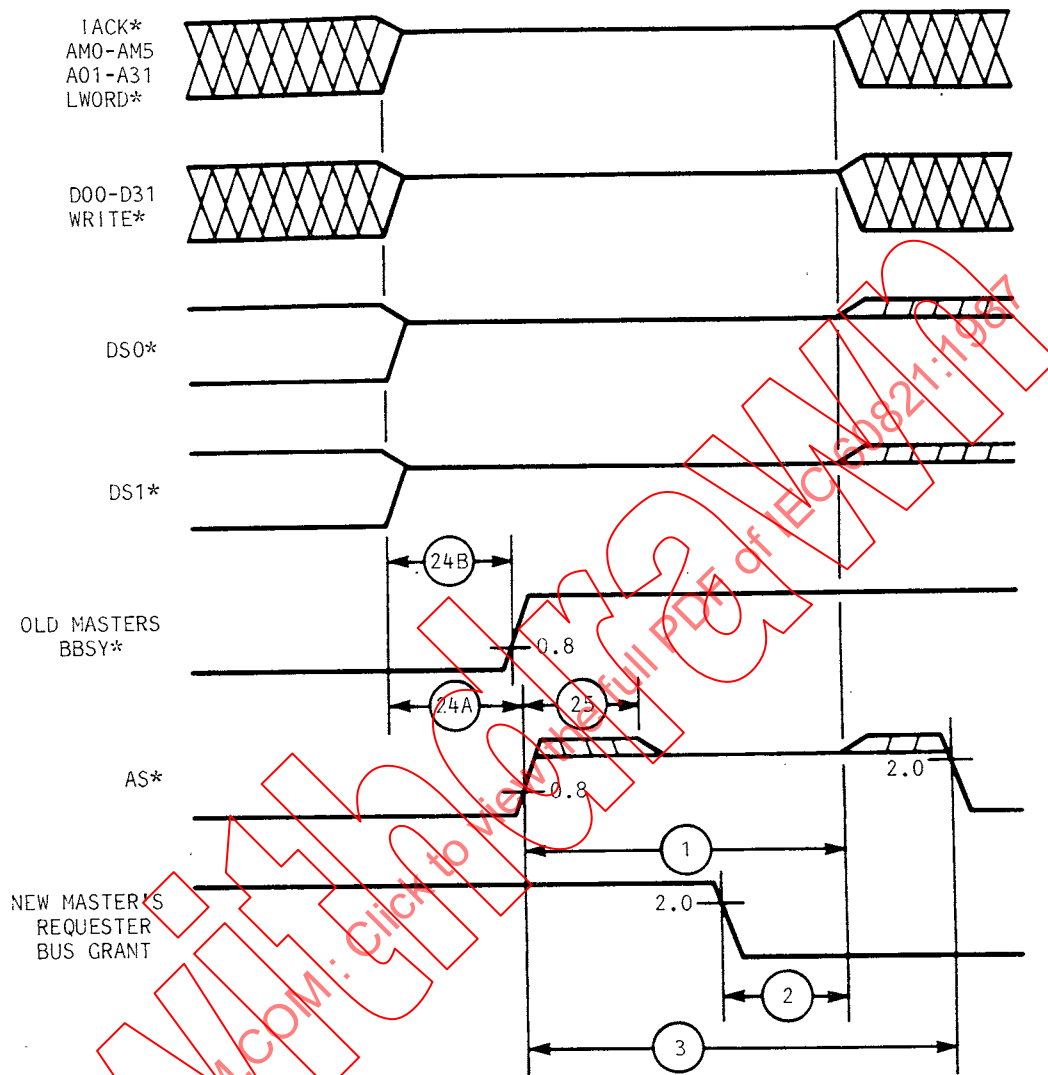


FIG. 2-26. - MASTER - DTB control transfer timing.

CHAPITRE 3: ARBITRAGE DU BUS DE TRANSFERT DE DONNEES DU BUS CEI 821

3.1 *Principes de l'arbitrage du bus*

Avec la réduction des coûts des microprocesseurs, il devient économiquement possible de concevoir des systèmes à plusieurs processeurs se partageant des ressources communes.

La plus importante de ces ressources communes est le bus de transfert de données, par lequel on a accès à toutes les autres ressources communes. Tout système multiprocesseur doit donc prévoir une méthode efficace d'allocation du bus de transfert de données. La rapidité de l'allocation étant essentielle, une méthode d'allocation par matériel est la seule possibilité. Le BUS CEI 821 répond à ce besoin grâce à son sous-système d'arbitrage. (Voir figure 3-1, page 194.)

Le sous-système d'arbitrage du BUS CEI 821:

- empêche l'utilisation simultanée du bus par deux MAITRES;
- planifie les demandes des MAITRES multiples pour optimiser l'utilisation du bus.

3.1.1 *Types d'arbitrage*

Lorsque plusieurs cartes demandent simultanément l'usage du DTB, le sous-système d'arbitrage détecte ces demandes et accorde le bus à une seule carte à la fois. La décision d'octroyer le bus à une carte en premier dépend de l'algorithme d'affectation qui est utilisé. Plusieurs algorithmes sont possibles. Le BUS CEI 821 en décrit trois: prioritaire, à priorité tournante et à niveau unique.

L'arbitrage prioritaire affecte le bus selon un schéma de priorité fixe où chacune des quatre lignes de demandes du bus a une priorité allant du plus haut niveau (BR3*) au plus bas (BR0*).

L'arbitrage à priorité tournante affecte le bus sur une base de rotation du niveau de priorité. S'il est affecté au DEMANDEUR par la ligne de demande du bus "BR(n)*", la priorité la plus élevée pour l'arbitration suivante passe alors à la ligne de demande du bus "BR(n-1)*".

L'arbitrage à niveau unique n'accepte des demandes que sur BR3* et se base sur la chaîne d'allocation de BR3* pour hiérarchiser les demandes.

AUTORISATION 3.1:

Des algorithmes d'affectation autres que prioritaires, à priorité tournante ou à niveau unique PEUVENT être utilisés. Par exemple, un schéma d'arbitrage peut donner la priorité la plus haute à BR3*, alors que les autres niveaux BR0* à BR2* restent sur une base d'arbitrage à priorité tournante.

CHAPTER 3: IEC 821 BUS DATA TRANSFER BUS ARBITRATION

3.1 *Bus arbitration philosophy*

As microprocessor costs decrease, it is becoming more cost effective to design systems with multiple processors sharing global resources.

The most fundamental of these global resources is the Data Transfer Bus through which all other global resources are accessed. Therefore, any system that supports multiprocessing needs to provide an efficient allocation method for the Data Transfer Bus. Because speed of allocation is vital, a hardware allocation scheme is the only practical alternative. The IEC 821 BUS meets this need with its arbitration subsystem. (See Figure 3-1, page 195.)

The IEC 821 BUS arbitration subsystem:

- prevents simultaneous use of the bus by two MASTERS;
- schedules requests from multiple MASTERS for optimum bus use.

3.1.1 *Types of arbitration*

When several boards request use of the DTB simultaneously, the arbitration subsystem detects these requests and grants the bus to one board at a time. The decision of which board is granted the bus first depends upon what scheduling algorithm is used. Many algorithms are possible. The IEC 821 BUS describes three: prioritized, round-robin, and single level.

Prioritized arbitration assigns the bus according to a fixed priority scheme where each of four bus request lines has a priority from highest (BR3*) to lowest (BR0*).

Round-robin arbitration assigns the bus on a rotating priority basis. When the bus is granted to the REQUESTER on bus request line "BR(n)*", then the highest priority for the next arbitration is assigned to bus request line "BR(n-1)*".

Single level arbitration only accepts requests on BR3*, and relies on BR3*'s bus grant daisy-chain to arbitrate the requests.

PERMISSION 3.1:

Scheduling algorithms other than priority, round-robin, or single level MAY be used. For example, an ARBITER's algorithm might give highest priority to BR3*, but grant the bus to BR0* through BR2* on a round-robin basis.

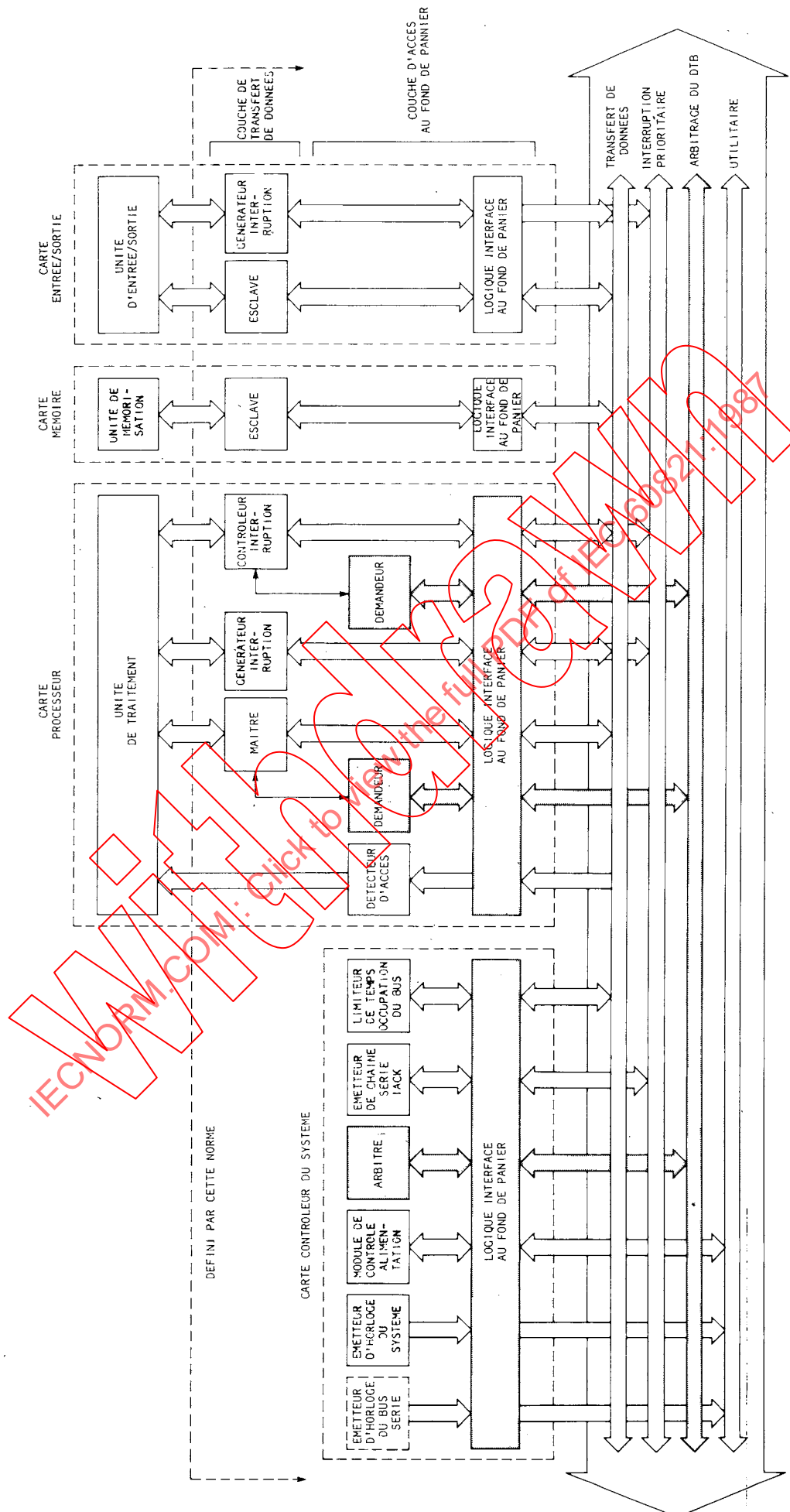


FIG. 3-1. - Schéma-bloc fonctionnel de l'arbitrage du bus.

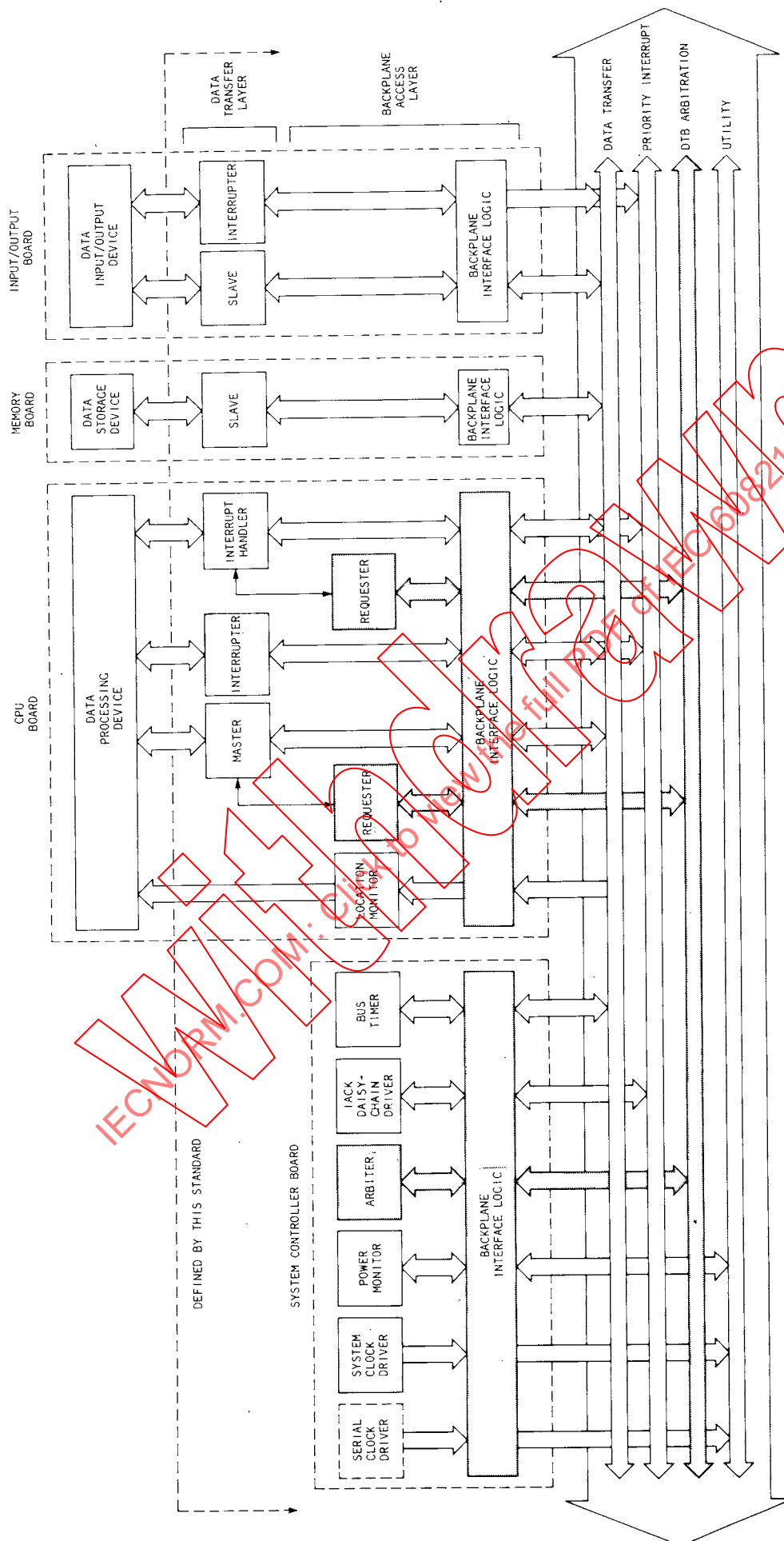


FIG. 3-1. - Arbitration bus functional block diagram.

3.2 Lignes du bus d'arbitrage

Le bus d'arbitrage est constitué de six lignes du BUS CEI 821 et quatre lignes en chaîne série qui nécessitent des noms spéciaux de signaux. Les signaux qui entrent dans chaque carte sont appelés lignes d'"ENTREE d'allocation du bus"(BGxIN*), et les signaux qui sortent de chaque carte sont appelés lignes de "SORTIE d'allocation du bus" (BGxOUT*). Les lignes qui sortent de l'emplacement n sous le nom de BGxOUT* entrent dans l'emplacement n+1 désignées par BGxIN*. Cela est illustré dans la figure 3-2, page 198.

OBSERVATION 3.1:

Dans tout ce chapitre, les termes BRx*, BGxIN* et BGxOUT* sont utilisés pour décrire les lignes de demandes et d'allocation du bus, où x prend une valeur comprise entre zéro et trois.

Dans le système d'arbitrage du BUS CEI 821 un module DEMANDEUR commande les lignes suivantes:

- 1 ligne de demande du bus (une parmi BR0* à BR3*)
- 1 ligne de sortie d'allocation du bus (une parmi BG0OUT* à BG3OUT*)
- 1 ligne de bus occupé (BBSY*).

REGLE 3.1:

SI une carte de BUS CEI 821 ne génère pas de demandes de bus sur l'un des niveaux,
ALORS elle DOIT propager les signaux de la chaîne série relatifs à ces niveaux provenant des lignes BGxIN* vers les lignes BGxOUT* correspondantes.

AUTORISATION 3.2:

La propagation pour les lignes non utilisées de la chaîne série d'allocation du bus PEUT être faite en utilisant des cavaliers ou des circuits logiques actifs. La dernière approche permet la sélection du niveau de demande sous le contrôle logiciel, alors que de la première résulte une propagation plus rapide à travers la chaîne série.

Trois types d'ARBITRES sont décrits dans la présente norme du BUS CEI 821:

- Prioritaire (PRI)
- Sélection à priorité tournante (RRS)
- Niveau unique (SGL).

Le fonctionnement de ces trois types d'ARBITRES est décrit dans la section 3.3.

L'ARBITRE PRI commande les lignes suivantes:

- 1 ligne de libération du bus (BCLR*)
- 4 lignes d'entrée d'allocation du bus (emplacement 1 BG0IN* à BG3IN*).

3.2 Arbitration bus lines

The arbitration bus consists of six bused IEC 821 BUS lines and four daisy-chained lines. These daisy-chained lines require special signal names. The signals entering each board are called "Bus Grant IN" lines (BGxIN*), while the signals leaving each board are called "Bus Grant OUT" lines (BGxOUT*). The lines which leave slot n as BGxOUT* enter slot n+1 as BGxIN*. This is illustrated in Figure 3-2, page 199.

OBSERVATION 3.1:

In all descriptions in this chapter, the terms BRx*, BGxIN*, and BGxOUT* are used to describe the bus request and bus grant lines, where x takes on any value from zero to three.

In the IEC 821 BUS arbitration system, a REQUESTER module drives the following lines:

- 1 bus request line (one of BR0* through BR3*)
- 1 bus grant out line (one of BG0OUT* through BG3OUT*)
- 1 bus busy line (BBSY*).

RULE 3.1:

IF an IEC 821 BUS board does not generate bus requests on some bus request levels,
THEN it MUST propagate the daisy-chain signals for those levels from its BGxIN* lines to its respective BGxOUT* lines.

PERMISSION 3.2:

The propagation for the unused lines of the bus grant daisy-chain MAY be done using jumpers or active logic. The latter approach allows selection of the request level under software control, while the former results in faster propagation through the daisy-chain.

Three types of ARBITERS are described in the IEC 821 BUS standard. They are:

- Prioritized (PRI)
- Round-Robin Select (RRS)
- Single Level (SGL).

The operation of these three types of ARBITERS is described in Section 3.3.

A PRI ARBITER drives the following:

- 1 bus clear line (BCLR*)
- 4 bus grant lines (slot 1 BG0IN* through BG3IN*).

Un ARBITRE RRS commande les quatre lignes BGxIN* de l'emplacement 1 et optionnellement la ligne BCLR*.

Un ARBITRE SGL commande seulement BG3IN* à l'emplacement 1.

Deux lignes supplémentaires sont importantes pour le séquençage de mise en ou hors tension du système d'arbitrage: SYSRESET* et ACFAIL*. Bien que leur action sur le système d'arbitrage soit traitée dans ce chapitre, ces lignes sont décrites dans le chapitre 5.

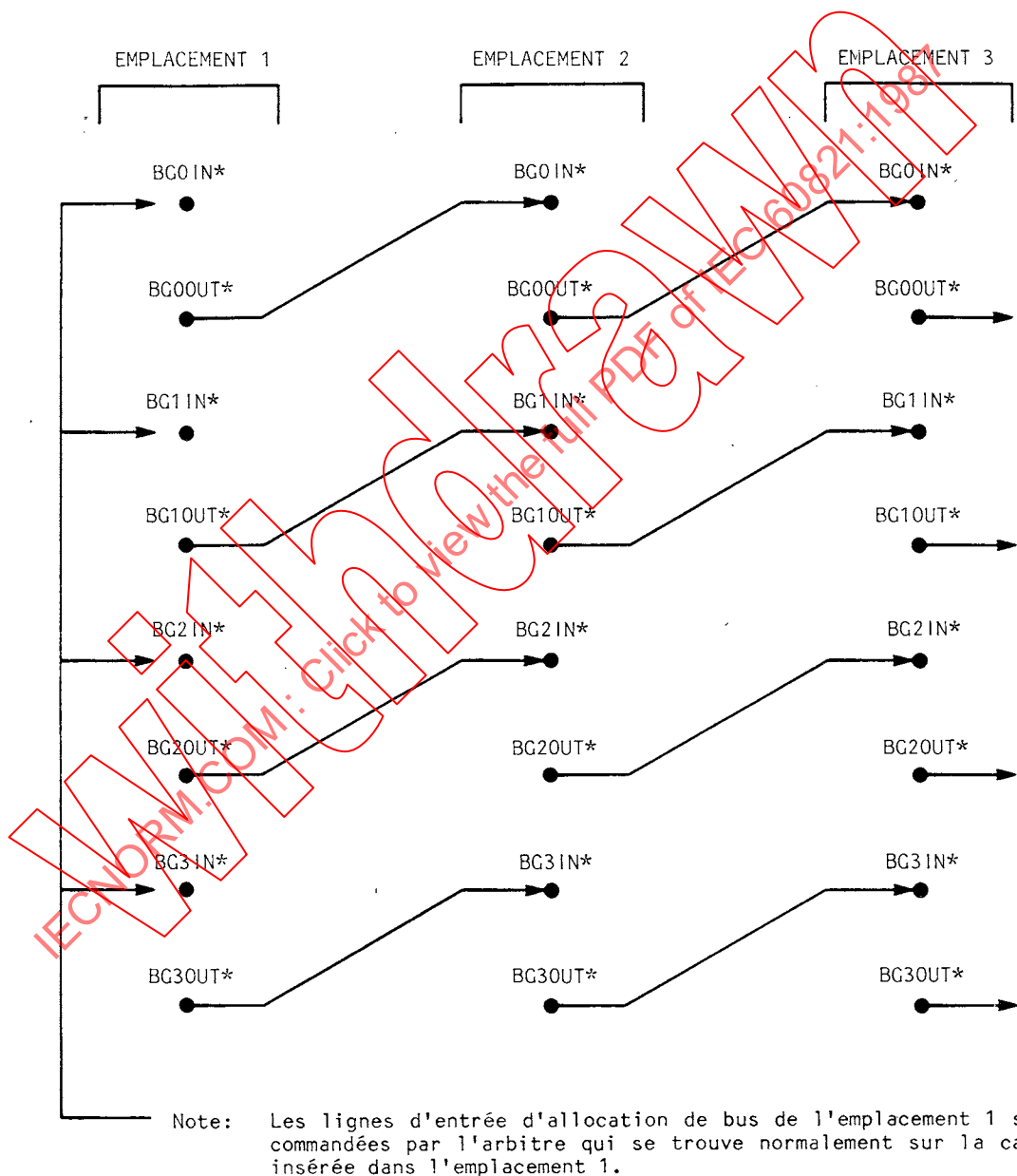


FIG. 3-2. - Illustration des lignes de la chaîne série d'allocation du bus.

An RRS ARBITER drives the four slot 1 BGxIN* lines and, optionally, the BCLR* line.

A SGL ARBITER drives only BG3IN* at slot 1.

Two additional lines are intimately connected with the arbitration system during power-up and power-down sequencing: SYSRESET*, and ACFAIL*. While their impact on the arbitration system is included in this chapter, these lines are discussed in Chapter 5.

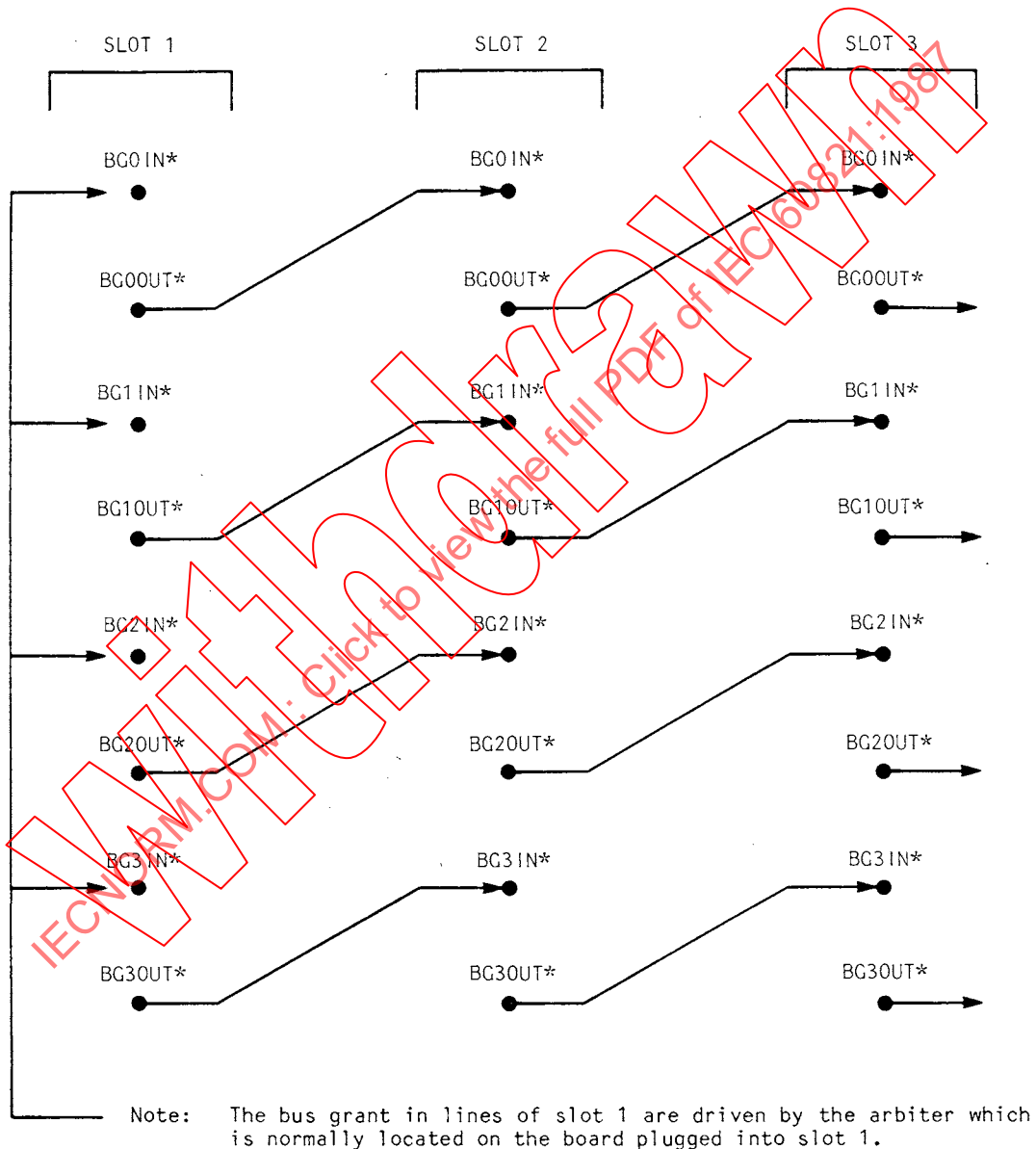


FIG. 3-2. - Illustration of the daisy-chained bus grant lines.

3.2.1 Lignes de demande et d'allocation du bus

Les lignes de demande du bus sont utilisées par chaque DEMANDEUR pour requérir l'utilisation du DTB. Les lignes d'allocation du bus permettent à l'ARBITRE d'accorder l'utilisation du bus. Il le fait en commandant au niveau bas une ligne de chaîne série d'allocation du bus. Ce niveau bas se propage le long de la chaîne série au travers de chaque carte. Si une carte n'utilise jamais un certain niveau de demande/-allocation x , le signal doit être transmis par cette carte. Lorsque la carte utilise un niveau de demande/allocation x , le signal $BGxIN^*$ correspondant est pris en compte par la carte. Si le DEMANDEUR de la carte a une demande du DTB en cours sur ce niveau, il ne transmet pas sur $BGxOUT$ le niveau bas. Dans le cas contraire, il transmet ce niveau.

REGLE 3.2:

SI un emplacement de fond de panier du BUS CEI 821 n'est pas occupé par une carte, et s'il y a des cartes en aval sur la chaîne série,

ALORS des cavaliers DOIVENT être installés dans l'emplacement vide pour propager le signal à travers la chaîne série.

OBSERVATION 3.2:

La spécification mécanique du fond de panier du chapitre 7 décrit les moyens d'installation de cavaliers dans chaque emplacement.

REGLE 3.3:

L'ARBITRE DOIT toujours être situé dans l'emplacement 1.

3.2.2 Ligne d'occupation du bus ($BBSY^*$)

Lorsqu'un DEMANDEUR a reçu le contrôle du bus de transfert de données par la chaîne série d'allocation du bus, il commande $BBSY^*$ au niveau bas. Ensuite, il a le contrôle du DTB jusqu'à ce qu'il libère $BBSY^*$, ce qui permet à l'ARBITRE d'accorder le DTB à un autre DEMANDEUR.

3.2.3 Ligne de libération du bus ($BCLR^*$)

L'ARBITRE PRI commande $BCLR^*$ au niveau bas pour informer le MAITRE qui contrôle actuellement le DTB qu'une demande de priorité plus haute est en attente. Le MAITRE actuel n'est pas obligé de libérer le bus dans une limite de temps prescrite. Généralement, il continue de transférer les données jusqu'à ce qu'il atteigne un point d'arrêt approprié, et alors permet à son DEMANDEUR placé sur la carte, de libérer $BBSY^*$.

AUTORISATION 3.3:

Bien qu'il ne soit pas nécessaire que les ARBITRES RRS commandent la ligne $BCLR^*$, ils PEUVENT le faire.

SUGGESTION 3.1:

SI un ARBITRE RRS commande la ligne $BCLR^*$ au niveau bas,

ALORS le concevoir pour agir ainsi dès qu'il y a une demande en attente sur n'importe quelle ligne non acquittée de demande du bus.

3.2.1 *Bus request and bus grant lines*

The bus request lines are used by each REQUESTER to request use of the DTB. The bus grant lines allow the ARBITER to award use of the bus. It does this by driving a bus grant daisy-chain line low. This low level propagates down the daisy-chain, typically passing through several boards in the process. If a board never uses a particular request/grant level, the signal is passed through that board. Where the board uses a request/grant level x , the corresponding signal $BGxIN^*$ is gated on board. If its on-board REQUEST is currently requesting the DTB on that level, it does not pass the low level on to its $BGxOUT^*$. Otherwise, it passes on the low level.

RULE 3.2:

IF an IEC 821 BUS backplane slot is not occupied by a board, and if there are boards farther down the daisy-chain,

THEN jumpers MUST be installed at the empty slot to pass through the daisy-chain signal.

OBSERVATION 3.2:

The backplane mechanical specification in Chapter 7 describes a provision for the installation of jumpers at each slot.

RULE 3.3

The ARBITER MUST always be located in slot 1.

3.2.2 *Bus busy line (BBSY*)*

Once a REQUESTER has been granted control of the Data Transfer Bus via the bus grant daisy-chain, it drives $BBSY^*$ low. It then has control of the DTB until it releases $BBSY^*$, allowing the ARBITER to grant the DTB to some other REQUESTER.

3.2.3 *Bus clear line (BCLR*)*

The PRI ARBITER drives $BCLR^*$ low to inform the MASTER, currently in control of the DTB, when a higher priority request is pending. The current MASTER does not have to relinquish the bus within any prescribed time limit. It can continue transferring data until it reaches an appropriate stopping point, and then allows its on-board REQUESTER to release $BBSY^*$.

PERMISSION 3.3:

Although RRS ARBITERS are not required to drive the $BCLR^*$ line they MAY do so.

SUGGESTION 3.1:

IF a RRS ARBITER drives the $BCLR^*$ line low,

THEN design it to do so whenever there is a request pending on any of the non-granted bus request lines.

3.3 Modules fonctionnels

Le sous-système d'arbitrage comprend plusieurs modules:

- un ARBITRE;
- un ou plusieurs DEMANDEURS.

Les figures 3-3 et 3-4, pages 206 et 212, fournissent les schéma-blocs des deux types de modules du bus d'arbitration.

REGLE 3.4:

Les lignes de signaux de sortie indiquées en lignes continues dans les figures 3-3 et 3-4 DOIVENT être commandées par le module, sinon il devrait toujours les commander au niveau haut.

REGLE 3.5:

Les lignes de signaux d'entrée indiquées en lignes continues dans les figures 3-3 et 3-4 DOIVENT être surveillées et on doit y répondre de façon appropriée.

OBSERVATION 3.3:

Les REGLES et les AUTORISATIONS pour commander et surveiller les lignes de signaux indiquées en lignes pointillées dans les figures 3-3 et 3-4 sont énumérées dans les tableaux 3-1 et 3-2.

OBSERVATION 3.4:

Si une ligne de signal de sortie n'est pas commandée, ALORS des adaptateurs d'impédance sur le fond de panier assurent qu'elle est au niveau haut.

OBSERVATION 3.5:

Bien que SYSRESET* et ACFAIL* ne soient pas spécifiés comme faisant partie du bus d'arbitrage, ils sont importants ici parce que des MAITRES associés à des DEMANDEURS répondent à ces lignes de signaux. (SYSRESET* et ACFAIL* sont commandés par le module de contrôle d'alimentation qui est présenté dans le chapitre 5.)

3.3.1 ARBITRE

L'ARBITRE est un module fonctionnel qui décide à quel DEMANDEUR sera alloué le contrôle du DTB quand plusieurs demandes simultanées ont lieu. Plusieurs algorithmes possibles pourront être utilisés pour prendre cette décision. Trois types d'ARBITRES sont décrits dans cette norme: ARBITRE prioritaire (PRI), ARBITRE à priorité tournante (RRS), ARBITRE à niveau unique (SGL).

Un ARBITRE répond à une demande de bus lui parvenant et accorde le DTB au DEMANDEUR approprié avec l'une des lignes du bus d'allocation.

Quand un ARBITRE détecte BBSY* au niveau haut, et après avoir détecté une ou plusieurs demandes du bus, il génère une allocation du bus, correspondant à la demande de bus de priorité la plus haute.

Lorsque le DEMANDEUR reçoit l'allocation du bus, il commande BBSY* au niveau bas et signale à son MAITRE sur sa carte ou au CONTROLEUR d'INTERRUPTION qu'il a alloué le DTB. Après que son MAITRE sur sa carte ou le CONTROLEUR d'INTERRUPTION eut terminé d'utiliser le DTB, le DEMANDEUR libère BBSY*. Le front montant résultant de BBSY* autorise l'ARBITRE à fournir une autre allocation de bus, établie sur les niveaux des lignes de demandes de bus à cet instant.

3.3 Functional modules

The arbitration subsystem is composed of several modules:

- one ARBITER;
- one or more REQUESTERS.

Figures 3-3 and 3-4, pages 207 and 213, provide block diagrams for the two types of arbitration bus modules.

RULE 3.4:

Output signal lines shown with solid lines in Figures 3-3 and 3-4 MUST be driven by the module, unless it would always drive them high.

RULE 3.5:

Input signal lines shown with solid lines in Figures 3-3 and 3-4 MUST be monitored and responded to in the appropriate fashion.

OBSERVATION 3.3:

RULES and PERMISSIONS for driving and monitoring the signal lines shown with dotted lines in Figures 3-3 and 3-4 are given in Tables 3-1 and 3-2.

OBSERVATION 3.4:

IF an output signal line is not driven,
THEN terminators on the backplane ensure that it is high.

OBSERVATION 3.5:

Although SYSRESET* and ACFAIL* are not specified as part of the arbitration bus, they are important here because MASTERS, which are paired with REQUESTERS, respond to these signal lines. (SYSRESET* and ACFAIL* are driven by the power monitor module which is discussed in Chapter 5.)

3.3.1 ARBITER

The ARBITER is a functional module that decides which REQUESTER should be granted control of the DTB when several request it simultaneously. There are many possible algorithms that could be used to make this decision. Three types of ARBITERS are described in this standard: a prioritized (PRI) ARBITER, a round-robin (RRS) ARBITER, and a single level (SGL) ARBITER.

An ARBITER responds to incoming bus requests and grants the DTB to the appropriate REQUESTER with one of the bus grant lines.

When the ARBITER detects BBSY* high, and after it detects one or more bus requests, it issues a bus grant, corresponding to the highest priority bus request.

When the REQUESTER receives the bus grant, it drives BBSY* low and signals to its on-board MASTER or INTERRUPT HANDLER that it has been granted the DTB. After its on-board MASTER or INTERRUPT HANDLER finishes using the DTB, the REQUESTER releases BBSY*. The resulting rising edge of BBSY* enables the ARBITER to issue another bus grant, based upon the levels of the bus request lines at that time.

En plus de l'arbitrage fourni par l'ARBITRE, un niveau secondaire d'arbitre est réalisé par les chaînes séries d'allocation du bus. Grâce à ces chaînes d'allocations, les DEMANDEURS qui partagent une ligne de demande commune sont hiérarchisés en fonction de leur emplacement. Le DEMANDEUR le plus proche de l'emplacement 1 a la priorité la plus haute.

Les arbitres SGL ne répondent qu'aux demandes du bus sur BR3* et tiennent compte de la chaîne série d'allocation BG3IN*/BG3OUT* pour effectuer l'arbitrage.

L'ARBITRE PRI établit la priorité entre les quatre lignes de demandes du bus de BR0* (la plus basse) à BR3* (la plus haute) et il répond comme il convient par BG3IN*. Un ARBITRE PRI informe aussi un MAITRE, ayant le contrôle du bus, lorsqu'un niveau supérieur de demande est en attente, en commandant BCLR* au niveau bas.

Pour se représenter un ARBITRE à priorité tournante RRS, on peut considérer un commutateur mécanique commandé par un moteur pas à pas. Chaque position de ce commutateur connecte une ligne de demande de bus à la ligne d'allocation correspondante. Quand le bus est occupé, le commutateur est arrêté sur le niveau qui est actuellement utilisé sur le bus. A la libération du bus, le commutateur passe au pas inférieur c'est-à-dire BR(n)* à BR(n-1)*) et teste s'il y a une demande. Il continue cette scrutation jusqu'à ce qu'une demande soit trouvée, et envoie alors une allocation de bus sur la ligne appropriée.

AUTORISATION 3.4:

Un ARBITRE PEUT être conçu avec un limiteur de temps qui déclenche le retrait par l'ARBITRE de l'allocation du bus si BBSY* n'est pas commandé au niveau bas par un DEMANDEUR dans un délai déterminé.

OBSERVATION 3.6:

Le temps utilisé par l'ARBITRE permis par l'AUTORISATION 3.4 nécessite d'être plus long que le plus long délai possible de propagation de la chaîne série d'allocation du bus, plus le temps pris par le DEMANDEUR le plus lent pour générer BBSY*.

REGLE 3.6:

Excepté le cas d'une situation de limitation de temps où aucun DEMANDEUR ne répond, une fois que l'ARBITRE octroie le bus à un DEMANDEUR, il NE DOIT PAS allouer le bus avant que le DEMANDEUR ne génère un front de montée sur la ligne BBSY*. (Le DEMANDEUR génère un front de montée en commandant BBSY* au niveau bas puis en le libérant.)

OBSERVATION 3.7:

SI un ARBITRE utilise un échantillonnage et une mémorisation des lignes de demandes avant le front montant de BBSY*,
ALORS il pourrait allouer le bus à un DEMANDEUR qui a déjà retiré sa demande.

In addition to the arbitration provided by the ARBITER, a secondary level of arbitration is provided by the bus grant daisy-chains. Because of these daisy-chains, REQUESTERS sharing a common request line are prioritized by slot position. The REQUESTER closest to slot 1 has the highest priority.

SGL ARBITERS respond only to bus requests on BR3* and depend on the BG3IN*/BG3OUT* daisy-chain to do the arbitration.

The PRI ARBITER prioritizes the four bus request lines, from BRO* (the lowest) to BR3* (the highest), and responds with BGOIN* through BG3IN*, as appropriate. A PRI ARBITER also informs any MASTER, currently in control of the bus, when a higher level request is pending by driving BCLR* low.

To visualize an RRS ARBITER, consider a mechanical switch being driven by a stepping motor. Each position on the switch connects a bus request line to its corresponding bus grant line. When the bus is busy, the switch is stopped on the current level. Upon release of the bus, the switch steps one position lower (i.e. from BR(n)* to BR(n-1)*) and tests for a request. It continues this scanning operation until a request is found, sending a bus grant over the appropriate line.

PERMISSION 3.4:

An ARBITER MAY be designed with a built-in time-out feature that causes it to withdraw a bus grant if BBSY* is not driven low by a REQUESTER within a prescribed time.

OBSERVATION 3.6:

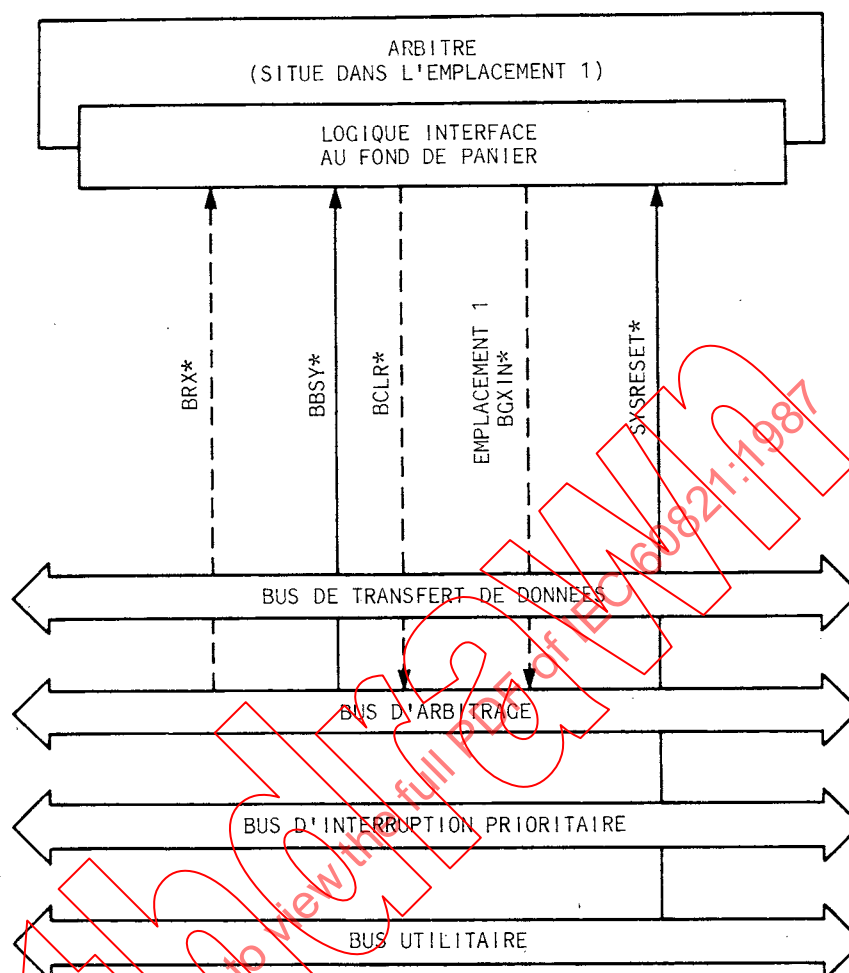
The time used by the ARBITER allowed by PERMISSION 3.4 needs to be longer than the longest possible bus grant daisy-chain propagation delay time, plus the time the slowest REQUESTER takes to generate BBSY*.

RULE 3.6:

Except for a time-out situation where no REQUESTER responds, once the ARBITER grants the bus to a REQUESTER it MUST NOT generate a new Bus Grant before that REQUESTER generates a rising edge on the BBSY* line. (The REQUESTER generates a rising edge by driving BBSY* low and then releasing it.)

OBSERVATION 3.7:

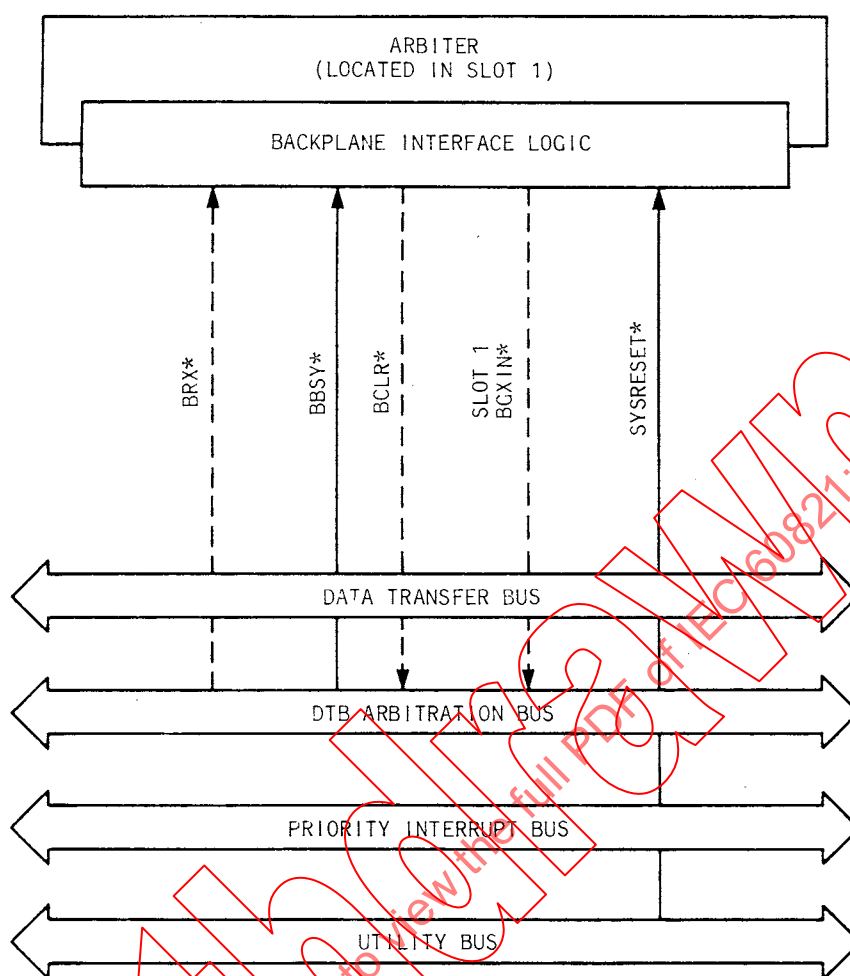
IF an ARBITER uses a "snapshot" of the request lines taken prior to the rising edge of BBSY*,
THEN it might grant the bus to a REQUESTER that has since removed its request.



Note:

Les REGLES et OBSERVATIONS pour commander et surveiller les lignes pointillées sont données dans le tableau 3-1.

FIG. 3-3. - Schéma-bloc: ARBITRE.



Note:

The RULES and PERMISSIONS for monitoring and driving the dotted lines are given in Table 3-1.

FIG. 3-3. - Block diagram: ARBITER.

Tableau 3-1

REGLES et OBSERVATIONS qui spécifient l'utilisation des lignes pointillées par les différents types d'ARBITRES définis dans cette norme

Type d'ARBITRE	Utilisation des lignes pointillées
SGL	DOIT commander BG3IN* de l'emplacement 1. DOIT surveiller BR3*. DOIT assurer que BG0IN*-BG2IN* de l'emplacement 1 sont au niveau haut. PEUT ou NON commander BCLR*, ou BG0IN*-BG2IN* de l'emplacement 1. PEUT ou NON surveiller BR0*-BR2*
RRS	DOIT commander BG0IN*-BG3IN* de l'emplacement 1. DOIT surveiller BR0*-BR3*. PEUT ou NON commander BCLR*.
PRI	DOIT commander BG0IN*-BG3IN* de l'emplacement 1 et BCLR*. DOIT surveiller BR0*-BR3*.

3.3.2 DEMANDEUR

Chaque DEMANDEUR dans le système:

Surveille le signal DISPOSITIF DEMANDE LE BUS du MAITRE sur sa carte ou du CONTROLEUR d'INTERRUPTION et génère une demande DTB en cas de besoin.

S'il détecte un niveau bas sur sa ligne BGxIN* et que son MAITRE sur sa carte ou le CONTROLEUR d'INTERRUPTION n'a pas besoin du DTB, il transmet ce niveau bas sur BGxOUT*.

S'il détecte un niveau bas sur la ligne BGxIN* et que son MAITRE sur sa carte ou le CONTROLEUR d'INTERRUPTION a besoin du DTB, il génère un signal sur sa carte DISPOSITIF RECOIT LE BUS qui indique que le DTB est disponible, et commande le signal BBSY* au niveau bas.

Deux types de DEMANDEURS sont décrits dans cette norme: le DEMANDEUR (RWD) qui libère le bus lorsqu'il a terminé et le DEMANDEUR (ROR) qui libère le bus sur demande.

Le DEMANDEUR RWD libère BBSY* lorsque le MAITRE ou le CONTROLEUR d'INTERRUPTION désactive le signal sur sa carte DISPOSITIF DEMANDE LE BUS.

Le DEMANDEUR ROR ne libère pas BBSY* lorsque le signal sur sa carte DISPOSITIF DEMANDE LE BUS devient inactif tant qu'un autre DEMANDEUR sur le bus ne commande pas une des lignes de demande du bus au niveau bas. Il surveille les quatre lignes de demande du bus et libère BBSY* seulement si une autre demande de bus est en attente. Les DEMANDEURS ROR réduisent le nombre d'arbitrages déclenchés par un MAITRE qui prend une grande part du trafic sur le bus. Voir la figure 3-4, page 212.

Table 3-1

*RULES and PERMISSIONS that specify the use of the dotted lines
by the various types of ARBITERS defined by this standard*

Type of ARBITER	Use of dotted line
SGL	MUST drive slot 1 BG3IN*. MUST monitor BR3*. MUST ensure slot 1 BG0IN*-BG2IN* are high. MAY or MAY NOT drive BCLR*, or slot 1 BG0IN*-BG2IN*. MAY or MAY NOT monitor BR0*-BR2
RRS	MUST drive slot 1 BG0IN-BG3IN*. MUST monitor BR0*-BR3*. MAY or MAY NOT drive BCLR*
PRI	MUST drive slot 1 BG0IN*-BG3IN*, and BCLR*. MUST monitor BR0*-BR3*

3.3.2 REQUESTER

Each REQUESTER in the system:

Monitors the DEVICE WANTS BUS signal from its on-board MASTER or INTERRUPT HANDLER and generates a bus request when the DTB is needed.

If it detects a low level on its BGxIN* line, and its on-board MASTER or INTERRUPT HANDLER does not need the DTB, then it passes on that low level to its BGxOUT*.

If it detects a low level on its BGxIN* line, and its on-board MASTER or INTERRUPT HANDLER needs the DTB, it generates an on-board DEVICE GRANTED BUS signal to indicate the DTB is available, and drives the BBSY* signal low.

Two types of REQUESTERS are described in this standard: a release when done (RWD) REQUESTER and a release on request (ROR) REQUESTER.

The RWD REQUESTER releases BBSY* when the MASTER or INTERRUPT HANDLER drives the on-board DEVICE WANTS BUS signal false.

The ROR REQUESTER does not release BBSY* when its on-board DEVICE WANTS BUS signal goes false unless some other REQUESTER on the bus drives one of the bus request lines low. It monitors the four bus request lines and releases BBSY* only if another bus request is pending. ROR REQUESTERS reduce the number of arbitrations initiated by a MASTER which is generating a large percentage of the bus traffic. See Figure 3-4, page 213.

En considérant que l'entrée DISPOSITIF DEMANDE LE BUS du DEMANDEUR est vraie lorsqu'il reçoit l'allocation du bus, ce dernier exécute trois opérations.

- a) Il commande BBSY* au niveau bas.
- b) Il libère son niveau bas sur BRx.
- c) Il commande le signal sur sa carte DISPOSITIF RECOIT LE BUS vrai permettant au MAITRE ou au CONTROLEUR d'INTERRUPTION de lancer les transferts sur le bus.

Ces opérations peuvent avoir lieu dans n'importe quel ordre. Il est même possible, bien que dénué de sens, que le MAITRE ou le CONTROLEUR d'INTERRUPTION n'utilise pas le bus en réponse à une allocation.

Les REGLES suivantes s'appliquent, cependant:

REGLE 3.7:

Le DEMANDEUR DOIT commander BBSY* au niveau bas pendant au moins 90 ns.

REGLE 3.8:

Le DEMANDEUR DOIT libérer DEMANDE DU BUS au niveau haut.

REGLE 3.9:

Le DEMANDEUR DOIT maintenir BBSY* au niveau bas pendant au moins 30 ns après avoir libéré sa demande du bus.

OBSERVATION 3.8:

Le délai de 30 ns entre le front montant de la demande du bus et le front montant de BBSY* garantit que l'ARBITRE n'interprète pas par erreur l'ancienne demande du bus comme une nouvelle et ne fasse pas une autre allocation.

REGLE 3.10:

Le DEMANDEUR DOIT maintenir BBSY* au niveau bas jusqu'à ce que son allocation du bus passe au niveau haut.

OBSERVATION 3.9:

La REGLE 3.10 assure que la transition BBSY* au niveau bas a été vue par l'ARBITRE et que toutes les lignes de la chaîne série d'allocation du bus sont ramenées au niveau haut, en préparation du prochain arbitrage.

AUTORISATION 3.5:

SI un DEMANDEUR a une demande de bus en attente et s'il voit un autre DEMANDEUR commander BBSY* à l'état bas,
ALORS il PEUT retirer sa demande en libérant la ligne de demande du bus.

REGLE 3.11:

SI un DEMANDEUR retire une demande de bus sans avoir eu d'abord le bus,
ALORS il DOIT attendre pour le faire jusqu'à ce que BBSY* passe au niveau bas et il doit le faire au plus tard 50 ns après que BBSY* eut passé au niveau bas.

Assuming that the REQUESTER'S DEVICE WANTS BUS input is true, when it receives a bus grant it does three things:

- a) It drives BBSY* to low.
- b) It releases its low on BRx.
- c) It drives the DEVICE GRANTED BUS on-board signal true, allowing the MASTER or INTERRUPT HANDLER to initiate bus transfers.

These events might occur in any order. It is even possible, although meaningless, that the MASTER or INTERRUPT HANDLER might not use the bus in response to this particular grant.

However, the following RULES apply:

RULE 3.7:

The REQUESTER MUST drive BBSY* to low for at least 90 ns.

RULE 3.8:

The REQUESTER MUST release BUS REQUEST to high.

RULE 3.9:

The REQUESTER MUST maintain BBSY* low for at least 30 ns after it releases its bus request.

OBSERVATION 3.8:

The 30 ns delay between the bus request's rising edge and the BBSY* rising edge ensures that the ARBITER does not mistakenly interpret the old bus request as a new one and issue another grant.

RULE 3.10:

The REQUESTER MUST hold BBSY* low until its bus grant goes high.

OBSERVATION 3.9:

RULE 3.10 ensures that the BBSY* transition to low has been seen by the ARBITER and that all segments of the bus grant daisy-chain have returned to high, in preparation for the next arbitration.

PERMISSION 3.5:

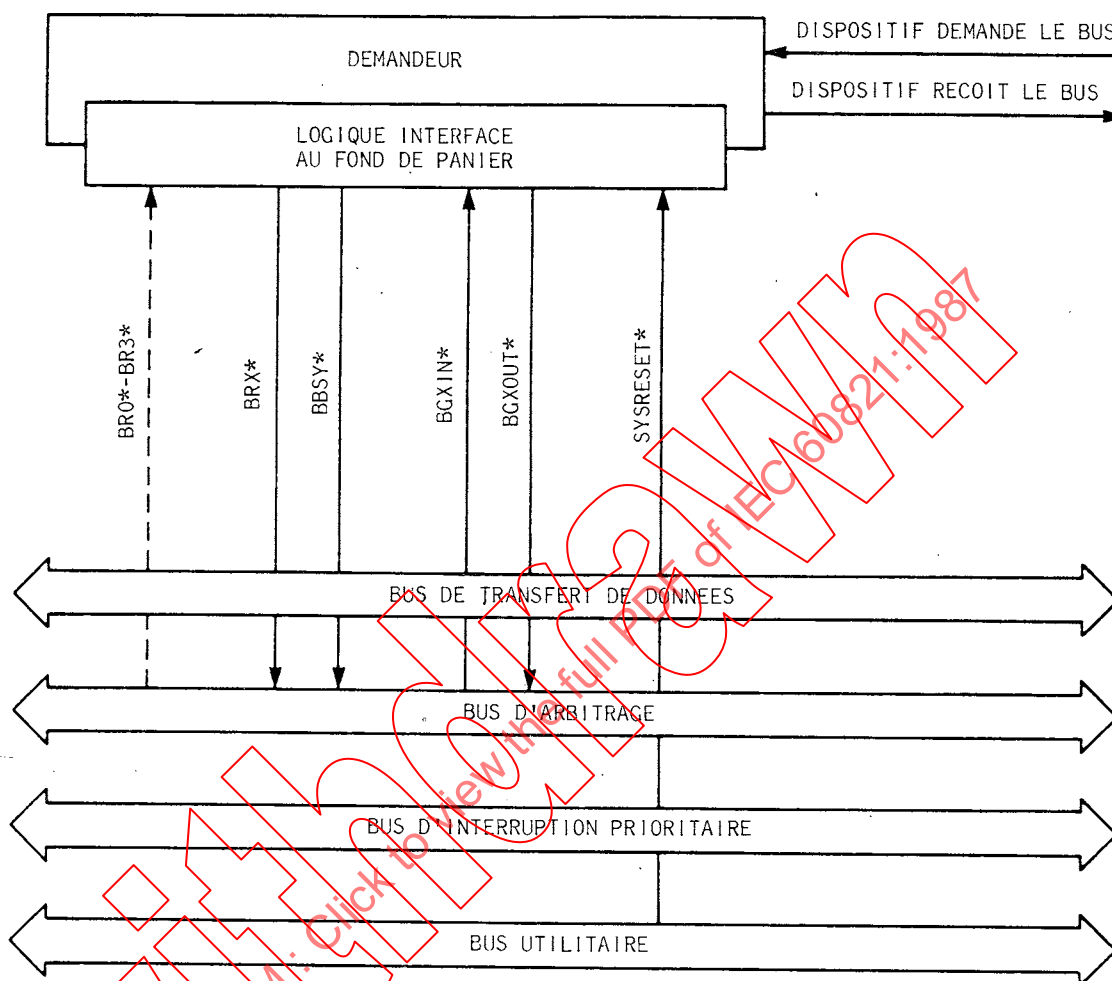
IF a REQUESTER has a bus request pending and, if it sees some other REQUESTER drive BBSY* low,
THEN it MAY withdraw its request by releasing the bus request line.

RULE 3.11:

IF a REQUESTER withdraws a bus request without having first been granted the bus,
THEN it MUST wait to do so until BBSY* goes low and it MUST do so within 50 ns after BBSY* goes low.

SUGGESTION 3.2:

Concevoir les DEMANDEURS pour qu'ils transmettent l'allocation à travers la chaîne série d'allocation du bus aussi rapidement que possible après réception du signal d'allocation du bus. Cela améliorera la performance du système.



Note:

Les REGLES et les AUTORISATIONS concernant la surveillance des lignes pointillées sont données dans le tableau 3-2.

FIG. 3-4. - Schéma-bloc: DEMANDEUR.

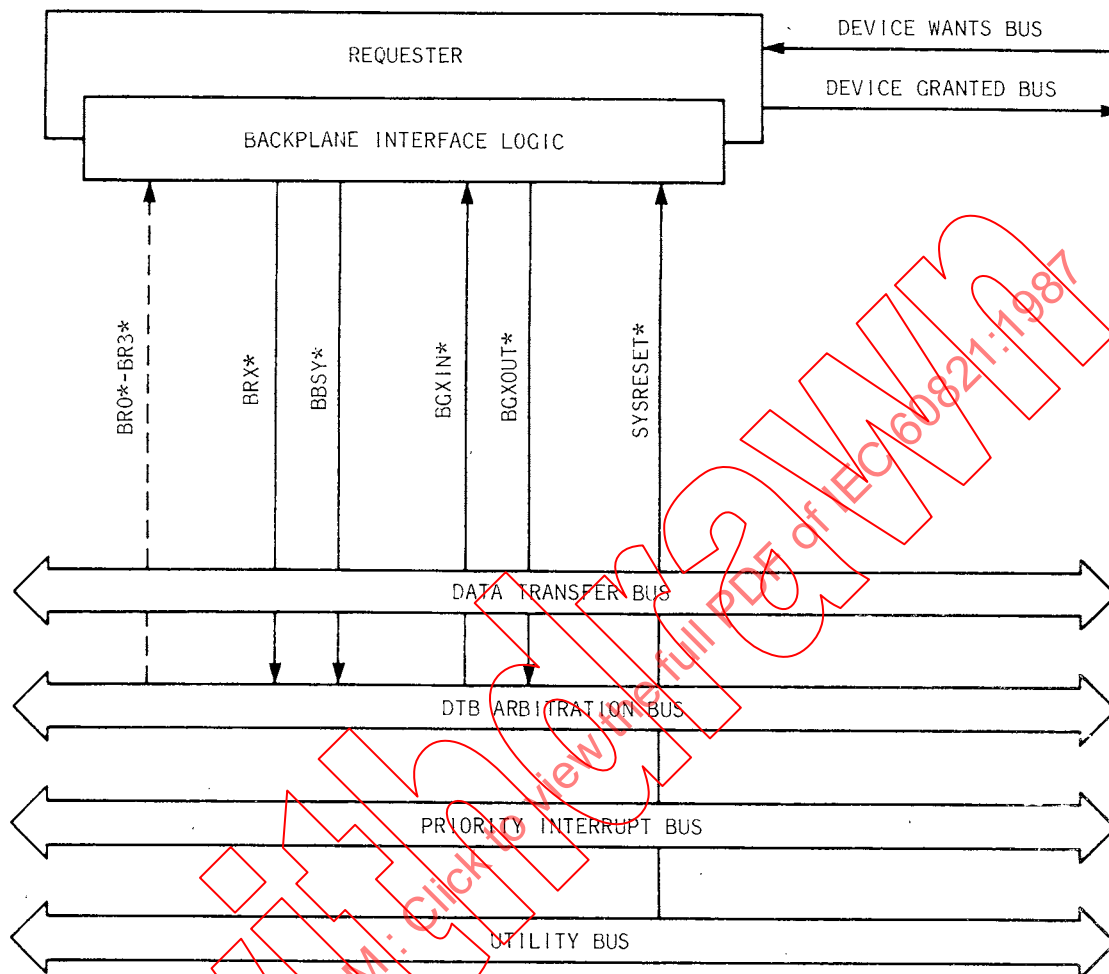
Tableau 3-2

REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de DEMANDEURS définis dans cette norme

Type de DEMANDEUR	Utilisation de la ligne pointillée
RWD	PEUT ou NON surveiller BR0*-BR3*
ROR	DOIT surveiller BR0*-BR3*

SUGGESTION 3.2:

Design REQUESTERS so that they pass on the bus grant daisy-chain as fast as possible after receipt of a bus grant. This will improve system performance.

**Note:**

The RULES and PERMISSIONS for monitoring the dotted lines are given in Table 3-2.

FIG. 3-4. - Block diagram: REQUESTER.

Table 3-2

RULES and PERMISSIONS that specify the use of the dotted lines by the various types of REQUESTERS defined by this standard

Type of REQUESTER	Use of dotted line
RWD	MAY or MAY NOT monitor BR0*-BR3*
ROR	MUST monitor BR0*-BR3*

3.3.3 MAITRE du bus de transfert de données

3.3.3.1 Libération du DTB

Le protocole du bus d'arbitrage détermine comment et quand le DTB est accordé aux différents MAITRES et aux CONTROLEURS d'INTERRUPTION du système. Il ne contrôle pas, cependant, le moment de la libération du DTB par le MAITRE ou le CONTROLEUR d'INTERRUPTION.

Les MAITRES et les CONTROLEURS d'INTERRUPTION utilisent plusieurs critères pour décider du moment de la libération du DTB. Les CONTROLEURS d'INTERRUPTION restituent le bus après leur cycle de reconnaissance d'interruption. Les MAITRES restituent le bus lorsqu'ils terminent leurs transferts de données.

Certains MAITRES surveillent aussi les signaux ACFAIL* et BCLR* du BUS CEI 821. Ces deux signaux informent le MAITRE que le DTB est demandé pour une activité de priorité supérieure. Dans le cas de BCLR*, le délai avant libération du bus est défini à la conception du MAITRE. Par exemple, un MAITRE sur une carte contrôleur de disque pourrait ne pas être en état de libérer le bus pendant le transfert d'un secteur de disque sans perte de données, il devrait garder ainsi le bus jusqu'à ce que le transfert du secteur soit terminé. ACFAIL* informe le MAITRE qu'une perte de courant alternatif a été détectée, et quels que soient les problèmes que rencontrera le MAITRE, en abandonnant le bus, ils sont insignifiants en comparaison des besoins du système total.

RECOMMANDATION 3.1:

Concevoir les MAITRES pour libérer le DTB dans les 200 μ s qui suivent le passage au niveau bas de ACFAIL*, sauf pour participer aux activités consécutives à la défaillance d'alimentation.

OBSERVATION 3.10:

Les 200 μ s spécifiées dans la RECOMMANDATION 3.1 ont pour but de laisser le temps pour accomplir un arrêt correct du système.

Quels que soient les critères utilisés pour décider de la libération du DTB, l'arbitrage a lieu avant qu'un autre MAITRE ou CONTROLEUR d'INTERRUPTION ne commence à l'utiliser. Cet arbitrage a lieu, soit pendant le dernier transfert de données, soit après ce transfert, selon l'attention que le MAITRE ou le CONTROLEUR d'INTERRUPTION accorde à son DEMANDEUR sur sa carte.

AUTORISATION 3.6:

Les MAITRES et les CONTROLEURS d'INTERRUPTION PEUVENT libérer le DTB aussi bien pendant qu'après le dernier transfert de données.

Par exemple, si le MAITRE avertit son DEMANDEUR sur sa carte qu'il n'a plus besoin du bus pendant le dernier transfert de données, le DEMANDEUR libère BBSY* et l'arbitrage a lieu pendant le dernier transfert. Mais si le MAITRE attend que le dernier transfert soit terminé avant de le signaler à son DEMANDEUR sur sa carte, le DTB reste au repos pendant l'arbitrage. (Cela est présenté dans la section 2.5.)

Les chapitres 2 et 4 contiennent des REGLES qui appartiennent à la libération du DTB.

3.3.3 Data Transfer Bus MASTER

3.3.3.1 Release of the DTB

The bus arbitration protocol determines how and when the DTB is granted to the various MASTERS and INTERRUPT HANDLERS in the system. It does not, however, control when MASTERS and INTERRUPT HANDLERS release the DTB.

MASTERS and INTERRUPT HANDLERS use several criteria in deciding when to release the DTB. INTERRUPT HANDLERS give up the bus after their interrupt acknowledge cycle. MASTERS give up the bus when they finish their data transfers.

Some MASTERS also monitor the ACFAIL* and BCLR* IEC 821 BUS signals. Both of these signals inform the MASTER that the DTB is needed for some higher priority activity. In the case of BCLR*, the MASTER'S design determines how long it takes to release the bus. For example, a MASTER on a disk controller board might not be able to relinquish the bus during a disk sector transfer without loss of data, so it might keep the bus until the sector transfer is finished. ACFAIL* informs the MASTER that an a.c. power loss has been detected, and whatever problems the MASTER will face in surrendering the bus are insignificant compared to the needs of the total system.

RECOMMENDATION 3.1:

Design MASTERS to release the DTB within 200 μ s after ACFAIL* goes low, except to participate in the ensuing power failure activities.

OBSERVATION 3.10:

The 200 μ s specified in RECOMMENDATION 3.1 is intended to provide time for an orderly shut-down of the system.

Whatever criteria are used to decide when to release the DTB, arbitration is done before some other MASTER or INTERRUPT HANDLER begins using it. This arbitration takes place either during the last data transfer or after that transfer, depending on how much notice the MASTER or INTERRUPT HANDLER gives to its on-board REQUESTER.

PERMISSION 3.6:

MASTERS and INTERRUPT HANDLERS MAY release the DTB either during or after their last data transfer.

For example, if the MASTER notifies its on-board REQUESTER that it no longer wants the bus during its last data transfer, the REQUESTER releases BBSY* and arbitration takes place during the last transfer. But if the MASTER waits until the last transfer has completed before signalling its on-board REQUESTER, the DTB remains idle while the arbitration is done. (This was illustrated in Section 2.5.)

Chapters 2 and 4 contain RULES that pertain to the release of the DTB.

SUGGESTION 3.3:

Concevoir les cartes MAITRES avec transfert par bloc de telle manière qu'elles signalent à leur DEMANDEUR de relâcher BBSY* pendant le dernier signal de validation de donnée du transfert par bloc. S'il est relâché au début du transfert par bloc, des demandes de bus à haute priorité lancées pendant le transfert par bloc pourraient ne pas être prises en compte par l'ARBITRE avant le cycle d'arbitrage suivant.

3.3.3.2 Acquisition du DTB

Certaines règles sont imposées pour s'assurer qu'aucune ligne du DTB ne sera commandée vers deux états opposés par deux MAITRES ou CONTROLEURS D'INTERRUPTION prenant le contrôle du DTB.

REGLE 3.12:

Lorsqu'un MAITRE ou un CONTROLEUR D'INTERRUPTION a reçu le contrôle du DTB par son DEMANDEUR sur sa carte, il DOIT attendre de détecter AS* au niveau haut avant de commander ses émetteurs du DTB.

OBSERVATION 3.11:

SI le MAITRE ou le précédent CONTROLEUR D'INTERRUPTION a libéré le bus PENDANT son dernier transfert de données,
ALORS la REGLE 3.12 assure que le transfert de données sera terminé avant qu'un nouveau MAITRE ou CONTROLEUR D'INTERRUPTION ne commence à utiliser le DTB. (Si le MAITRE ou le précédent CONTROLEUR D'INTERRUPTION attendait que le transfert de données soit terminé avant de libérer le bus, AS* serait déjà au niveau haut.)

3.3.3.3 Autres informations**RECOMMANDATION 3.2:**

Pour permettre un traitement rapide aux demandes d'interruptions et pour l'utilisation optimale du DTB, concevoir des MAITRES de telle manière qu'ils libèrent le DTB dès que possible après qu'ils eurent détecté BCLR* au niveau bas.

AUTORISATION 3.7:

Un MAITRE ou un CONTROLEUR D'INTERRUPTION PEUT avoir plus d'un DEMANDEUR, chacun générant des demandes de bus sur une ligne différente.

OBSERVATION 3.12:

Lorsqu'un MAITRE ou un CONTROLEUR D'INTERRUPTION a deux ou plusieurs DEMANDEURS, il peut exécuter des transferts de données de priorité supérieure en utilisant un DEMANDEUR et des transferts de priorité inférieure en utilisant un autre DEMANDEUR.

3.4 Fonctionnement typique**3.4.1 Arbitrage de deux niveaux différents de demande du bus**

Les figures 3-5 et 3-6, pages 220 à 224, illustrent la suite d'opérations qui a lieu lorsque deux DEMANDEURS envoient des demandes du bus simultanées à l'ARBITRE PRI sur des lignes de demande de bus différentes. Au début, chacun des DEMANDEURS commande sa ligne de demande de bus respective au niveau bas (le DEMANDEUR A commande BR1* et le DEMANDEUR B commande BR2*). L'ARBITRE détecte BR1* et BR2* au niveau bas simultanément et il commande BG2IN* de son propre

SUGGESTION 3.3:

Design block transfer MASTER boards so that they signal their REQUESTER to release BBSY* during the last data strobe of the block transfer. If it is released at the beginning of the block transfer, high priority bus requests initiated during the block transfer might not be taken into account by the ARBITER until the next arbitration cycle.

3.3.3.2 Acquisition of the DTB

To ensure that no DTB line is ever driven to opposite states by two MASTERS or INTERRUPT HANDLERS, when these modules take control of the DTB they are constrained by certain rules.

RULE 3.12:

When a MASTER or INTERRUPT HANDLER is given control of the DTB by its on-board REQUESTER, it MUST wait until it detects AS* high before turning on its DTB drivers.

OBSERVATION 3.11:

IF the previous MASTER or INTERRUPT HANDLER releases the bus DURING its last data transfer,
 THEN RULE 3.12 ensures that the data transfer will be finished before the new MASTER or INTERRUPT HANDLER starts using the DTB. (If the previous MASTER or INTERRUPT HANDLER waited until the data transfer was finished before releasing the bus, AS* will already be high).

3.3.3.3 Other information**RECOMMENDATION 3.2:**

To allow for prompt servicing of interrupt requests and for optimum use of the DTB, design MASTERS so that they release the DTB as soon as possible after they detect BCLR* low.

PERMISSION 3.7:

A MASTER or INTERRUPT HANDLER MAY have more than one REQUESTER, where each REQUESTER generates bus requests on a different bus request line.

OBSERVATION 3.12:

Where a MASTER or INTERRUPT HANDLER has two or more REQUESTERS, it can do high priority data transfers using one REQUESTER and low priority transfers using another REQUESTER.

3.4 Typical operation**3.4.1 Arbitration of two different levels of bus request**

Figures 3-5 and 3-6, pages 221 to 225, illustrate the sequence of events that takes place when two REQUESTERS send simultaneous bus requests to a PRI ARBITER on different bus request lines. When the sequence begins, each of the REQUESTERS drives its respective bus request line low (REQUESTER A drives BR1* and REQUESTER B drives BR2*). The ARBITER detects BR1* and BR2* low simultaneously, and it drives BG2IN* low to its own slot (slot 1). That BG2IN* signal is monitored by REQUESTER B (also in slot 1). When REQUESTER B

emplacement (emplacement 1) au niveau bas. Ce signal BG2IN* est surveillé par le DEMANDEUR B (situé aussi dans l'emplacement 1). Lorsque le DEMANDEUR B détecte BG2IN* au niveau bas, il répond en commandant BBSY* au niveau bas. Le DEMANDEUR B libère alors la ligne BR2* et informe son propre MAITRE (MAITRE B) que le DTB est disponible.

Lorsque BBSY* passe au niveau bas, l'ARBITRE commande BG2IN* de l'emplacement 1 au niveau haut.

Lorsque le MAITRE B termine son ou ses transferts de données et signale ce fait en commandant DISPOSITIF DEMANDE LE BUS à l'état faux, le DEMANDEUR B libère BBSY*, pourvu que BG2IN* ait été reçu au niveau haut et que 30 ns se soient écoulées depuis la libération de BR2*.

L'ARBITRE interprète la libération de BBSY* comme un signal pour arbitrer n'importe quelle demande de bus en cours. Puisque BR1* est la seule demande de bus commandée au niveau bas, l'ARBITRE accorde le DTB au DEMANDEUR A en commandant BG1IN* au niveau bas. Le DEMANDEUR A répond en activant BBSY* au niveau bas. Lorsque le MAITRE A termine son ou ses transferts de données et signale ce fait en commandant DISPOSITIF DEMANDE LE BUS à l'état faux, le DEMANDEUR A libère BBSY*, pourvu que BG1IN* ait été reçu au niveau haut et que 30 ns se soient écoulées depuis la libération de BR1*.

Dans cet exemple, puisqu'aucune ligne de demande de bus n'est commandée au niveau bas lorsque le DEMANDEUR A libère BBSY*, l'ARBITRE attend jusqu'à ce qu'il détecte une demande de bus.

OBSERVATION 3.13:

La description illustrée dans les figures 3-5 et 3-6, pages 220 à 224, serait valable pour les ARBITRES PRI et RRS, à moins de considérer un ARBITRE RRS où la dernière demande active serait le niveau BR2*. Dans ce cas, l'ARBITRE traiterait la demande de BR1* d'abord et la demande de BR2* après.

OBSERVATION 3.14:

BBSY* et l'allocation du bus sont interverrouillés comme indiqué sur la figure 3-6.

- 1) L'ARBITRE ne commande pas l'allocation du bus au niveau haut tant qu'il ne détecte pas BBSY* au niveau bas.
- 2) Le DEMANDEUR ne libère BBSY* au niveau haut que lorsqu'il détecte l'allocation du bus au niveau haut.
- 3) L'ARBITRE ne commande la prochaine allocation du bus que lorsqu'il détecte BBSY* au niveau haut.
- 4) Le DEMANDEUR ne commande BBSY* au niveau bas que lorsqu'il détecte l'allocation du bus au niveau bas.

detects BG2IN* low, it responds by driving BBSY* low. REQUESTER B then releases the BR2* line and informs its own MASTER (MASTER B) that the DTB is available.

When BBSY* goes low, the ARBITER drives BG2IN* of slot 1 high.

When MASTER B completes its data transfer(s), and signals that fact by driving DEVICE WANTS BUS false, REQUESTER B releases BBSY*, provided that its BG2IN* has been received high and 30 ns also elapsed since it released BR2*.

The ARBITER interprets the release of BBSY* as a signal to arbitrate any current bus requests. Since BR1* is the only bus request being driven low, the ARBITER grants the DTB to REQUESTER A by driving BG1IN* low. REQUESTER A responds by driving BBSY* low. When MASTER A completes its data transfer(s), and signals that fact by driving DEVICE WANTS BUS false, REQUESTER A releases BBSY*, provided that its BG1IN* has been received high and 30 ns have elapsed since it released BR1*.

In this example, since no bus request lines are low when REQUESTER A releases BBSY*, the ARBITER waits until it detects a bus request.

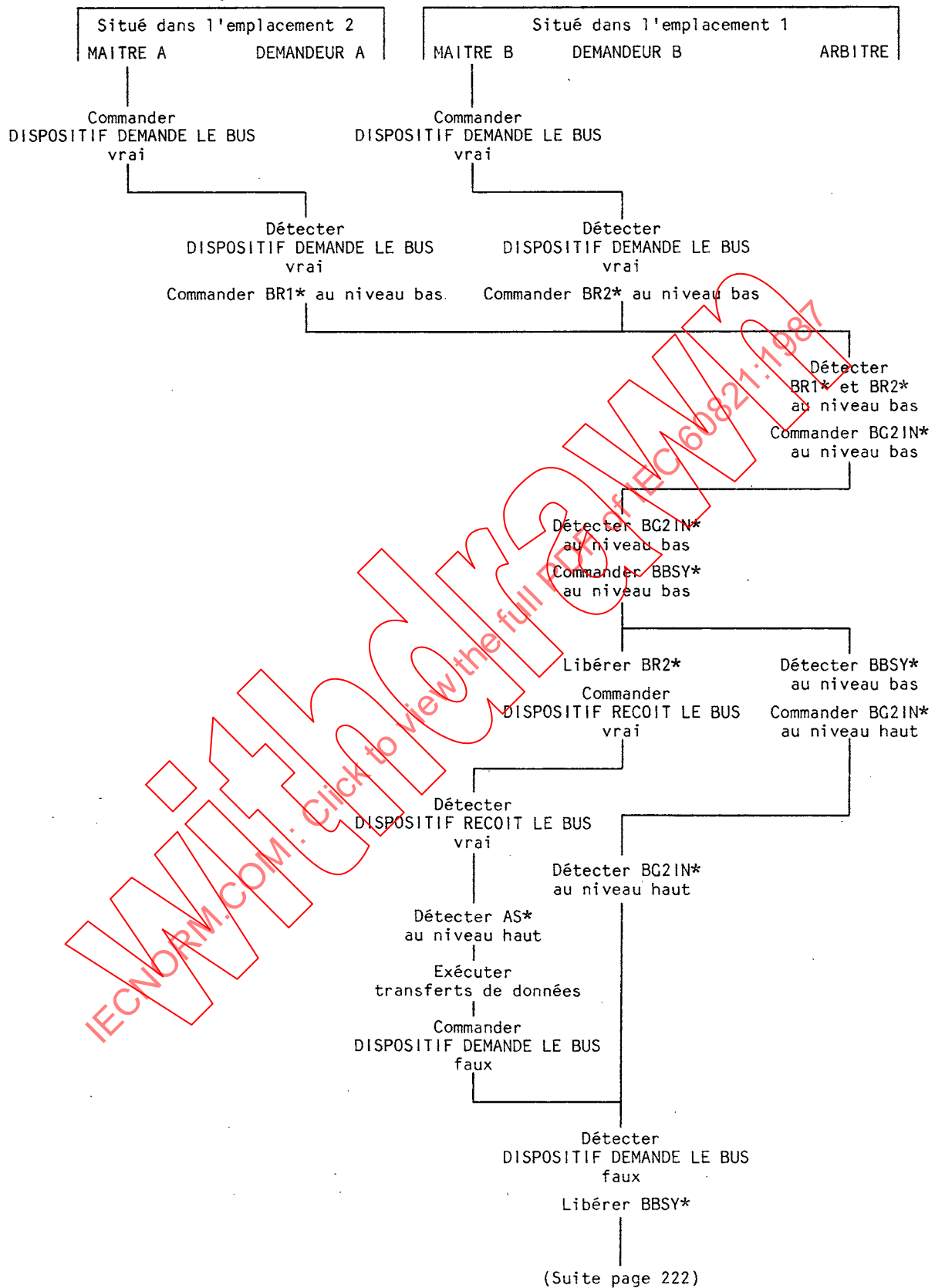
OBSERVATION 3.13:

The description illustrated in Figures 3-5 and 3-6, pages 221 to 225, would hold for both PRI and RRS ARBITERS, unless we consider an RRS ARBITER where the last active request was level BR2*. In this case, the ARBITER would process the BR1* request first and then proceed to the BR2* request.

OBSERVATION 3.14:

BBSY* and the bus grants are fully interlocked as shown in Figure 3-6.

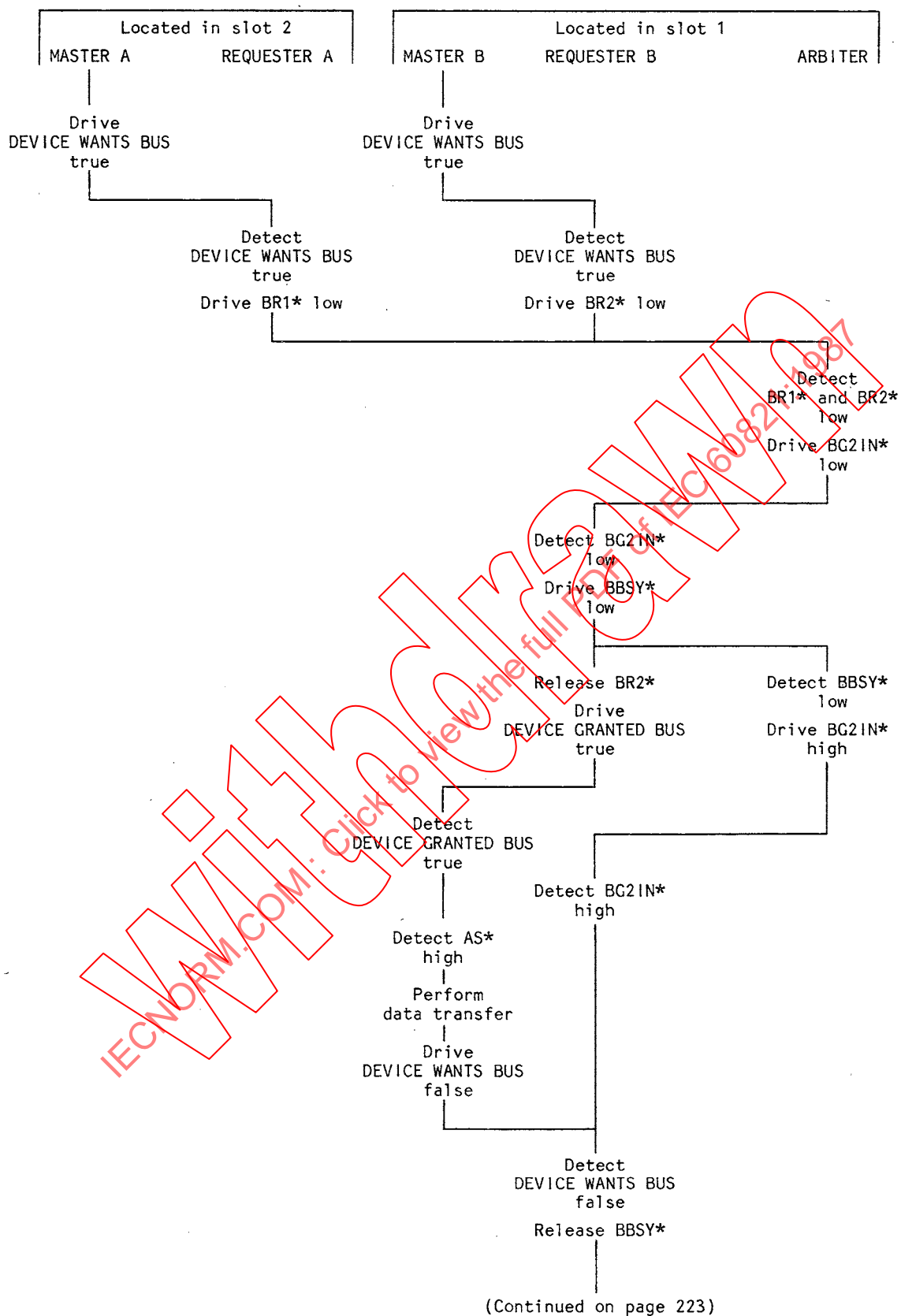
- 1) The ARBITER does not drive the bus grant high until it detects BBSY* low.
- 2) The REQUESTER does not release BBSY* to high until it detects the bus grant high.
- 3) The ARBITER does not drive the next bus grant low until it detects BBSY* high.
- 4) The next REQUESTER does not drive BBSY* low until it detects the bus grant low.



Note:

Le DISPOSITIF DEMANDE LE BUS et le DISPOSITIF REÇOIT LE BUS sont des signaux internes à la carte entre le MAITRE et son DEMANDEUR (voir figure 3-4, page 212).

FIG. 3-5. - Organigramme de l'arbitrage: deux DEMANDEURS, deux niveaux de demande



Note:
DEVICE WANTS BUS and DEVICE GRANTED BUS are on-board signals between the MASTER and its REQUESTER (see Figure 3-4, page 213).

FIG. 3-5. - Arbitration flow diagram: two REQUESTERS, two request levels.

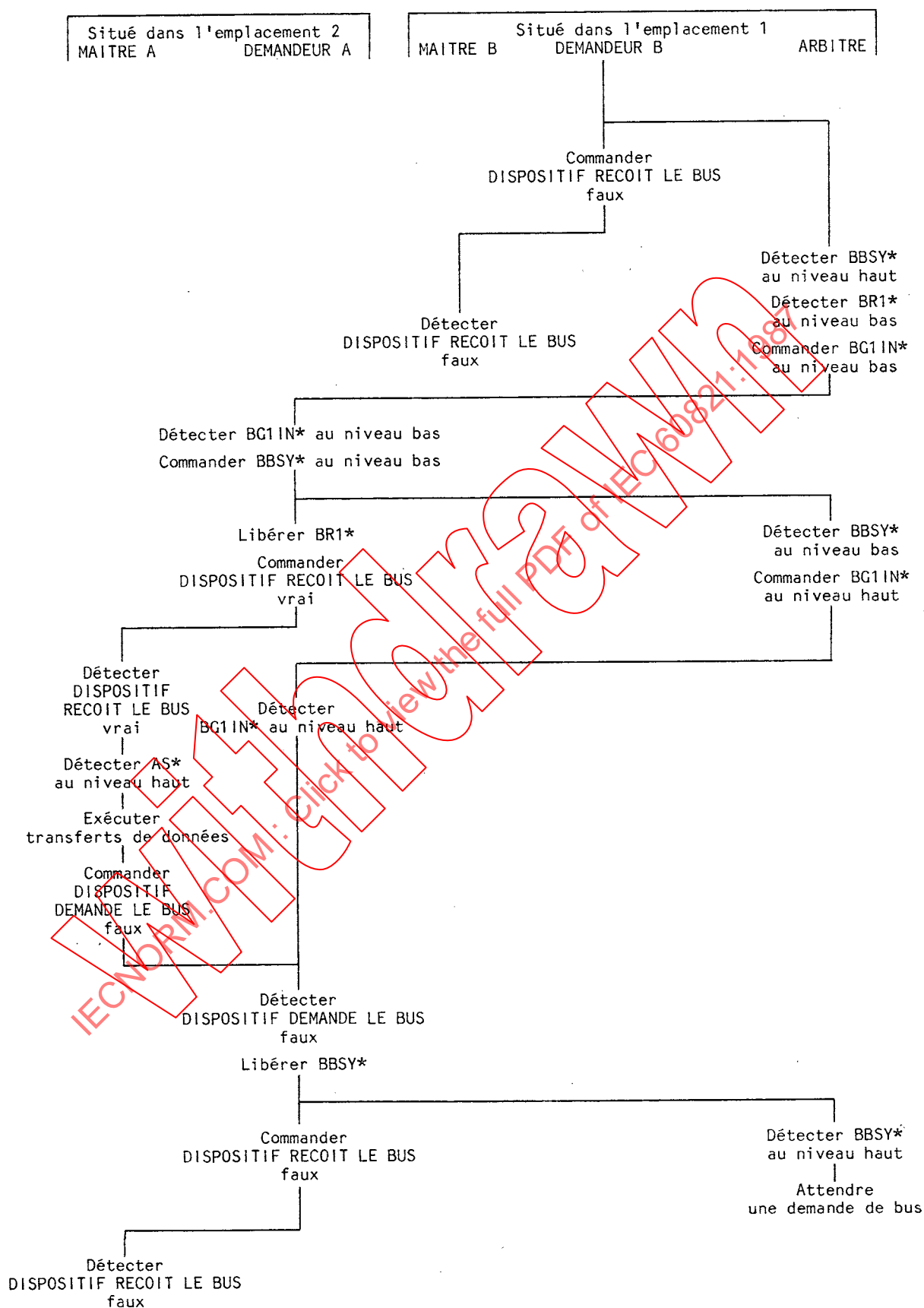


FIGURE 3-5 (fin).

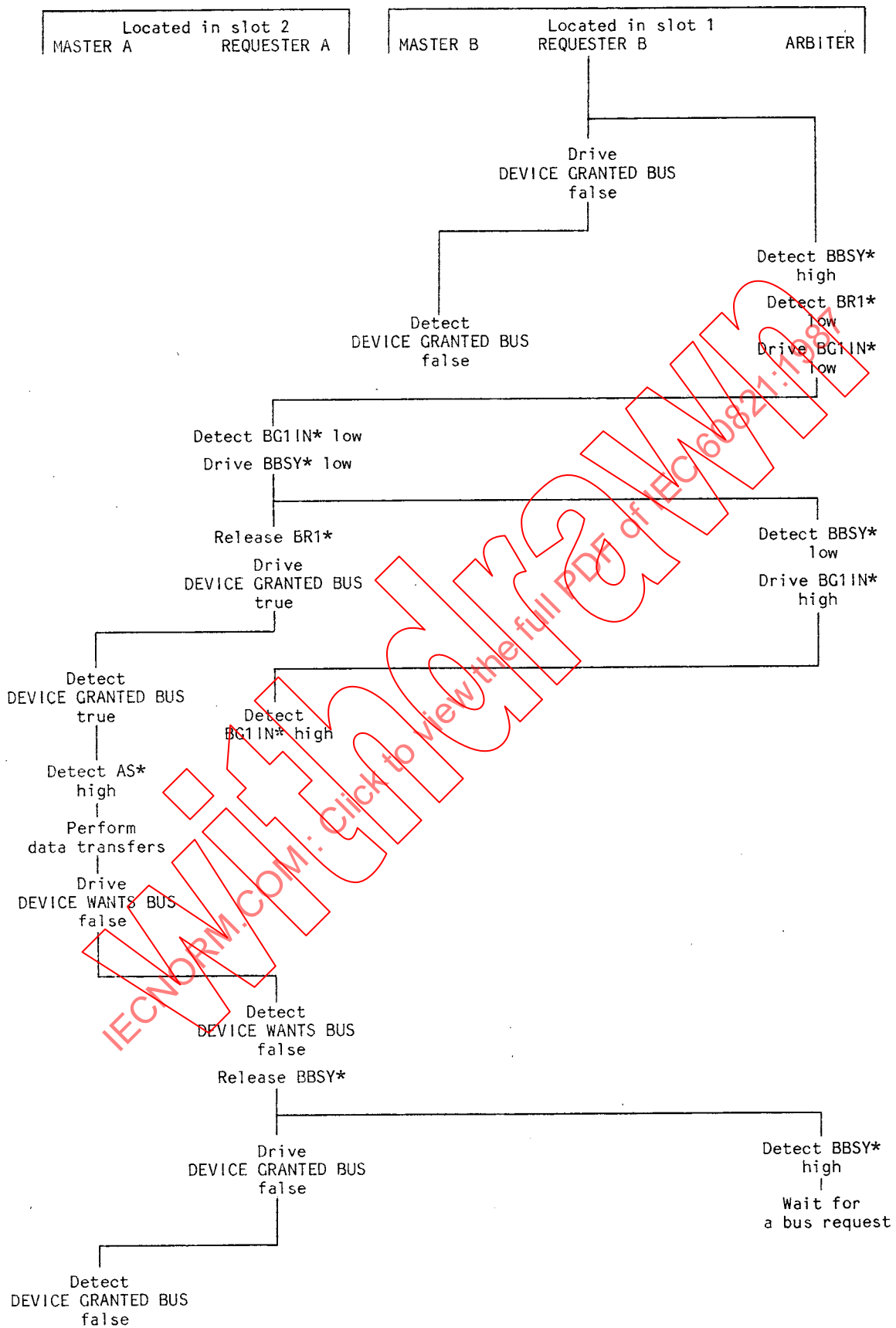
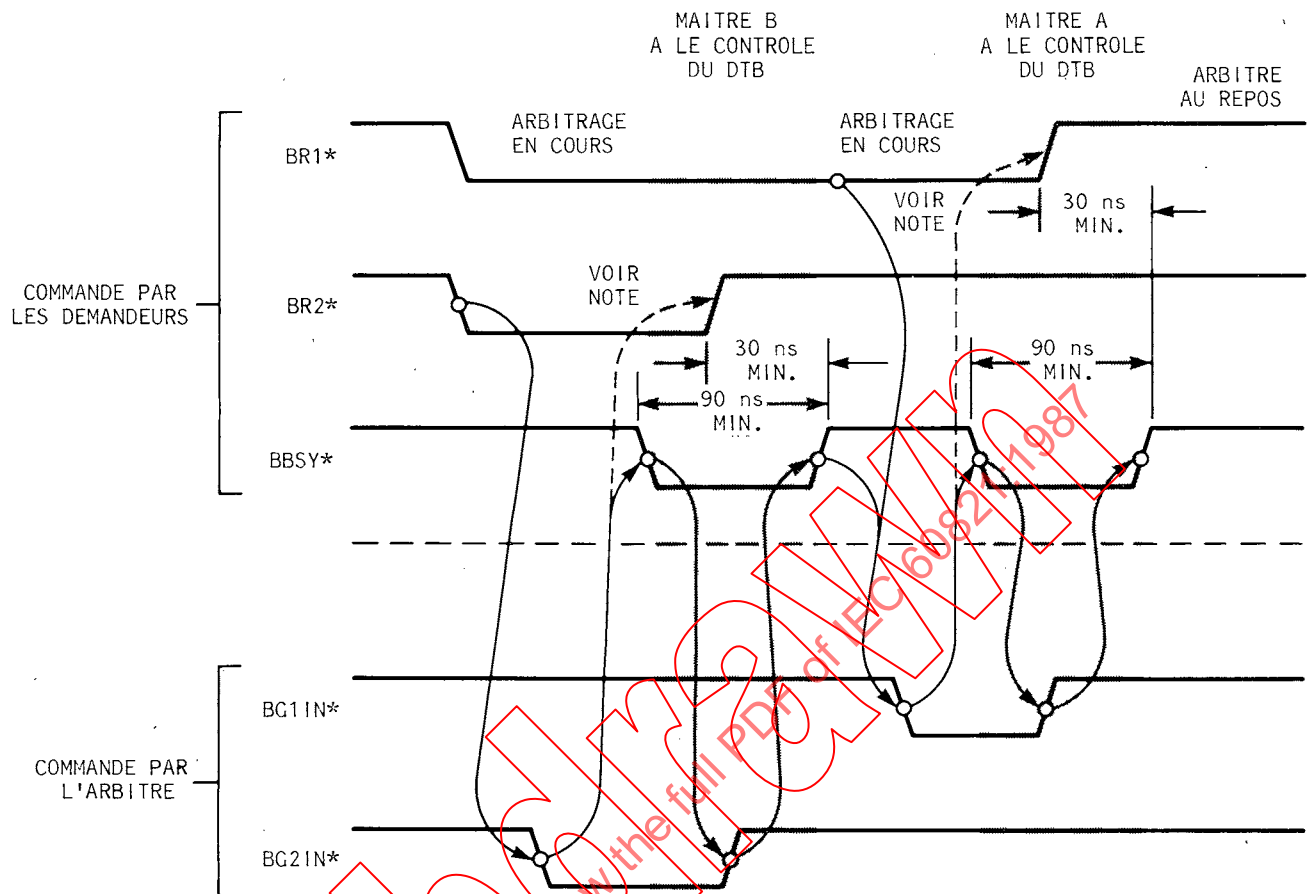


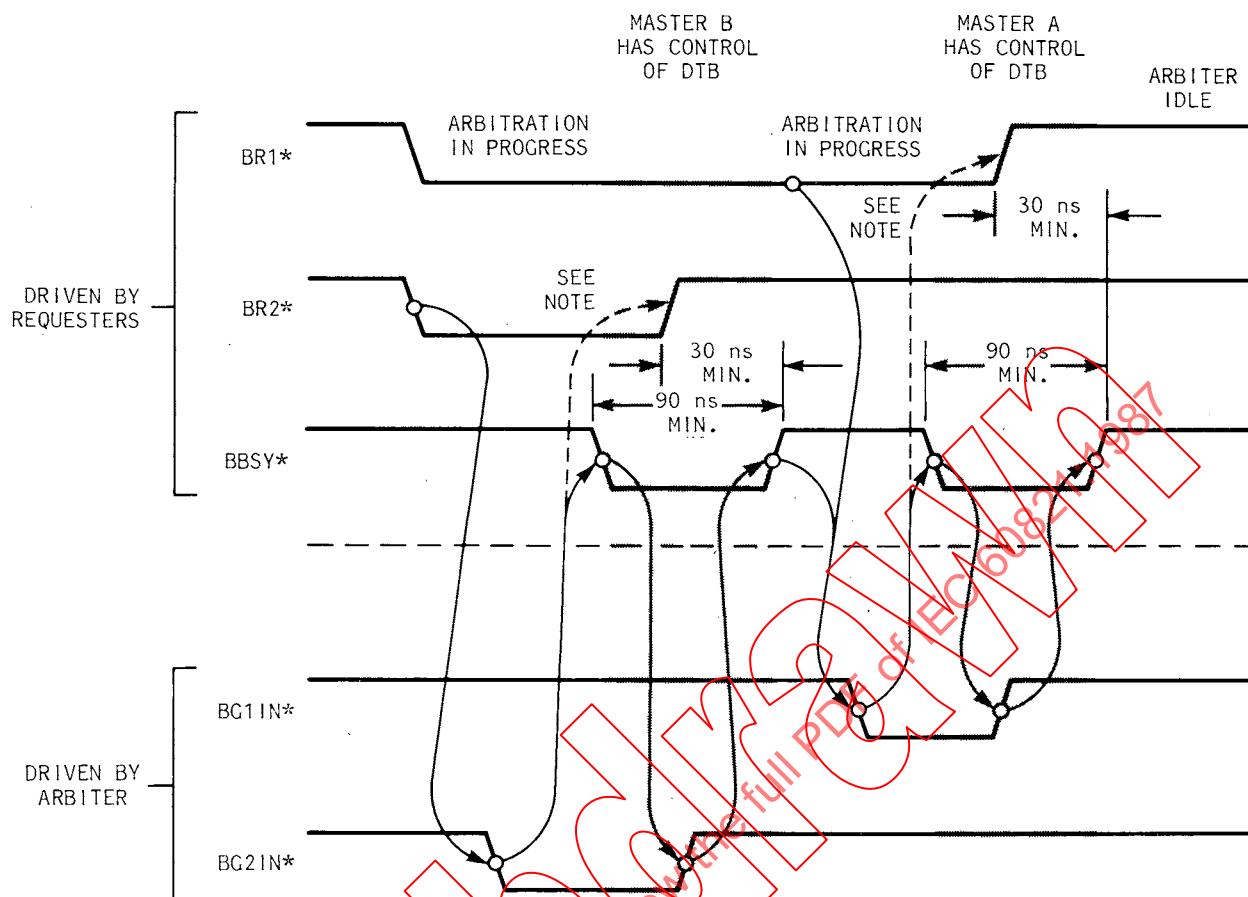
FIGURE 3-5 (concluded).



Note:

Dans cet exemple, chaque DEMANDEUR maintient sa ligne de demande de bus au niveau bas tant qu'on ne lui a pas alloué le DTB. Dans certains cas, un DEMANDEUR pourrait libérer sa ligne de demande de bus sans recevoir un signal d'allocation (voir paragraphe 3.3.2).

FIG. 3-6. - Séquence d'arbitrage: deux DEMANDEURS, deux niveaux de demande.



Note:

In this example, each REQUESTER maintains its bus request line low until it is granted the DTB. In some cases, a REQUESTER might release its bus request line without receiving a bus grant (see Paragraph 3.3.2).

FIG. 3-6. - Arbitration sequence diagram: two REQUESTERS, two request levels.

3.4.2 Arbitrage de deux demandes du bus sur la même ligne

Les figures 3-7 et 3-8, pages 228 et 232, illustrent la suite des opérations qui ont lieu lorsqu'un DEMANDEUR ROR et un DEMANDEUR RWD envoient des demandes simultanées à l'ARBITRE PRI sur une ligne commune de demande de bus. Dans cet exemple, l'ARBITRE et le DEMANDEUR RWD sont situés sur la carte de contrôle du système dans l'emplacement 1, avec le DEMANDEUR ROR situé dans l'emplacement 2. Au début, les deux DEMANDEURS commandent BR1* au niveau bas simultanément. L'ARBITRE commande ensuite BG1IN* de son propre emplacement, l'emplacement 1 au niveau bas. Ce signal BG1IN* est surveillé par le DEMANDEUR B (également dans l'emplacement 1). Lorsque le DEMANDEUR B détecte BG1IN* au niveau bas, il répond en commandant BBSY* au niveau bas. Il libère alors BR1* et informe le MAITRE B que le DTB est disponible.

OBSERVATION 3.15:

Même si le DEMANDEUR B libère BR1*, le DEMANDEUR A continue à le commander au niveau bas (voir figures 3-7 et 3-8).

Après avoir détecté BBSY* au niveau bas, l'ARBITRE commande BG1IN* au niveau haut. Lorsque le MAITRE B a terminé son ou ses transferts de données, il commande le DISPOSITIF DEMANDE LE BUS à l'état faux. Lorsque le DEMANDEUR B le détecte, et 30 ns après que la libération de BR1* eut été satisfaite, le DEMANDEUR B libère BBSY*.

L'ARBITRE interprète la commande de BBSY* à l'état haut comme un signal pour arbitrer n'importe quelle demande de bus en cours. Puisque la ligne BR1* est toujours au niveau bas, l'ARBITRE commande de nouveau BG1IN* au niveau bas. Lorsque le DEMANDEUR B détecte BG1IN* au niveau bas, il commande BG1OUT* au niveau bas parce qu'il n'a pas besoin du DTB. Ensuite, le DEMANDEUR A détecte le niveau bas sur BG1IN* et répond en commandant BBSY* au niveau bas. Quand l'ARBITRE détecte le niveau bas de BBSY*, il commande BG1IN* au niveau haut ce qui impose au DEMANDEUR B de commander BG1OUT* au niveau haut.

Plus tard, lorsque le MAITRE A a terminé ses transferts de données, il commande LE DISPOSITIF DEMANDE LE BUS à l'état faux, indiquant qu'il a terminé l'utilisation du DTB.

Puisque le DEMANDEUR A est un DEMANDEUR ROR, il ne libère pas BBSY*, mais le maintient au niveau bas. Dans le cas où le MAITRE A a de nouveau besoin du DTB, aucun arbitrage ne sera requis. Dans cet exemple cependant, le DEMANDEUR B commande BR1* au niveau bas indiquant un besoin, d'utiliser le DTB, et le DEMANDEUR A (qui surveille les lignes de demande de bus) libère la ligne BBSY*. Ensuite, l'ARBITRE accorde le DTB au DEMANDEUR B.

3.4.2 Arbitration of two bus requests on the same bus request line

Figures 3-7 and 3-8, pages 229 to 233, illustrate the sequence of events which takes place when an ROR REQUESTER and an RWD REQUESTER send simultaneous requests to a PRI ARBITER on a common bus request line. In this example, the ARBITER and RWD REQUESTER are located on the system controller board in slot 1, with the ROR REQUESTER located in slot 2. When the sequence begins, both of the REQUESTERS drive BR1* low simultaneously. The ARBITER then drives BG1IN* low to its own slot (slot 1). That BG1IN* signal is monitored by REQUESTER B (also in slot 1). When REQUESTER B detects BG1IN* low, it responds by driving BBSY* low. REQUESTER B then releases BR1* and informs MASTER B that the DTB is available.

OBSERVATION 3.15:

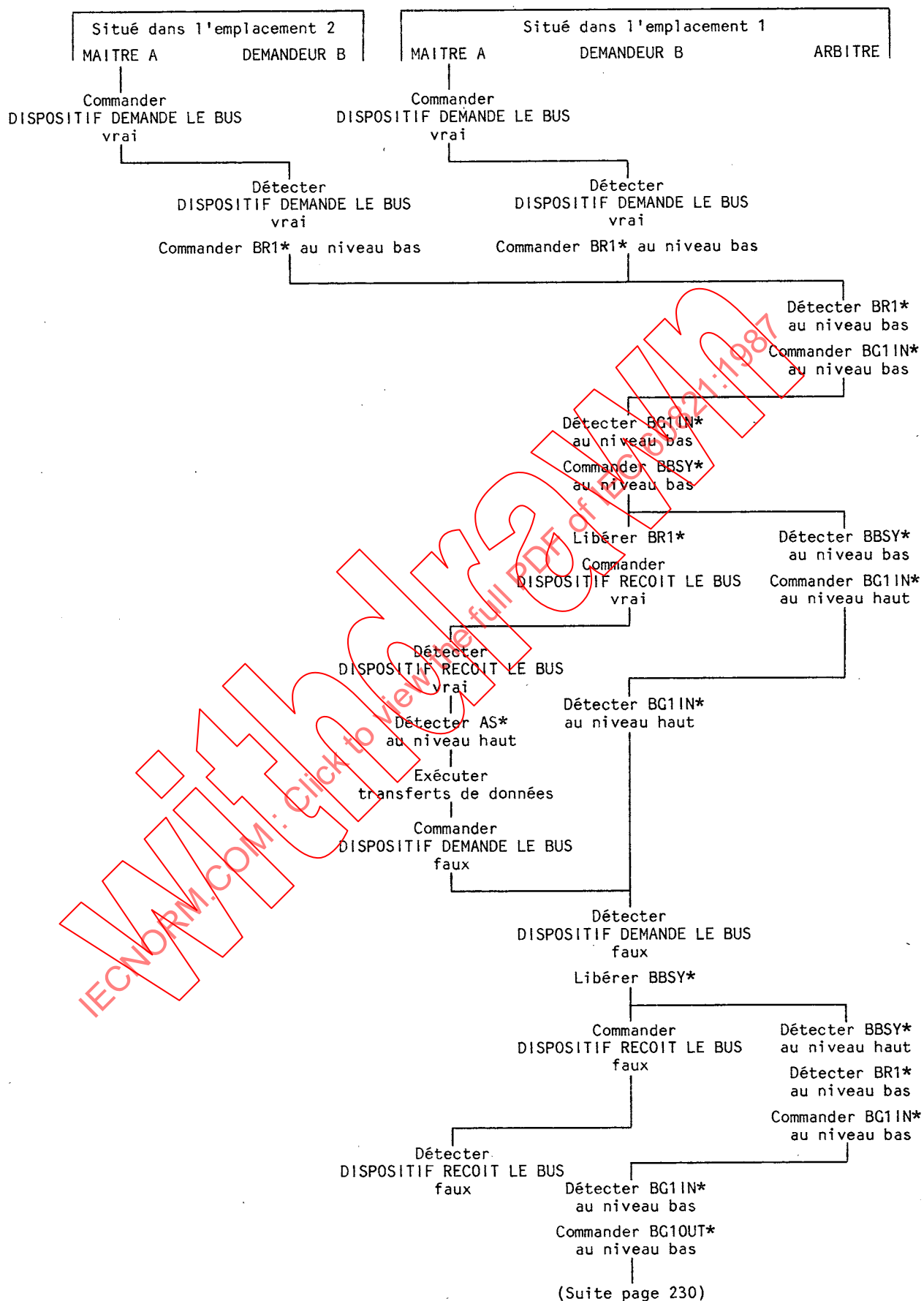
Even though REQUESTER B releases BR1*, REQUESTER A continues to drive it low (see Figures 3-7 and 3-8).

After detecting BBSY* low, the ARBITER drives BG1IN* high. When MASTER B has completed its data transfer(s), it drives DEVICE WANTS BUS false. When REQUESTER B detects this, and when the 30 ns delay since the release of BR1* has been satisfied, REQUESTER B releases BBSY*.

The ARBITER interprets the release of BBSY* as a signal to arbitrate any current bus requests. Since the BR1* line is still low, the ARBITER drives BG1IN* low again. When REQUESTER B detects BG1IN* low, it drives its BG1OUT* low because it does not need the DTB. REQUESTER A then detects the low on its BG1IN* and responds by driving BBSY* low. When the ARBITER detects the low on BBSY*, it drives BG1IN* high, which causes REQUESTER B to drive its BG1OUT* high.

Some time later, when MASTER A has finished its data transfers, it drives DEVICE WANTS BUS false, indicating that it has finished using the DTB.

Since REQUESTER A is an ROR REQUESTER, it does not release BBSY*, but keeps it driven low. In the event that MASTER A needs to use the DTB again, no arbitration will be required. In this example, however, REQUESTER B drives BR1* low, indicating a need to use the DTB, and REQUESTER B (which is monitoring the bus request lines) releases the BBSY* line. The ARBITER then grants the DTB to REQUESTER B.



Note:

Note:
Le DISPOSITIF DEMANDE LE BUS et le DISPOSITIF RECOIT LE BUS sont des signaux internes à la carte entre le MAÎTRE et son DEMANDEUR (voir figure 3-4, page 212).

FIG. 3-7. - Organigramme de l'arbitrage: deux DEMANDEURS, même niveau de demande

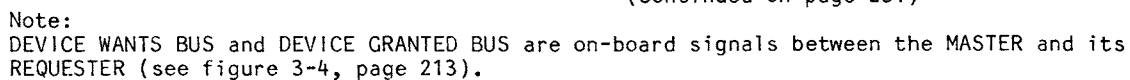


FIG. 3-7. - Arbitration flow diagram: two REQUESTERS, same request level.

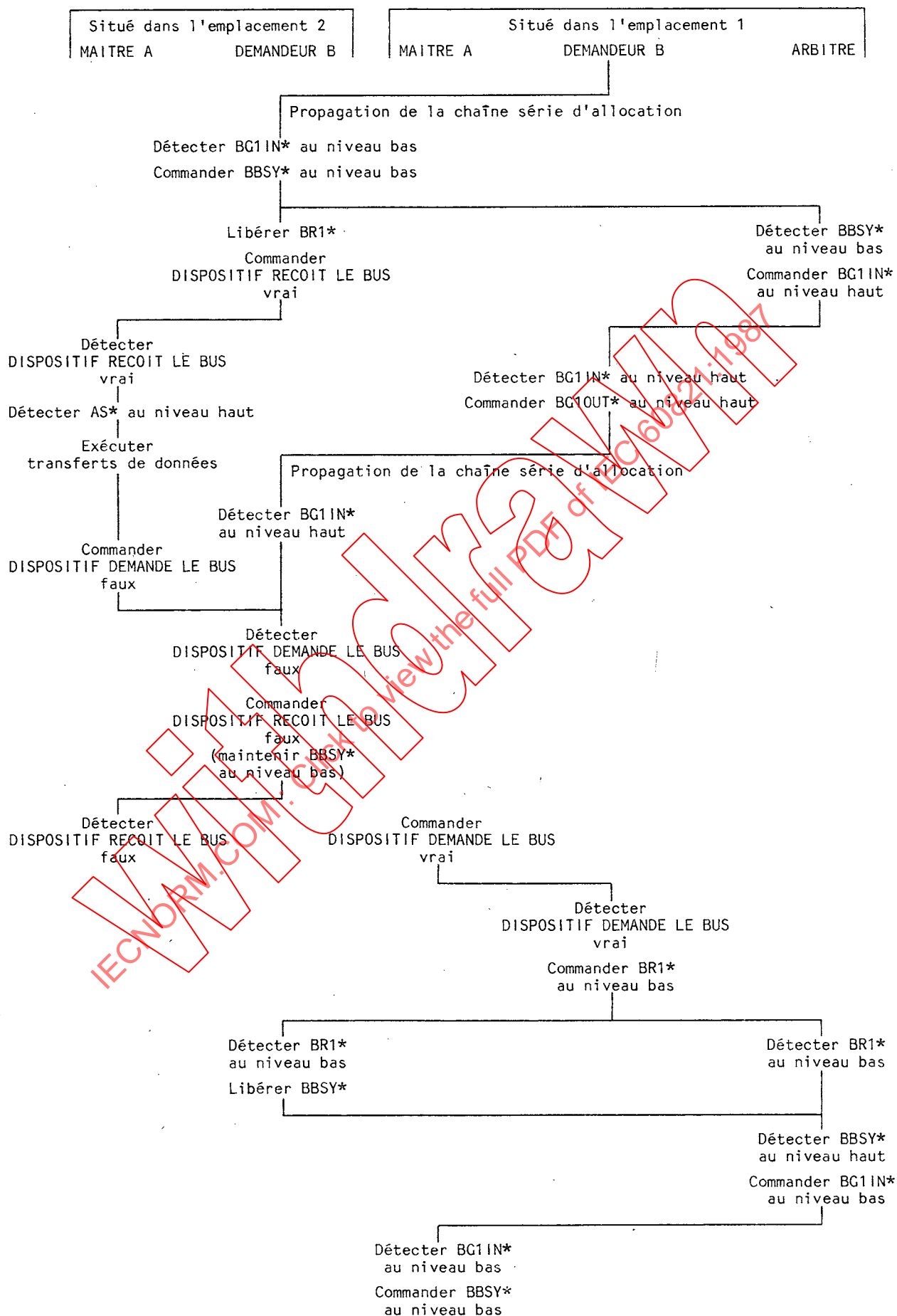


FIGURE 3-7 (fin).

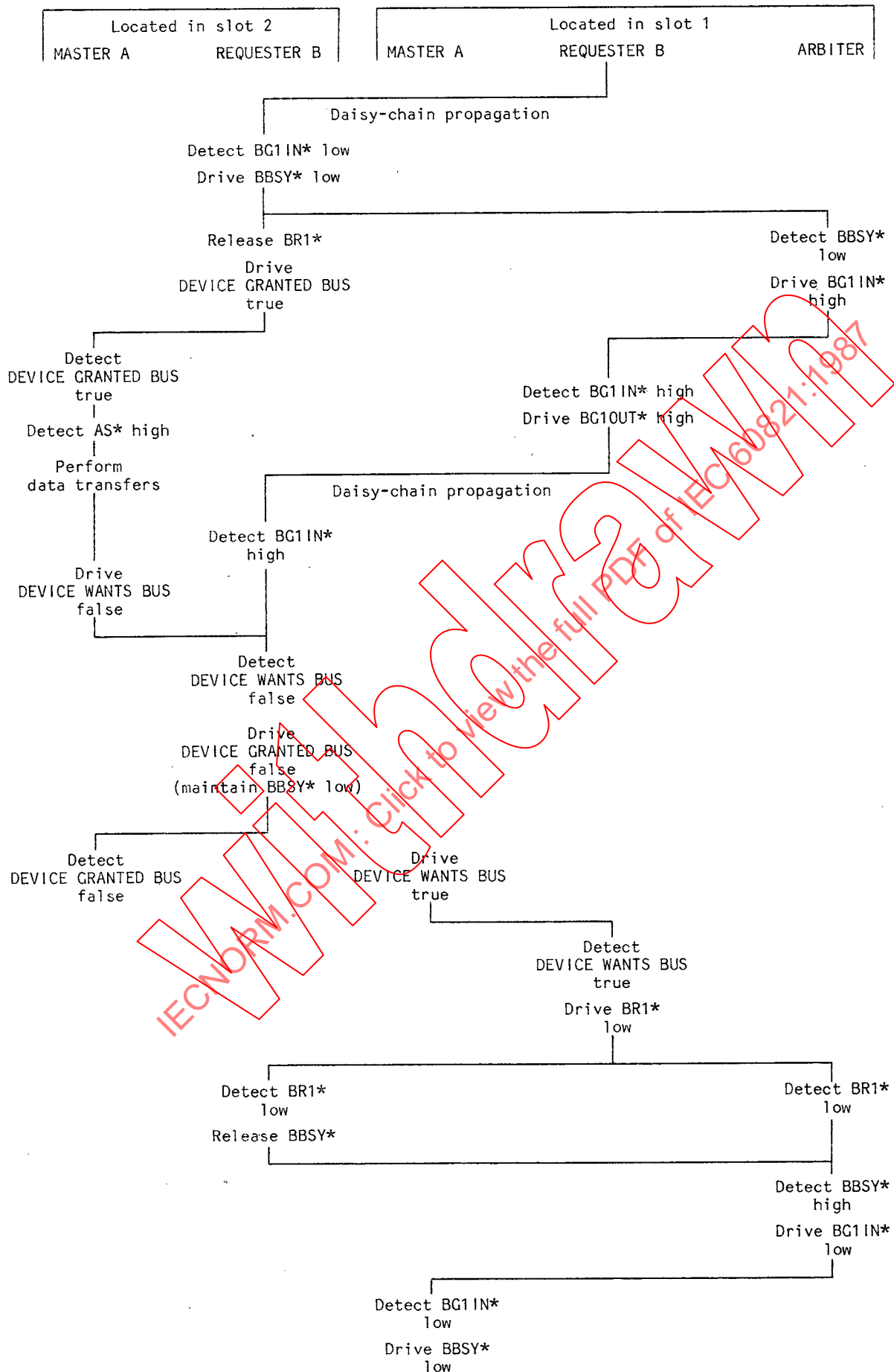
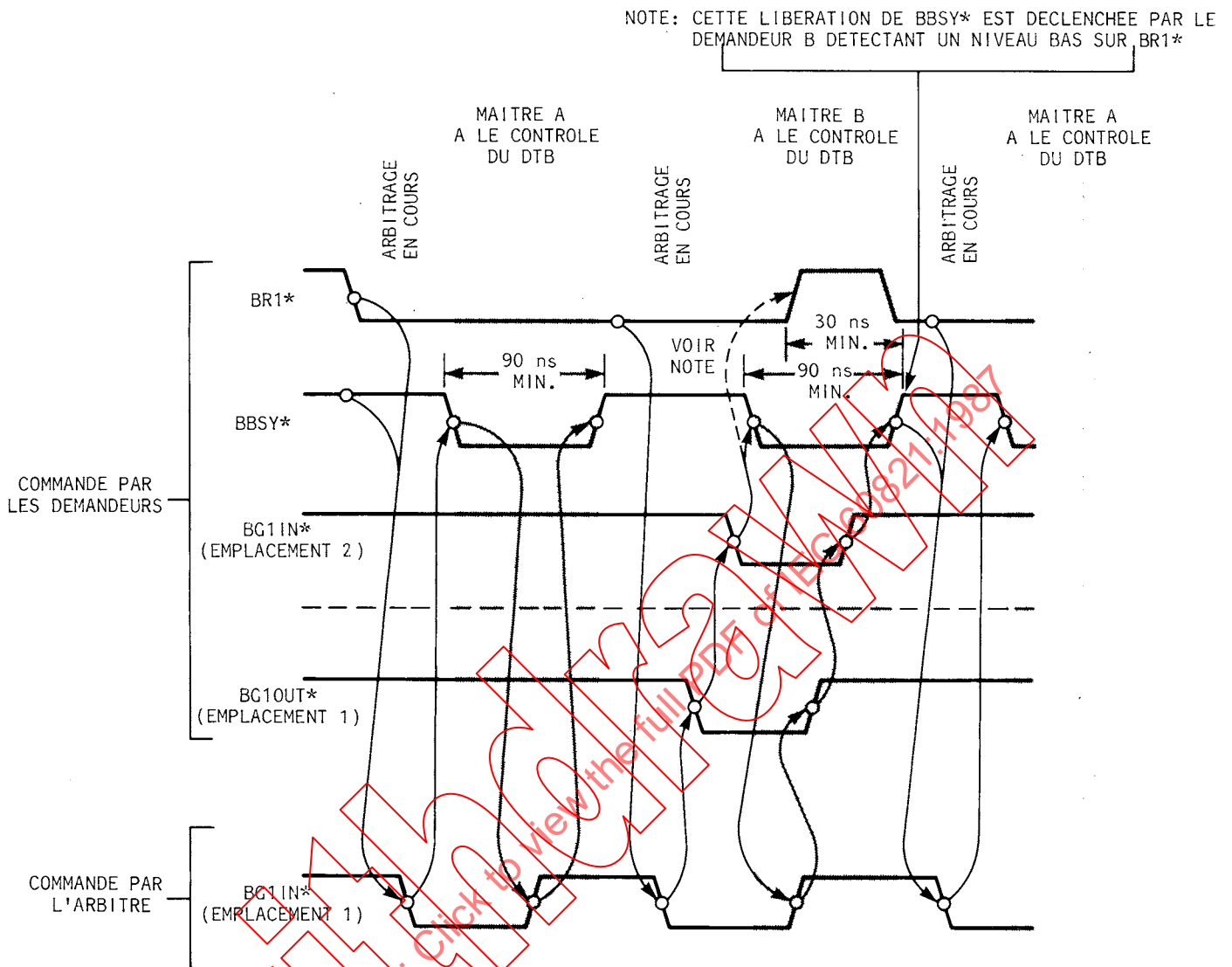


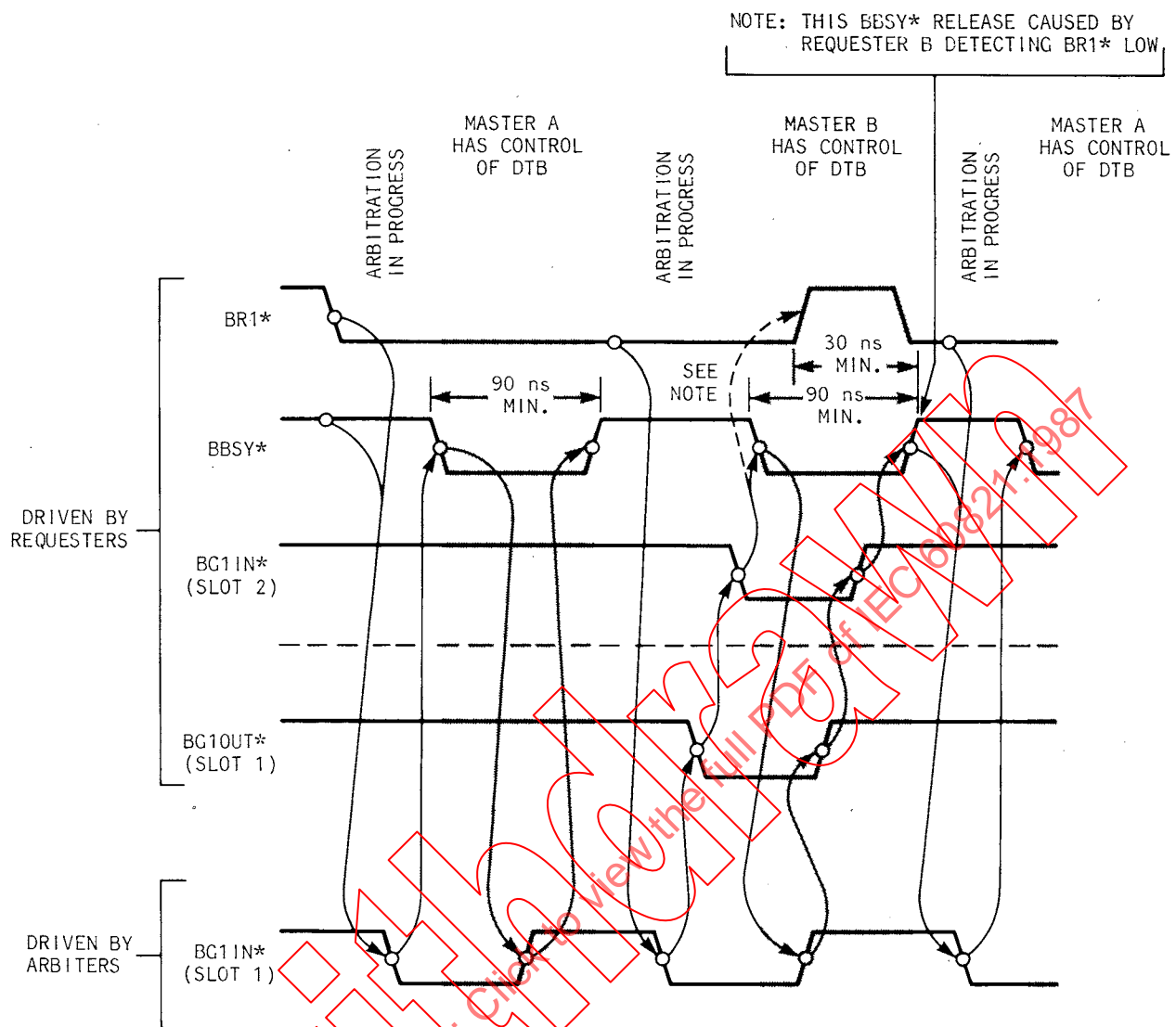
FIGURE 3-7 (concluded).



Note:

Dans cet exemple, chaque DEMANDEUR maintient sa ligne de demande de bus au niveau bas tant qu'on ne lui a pas alloué le DTB. Dans certains cas, un DEMANDEUR pourrait libérer sa ligne de demande de bus sans recevoir un signal d'allocation (voir paragraphe 3.3.2).

FIG. 3-8. - Séquence d'arbitrage: deux DEMANDEURS, même niveau de demande.



Note:

In this example, each REQUESTER maintains its bus request line low until it is granted the DTB. In some cases, a REQUESTER might release its bus request line without receiving a bus grant (see Paragraph 3.3.2).

FIG. 3-8. Arbitration sequence diagram: two REQUESTERS, same request level.

3.5 *Course critique entre les demandes du MAITRE et les accords de l'ARBITRE*

Supposons qu'il y ait deux DEMANDEURS: le DEMANDEUR A et le DEMANDEUR B qui partagent une ligne commune de demande de bus. Le DEMANDEUR B, qui est en aval sur la chaîne série demande le bus et l'ARBITRE commande au niveau bas la ligne d'allocation du bus correspondante. Cette allocation du bus arrive au DEMANDEUR A au moment où le MAITRE A signale qu'il a besoin du bus. Si le DEMANDEUR A est mal conçu, cette situation pourrait le faire commander momentanément la ligne BGxOUT* au niveau bas puis au niveau haut de nouveau y provoquant une impulsion transitoire de très courte durée.

REGLE 3.13:

Les DEMANDEURS DOIVENT être conçus pour assurer qu'aucune impulsion transitoire de niveau bas ne soit pas générée sur leur ligne BGxOUT*.

OBSERVATION 3.16:

Si le DEMANDEUR est conçu pour mémoriser l'état de la ligne DISPOSITIF DEMANDE LE BUS interne à sa carte sur le front descendant de sa ligne d'entrée de l'allocation du bus et si ce signal est en transition lorsque ce front descendant a lieu, les sorties de cette mémoire pourront parfois osciller ou resteront pendant une courte durée dans la région de seuil entre les niveaux haut et bas. A cause de cela, la norme du BUS CEI 821 n'impose pas de limite de temps au DEMANDEUR pour transmettre l'allocation du bus. Il leur interdit seulement de générer une impulsion transitoire au niveau bas sur sa ligne BGxOUT* qui pourrait être interprétée comme une allocation du bus par le DEMANDEUR en aval dans la chaîne série d'allocation.

AUTORISATION 3.8:

SI un DEMANDEUR détecte que son MAITRE sur sa carte a besoin du bus après qu'il eut reçu l'allocation du bus destinée à un autre DEMANDEUR et avant de passer cette allocation du bus,
ALORS il PEUT considérer l'allocation de bus comme la sienne. Dans ce cas, l'autre DEMANDEUR maintiendra sa demande de bus jusqu'à ce qu'une autre allocation de bus soit donnée.

3.5 Race conditions between MASTER requests and ARBITER grants

Suppose that there are two REQUESTERS: REQUESTER A and REQUESTER B, that share a common bus request line. REQUESTER B, which is further down the daisy-chain, requests the bus and the ARBITER drives the corresponding bus grant line low. This bus grant arrives at REQUESTER A just as MASTER A signals that it wants the bus. If REQUESTER A is improperly designed, this situation might cause it to momentarily drive its BGxOUT* line low and then high again resulting in a low-going transient.

RULE 3.13:

REQUESTERS MUST be designed to ensure that no momentarily low-going transients are generated on their BGxOUT* line.

OBSERVATION 3.16:

If the REQUESTER is designed such that it latches the state of the on-board DEVICE WANTS BUS line upon the falling edge of its bus grant in line, and if that signal is in transition when the falling edge occurs, the outputs of the latch will sometimes oscillate, or remain in the threshold region between the high and low levels, for a short time. Because of this, the IEC 821 BUS standard does not set a time limit for the REQUESTER to pass along the bus grant. It only prohibits the REQUESTER from generating a low-going transient on its BGxOUT* line which might be interpreted as a bus grant by a REQUESTER further down the daisy-chain.

PERMISSION 3.8:

IF a REQUESTER detects that its on-board MASTER needs the bus between the time that it receives a bus grant intended for another REQUESTER, and the time it would pass that bus grant on,
THEN it MAY treat the bus grant as its own. In this case, the other REQUESTER will maintain its bus request until another bus grant is issued.

CHAPITRE 4: BUS D'INTERRUPTION PRIORITAIRE DU BUS CEI 821

4.1 *Introduction*

Le BUS CEI 821 inclut un bus d'interruption prioritaire qui fournit les lignes de signaux nécessaires pour générer et servir des interruptions. La figure 4-1, page 238, représente un système typique de BUS CEI 821. Des GENERATEURS D'INTERRUPTION utilisent le bus d'interruption prioritaire pour envoyer des demandes d'interruption aux CONTROLEURS D'INTERRUPTION qui répondent à ces demandes.

Un système qui possède une possibilité d'interruption inclut des sous-programmes qui sont invoqués par les interruptions et qui alors s'appellent sous-programmes de service d'interruption. Les systèmes d'interruption peuvent se classer en deux groupes:

- a) Les systèmes à contrôleur unique ayant un seul CONTROLEUR D'INTERRUPTION qui reçoit et sert toutes les interruptions.
- b) Les systèmes à gestion répartie ayant deux ou plusieurs CONTROLEURS D'INTERRUPTION qui reçoivent et servent les interruptions du bus.

4.1.1 *Systèmes à contrôleur unique*

Dans un système à contrôleur unique, toutes les interruptions sont reçues par un seul CONTROLEUR D'INTERRUPTION et tous les sous-programmes de service d'interruption sont exécutés par un seul processeur. La figure 4-2, page 240, représente la structure relative aux interruptions d'un système à contrôleur unique. Ce type d'architecture est bien adapté aux applications de commande de processus et de la machine où un processeur de supervision coordonne les activités des processeurs dédiés. Les processeurs dédiés possèdent généralement une interface à la machine ou au processus commandé.

Le processeur de supervision est le destinataire de toutes les interruptions du bus, les traitant en tenant compte de leur priorité. Les processeurs dédiés n'ont pas à traiter les interruptions du bus et peuvent se consacrer à la commande de la machine ou du processus.

4.1.2 *Systèmes distribués*

La figure 4-3, page 242, représente la structure relative aux interruptions d'un système distribué. Ce système inclut deux ou plusieurs CONTROLEURS D'INTERRUPTION, chacun servant un seul sous-ensemble d'interruption du bus. En général, chaque CONTROLEUR D'INTERRUPTION réside sur une carte processeur différente. Ce type d'architecture est bien adapté aux applications de traitement réparti où des processeurs multiples et égaux exécutent le logiciel d'application. Comme chacun de ces processeurs égaux exécute une partie du logiciel du système, il peut avoir besoin de communiquer avec les autres processeurs. Dans un système distribué, chaque processeur sert seulement les interruptions qui lui sont envoyées en établissant des voies de communication dédiées parmi tous les processeurs.

CHAPTER 4: IEC 821 BUS PRIORITY INTERRUPT BUS

4.1 *Introduction*

The IEC 821 BUS includes a Priority Interrupt Bus which provides the signal lines needed to generate and service interrupts. Figure 4-1, page 239, shows a typical IEC 821 BUS system. INTERRUPTERS use the Priority Interrupt Bus to send interrupt requests to INTERRUPT HANDLERS which respond to these requests.

Any system which has interrupt capability includes software routines that are called interrupt service routines, and are invoked by the interrupts. Interrupt subsystems can be classified into two groups.

- a) Single handler systems, which have only one INTERRUPT HANDLER that receives and services all bus interrupts.
- b) Distributed systems, which have two or more INTERRUPT HANDLERS that receive and service bus interrupts.

4.1.1 *Single handler systems*

In a single handler system, all interrupts are received by one INTERRUPT HANDLER, and all interrupt service routines executed by one processor. Figure 4-2, page 241, shows the interrupt structure of a single handler system. This type of architecture is well suited to machine or process control applications, where a supervisory processor coordinates the activities of dedicated processors. The dedicated processors are typically interfaced to the machine or the process being controlled.

The supervisory processor is the destination for all bus interrupts, servicing them in a prioritized manner. The dedicated processors are not required to service interrupts from the bus, and can give primary attention to controlling a machine or process.

4.1.2 *Distributed systems*

Figure 4-3, page 243, shows the interrupt structure of a distributed system. This system includes two or more INTERRUPT HANDLERS, each servicing only a subset of the bus interrupts. In a typical implementation, each of the INTERRUPT HANDLERS resides on a different processor board. This type of architecture is well suited to distributed computing applications, where multiple, co-equal processors execute the application software. As each of the co-equal processors executes part of the system software, it might need to communicate with the other processors. In the distributed system, each processor services only those interrupts directed to it, establishing dedicated communication paths among all processors.

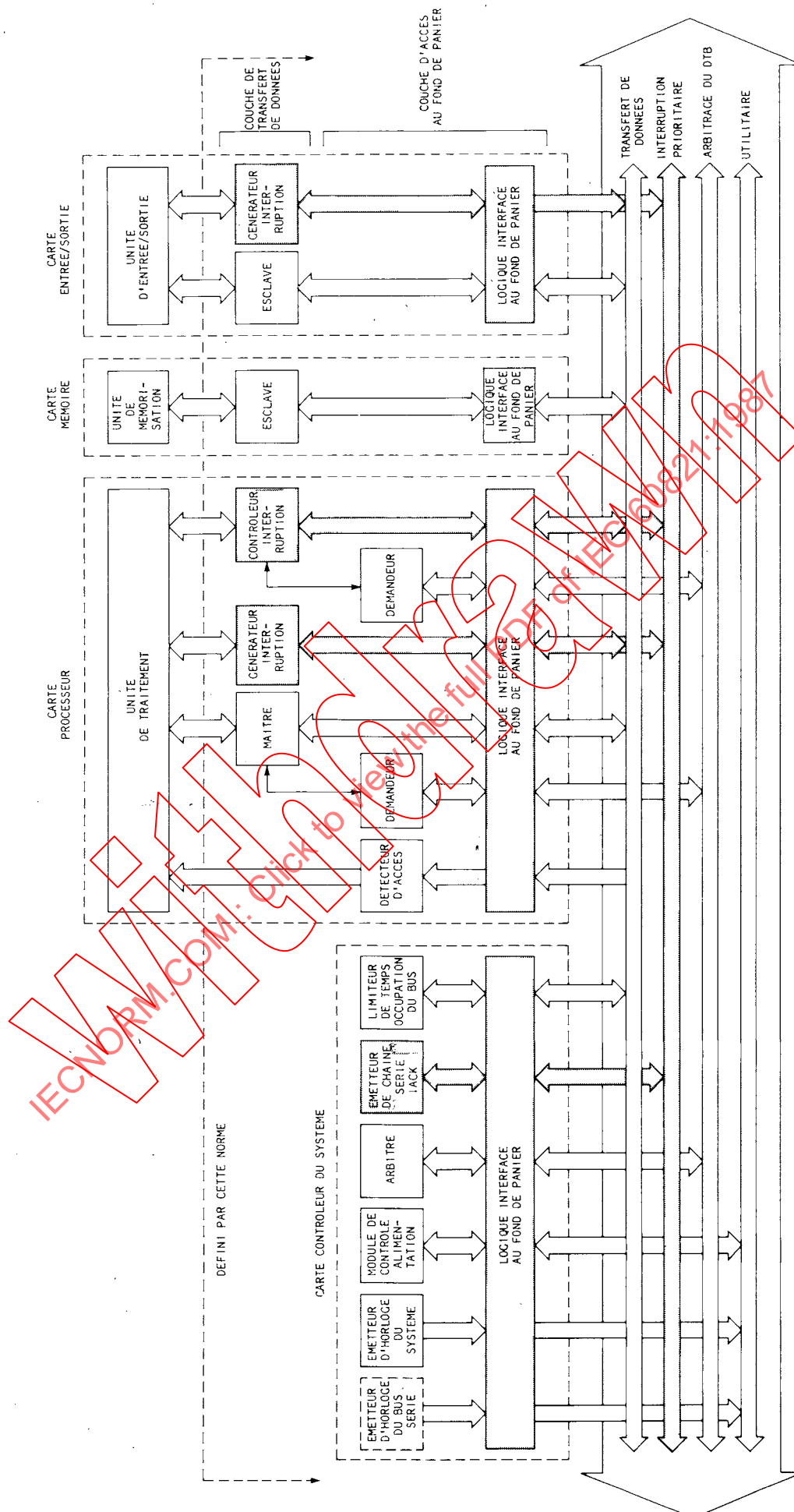


FIG. 4-1. - Schéma-bloc fonctionnel du système d'interruption prioritaire du BUS CEI 821.

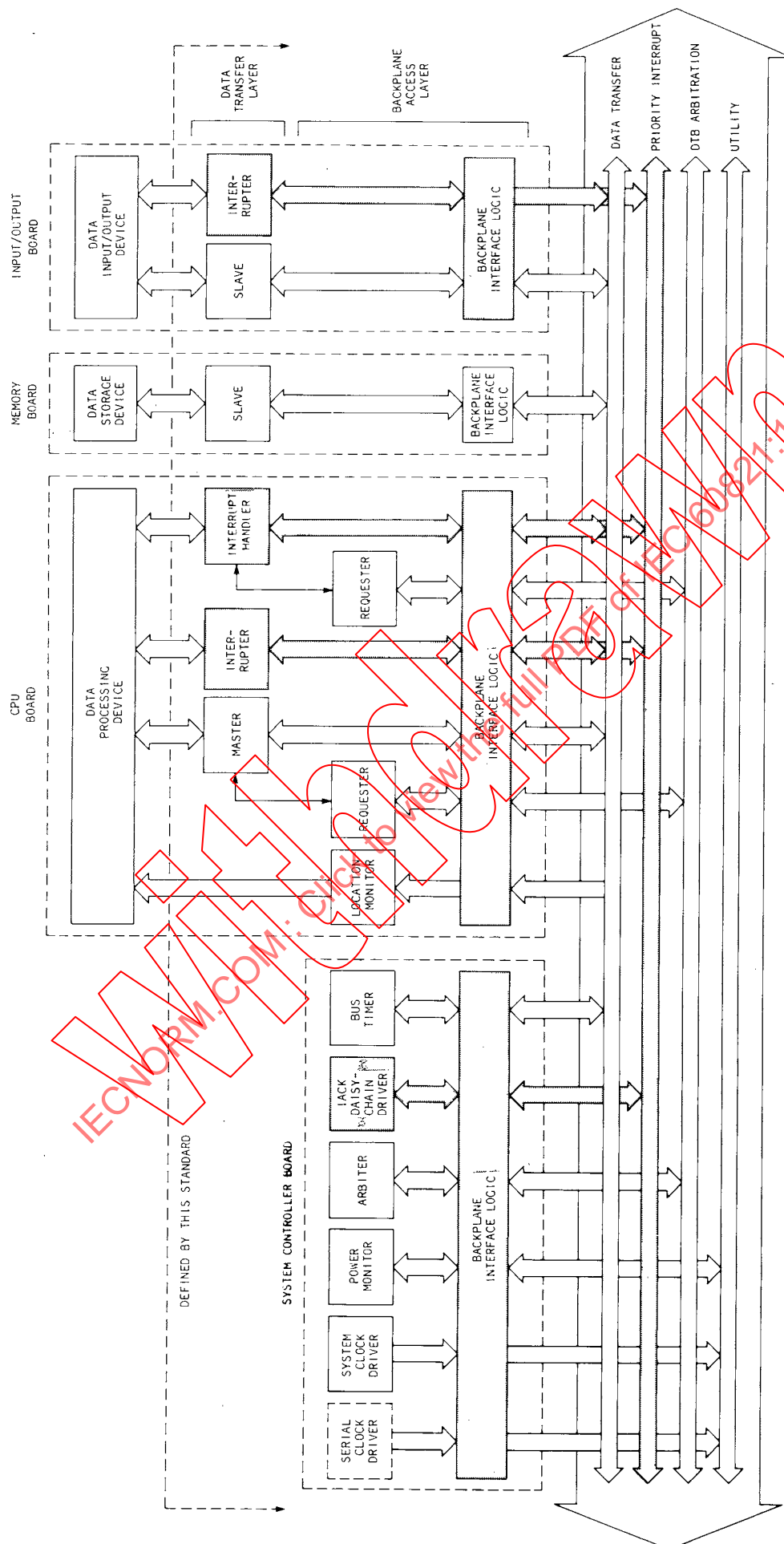


FIG. 4-1. - Priority Interrupt Bus functional block diagram.

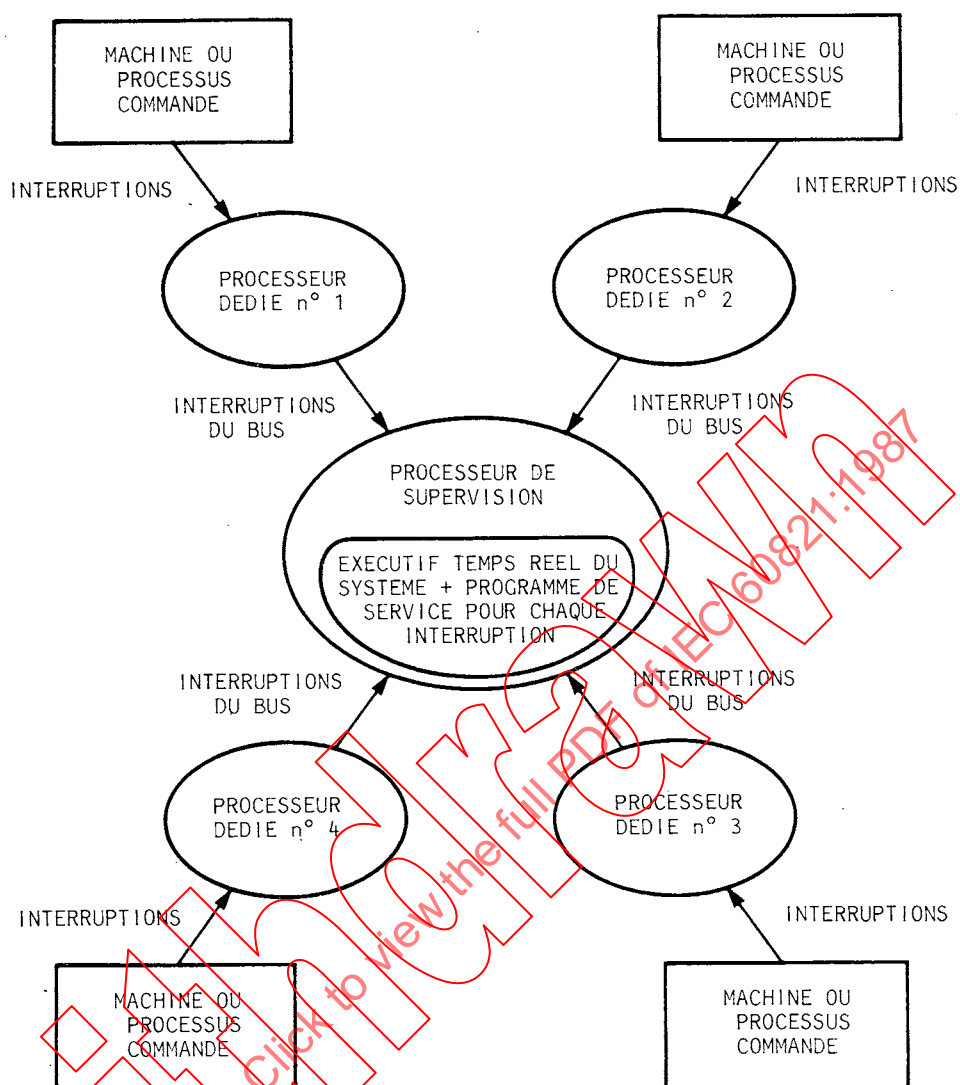


FIG. 4-2. - Structure du sous-système d'interruption: système à contrôleur unique.

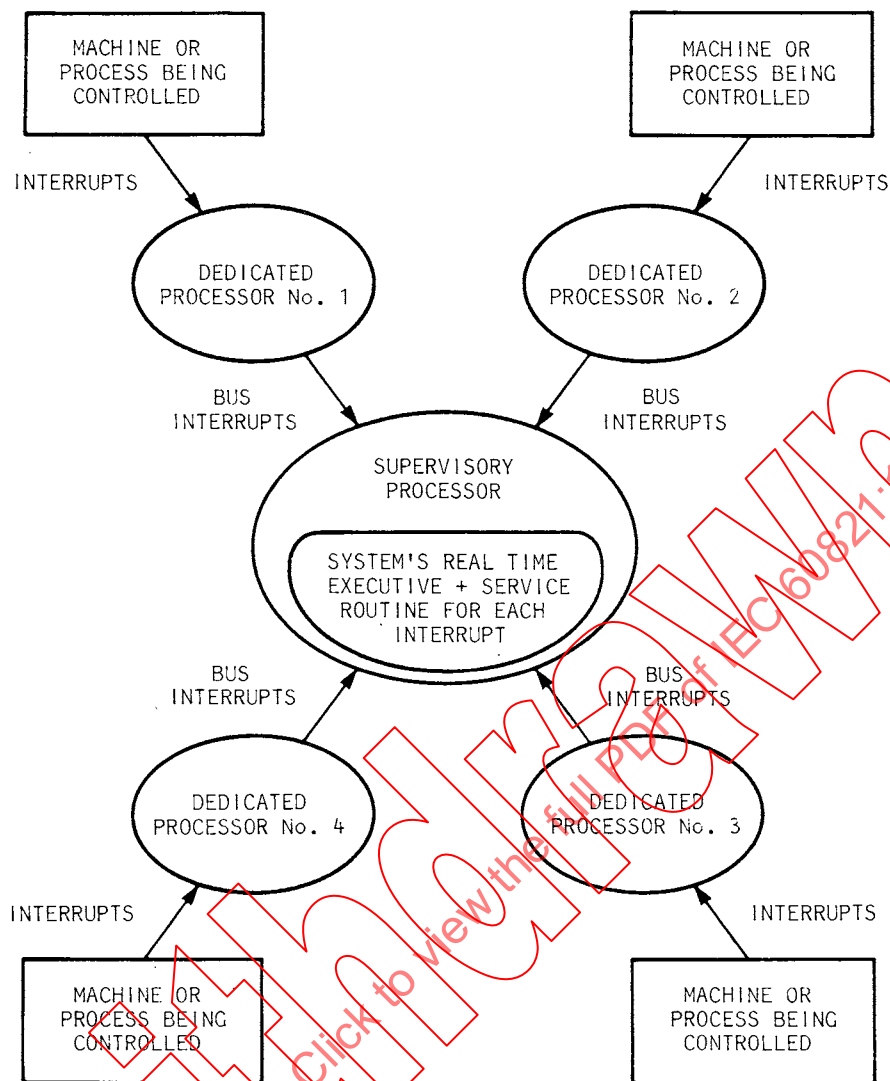


FIG. 4-2. - Interrupt subsystem structure: single handler system.

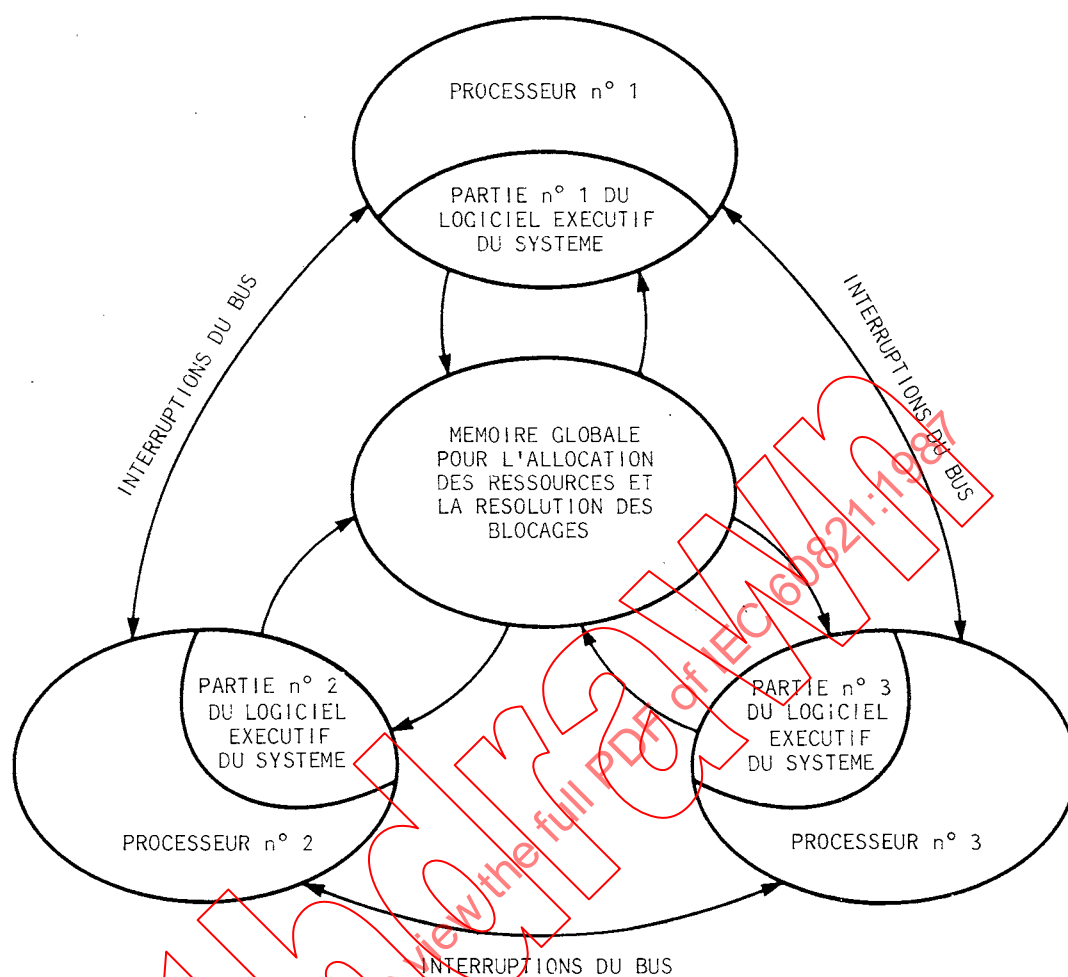


FIG. 4-3 - Structure du sous-système d'interruption: système distribué.

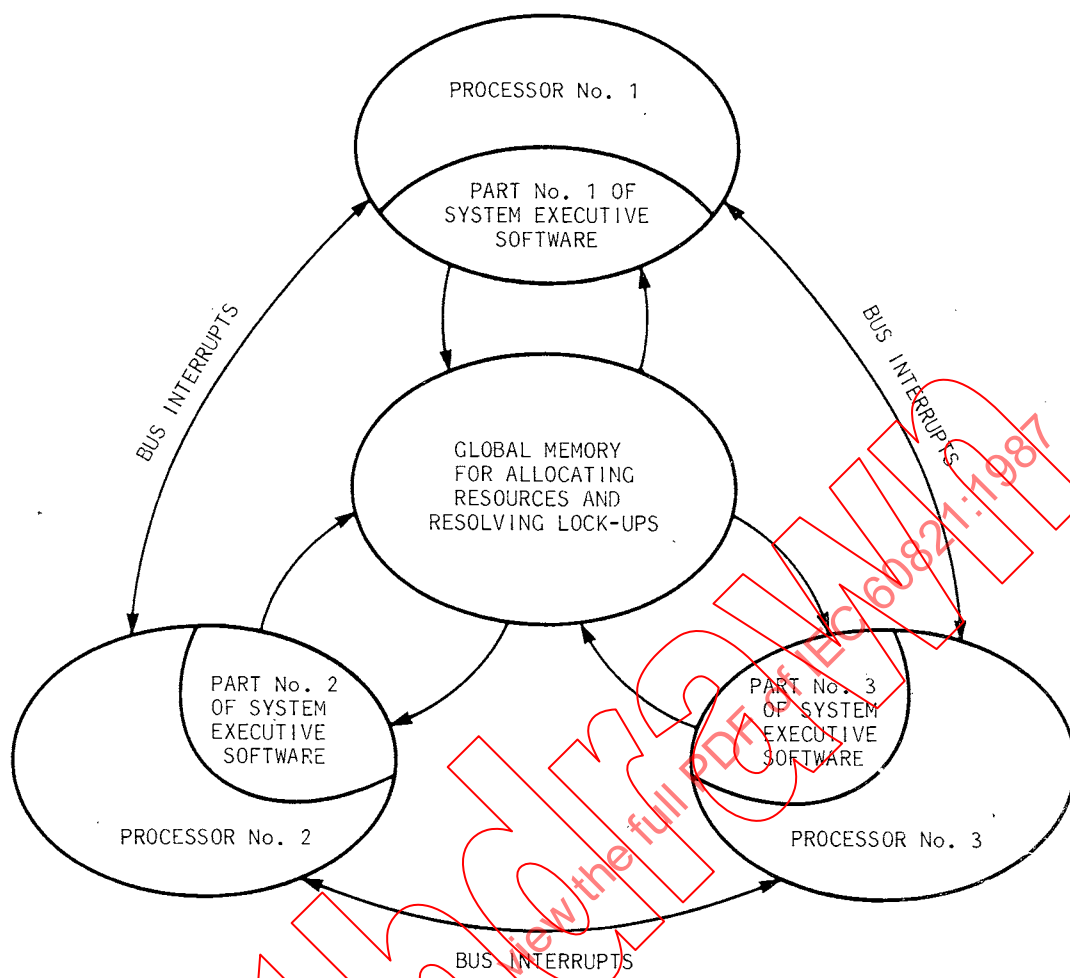


FIG. 4-3. - Interrupt subsystem structure: distributed system.

4.2 Lignes du bus d'interruption prioritaire

Le bus de transfert de données, le bus d'arbitrage et le bus d'interruption prioritaire sont tous utilisés dans le processus de génération et de traitement des interruptions du bus.

La description qui suit, relative au bus d'interruption prioritaire, suppose que le lecteur a compris le fonctionnement du bus de transfert de données décrit dans le chapitre 2, et du bus d'arbitrage décrit dans le chapitre 3.

Le bus d'interruption prioritaire comprend sept lignes de demande d'interruption, une ligne d'acquiescement d'interruption et une chaîne série de reconnaissance d'interruption:

IRQ1*	demande d'interruption 1
IRQ2*	demande d'interruption 2
IRQ3*	demande d'interruption 3
IRQ4*	demande d'interruption 4
IRQ5*	demande d'interruption 5
IRQ6*	demande d'interruption 6
IRQ7*	demande d'interruption 7
IACK*	reconnaissance d'interruption
IACKIN*/IACKOUT*	chaîne série de reconnaissance d'interruption

4.2.1 Lignes de demande d'interruption

Les GENERATEURS D'INTERRUPTION émettent des interruptions en commandant une ligne de demandes d'interruption au niveau bas. Dans un système à contrôleur unique, les lignes de requête d'interruption sont hiérarchisées, IRQ7* ayant la priorité la plus élevée.

4.2.2 Ligne de reconnaissance d'interruption

La ligne IACK* occupe toute la longueur du bus et se connecte à la broche IACKIN* de l'emplacement 1 (voir figure 4-4, page 246). Lorsqu'elle est commandée au niveau bas, la broche IACKIN* oblige l'EMETTEUR DE CHAÎNE SÉRIE IACK situé dans l'emplacement 1, à propager un front descendant le long de la chaîne série de reconnaissance des interruptions.

4.2.3 Chaîne série de reconnaissance d'interruption - IACKIN*/IACKOUT*

Chacune des sept lignes de demande d'interruption peut être partagée entre deux ou plusieurs modules GENERATEURS D'INTERRUPTION. La chaîne série de reconnaissance d'interruption garantit qu'un seul GENERATEUR D'INTERRUPTION répond au cycle de reconnaissance d'interruption. Cette ligne en chaîne série traverse toutes les cartes sur le BUS CEI 821. Chaque GENERATEUR D'INTERRUPTION commandant au niveau bas une ligne de demande d'interruption attend l'arrivée d'un front descendant sur son entrée IACKIN* de la chaîne série. Ce n'est qu'après avoir reçu ce front descendant qu'un GENERATEUR D'INTERRUPTION répondra à un cycle de reconnaissance d'interruption. Il ne transmet pas le front descendant en aval sur la chaîne série, ce qui garantit qu'aucun autre GENERATEUR D'INTERRUPTION ne répondra à un cycle de reconnaissance d'interruption.

4.2 *Priority Interrupt Bus lines*

The Data Transfer Bus, the Arbitration Bus, and the Priority Interrupt Bus are all used in the process of generating and handling bus interrupts.

The following discussion of the Priority Interrupt Bus assumes that the reader understands the operation of the Data Transfer Bus described in Chapter 2, and the Arbitration Bus described in Chapter 3.

The Priority Interrupt Bus consists of seven interrupt request signal lines, one interrupt acknowledge line, and one interrupt acknowledge daisy-chain.

IRQ1*	interrupt request 1
IRQ2*	interrupt request 2
IRQ3*	interrupt request 3
IRQ4*	interrupt request 4
IRQ5*	interrupt request 5
IRQ6*	interrupt request 6
IRQ7*	interrupt request 7
IACK*	interrupt acknowledge
IACKIN*/IACKOUT*	interrupt acknowledge daisy-chain

4.2.1 *Interrupt request lines*

INTERRUPTERS requests interrupts by driving an interrupt request line low. In a single handler system, these interrupt request lines are prioritized, with IRQ7* having the highest priority.

4.2.2 *Interrupt acknowledge line*

The IACK* line runs the full length of the backplane and is connected to the IACKIN* pin of slot 1 (see Figure 4-4, page 247). When driven low, the IACKIN* pin causes the IACK DAISY-CHAIN DRIVER, located in slot 1, to propagate a falling edge down the interrupt acknowledge daisy-chain.

4.2.3 *Interrupt acknowledge daisy-chain - IACKIN*/IACKOUT**

Each of the seven interrupt request lines can be shared by two or more INTERRUPTER modules. The interrupt acknowledge daisy-chain assures that only one INTERRUPTER responds to the interrupt acknowledge cycle. This daisy-chain line passes through each board on the IEC 821 BUS. Each INTERRUPTER that is driving an interrupt request line low waits for a falling edge to arrive at its IACKIN* daisy-chain input. Only upon receiving this falling edge does an INTERRUPTER respond to an interrupt acknowledge cycle. It does not pass the falling edge on down the daisy-chain, preventing other INTERRUPTERS from responding to the interrupt acknowledge cycle.

REGLE 4.1:

SI un emplacement du fond de panier du BUS CEI 821 n'est pas occupé par une carte et s'il y a des cartes plus en aval le long de la chaîne série de reconnaissance d'interruption,
ALORS des cavaliers DOIVENT être installés à l'emplacement vide pour propager le signal de la chaîne série.

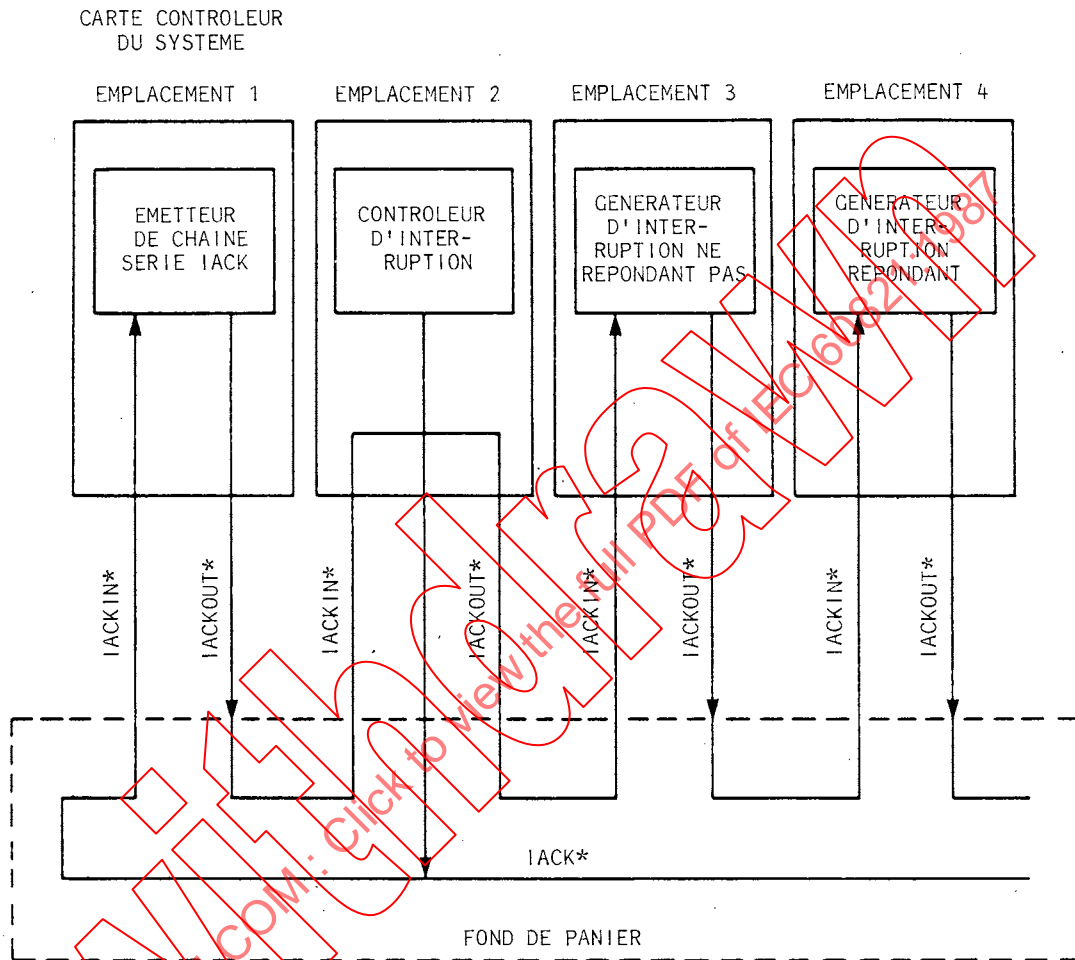


FIG. 4-4. - CHAINE SERIE IACKIN*/IACKOUT*.

4.3 Modules du bus d'interruption prioritaire - Description générale

Il y a trois types de modules fonctionnels associés au bus d'interruptions prioritaires: les GENERATEURS D'INTERRUPTION, les CONTROLEURS D'INTERRUPTION et les EMETTEURS DE CHAINE SERIE IACK. Les possibilités des GENERATEURS D'INTERRUPTION et CONTROLEURS D'INTERRUPTION sont décrites par une liste de mnémoniques qui montre quels types de cycle de reconnaissance d'interruption ils peuvent générer et accepter respectivement.

Les paragraphes 4.3.1 à 4.3.3 fournissent les schémas-blocs des trois types de modules du bus d'interruption prioritaire: CONTROLEUR D'INTERRUPTION, GENERATEUR D'INTERRUPTION et EMETTEUR DE CHAINE SERIE IACK.

RULE 4.1:

IF an IEC 821 BUS backplane slot is not occupied by a board, and if there are boards further down the interrupt acknowledge daisy-chain,

THEN jumpers MUST be installed at the empty slot to pass through the daisy-chain signal.

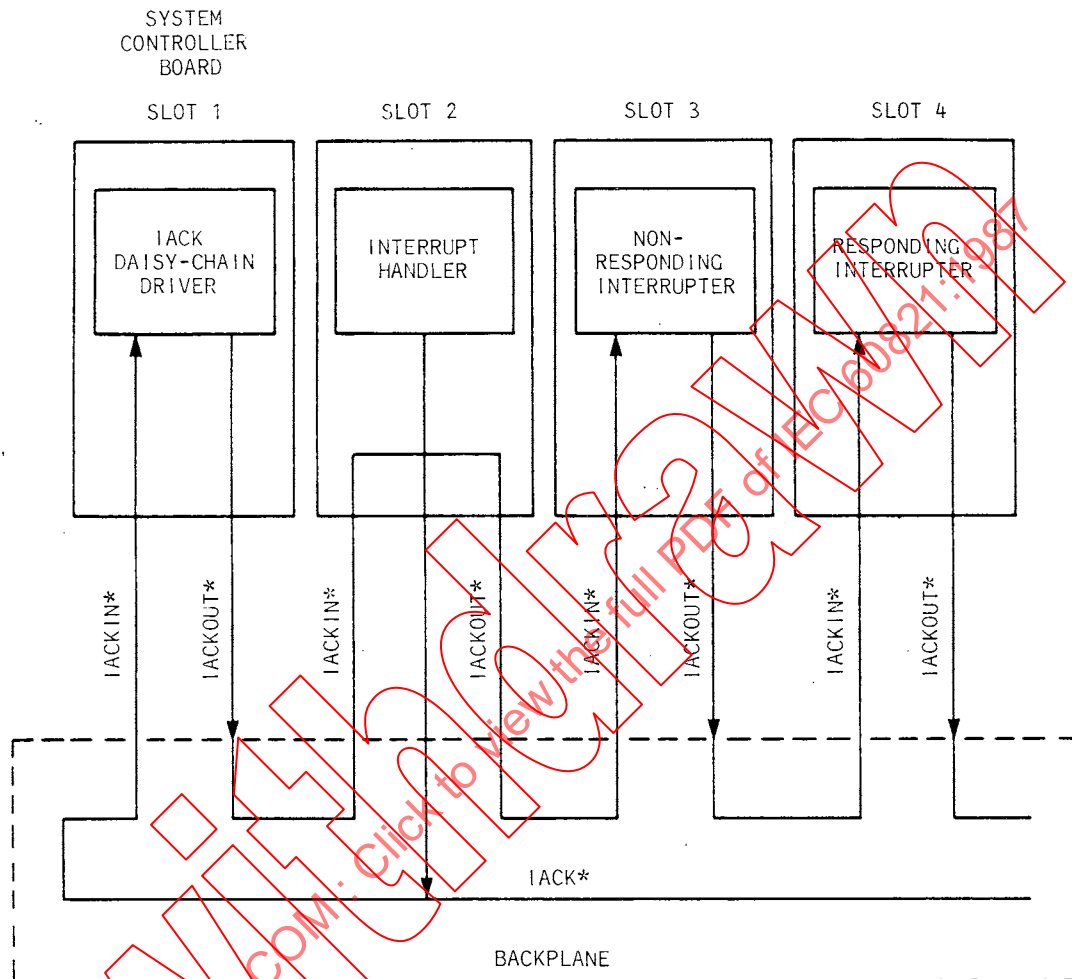


FIG. 4-4. - IACKIN*/IACKOUT* DAISY-CHAIN.

4.3 Priority Interrupt Bus modules - Basic description

There are three types of functional modules associated with the Priority Interrupt Bus: INTERRUPTERS, INTERRUPT HANDLERS, and IACK* DAISY-CHAIN DRIVERS. The capabilities of INTERRUPT HANDLERS and INTERRUPTERS are described in a list of mnemonics that show what interrupt acknowledge cycle types they can generate and accept, respectively.

Paragraphs 4.3.1 through 4.3.3 provide block diagrams for the three types of Priority Interrupt Bus Modules: INTERRUPT HANDLER, INTERRUPTER, and IACK* DAISY-CHAIN DRIVER.

REGLE 4.2:

Les lignes de signaux de sorties indiquées en lignes continues dans les figures 4-5 à 4-7, pages 250 à 262, DOIVENT être commandées par le module à moins qu'il les commandent toujours au niveau haut.

OBSERVATION 4.1:

SI une ligne de sortie n'est pas commandée,
ALORS des adaptateurs d'impédance sur le fond de panier assurent le maintien de la ligne au niveau haut.

REGLE 4.3:

Les lignes de signaux d'entrées indiquées en lignes continues dans les figures 4-5 à 4-7 DOIVENT être surveillées et donner lieu à une réponse appropriée.

OBSERVATION 4.2:

Les REGLES et OBSERVATIONS pour la commande et la surveillance des lignes de signaux indiquées en pointillé dans les figures 4-5 à 4-7 sont données dans les tableaux 4-1 et 4-2.

4.3.1 CONTROLEURS D'INTERRUPTION

Le CONTROLEUR D'INTERRUPTION est utilisé pour accomplir plusieurs tâches:

- a) Il hiérarchise les demandes d'interruption qui entrent dans le groupe de lignes de demande d'interruption qui lui sont assignées (la plus élevée des IRQ1* IRQ7*).
- b) Il utilise son DEMANDEUR sur sa carte pour demander le DTB et, quand l'utilisation du DTB lui est accordée, il lance un cycle de reconnaissance d'interruption lisant un MOT D'ETAT/IDentificateur provenant du GENERATEUR D'INTERRUPTION qui a été reconnu.
- c) Il lance la séquence appropriée de traitement d'interruption établie sur l'information reçue dans le MOT D'ETAT/IDentificateur.

OBSERVATION 4.3:

La norme du BUS CEI 821 ne précise pas ce qui arrivera pendant la séquence de traitement d'interruption. Le traitement de l'interruption peut ou non impliquer l'usage du BUS CEI 821.

Le CONTROLEUR D'INTERRUPTION utilise le DTB pour lire un MOT D'ETAT/IDentificateur du GENERATEUR D'INTERRUPTION. Dans ce cas, le CONTROLEUR D'INTERRUPTION agit comme un MAITRE et le GENERATEUR D'INTERRUPTION agit comme un ESCLAVE. Cependant, il y a quatre différences importantes. Le CONTROLEUR D'INTERRUPTION:

- a) commande toujours IACK* au niveau bas;
- b) n'est pas tenu de commander les lignes de modification d'adresse;
- c) utilise seulement les trois lignes d'adresse les plus basses (A01-A03);
- d) ne commande jamais le bus de données.

Le CONTROLEUR D'INTERRUPTION commande toujours IACK* au niveau bas quand il accède au bus. Le MAITRE le commande au niveau haut ou pas du tout.

RULE 4.2:

Output signal lines shown with solid lines in Figures 4-5 through 4-7, pages 251 to 263, **MUST** be driven by the module, unless it would always drive them high.

OBSERVATION 4.1:

IF an output line is not driven,
THEN terminators on the backplane ensure that it is high.

RULE 4.3:

Input signal lines shown with solid lines in Figures 4-5 through 4-7 **MUST** be monitored and responded to in the appropriate fashion.

OBSERVATION 4.2:

RULES and PERMISSIONS for driving and monitoring lines shown with dotted lines in Figures 4-5 through 4-7 are given in Tables 4-1 and 4-2.

4.3.1 INTERRUPT HANDLERS

The INTERRUPT HANDLER is used to accomplish several tasks:

- a) It prioritizes the incoming interrupt requests within its assigned group of interrupt request lines (highest of IRQ1*-IRQ7*)
- b) It uses its on-board REQUESTER to request the DTB and, when granted use of the DTB, initiates an interrupt acknowledge cycle, reading a STATUS/ID from the INTERRUPTER being acknowledged.
- c) It initiates the appropriate interrupt servicing sequence, based on the information received in the STATUS/ID.

OBSERVATION 4.3:

The IEC 821 BUS standard does not dictate what will happen during the interrupt servicing sequence. Servicing of the interrupt might or might not involve use of the IEC 821 BUS.

The INTERRUPT HANDLER uses the DTB to read a STATUS/ID from the INTERRUPTER. In this respect, the INTERRUPT HANDLER acts like a MASTER and the INTERRUPTER acts like a SLAVE. However, there are four important differences. The INTERRUPT HANDLER:

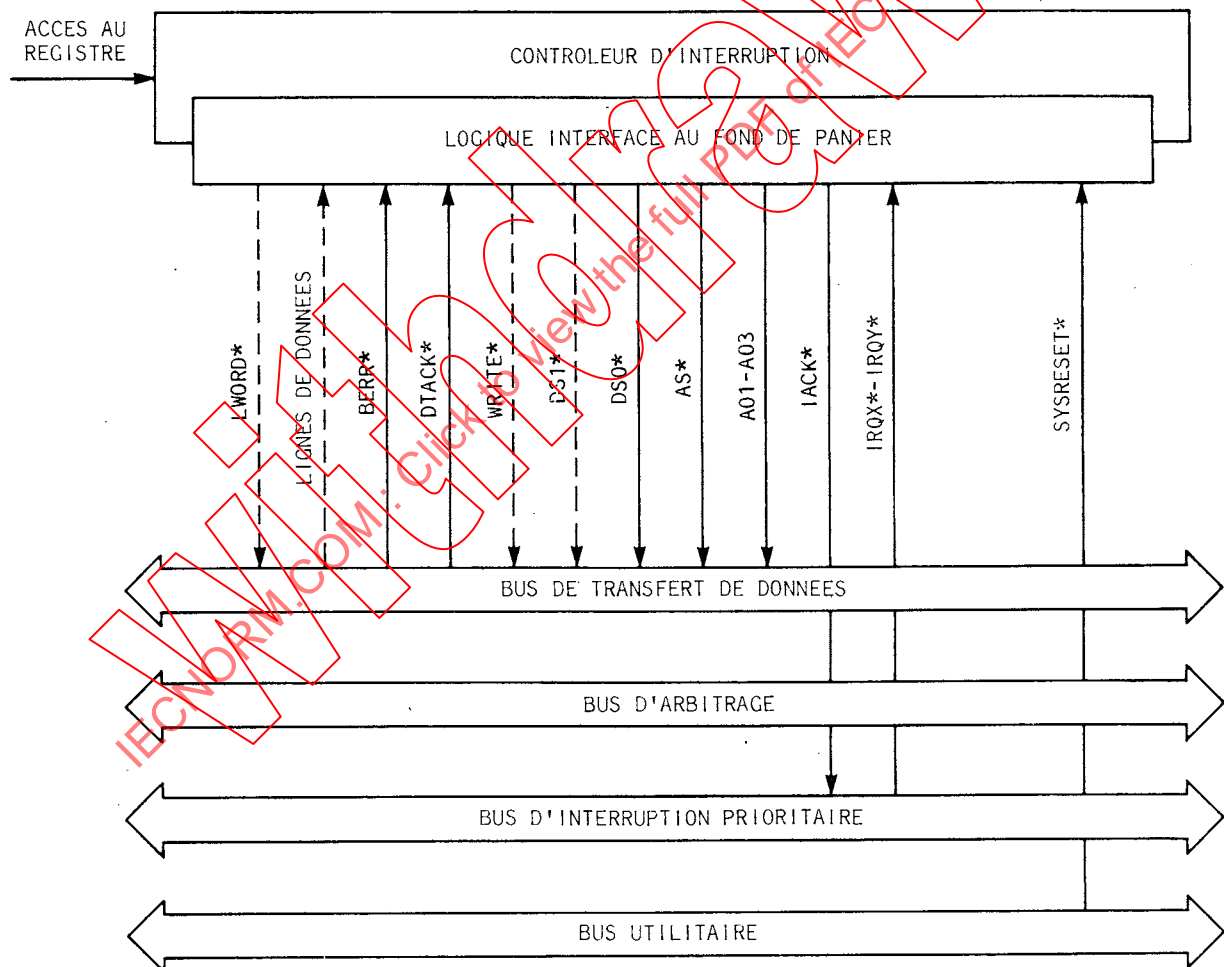
- a) always drives IACK* low;
- b) is not required to drive the address modifier lines;
- c) only uses the lowest three address lines (A01-A03);
- d) never drives the data bus.

The INTERRUPT HANDLER always drives IACK* low when it accesses the bus. The MASTER either drives it high or does not drive it at all.

Le CONTROLEUR D'INTERRUPTION n'a pas à commander les lignes de modification d'adresse avec un code valide et il commande seulement les trois lignes d'adresse les plus basses (A01-A03) avec une information valide. Les niveaux de ces trois lignes d'adresse indiquent laquelle des sept lignes de demande d'interruption a été reconnue comme indiqué dans le tableau 4-7. Un MAITRE commande 15, 23 ou 31 lignes d'adresse (en fonction du mode d'adressage) avec l'adresse de l'ESCLAVE à accéder et fournit un code modificateur d'adresse sur les lignes de modification d'adresse.

Le CONTROLEUR D'INTERRUPTION ne commande pas les lignes de données (c'est-à-dire il "n'écrit" pas vers le GENERATEUR D'INTERRUPTION) et n'a pas à commander la ligne WRITE*. Un MAITRE utilise les lignes de données d'un ESCLAVE de manière bidirectionnelle et, pendant l'utilisation normale, commande WRITE* au niveau bas ou haut, comme nécessaire.

Un schéma-bloc du CONTROLEUR D'INTERRUPTION est représenté à la figure 4-5.



Note:

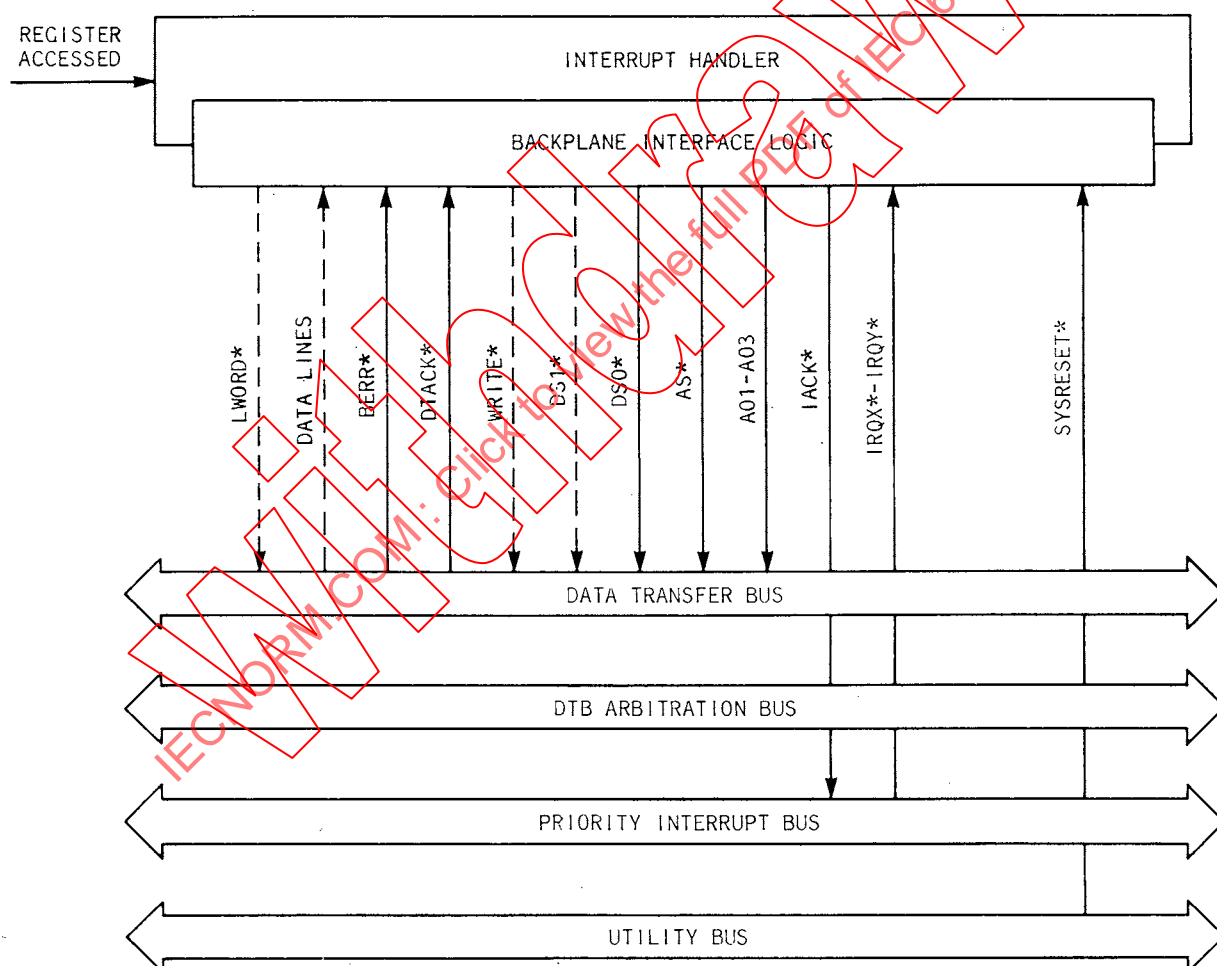
Les REGLES et OBSERVATIONS pour la surveillance et la commande des lignes pointillées sont données dans le tableau 4-1.

FIG. 4-5. - Schéma-bloc: CONTROLEUR D'INTERRUPTION.

The INTERRUPT HANDLER does not have to drive the address modifier lines with a valid code, and it only drives the lowest three address lines (A01-A03) with valid information. The levels of these three address lines indicate which of the seven interrupt request lines is being acknowledged, as shown in Table 4-7. A MASTER drives 15, 23, or 31 address lines (depending on the addressing mode) with the address of the SLAVE being accessed, and provides an address modifier code on the address modifier lines.

The INTERRUPT HANDLER does not drive the data lines (i.e. it does not "write" to the INTERRUPTER) and does not have to drive the WRITE* line. A MASTER uses the data lines to a SLAVE bidirectionally and, during normal use, drives WRITE* low or high as required.

A block diagram of the INTERRUPT HANDLER is shown in Figure 4-5.



Note:

The RULES and PERMISSIONS for monitoring and driving the dotted lines are given in Table 4-1.

FIG. 4-5. - Block diagram: INTERRUPT HANDLER.

Tableau 4-1

REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de
CONTROLEURS D'INTERRUPTION

Type de CONTROLEUR D'INTERRUPTION	Utilisation des lignes pointillées
D08(0)	DOIT surveiller D00-D07. PEUT ou NON commander LWORD* et DS1*. PEUT ou NON surveiller D08-D31
D16	DOIT commander DS1*. DOIT surveiller D00-D15. PEUT ou NON commander LWORD*. PEUT ou NON surveiller D16-D31
D32	DOIT commander DS1* et LWORD*. DOIT surveiller D00-D31
TOUS	NE DOIVENT PAS commander WRITE* au niveau bas

Note:

Les mnémoniques D08(0), D16 et D32 sont définis dans le tableau 4-5.

4.3.2 GENERATEUR D'INTERRUPTION

Le GENERATEUR D'INTERRUPTION fonctionne de la manière suivante:

- Il demande une interruption au CONTROLEUR D'INTERRUPTION qui surveille sa ligne de demande d'interruption.
- S'il reçoit un front descendant sur l'entrée de la chaîne série de reconnaissance d'interruption.

Alors s'il demande une interruption
 et que les niveaux sur les trois lignes valides d'adresse correspondent à la ligne de demande d'interruption qu'il utilise
 et que la taille du MOT D'ETAT/IDentificateur requis est égale ou supérieure à celle qu'il utilise,
 Alors il fournit un MOT D'ETAT/IDentificateur.

Sinon il transmet le front descendant en aval sur la chaîne série de reconnaissance d'interruption.

Chaque module GENERATEUR D'INTERRUPTION commande une seule ligne de demande d'interruption. La norme du BUS CEI 821 considère une carte qui génère des demandes d'interruption sur plusieurs lignes d'interruption comme possédant plusieurs modules GENERATEURS D'INTERRUPTION.

Table 4-1

RULES and PERMISSIONS that specify the use of the dotted lines by the various types of INTERRUPT HANDLERS

Type of INTERRUPT HANDLER	Use of dotted lines
D08(0)	MUST monitor D00-D07. MAY or MAY NOT drive LWORD* and DS1*. MAY or MAY NOT monitor D08-D31
D16	MUST drive DS1*. MUST monitor D00-D15. MAY or MAY NOT drive LWORD*. MAY or MAY NOT monitor D16-D31
D32	MUST drive DS1* and LWORD*. MUST monitor D00-D31
ALL	MUST NOT drive WRITE* low

Note:

The mnemonics D08(0), D16, and D32 are defined in Table 4-5.

4.3.2 INTERRUPTER

The INTERRUPTER functions as follows:

- It requests an interrupt from the INTERRUPT HANDLER which monitors its interrupt request line.
- If it receives a falling edge in the interrupt acknowledge daisy-chain input.

Then, if it is requesting an interrupt
and the levels on the three valid address lines correspond to the interrupt request line it is using

and the width of the requested STATUS/ID is either equal to, or greater than the size it is using,
Then it supplies a STATUS/ID.

Else it passes the falling edge down the interrupt acknowledge daisy-chain.

Each INTERRUPTER module drives only one interrupt request line. The IEC 821 BUS standard describes a board that generates interrupt requests on several interrupt lines as having several INTERRUPTER modules.

AUTORISATION 4.1:

Puisque le GENERATEUR D'INTERRUPTION est seulement un modèle conceptuel, la logique sur une carte de BUS CEI 821 PEUT être partagée entre plusieurs modules GENERATEURS D'INTERRUPTION.

Le GENERATEUR D'INTERRUPTION utilise l'une des sept lignes pour demander une interruption. Il surveille alors les trois lignes les plus basses du bus d'adresse (A01-A03), la chaîne série IACKIN*/IACKOUT* et optionnellement IACK* pour déterminer quand son interruption a été reconnue. Quand elle est reconnue, il place son MOT D'ETAT/IDentificateur sur le bus de données et signale au CONTROLEUR D'INTERRUPTION que le MOT D'ETAT/IDentificateur est valide en commandant DTACK* au niveau bas.

Il y a cinq différences principales dans l'utilisation du DTB par le GENERATEUR D'INTERRUPTION et par l'ESCLAVE. Le GENERATEUR D'INTERRUPTION:

- a) répond seulement quand son signal IACKIN* est au niveau bas;
- b) n'a pas à surveiller les lignes de modification d'adresse;
- c) surveille seulement les trois lignes d'adresse les plus basses;
- d) ne surveille pas la ligne WRITE*;
- e) est autorisé à répondre avec des données d'une taille différente à celle demandée.

L'ESCLAVE surveille AS* et interprète un front descendant sur AS* comme le moment indiquant qu'un cycle de bus valide est en cours. Il procède alors au décodage du nombre approprié de lignes d'adresse (15, 23 ou 31) et des lignes de modification d'adresse et, à partir de cette information, détermine s'il était sélectionné. Cependant, l'ESCLAVE répond seulement si IACK* est au niveau haut.

Le GENERATEUR D'INTERRUPTION interprète d'autre part le front descendant de sa ligne IACKIN* comme un signal qu'il peut répondre au cycle de reconnaissance d'interruption en cours. Il décode seulement les trois lignes d'adresse les plus basses (A01-A03) ignorant les lignes de modification d'adresse.

Le GENERATEUR D'INTERRUPTION n'est pas tenu de surveiller la ligne WRITE* puisqu'il n'y a jamais d'écriture vers lui. Les ESCLAVES doivent surveiller la ligne WRITE* pour faire une distinction entre les opérations de lecture et d'écriture.

Le GENERATEUR D'INTERRUPTION place son MOT D'ETAT/IDentificateur sur le bus et répond avec DTACK* même si les lignes LWORD*, DS1* et DS0* demandent un MOT D'ETAT/IDentificateur dont la taille est supérieure à celle que le GENERATEUR D'INTERRUPTION peut fournir. Par exemple, le CONTROLEUR D'INTERRUPTION peut commander LWORD* et les deux validations de données au niveau bas, indiquant qu'il lira un MOT D'ETAT/IDentificateur de 32 bits sur D00-D31, mais un GENERATEUR D'INTERRUPTION D08(O) répondrait encore avec son MOT D'ETAT/IDentificateur de 8 bits sur D00-D07. Par contre, lorsqu'un ESCLAVE ne peut pas présenter la taille demandée de la donnée, il répond en générant BERR* ou ne répond pas du tout, ce qui typiquement génère un signal de limitation de temps d'occupation du bus.

PERMISSION 4.1:

Since the INTERRUPTER is just a conceptual model, logic on an IEC 821 BUS board MAY be shared between several INTERRUPTER modules.

The INTERRUPTER uses one of seven lines to request an interrupt. It then monitors the lowest three lines of the address bus (A01-A03), the IACKIN*/IACKOUT* daisy-chain, and optionally IACK*, to determine when its interrupt is being acknowledged. When acknowledged, it places its STATUS/ID on the data bus and signals the INTERRUPT HANDLER that the STATUS/ID is valid by driving DTACK* low.

There are five primary differences in the use of the DTB by the INTERRUPTER and the SLAVE. The INTERRUPTER:

- a) only responds when its IACKIN* is low;
- b) does not have to monitor the address modifier lines;
- c) only monitors the lowest three address lines;
- d) does not monitor the WRITE* line;
- e) is permitted to respond with data of a different size than that requested.

The SLAVE monitors AS*, and interprets a falling edge on AS* as the signal that a valid bus cycle is in progress. It then proceeds to decode the appropriate number of address lines (15, 23, or 31), and the address modifier lines, and based on this information it determines whether it was addressed. However the SLAVE responds only if IACK* is high.

The INTERRUPTER on the other hand, interprets the falling edge on its IACKIN* line as a signal that it can respond to the interrupt acknowledge cycle in progress. It decodes only the lowest three address lines (A01-A03), ignoring the address modifier lines.

The INTERRUPTER does not need to monitor WRITE*, since it is never written to. SLAVES need to monitor WRITE* so that they can distinguish read cycles from write cycles.

The INTERRUPTER places a STATUS/ID on the bus, and responds with DTACK*, even if the LWORD*, DS1*, and DS0* lines call for a STATUS/ID whose width is greater than the INTERRUPTER is able to provide. For example, the INTERRUPT HANDLER might drive LWORD* and both data strobes low, indicating that it will read 32 bits of STATUS/ID from D00-D31, but a D08(O) INTERRUPTER would still respond with its 8-bit STATUS/ID on D00-D07. In contrast, when a SLAVE cannot provide the requested data width, it either responds with BERR* or does not respond at all, typically resulting in a bus time-out.

OBSERVATION 4.4:

Lorsque le GENERATEUR D'INTERRUPTION place un MOT D'ETAT/IDentificateur sur le bus de données, toutes les lignes de données qui ne sont pas commandées sont lues par le CONTROLEUR D'INTERRUPTION comme étant au niveau haut à cause des adaptations du bus. Par exemple, si un GENERATEUR D'INTERRUPTION D16 lance un cycle de reconnaissance d'interruption double octet, un GENERATEUR D'INTERRUPTION D08(O) placera un MOT D'ETAT/IDentificateur 8 bits sur D00-D07. Les 8 bits de rang plus élevé, lus par le CONTROLEUR D'INTERRUPTION de D08-D15, sont lus au niveau 1 (haut), car ils ne sont pas commandés par le GENERATEUR D'INTERRUPTION D08(O).

REGLE 4.4:

Avant de répondre à un cycle de reconnaissance d'interruption, le GENERATEUR D'INTERRUPTION:

- 1) DOIT avoir une demande d'interruption en attente.
- 2) Le niveau de cette demande DOIT correspondre au niveau indiqué sur A01-A03.
- 3) La taille du MOT D'ETAT/IDentificateur requis DOIT être égale ou plus grande que la taille avec laquelle il peut répondre.
- 4) Il DOIT avoir reçu un front descendant sur son entrée IACKIN* de la chaîne série.

SI une de ces quatre conditions n'est pas satisfaite,
ALORS le GENERATEUR D'INTERRUPTION ne DOIT PAS répondre au cycle de reconnaissance d'interruption.

SI la condition 4) est satisfaite mais si 1), 2) ou 3) ne l'est pas,
ALORS Le GENERATEUR D'INTERRUPTION DOIT propager le front descendant de IACKIN* vers le module GENERATEUR D'INTERRUPTION suivant dans la chaîne série.

Un schéma-bloc du GENERATEUR D'INTERRUPTION est indiqué dans la figure 4-6, page 258.

OBSERVATION 4.4:

When an INTERRUPTER places a STATUS/ID on the data bus, any undriven data lines are read by the INTERRUPT HANDLER as high because of the bus terminators. For example, if a D16 INTERRUPT HANDLER initiates a double byte interrupt acknowledge cycle, a D08(O) INTERRUPTER would place an 8-bit STATUS/ID on D00-D07. The upper 8 bits, read by the INTERRUPT HANDLER from D08-D15, are read as ones (high), since they are not driven by the D08(O) INTERRUPTER.

RULE 4.4:

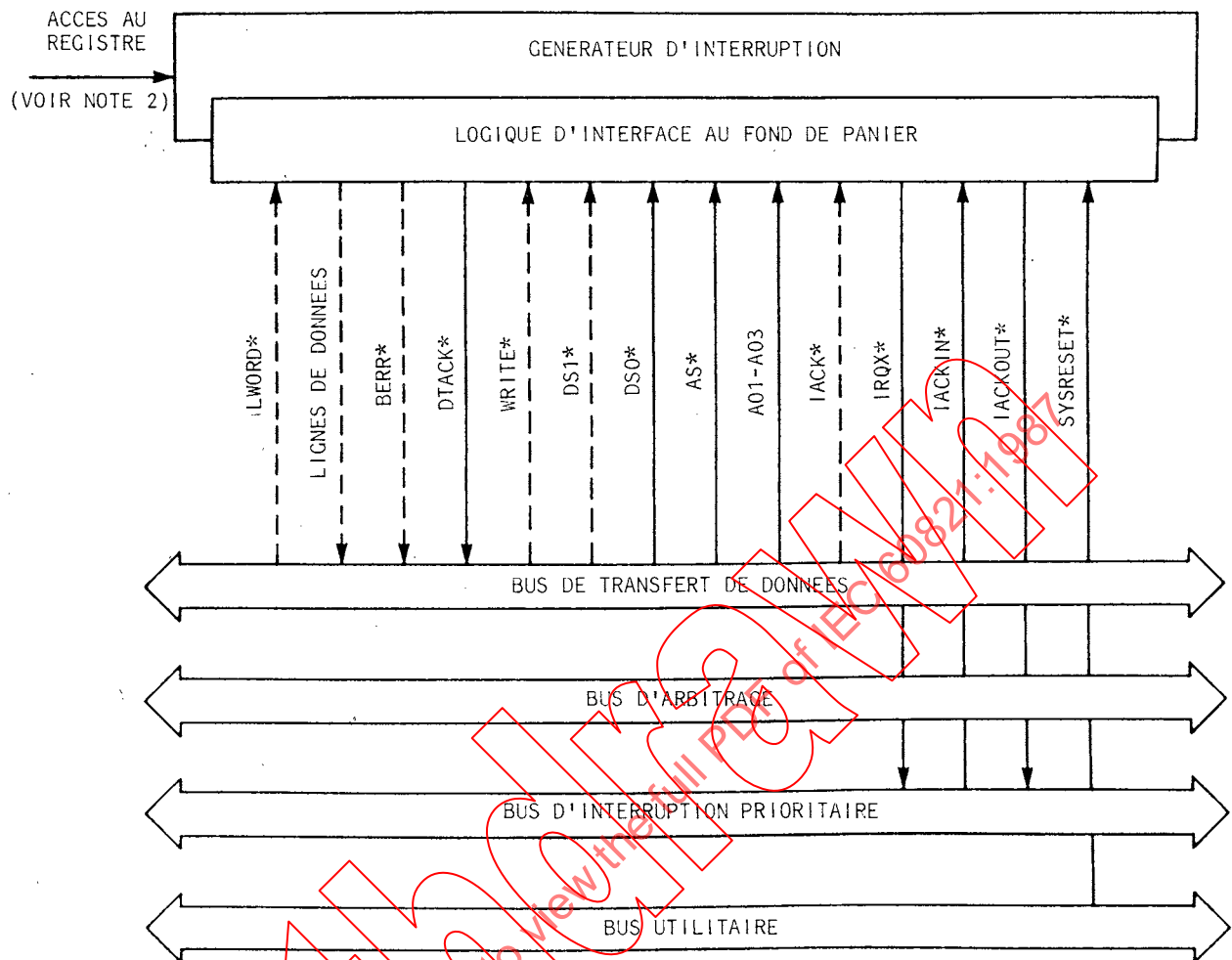
Before responding to an interrupt acknowledge cycle, the INTERRUPTER:

- 1) MUST have an interrupt request pending.
- 2) The level of that request MUST match the level indicated on A01-A03.
- 3) The width of the requested STATUS/ID MUST be equal to or greater than the size it can respond with.
- 4) It MUST have received an incoming falling edge on its IACKIN* daisy-chain input.

IF any one of these four conditions is not met,
THEN the INTERRUPTER MUST NOT respond to the interrupt acknowledge cycle.

IF condition 4) is met, but either 1), 2), or 3) is not,
THEN the INTERRUPTER MUST pass the falling edge of IACKIN* to the next INTERRUPTER module in the daisy-chain.

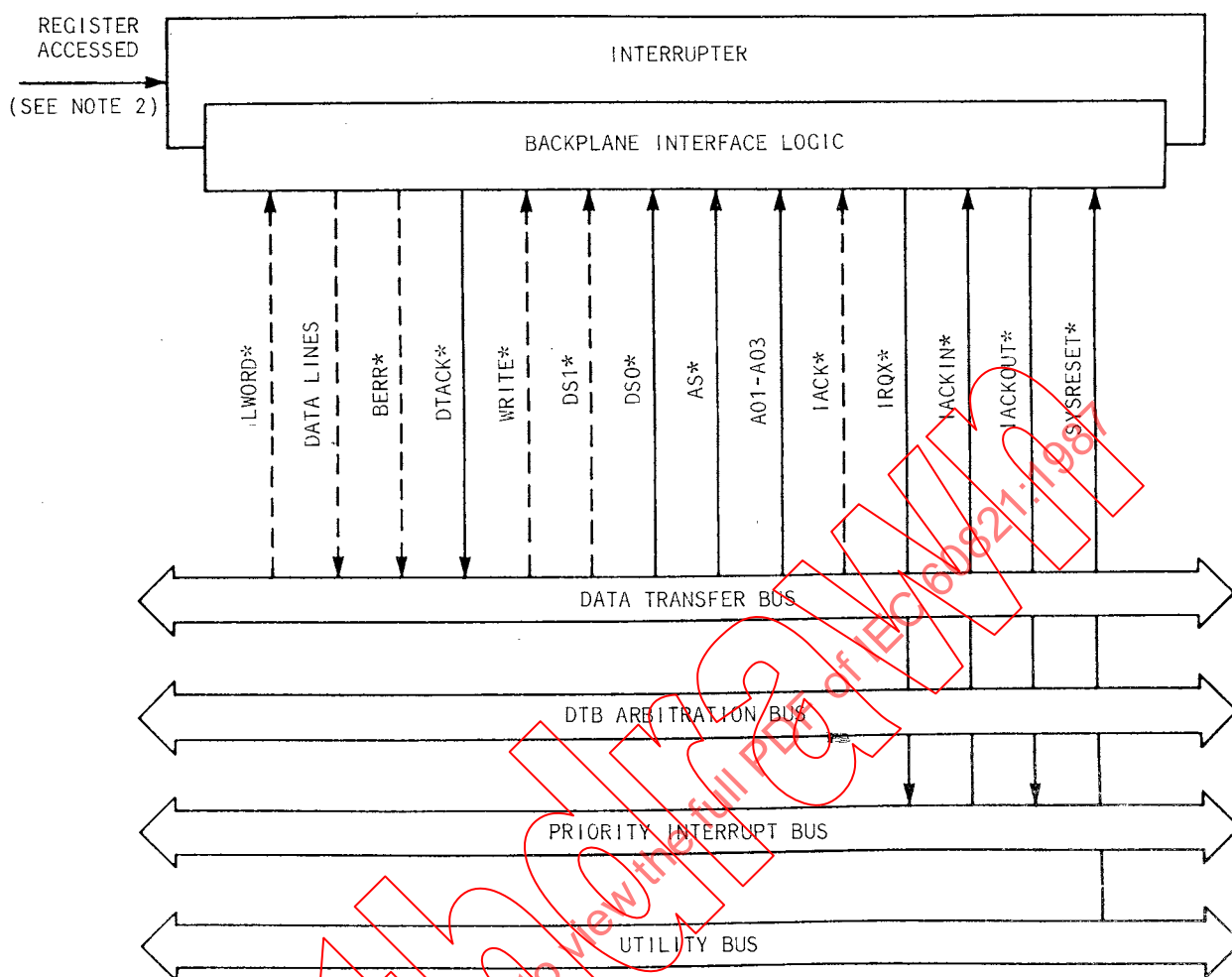
A block diagram of the INTERRUPTER is shown in Figure 4-6, page 259.



Notes:

- 1.- Les REGLES et OBSERVATIONS pour commander et contrôler les lignes pointillées sont données dans le tableau 4-2.
- 2.- Ce signal d'entrée est présent seulement sur les GENERATEURS D'INTERRUPTION RORA.

FIG. 4-6. - Schéma-bloc: GENERATEUR D'INTERRUPTION.



Notes:

- 1.- The RULES and PERMISSIONS for driving and monitoring the dotted lines are given in Table 4-2.
- 2.- This input signal is present on RORA interrupters only.

FIG. 4-6. - Block diagram: INTERRUPTER.

Tableau 4-2

REGLES et AUTORISATIONS qui spécifient l'utilisation des lignes pointillées par les différents types de GENERATEURS D'INTERRUPTION

Type de GENERATEUR D'INTERRUPTION	Utilisation des lignes pointillées
D08(0)	DOIT commander D00-D07. NE DOIT PAS commander D08-D31 au niveau bas. N'a aucune raison de surveiller LWORD* ou DS1*
D16	DOIT surveiller DS1*. DOIT commander D00-D15. NE DOIT PAS commander D16-D31 au niveau bas. N'a aucune raison de surveiller LWORD*
D32	DOIT surveiller DS1* et LWORD*. DOIT commander D00-D31
TOUS	PEUVENT ou NON surveiller WRITE* et IACK*. PEUVENT ou NON commander BERR*

Note:

Les mnémoniques D08(0), D16 et D32 sont définis dans le tableau 4-5.

4.3.3 EMETTEUR DE CHAÎNE SERIE IACK

L'EMETTEUR DE CHAÎNE SERIE IACK est un autre module qui interagit avec les CONTROLEURS D'INTERRUPTION et les GENERATEURS D'INTERRUPTION pour coordonner le traitement des interruptions. Il génère un front descendant sur la chaîne série de reconnaissance d'interruption chaque fois qu'un CONTROLEUR D'INTERRUPTION lance un cycle de reconnaissance d'interruption.

Un schéma-bloc de l'EMETTEUR DE CHAÎNE SERIE IACK est donné dans la figure 4-7, page 262.

Table 4-2

RULES and PERMISSIONS that specify the use of the dotted lines by the various types of INTERRUPTERS

Type of INTERRUPTER	Use of dotted lines
D08(0)	MUST drive D00-D07. MUST not drive D08-D31 low. Has no reason to monitor LWORD* or DS1*
D16	MUST monitor DS1*. MUST drive D00-D15. MUST NOT drive D16-D31 low. Has no reason to monitor LWORD*
D32	MUST monitor DS1* and LWORD*. MUST drive D00-D31
ALL	MAY or MAY NOT monitor WRITE* and JACK*. MAY or MAY NOT drive BERR*

Note:

The mnemonics D08(0), D16, and D32 are defined in Table 4-5.

4.3.3 IACK DAISY-CHAIN DRIVER

The IACK DAISY-CHAIN DRIVER is another module that interacts with INTERRUPT HANDLERS and INTERRUPTERS to coordinate the servicing of interrupts. It generates a falling edge on the interrupt acknowledge daisy-chain each time an INTERRUPT HANDLER initiates an interrupt acknowledge cycle.

A block diagram of the IACK DAISY-CHAIN DRIVER is given in Figure 4-7, page 263.

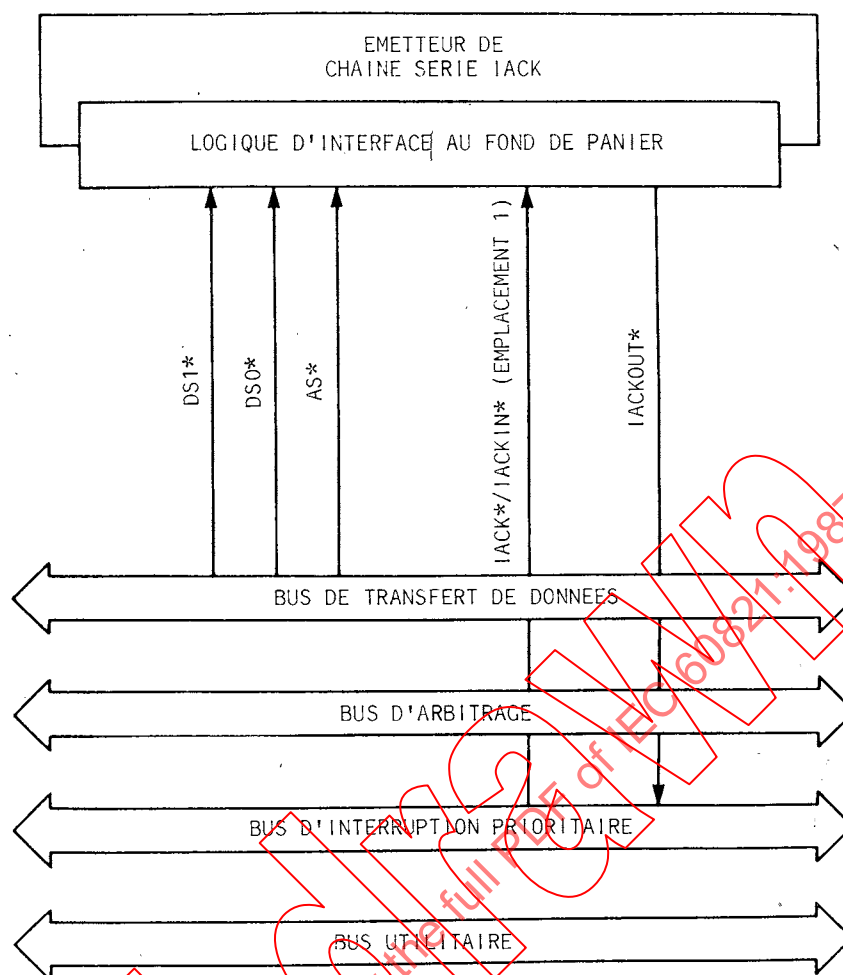


FIG. 4-7. - Schéma-bloc: EMETTEUR DE CHAÎNE SÉRIE IACK.

4.3.4 Possibilités de prise en compte d'une demande d'interruption

Les CONTRÔLEURS D'INTERRUPTION peuvent être conçus pour prendre en compte des demandes d'interruption reçues sur l'une des sept lignes de demande d'interruption. Le tableau 4-3 montre comment le mnémonique IH() est employé pour décrire les possibilités de prise en compte de l'interruption des CONTRÔLEURS D'INTERRUPTION.

Tableau 4-3

Utilisation du mnémonique IH() pour spécifier les possibilités de traitement d'interruption

Le mnémonique suivant	Quand appliqué à un	Signifie qu'il
IH(x-y)	CONTRÔLEUR D'INTERRUPTION	Peut générer des cycles de reconnaissance d'interruption en réponse à des demandes d'interruption sur les lignes IRQx* à IRQy*
IH(x)	CONTRÔLEUR D'INTERRUPTION	Peut générer des cycles de reconnaissance d'interruption en réponse à des demandes d'interruption sur les lignes IRQx*

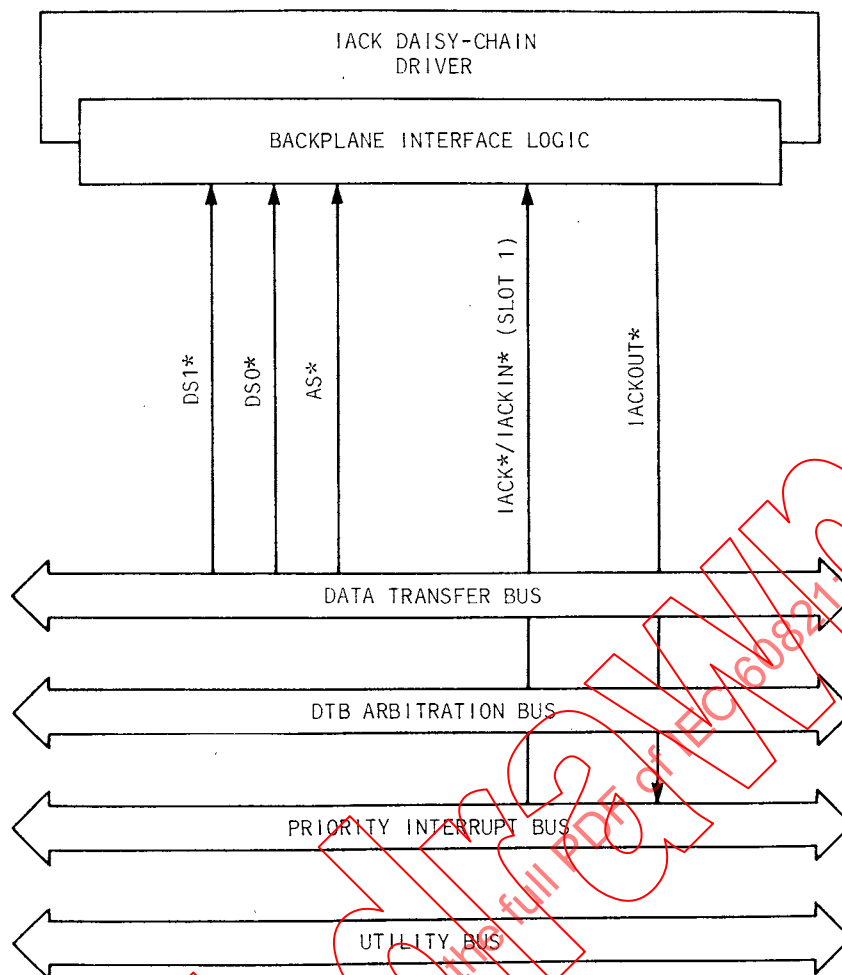


FIG. 4-7. - Block diagram: JACK DAISY-CHAIN DRIVER.

4.3.4 Interrupt request handling capabilities

INTERRUPT HANDLERS can be designed to handle interrupt requests received on one to seven interrupt request lines. Table 4-3 shows how the IH() mnemonic is used to describe the interrupt handling capabilities of INTERRUPT HANDLERS.

Table 4-3

Use of the IH() mnemonic to specify interrupt handling capabilities

The following mnemonic	When applied to an	Means that it
IH(x-y)	INTERRUPT HANDLER	Can generate interrupt acknowledge cycles in response to interrupt requests on lines IRQx* through IRQy*
IH(x)	INTERRUPT HANDLER	Can generate interrupt acknowledge cycles in response to interrupt requests on line IRQx*

4.3.5 Possibilités de génération d'une demande d'interruption

Les GENERATEURS D'INTERRUPTION peuvent être conçus pour générer une demande d'interruption sur l'une des sept lignes de demande d'interruption. Le tableau 4-4 montre comment le mnémonique I() est employé pour décrire les possibilités de génération de demande d'interruption des GENERATEURS D'INTERRUPTION.

Tableau 4-4

Utilisation du mnémonique I() pour spécifier les possibilités de génération d'une demande d'interruption

Le mnémonique suivant	Quand appliqué à un	Signifie qu'il
I(x)	GENERATEUR D'INTERRUPTION	Peut générer une demande d'interruption sur la ligne IRQx*

4.3.6 Possibilités de transferts du MOT D'ETAT/IDentificateur

Il y a trois possibilités de transferts du MOT D'ETAT/IDentificateur: D08(O), D16 et D32. Le tableau 4-5 montre comment ces mnémoniques sont utilisés pour décrire les possibilités de prise en compte des interruptions des GENERATEURS D'INTERRUPTION et CONTROLEURS D'INTERRUPTION.

Tableau 4-5

Mnémoniques qui spécifient les possibilités de transfert du MOT D'ETAT/IDentificateur

Le mnémonique suivant	Quand appliqué à un	Signifie qu'il
D08(O)	GENERATEUR D'INTERRUPTION	Répond à des cycles de reconnaissance d'interruption 8, 16 et 32 bits en fournissant un MOT D'ETAT IDentificateur 8 bits sur D00-D07.
	CONTROLEUR D'INTERRUPTION	Génère des cycles de reconnaissance d'interruption 8 bits en réponse à des demandes sur la ou les lignes de demande d'interruption et lit un MOT D'ETAT/IDentificateur 8 bits de D00-D07
D16	GENERATEUR D'INTERRUPTION	Répond à des cycles de reconnaissance d'interruption 16 et 32 bits en fournissant un MOT D'ETAT/IDentificateur 16 bits sur D00-D15.
	CONTROLEUR D'INTERRUPTION	Génère des cycles de reconnaissance d'interruption 16 bits en réponse à des demandes sur la ou les lignes de demande d'interruption et lit un MOT D'ETAT/IDentificateur 16 bits de D00-D15

4.3.5 Interrupt request generation capabilities

INTERRUPTERS can be designed to generate an interrupt request on any of the seven interrupt request lines. Table 4-4 shows how the I() mnemonic is used to describe the interrupt request generation capabilities of INTERRUPTERS.

Table 4-4

Use of the I() mnemonic to specify interrupt request generation capabilities

The following mnemonic	When applied to an	Means that it
I(x)	INTERRUPTER	Can generate interrupt request on line IRQx*

4.3.6 STATUS/ID transfer capabilities

There are three STATUS/ID transfer capabilities: D08(O), D16 and D32. Table 4-5 shows how these mnemonics are used to describe the interrupt handling capabilities of INTERRUPT HANDLERS and INTERRUPTERS.

Table 4-5

Mnemonics that specify STATUS/ID transfer capabilities

The following mnemonic	When applied to an	Means that it
D08(O)	INTERRUPTER	Responds to 8-bit, 16-bit, and 32-bit interrupt acknowledge cycles by providing an 8-bit STATUS/ID on D00-D07.
	INTERRUPT HANDLER	Generates 8-bit interrupt acknowledge cycles in response to the requests on the interrupt request line(s) and reads an 8-bit STATUS/ID from D00-D07
D16	INTERRUPTER	Responds to 16-bit and 32-bit interrupt acknowledge cycles by providing a 16-bit STATUS/ID on D00-D15.
	INTERRUPT HANDLER	Generates 16-bit interrupt acknowledge cycles in response to the requests on the interrupt request line(s) and reads a 16-bit STATUS/ID from D00-D15

Tableau 4-5 (suite)

Le mnémonique suivant	Quand appliqué à un	Signifie qu'il
D32	GENERATEUR D'INTERRUPTION	Répond à des cycles de reconnaissance d'interruption 32 bits en fournissant un MOT D'ETAT/IDentificateur 32 bits sur D00-D31.
	CONTROLEUR D'INTERRUPTION	Génère des cycles de reconnaissance d'interruption 32 bits en réponse à des demandes sur la ou les lignes de demande d'interruption et lit un MOT D'ETAT/IDentificateur 32 bits de D00-D31

4.3.7 Possibilités de libération de la demande d'interruption

De nombreux circuits intégrés de contrôleurs de périphériques génèrent des demandes d'interruption. Malheureusement, il n'y a aucune méthode normalisée pour indiquer à ces circuits à quel moment ils doivent retirer leur demande d'interruption. Trois méthodes sont utilisées:

- Lorsque le processeur concerné détecte une demande d'interruption provenant d'un dispositif périphérique, il entre dans une routine de traitement de l'interruption et LIT un registre d'état du périphérique. Ce cycle de lecture du registre d'état est considéré, par le dispositif périphérique, comme un signal pour retirer sa demande d'interruption.
- Lorsque le processeur concerné détecte une demande d'interruption provenant d'un dispositif périphérique, il entre dans une routine de traitement de l'interruption et ECRIT dans un registre de contrôle du périphérique. Ce cycle d'écriture dans le registre de contrôle est considéré, par le dispositif périphérique comme un signal pour retirer sa demande d'interruption.
- Lorsque le processeur concerné détecte une demande d'interruption provenant d'un dispositif périphérique, il lit un MOT D'ETAT/IDentificateur du dispositif périphérique. Ce cycle de lecture est considéré, par le dispositif périphérique, comme un signal pour retirer sa demande d'interruption.

La norme du BUS CEI 821 désigne les GENERATEURS D'INTERRUPTION qui utilisent les méthodes a) et b) par GENERATEURS D'INTERRUPTION à libération lors de l'accès à un registre (RORA) et ceux qui utilisent la méthode c) par GENERATEURS D'INTERRUPTION à libération lors de la reconnaissance de l'interruption (ROAK). La figure 4-8, page 270, montre comment un GENERATEUR D'INTERRUPTION ROAK libère sa ligne de demande d'interruption quand le CONTROLEUR D'INTERRUPTION lit le MOT D'ETAT/IDentificateur, tandis qu'un GENERATEUR D'INTERRUPTION RORA libère sa ligne de demande d'interruption lorsque le sous-programme de traitement de l'interruption accède à un registre de contrôle ou d'état.

OBSERVATION 4.5:

L'ESCLAVE qui a fourni l'accès au registre de contrôle ou d'état du GENERATEUR D'INTERRUPTION se trouve généralement sur la même carte que le GENERATEUR D'INTERRUPTION, et lui envoie un signal sur la carte après avoir accédé aux registres.

Table 4-5 (continued)

The following mnemonic	When applied to an	Means that it
D32	INTERRUPTER	Responds to 32-bit interrupt acknowledge cycles by providing a 32-bit STATUS/ID on D00-D31.
	INTERRUPT HANDLER	Generates 32-bit interrupt acknowledge cycles in response to the requests on the interrupt request line(s) and reads a 32-bit STATUS/ID from D00-D31

4.3.7 Interrupt request release capabilities

Many widely used peripheral ICs generate interrupt requests. Unfortunately, there is no standard method for indicating to these ICs when it is time for them to remove their interrupt request from the bus. Three methods are used:

- When the relevant processor senses an interrupt request from a peripheral device, it enters an interrupt service routine and READS a status register in the device. The peripheral device interprets this read cycle on its status register as a signal to remove its interrupt request.
- When the relevant processor senses an interrupt request from a peripheral device, it enters an interrupt service routine and WRITES to a control register in the device. The peripheral device interprets this write cycle on its control register as a signal to remove its interrupt request.
- When the relevant processor senses an interrupt request from a peripheral device, it reads a STATUS/ID from the device. The peripheral device interprets this read cycle as a signal to remove its interrupt request.

The IEC 821 BUS standard calls INTERRUPTERS that use methods *a)* and *b)* Release On Register Access (RORA) INTERRUPTERS, and those that use method *c)* Release On Acknowledge (ROAK) INTERRUPTERS. Figure 4-8, page 271, shows how an ROAK INTERRUPTER releases its interrupt request line when the INTERRUPT HANDLER reads its STATUS/ID and how an RORA INTERRUPTER releases its interrupt request upon an access to a control or status register.

OBSERVATION 4.5:

The SLAVE that provided the access to the INTERRUPTER's control or status register is typically on the same board as the INTERRUPTER, and it generates an on-board signal to the INTERRUPTER when it has completed the register access.

REGLE 4.5:

Un GENERATEUR D'INTERRUPTION RORA ne DOIT PAS libérer sa ligne de demande d'interruption avant que DSA* soit descendu au niveau bas pendant le cycle d'accès au registre. Il DOIT libérer la ligne de demande d'interruption dans les 2 μ s après que le dernier signal de validation de donnée fut passé au niveau haut à la fin du cycle d'accès au registre.

REGLE 4.6:

Un GENERATEUR D'INTERRUPTION ROAK ne DOIT PAS libérer sa ligne de demande d'interruption avant que DSA* soit descendu au niveau bas pendant le cycle de reconnaissance qui acquitte son interruption, et DOIT libérer la ligne de demande d'interruption dans les 500 ns après que le dernier signal de validation de donnée fut passé au niveau haut à la fin du cycle lecture du MOT D'ETAT/IDentificateur.

REGLE 4.7:

Les deux GENERATEURS D'INTERRUPTION RORA et ROAK DOIVENT fournir un MOT D'ETAT/IDentificateur lors du cycle de reconnaissance d'interruption qui a été déclenché suite à leur demande d'interruption.

REGLE 4.8:

Après qu'un CONTROLEUR D'INTERRUPTION eut déclenché un cycle de reconnaissance d'interruption et lu le MOT D'ETAT/IDentificateur d'un GENERATEUR D'INTERRUPTION RORA, il DOIT ignorer le niveau bas sur la ligne de demande d'interruption pendant 2 μ s après que le signal REGISTRE ACCEDE sur sa carte fut passé à l'état vrai.

OBSERVATION 4.6:

La REGLE 4.8 évite que le CONTROLEUR D'INTERRUPTION interprète de manière erronée le niveau bas sur cette ligne comme une nouvelle demande d'interruption.

OBSERVATION 4.7:

Le MAITRE qui accède au registre d'état ou de contrôle du GENERATEUR D'INTERRUPTION se trouve généralement sur la même carte que le CONTROLEUR D'INTERRUPTION et il génère, sur la carte, un signal au CONTROLEUR D'INTERRUPTION quand il a terminé l'accès au registre.

AUTORISATION 4.2:

Si une procédure est établie pour permettre au MAITRE de signaler au CONTROLEUR D'INTERRUPTION un accès aux registre d'état ou de contrôle du GENERATEUR D'INTERRUPTION.

ALORS le MAITRE et le CONTROLEUR D'INTERRUPTION PEUVENT résider sur des cartes différentes.

RULE 4.5:

An RORA INTERRUPTER MUST NOT release its interrupt request line before DSA* has fallen during the register access cycle. It MUST release the interrupt request line within 2 μ s after the last data strobe goes high at the end of the register access cycle.

RULE 4.6:

An ROAK INTERRUPTER MUST NOT release its interrupt request line before DSA* has fallen during the interrupt acknowledge cycle which acknowledges its interrupt, and it MUST release interrupt request line within 500 ns after the last data strobe goes high at the end of the STATUS/ID read cycle.

RULE 4.7:

Both RORA and ROAK INTERRUPTERS MUST provide a STATUS/ID during the interrupt acknowledge cycle that was initiated in response to their interrupt request.

RULE 4.8:

After an INTERRUPT HANDLER initiates an interrupt acknowledge cycle, and reads the STATUS/ID from an RORA INTERRUPTER, it MUST ignore the low level on the interrupt request line for 2 μ s after its on-board signal REGISTER ACCESSED goes true.

OBSERVATION 4.6:

RULE 4.8 prevents the INTERRUPT HANDLER from misinterpreting the low level on that line as a new interrupt request.

OBSERVATION 4.7:

The MASTER that accesses the INTERRUPTER'S control or status register is typically on the same board as the INTERRUPT HANDLER, and it generates an on-board signal to the INTERRUPT HANDLER when it has completed the register access.

PERMISSION 4.2:

IF a procedure is established to allow the MASTER to signal the INTERRUPT HANDLER an access to the INTERRUPTER'S control or status registers,
THEN the MASTER and INTERRUPT HANDLER MAY reside on different boards.

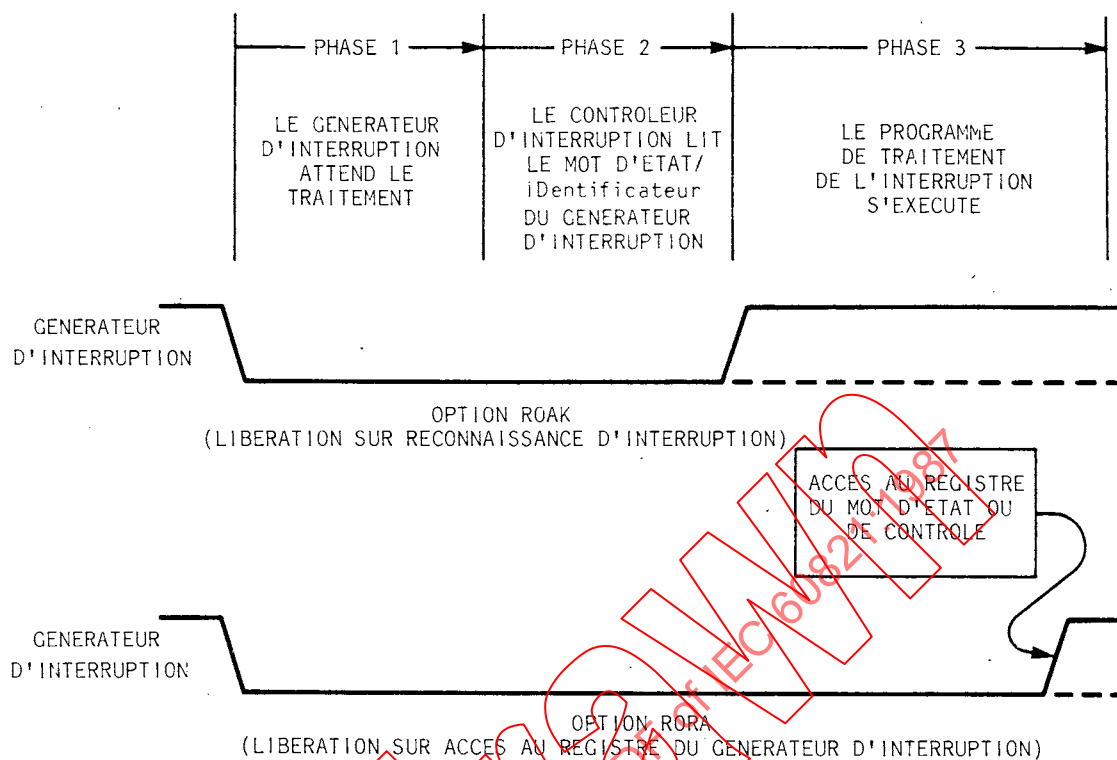


FIG. 4-8. - Libération des lignes de demande d'interruption par les GENERATEURS D'INTERRUPTION ROAK et RORA

Le tableau 4-6 montre comment les mnémoniques RORA et ROAK sont utilisés pour décrire les GENERATEURS D'INTERRUPTION.

Tableau 4-6

Mnémoniques qui spécifient les possibilités de libération des demandes d'interruption

Le mnémonique suivant	Quand appliqué à un	Signifie qu'il
RORA	GENERATEUR D'INTERRUPTION	Libère sa ligne de demande d'interruption quand un MAÎTRE accède à un registre d'état ou de contrôle placé sur sa carte
ROAK	GENERATEUR D'INTERRUPTION	Libère sa ligne de demande d'interruption quand son MOT D'ETAT/IDENTIFICATEUR est lu pendant un cycle de reconnaissance d'interruption

4.3.8 Interaction entre les modules du bus d'interruption prioritaire

Dans les discussions suivantes, plusieurs signaux locaux sont définis pour décrire l'interaction entre les modules GENERATEUR D'INTERRUPTION, CONTROLEUR D'INTERRUPTION et le reste de la logique

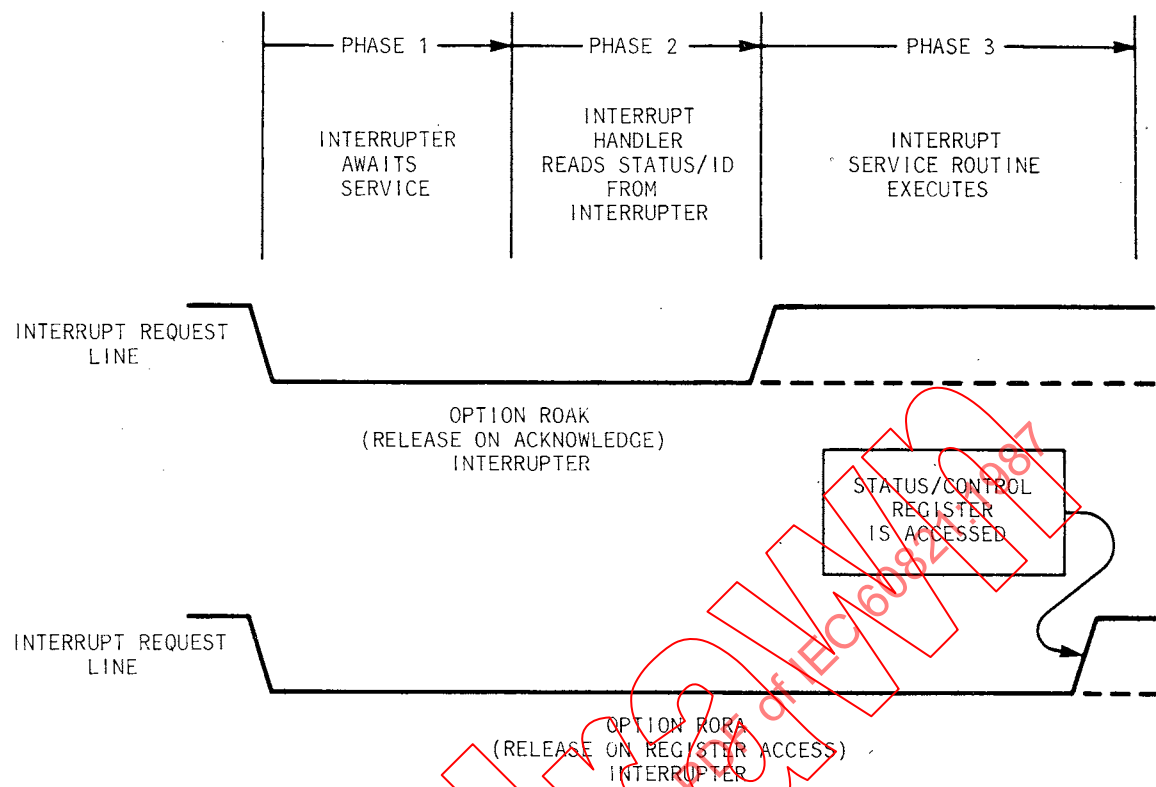


FIG. 4-8. - Release of the interrupt request lines by ROAK and RORA INTERRUPTERS.

Table 4-6 shows how the RORA and ROAK mnemonics are used to describe INTERRUPTERS.

Table 4-6

Mnemonics that specify interrupt request release capabilities

The following mnemonic	When applied to an	Means that it
RORA	INTERRUPTER	Releases its interrupt request line when some MASTER accesses an on-board status or control register
ROAK	INTERRUPTER	Releases its interrupt request line when its STATUS/ID is read during an interrupt acknowledge cycle

4.3.8 Interaction between Priority Interrupt Bus modules

In the following discussions, several on-board signals are defined to describe the interaction between the INTERRUPTER and INTERRUPT HANDLER modules and other on-board logic. These signals are only

locale de la carte. Ces signaux sont seulement destinés à illustrer l'information qui est échangée avec les modules, plutôt qu'à définir leur conception.

AUTORISATION 4.3:

Les cartes de BUS CEI 821 PEUVENT être conçues avec des signaux locaux qui diffèrent de ceux utilisés dans les discussions suivantes.

La figure 4-4, page 246, représente un exemple de routage de la chaîne série IACKIN*/IACKOUT* à travers une configuration typique de cartes sur le BUS CEI 821.

La ligne IACK* parcourt toute la longueur du fond de panier et peut être commandé par n'importe quel CONTROLEUR D'INTERRUPTION ayant le contrôle du DTB. Le fond de panier relie la ligne IACK* à la broche IACKIN* de l'emplacement 1. L'EMETTEUR DE CHAÎNE SÉRIE IACK est situé dans l'emplacement 1 et surveille le niveau de sa ligne IACKIN*.

Lorsque le CONTROLEUR D'INTERRUPTION commande IACK* (et la broche IACKIN* de l'emplacement 1) au niveau bas, puis commande DSA* avec un niveau bas, l'EMETTEUR DE CHAÎNE SÉRIE IACK génère un front descendant sur sa broche IACKOUT*. Cette broche est connectée à la broche IACKIN* de l'emplacement 2. Un cavalier sur la carte de l'emplacement 2 propage le front descendant sur la broche IACKIN* vers la broche IACKOUT* et, à travers le fond de panier, vers la broche IACKIN* de la carte de l'emplacement 3. Le GÉNÉRATEUR D'INTERRUPTION de l'emplacement 3 n'ayant pas de demande d'interruption en attente, il transmet alors le front descendant vers sa broche IACKOUT*. Le GÉNÉRATEUR D'INTERRUPTION de l'emplacement 4 détecte le front descendant sur sa ligne IACKIN* et répond en plaçant son MOT D'ÉTAT/Identificateur sur le bus de données, puis en commandant DTACK* au niveau bas.

AUTORISATION 4.4:

Un GÉNÉRATEUR D'INTERRUPTION PEUT être placé sur la carte contrôleur du système dans l'emplacement 1 avec l'EMETTEUR DE LA CHAÎNE SÉRIE IACK. La figure 4-9, page 274, montre comment ces deux modules seraient connectés.

AUTORISATION 4.5:

Plusieurs GÉNÉRATEURS D'INTERRUPTION PEUVENT se trouver sur une carte. La figure 4-10, page 276, montre comment cela pourrait être réalisé.

OBSERVATION 4.8:

Dans certains cas, les concepteurs de carte ne peuvent pas savoir si la carte en cours de conception sera installée dans l'emplacement 1 ou dans un autre emplacement du BUS CEI 821.

RECOMMANDATION 4.1:

SI une carte possède à la fois un EMETTEUR DE CHAÎNE SÉRIE IACK et un GÉNÉRATEUR D'INTERRUPTION et PEUT ou non être installée dans l'emplacement 1,
ALORS la concevoir comme indiqué dans la figure 4-9.

AUTORISATION 4.6:

Plusieurs cartes comportant des EMETTEURS DE CHAÎNES SÉRIES IACK PEUVENT être installées dans un système de BUS CEI 821.

intended to illustrate the information which is passed to and from the modules, rather than to define their designs.

PERMISSION 4.3:

IEC 821 BUS boards MAY be designed with on-board signals that differ from those used in the following discussions.

Figure 4-4, page 247, shows how the IACKIN*/IACKOUT* daisy-chain is routed through a typical configuration of boards on the IEC 821 BUS.

The IACK* line runs the full length of the backplane and can be driven by any INTERRUPT HANDLER that has control of the DTB. The backplane connects IACK* to the IACKIN* pin of slot 1. The IACK DAISY-CHAIN DRIVER resides in slot 1 and monitors the level of slot 1's IACKIN* line.

When an INTERRUPT HANDLER drives IACK* (and slot 1's IACKIN*) low, and then drives DSA* low, the IACK DAISY-CHAIN DRIVER generates a falling edge on its IACKOUT* pin. This pin is connected to the IACKIN* pin of slot 2. A jumper on the board in slot 2 routes the falling edge on the IACKIN* pin to the IACKOUT* pin, and through the backplane to the IACKIN* pin of the board in slot 3. The INTERRUPTER in slot 3 does not have a pending interrupt request, so it passes on the falling edge to its IACKOUT* pin. The INTERRUPTER in slot 4 then detects the falling edge on its IACKIN* line and responds by placing its STATUS/ID on the data bus, and then driving DTACK* low.

PERMISSION 4.4:

An INTERRUPTER MAY reside on the system controller board, installed in slot 1, along with the IACK DAISY-CHAIN DRIVER. Figure 4-9, page 275, shows how the two modules would be connected.

PERMISSION 4.5:

More than one INTERRUPTER MAY reside on a board. Figure 4-10, page 277, shows how this might be done.

OBSERVATION 4.8:

In some cases, board designers might not know whether or not the board they are designing will be installed in slot 1, or in some other slot of an IEC 821 BUS system.

RECOMMENDATION 4.1:

IF a board includes both an IACK DAISY-CHAIN DRIVER and an INTERRUPTER, and might or might not be installed in slot 1,
THEN design it as shown in Figure 4-9.

PERMISSION 4.6:

Several boards containing IACK DAISY-CHAIN DRIVERS MAY be installed in an IEC 821 BUS system.

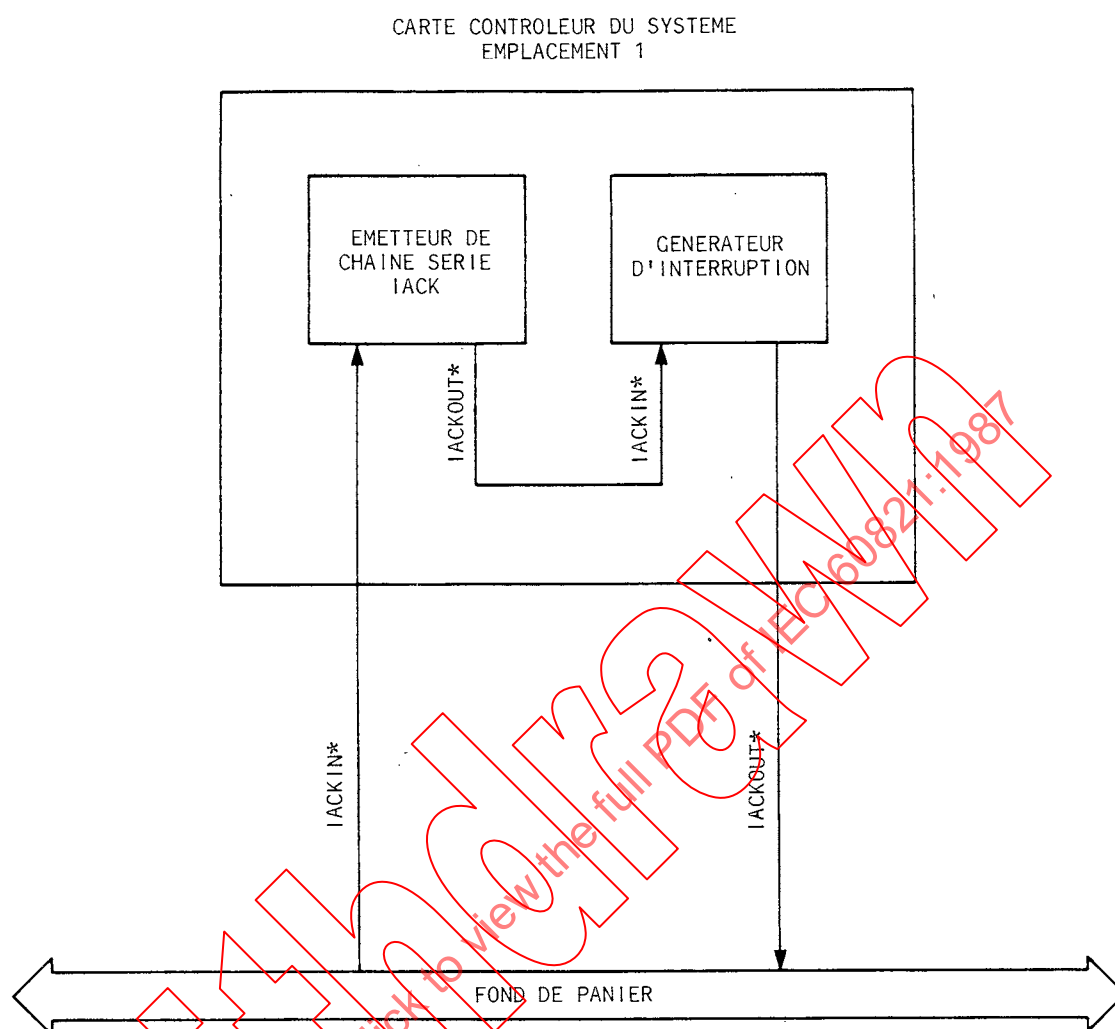


FIG. 4-9. - Un EMETTEUR DE CHAINE SERIE IACK et un GENERATEUR D'INTERRUPTION sur la même carte.

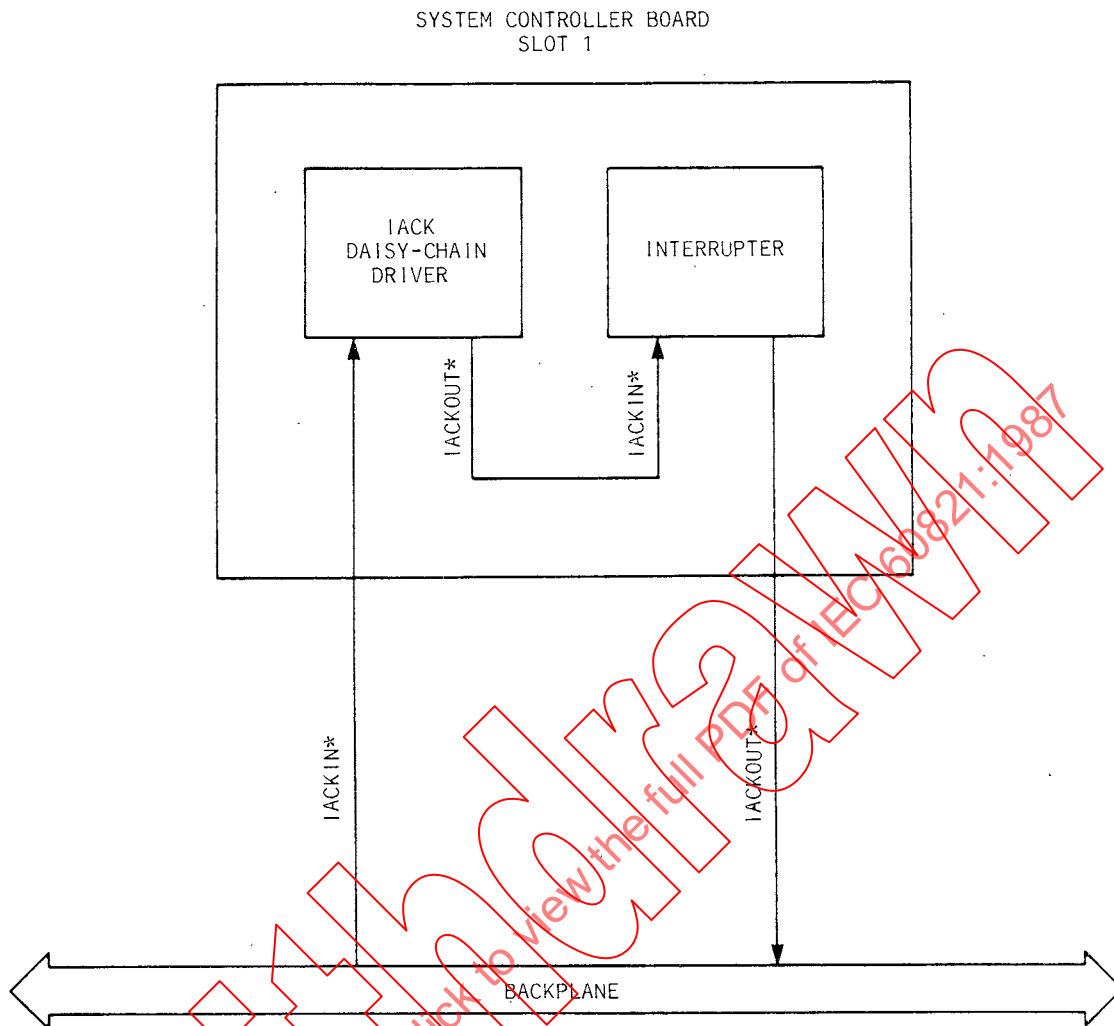


FIG. 4-9. - An JACK DAISY-CHAIN DRIVER and an INTERRUPTER on the same board.

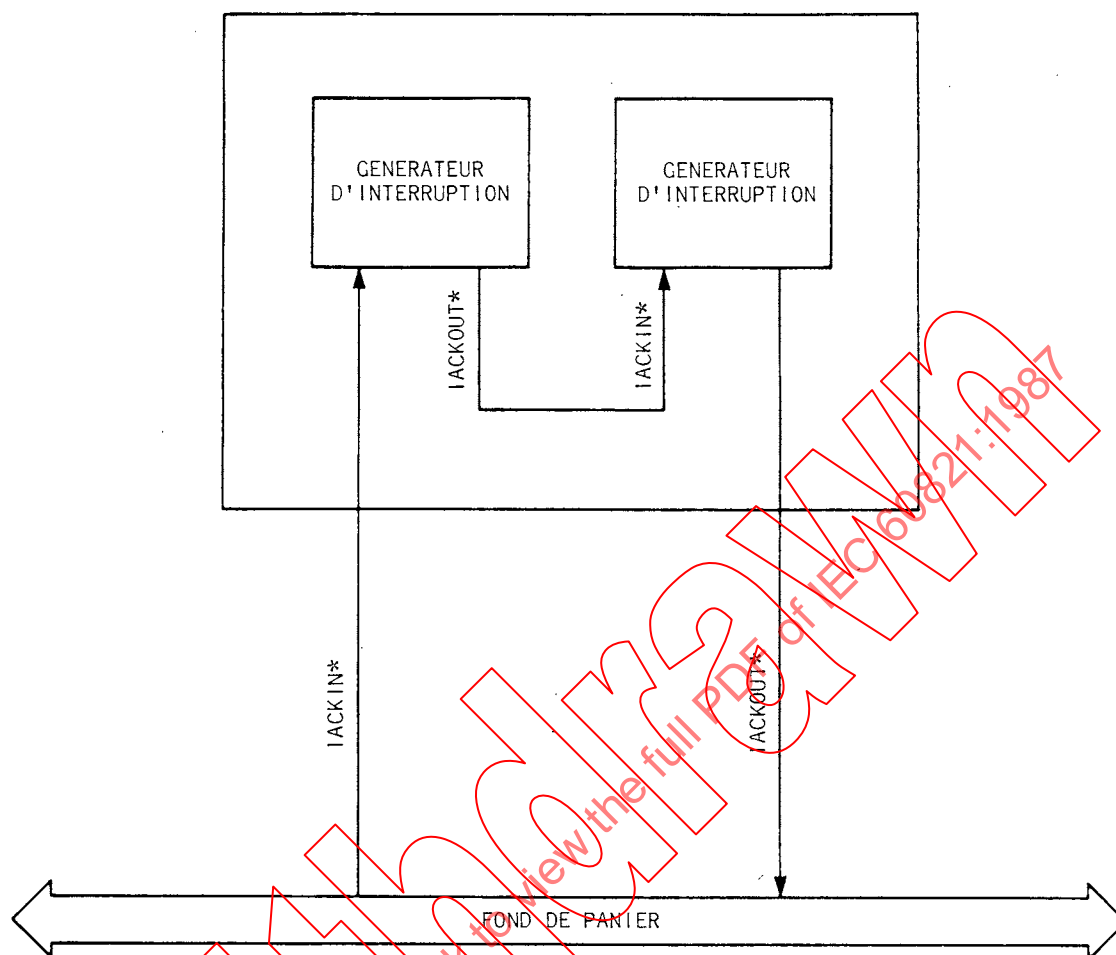


FIG. 4-10. - Deux GENERATEURS D'INTERRUPTION sur la même carte.

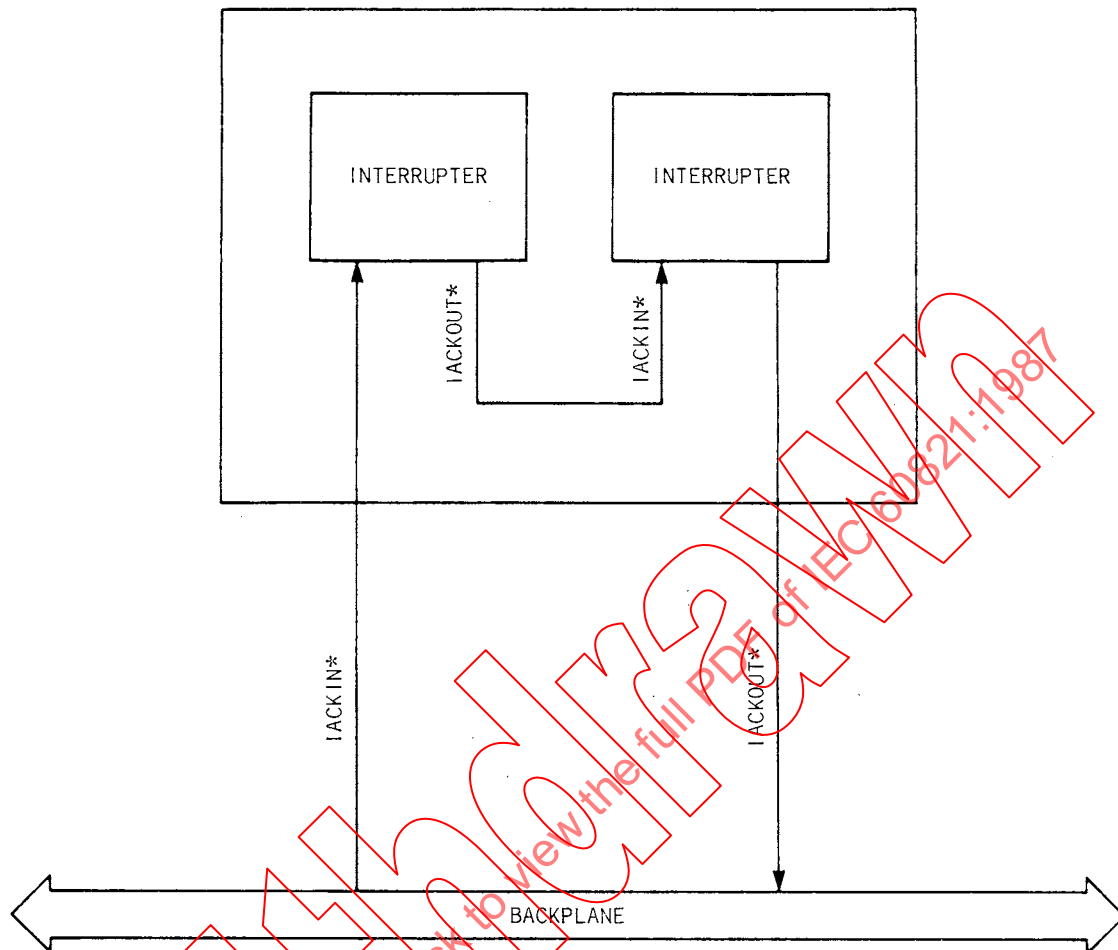


FIG. 4-10. - Two INTERRUPTERS on the same board.

4.4 Fonctionnement typique

Une séquence typique d'interruption peut être divisée en trois phases:

Phase 1: la phase de demande d'interruption.

Phase 2: la phase de reconnaissance d'interruption.

Phase 3: la phase de traitement d'interruption.

La figure 4-11 illustre les relations de temps entre ces trois phases.

La phase 1 commence lorsqu'un GENERATEUR D'INTERRUPTION commande au niveau bas une ligne de demande d'interruption et se termine lorsque le CONTROLEUR D'INTERRUPTION obtient le contrôle du DTB. Pendant la phase 2, le CONTROLEUR D'INTERRUPTION utilise le DTB pour lire le MOT D'ETAT/IDENTIFICATEUR du GENERATEUR D'INTERRUPTION. Pendant la phase 3, un sous-programme de traitement d'interruption est exécuté (cela peut ou non entraîner des transferts de données sur le BUS CEI 821).

Le protocole du sous-système d'interruption décrit l'interaction du module requise pendant les phases 1 et 2. Tout transfert de données ayant lieu pendant la phase 3 sera conforme au protocole du bus de transfert de données décrit dans le chapitre 2.

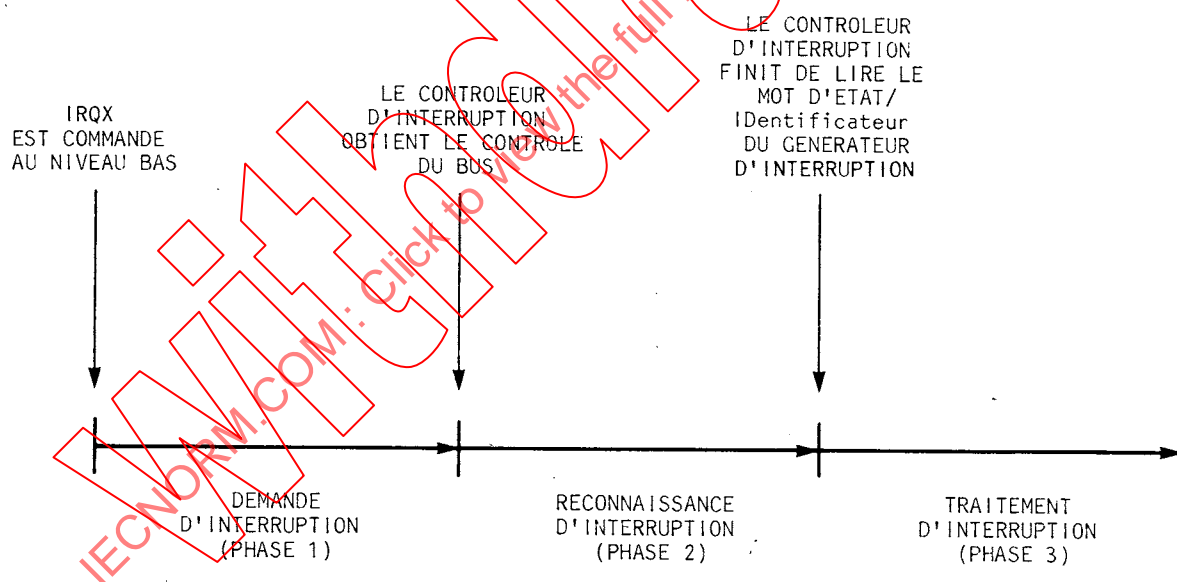


FIG. 4-11. - Les trois phases d'une séquence d'interruption.

4.4.1 Fonctionnement des interruptions à contrôleur unique

Dans les systèmes d'interruptions à contrôleur unique, les sept lignes de demande d'interruption sont toutes surveillées par un seul CONTROLEUR D'INTERRUPTION. Les lignes de demande d'interruption ont un ordre de priorité de manière que IRQ7* ait la priorité la plus élevée et IRQ1* ait la priorité la plus basse. Lorsque le CONTROLEUR D'INTERRUPTION détecte des demandes simultanées sur deux lignes de demande d'interruption, il reconnaît en premier la demande ayant la priorité la plus élevée.

4.4 Typical operation

A typical interrupt sequence can be divided into three phases:

- Phase 1: the interrupt request phase.
- Phase 2: the interrupt acknowledge phase.
- Phase 3: the interrupt service phase.

Figure 4-11 illustrates the timing relationships between the three phases.

Phase 1 starts when an INTERRUPTER drives an interrupt request line low and ends when the INTERRUPT HANDLER gains control of the DTB. During phase 2 the INTERRUPT HANDLER uses the DTB to read the INTERRUPTER'S STATUS/ID. During phase 3 an interrupt servicing routine is executed (this might or might not involve data transfers on the IEC 821 BUS.)

The protocol for the interrupt subsystem describes the module interaction required during phase 1 and phase 2. Any data transfers which take place during phase 3 will follow the Data Transfer Bus protocol described in Chapter 2.

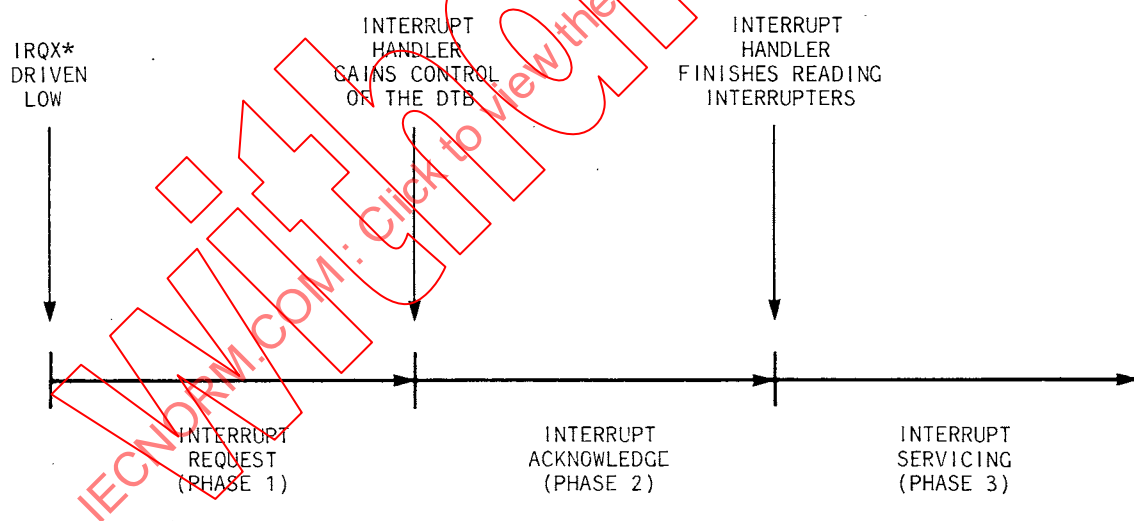


FIG. 4-11. - The three phases of an interrupt sequence.

4.4.1 Single handler interrupt operation

In single handler interrupt systems, the seven interrupt request lines are all monitored by a single INTERRUPT HANDLER. The interrupt request lines are prioritized such that IRQ7* has the highest priority, and IRQ1* has the lowest priority. When the INTERRUPT HANDLER detects simultaneous requests on two interrupt request lines, it acknowledges the highest priority request first.

4.4.2 *Fonctionnement des interruptions distribuées*

Les systèmes à interruptions distribuées contiennent deux à sept **CONTROLEURS D'INTERRUPTION**. Dans la description suivante, les systèmes à interruptions distribuées seront séparés en deux groupes:

- a) les systèmes à interruptions distribuées avec sept **CONTROLEURS D'INTERRUPTION**,
- b) les systèmes à interruptions distribuées avec deux à six **CONTROLEURS D'INTERRUPTION**.

4.4.2.1 *Systèmes à interruptions distribuées avec sept CONTROLEURS D'INTERRUPTION*

Dans des systèmes à interruptions distribuées avec sept **CONTROLEURS D'INTERRUPTION**, chacune des lignes de demande d'interruption est surveillée par un **CONTROLEUR D'INTERRUPTION** séparé. Chaque **CONTROLEUR D'INTERRUPTION** obtient le contrôle du DTB avant de lire le MOT D'ETAT/IDentificateur du **GENERATEUR D'INTERRUPTION** qui commande sa ligne de demande d'interruption.

OBSERVATION 4.9:

Il n'y a aucune relation spécifiée entre la ligne de demande d'interruption traitée par un **CONTROLEUR D'INTERRUPTION** et la ligne de demande de bus utilisée par le **DEMANDEUR** sur sa carte. Par exemple, un **CONTROLEUR D'INTERRUPTION** qui traite IRQ7* peut avoir un **DEMANDEUR** qui utilise BR0*, et un **CONTROLEUR D'INTERRUPTION** qui traite IRQ1* peut avoir un **DEMANDEUR** qui utilise BR3*. Cela montre clairement qu'il n'y a pas de priorité d'interruption évidente entre les lignes traitées par différents **CONTROLEURS D'INTERRUPTION**.

La figure 4-12, page 282, représente un système à interruptions distribuées où le **CONTROLEUR D'INTERRUPTION A** surveille la ligne IRQ2* et a un **DEMANDEUR** sur sa carte qui demande le DTB à l'aide de BR2*. Le **CONTROLEUR D'INTERRUPTION B** surveille IRQ5* et a un **DEMANDEUR** sur sa carte qui demande le DTB à l'aide de BR3*. Deux **GENERATEURS D'INTERRUPTION** commandent simultanément IRQ2* et IRQ5* au niveau bas et les deux **CONTROLEURS D'INTERRUPTION** entraînent leurs **DEMANDEURS** locaux à commander BR2* et BR3* au niveau bas simultanément. Dans cet exemple, l'arbitrage prioritaire est utilisé et, les deux demandes de bus étant au niveau bas, l'**ARBITRE** alloue le contrôle du DTB d'abord au **DEMANDEUR** du **CONTROLEUR D'INTERRUPTION B** et le **CONTROLEUR D'INTERRUPTION A** attend que B finisse d'utiliser le DTB.

OBSERVATION 4.10:

Si l'arbitrage à priorité tournante est utilisé, l'un ou l'autre des **CONTROLEURS D'INTERRUPTION** décrit dans la figure 4-12 peut recevoir le bus en premier.

4.4.2.2 *Systèmes à interruptions distribuées avec deux à six CONTROLEURS D'INTERRUPTION*

Il est aussi possible de définir une configuration de système à interruptions distribuées où deux ou plusieurs lignes de demandes d'interruption sont surveillées par un seul **CONTROLEUR D'INTERRUPTION**. La figure 4-13, page 284, présente une configuration de système avec deux **CONTROLEURS D'INTERRUPTION** dans laquelle le

4.4.2 *Distributed interrupt operation*

Distributed interrupt systems contain from two to seven INTERRUPT HANDLERS. For purposes of the following discussion, distributed interrupt systems will be considered in two groups:

- a) distributed interrupt systems with seven INTERRUPT HANDLERS,
- b) distributed interrupt systems with two to six INTERRUPT HANDLERS.

4.4.2.1 *Distributed interrupt systems with seven INTERRUPT HANDLERS*

In distributed interrupt systems with seven INTERRUPT HANDLERS, each of the interrupt request lines is monitored by a separate INTERRUPT HANDLER. Each INTERRUPT HANDLER gains control of the DTB before it reads the STATUS/ID from INTERRUPTERS driving its interrupt request line.

OBSERVATION 4.9:

There is no specified relationship between the interrupt request line that an INTERRUPT HANDLER services and the bus request line used by its on-board REQUESTER. For example, an INTERRUPT HANDLER that services IRQ7* might have a REQUESTER that uses BR0*, and an INTERRUPT HANDLER that services IRQ1* might have a REQUESTER that uses BR3*. It is clear from this that there is no implied interrupt priority between lines serviced by different INTERRUPT HANDLERS.

Figure 4-12, page 283, illustrates a distributed interrupt system where INTERRUPT HANDLER A monitors IRQ2* and has an on-board REQUESTER which requests the DTB on BR2*. INTERRUPT HANDLER B monitors IRQ5* and has an on-board REQUESTER which requests the DTB on BR3*. Two INTERRUPTERS simultaneously drive IRQ2* and IRQ5* low, and the two INTERRUPT HANDLERS cause their on-board REQUESTERS to drive BR2* and BR3* low simultaneously. In this example, priority arbitration is used and, since both bus requests go low together, the ARBITER first grants control of the DTB to INTERRUPT HANDLER B's REQUESTER, and INTERRUPT HANDLER A waits until B has finished using the DTB.

OBSERVATION 4.10:

If round-robin arbitration is used, either of the INTERRUPT HANDLERS described in Figure 4-12 might be granted the bus first.

4.4.2.2 *Distributed interrupt systems with two to six INTERRUPT HANDLERS.*

It is also possible to configure a distributed interrupt system in which two or more of the interrupt request lines are monitored by a single INTERRUPT HANDLER. Figure 4-13, page 285, illustrates a system configured with two INTERRUPT HANDLERS in which

CONTROLEUR D'INTERRUPTION A surveille les lignes IRQ1*-IRQ4* et le CONTROLEUR D'INTERRUPTION B surveille les lignes IRQ5*-IRQ7*. Dans ce cas, les lignes IRQ1*-IRQ4* sont hiérarchisées; IRQ4* = la priorité la plus élevée du CONTROLEUR D'INTERRUPTION A. Les lignes IRQ5*-IRQ7* sont aussi hiérarchisées; IRQ7* = la priorité la plus élevée du CONTROLEUR D'INTERRUPTION B. L'arbitrage du DTB détermine toujours lequel des CONTROLEURS D'INTERRUPTION reçoit en premier le contrôle du DTB.

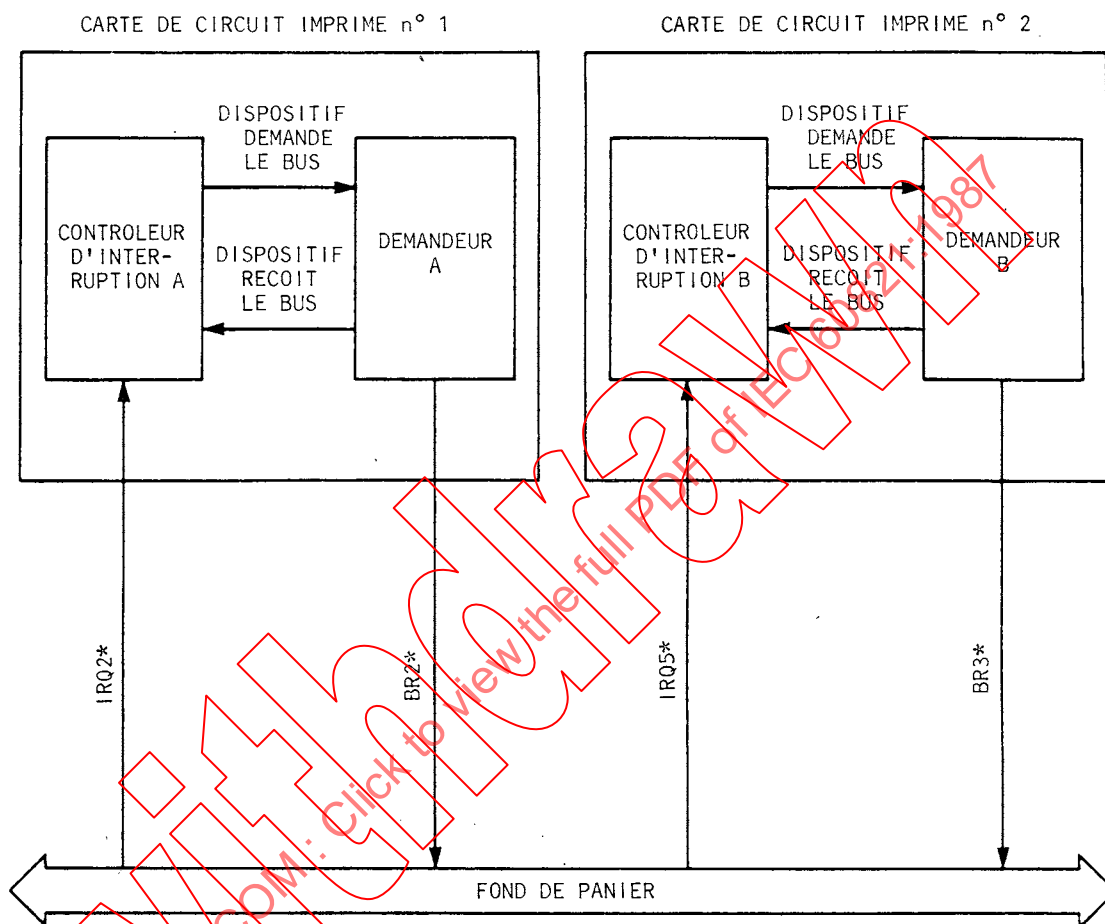


FIG. 4-12. - Deux CONTROLEURS D'INTERRUPTION surveillant chacun une ligne de demande d'interruption.

INTERRUPT HANDLER A monitors IRQ1*-IRQ4*, and INTERRUPT HANDLER B monitors IRQ5*-IRQ7*. In this case the IRQ1*-IRQ4* lines are prioritized: IRQ4* = highest priority for INTERRUPT HANDLER A, and the IRQ5*-IRQ7* lines are prioritized; IRQ7* = highest priority for INTERRUPT HANDLER B. The DTB arbitration still determines which INTERRUPT HANDLER is granted the use of the DTB first.

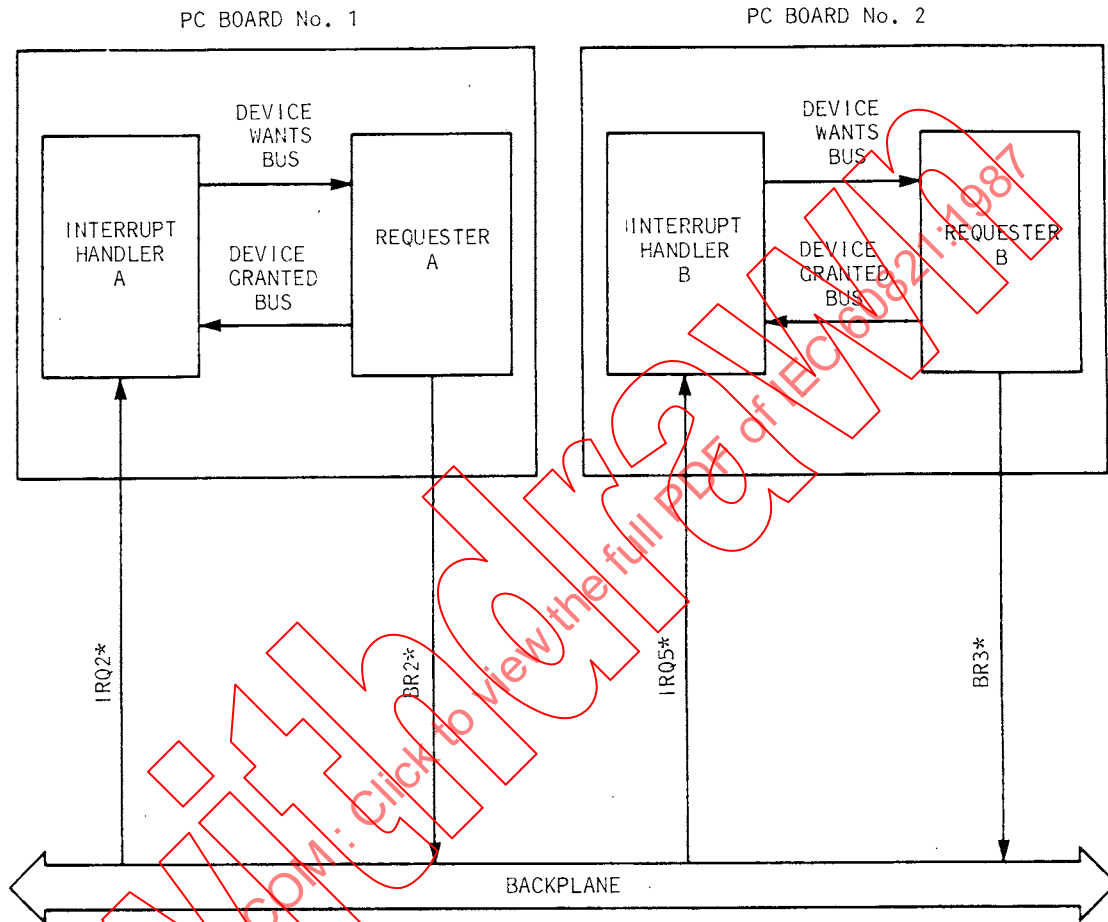


FIG. 4-12. - Two INTERRUPT HANDLERS, each monitoring one interrupt request line.

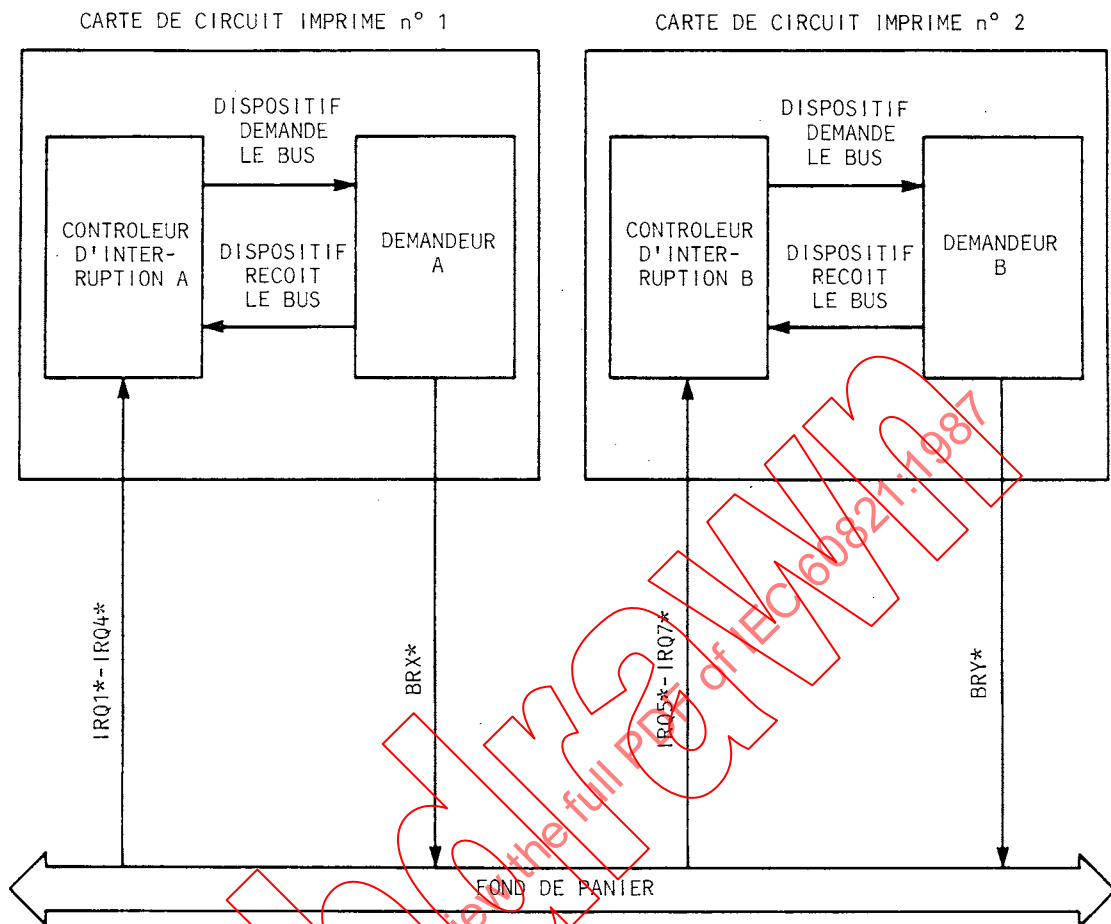


FIG. 4-13 - Deux CONTROLEURS D'INTERRUPTION surveillant chacun plusieurs lignes de demande d'interruption.

4.4.3 Exemple: Fonctionnement typique d'un système d'interruption à contrôleur unique

La figure 4-14, page 288, présente le fonctionnement d'un système d'interruption à contrôleur unique dans lequel un CONTROLEUR D'INTERRUPTION surveille et gère la priorité des sept lignes d'interruption. En haut du diagramme, un MAITRE utilise le DTB pour transférer des données dans le système avec le niveau 2 de demande. Un GENERATEUR D'INTERRUPTION à l'emplacement 4 demande une interruption en commandant IRQ4* au niveau bas. Lorsque le CONTROLEUR D'INTERRUPTION détecte IRQ4* au niveau bas, il envoie un signal à son DEMANDEUR sur sa carte pour indiquer qu'il a besoin du bus. Ce DEMANDEUR commande alors BR3* au niveau bas. Détectant la demande de bus, l'ARBITRE commande le signal BCLR* au niveau bas pour indiquer l'attente du DTB par un DEMANDEUR de niveau de priorité plus élevé. (Cet exemple s'applique à l'option d'ARBITRE PRI.) Lorsque le MAITRE A détecte le signal BCLR* au niveau bas, il arrête le transfert de données et permet à son DEMANDEUR d'abandonner le contrôle du DTB et de libérer le signal BBSY*.

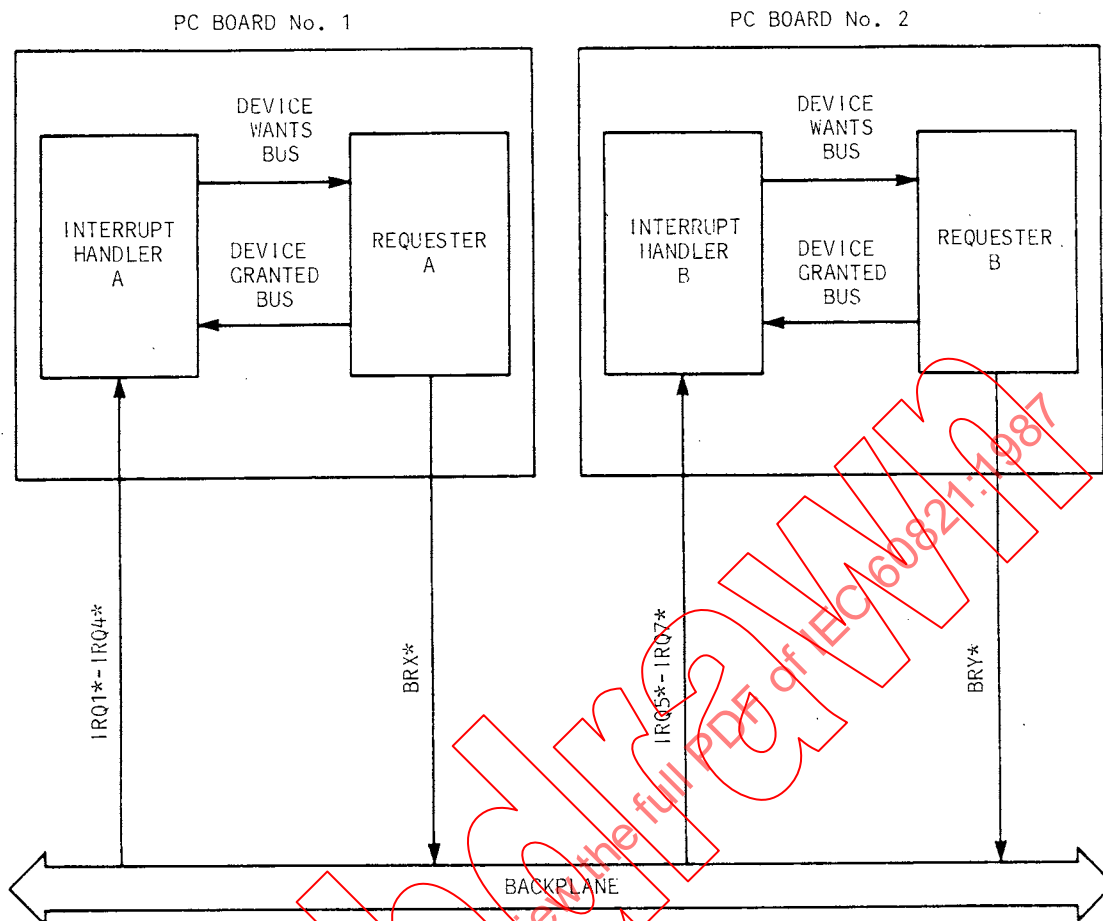


FIG. 4-13. Two INTERRUPT HANDLERS, each monitoring several interrupt request lines.

4.4.3 Example: typical single handler interrupt system operation

Figure 4-14, page 289, illustrates the operation of a single handler interrupt system in which one INTERRUPT HANDLER monitors and prioritizes all seven interrupt lines. At the top of the diagram, a MASTER is using the DTB to move data within the system at a bus request level of 2. An INTERRUPTER in slot 4 requests an interrupt by driving IRQ4* low. When the INTERRUPT HANDLER detects the low level on IRQ4* it sends a signal to its on-board REQUESTER, indicating that it needs the bus. This REQUESTER then drives BR3* low. Upon detecting the bus request, the ARBITER drives BCLR* low, indicating that a higher priority REQUESTER is waiting for the DTB. (This example assumes a PRI ARBITER.) When MASTER A detects the low level on BCLR*, it stops moving data and allows its requester to relinquish control of the DTB, and release BBSY*.

OBSERVATION 4.11:

Le MAITRE actif n'est pas obligé d'abandonner le DTB dans un temps spécifié mais une réponse rapide à la ligne BCLR* permet de traiter l'interruption plus rapidement.

Lorsque l'ARBITRE détecte BBSY* au niveau haut, il alloue le DTB au DEMANDEUR B, qui informe son CONTROLEUR D'INTERRUPTION que le DTB est disponible (voir figure 2-26, page 190). Le CONTROLEUR D'INTERRUPTION émet alors sur les trois lignes d'adresse les moins significatives, un code de 3 bits indiquant la reconnaissance de la demande d'interruption sur la ligne IRQ4* (voir tableau 4-7). En même temps, il commande la ligne IACK* au niveau bas, pour signaler la reconnaissance d'une interruption et commande AS* au niveau bas. Le niveau bas sur la ligne IACK* est transmis par une piste du fond de panier à la broche IACKIN* de l'emplacement 1, et provoque la génération par l'EMETTEUR DE CHAÎNE SERIE IACK* d'un front descendant qui se propage en aval sur la chaîne série IACKIN*/ IACKOUT*.

Lorsque le GENERATEUR D'INTERRUPTION détecte un front descendant sur sa ligne d'entrée de la chaîne série (IACKIN*), il vérifie si les 3 bits d'adresse les moins significatifs correspondent à la ligne de demande d'interruption qu'il commande au niveau bas. Puisque le codage des 3 bits est égal au niveau de la ligne sur laquelle il génère sa demande d'interruption lorsque le GENERATEUR D'INTERRUPTION détecte le ou les signaux de validation de donnée à l'état bas, il place son MOT D'ETAT/IDentificateur sur le bus de données et commande la ligne DTACK* au niveau bas. Lorsque le CONTROLEUR D'INTERRUPTION détecte le niveau bas de DTACK*, il lit le MOT D'ETAT/IDentificateur et active le sous-programme de traitement d'interruption correspondant.

OBSERVATION 4.11:

The active MASTER is not required to relinquish the DTB within any specified time, but a prompt response to the BCLR* line allows the interrupt to be serviced quicker.

When the ARBITER detects BBSY* high, it grants the DTB to REQUESTER B, which informs its INTERRUPT HANDLER that the DTB is available (see Figure 2-26, page 191). The INTERRUPT HANDLER then puts out a 3-bit code on the lower three address lines, indicating that it is acknowledging the interrupt request on the IRQ4* line (see table 4-7). At the same time, it drives IACK* low, indicating that it is acknowledging an interrupt, and drives AS* low. The low level on IACK* is connected, by a signal trace in the backplane, to the IACKIN* pin of slot 1 and causes the IACK DAISY-CHAIN DRIVER to generate a falling edge down the IACKIN*/IACKOUT* daisy-chain.

When the INTERRUPTER detects a falling edge on its incoming daisy-chain line (IACKIN*) it checks the lower 3 address bits to see if they match the interrupt request line which it is driving low. Since the 3-bit code matches the line on which it is making its interrupt request, when the INTERRUPTER detects the data strobe(s) low, it places its STATUS/ID on the data bus and drives the DTACK* line low. When the INTERRUPT HANDLER detects the low on DTACK*, it reads the STATUS/ID and activates the appropriate interrupt service routine.

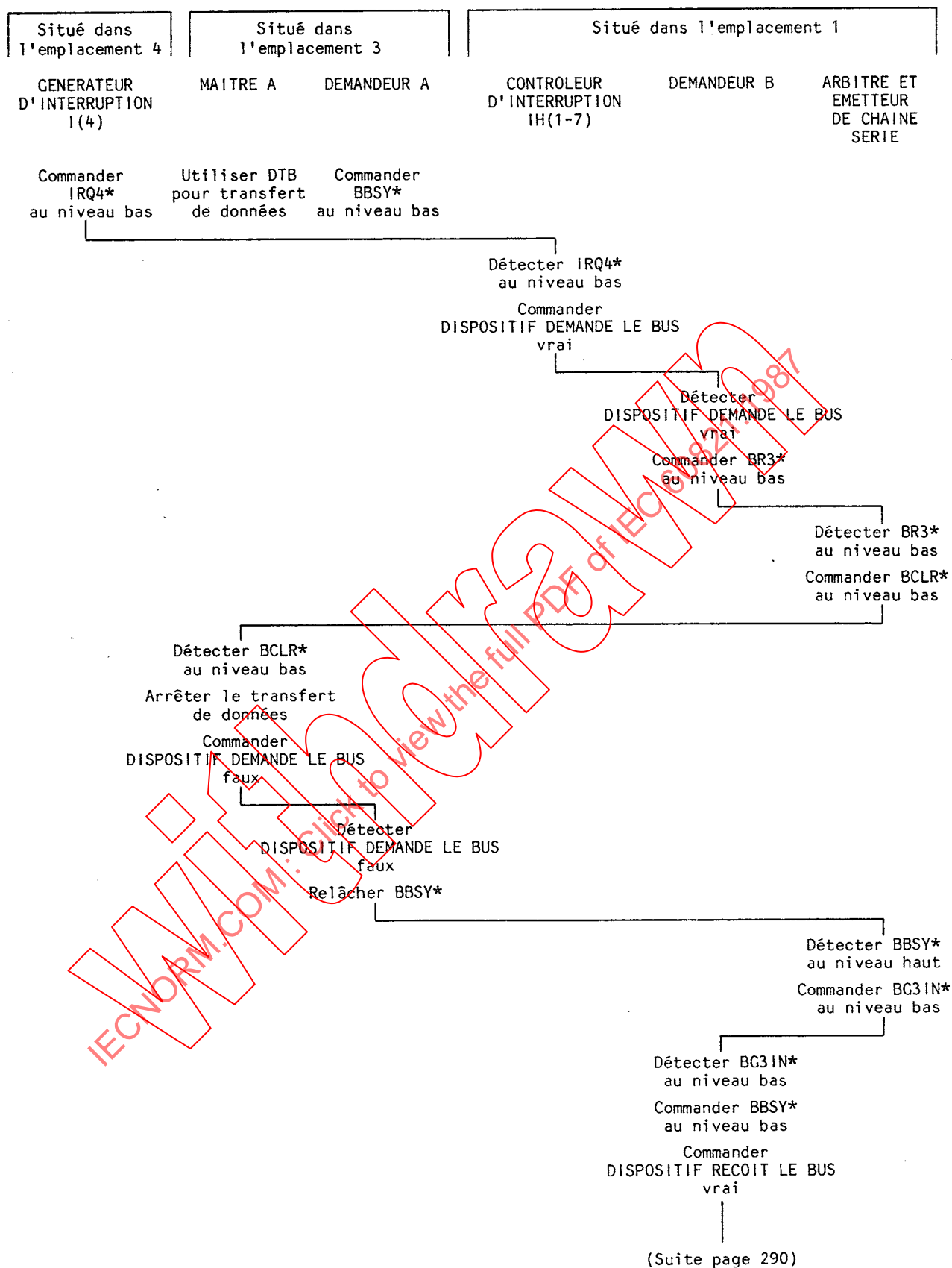


Fig. 4-14. - Organigramme du fonctionnement typique d'un système d'interruption à contrôleur unique d'interruption.

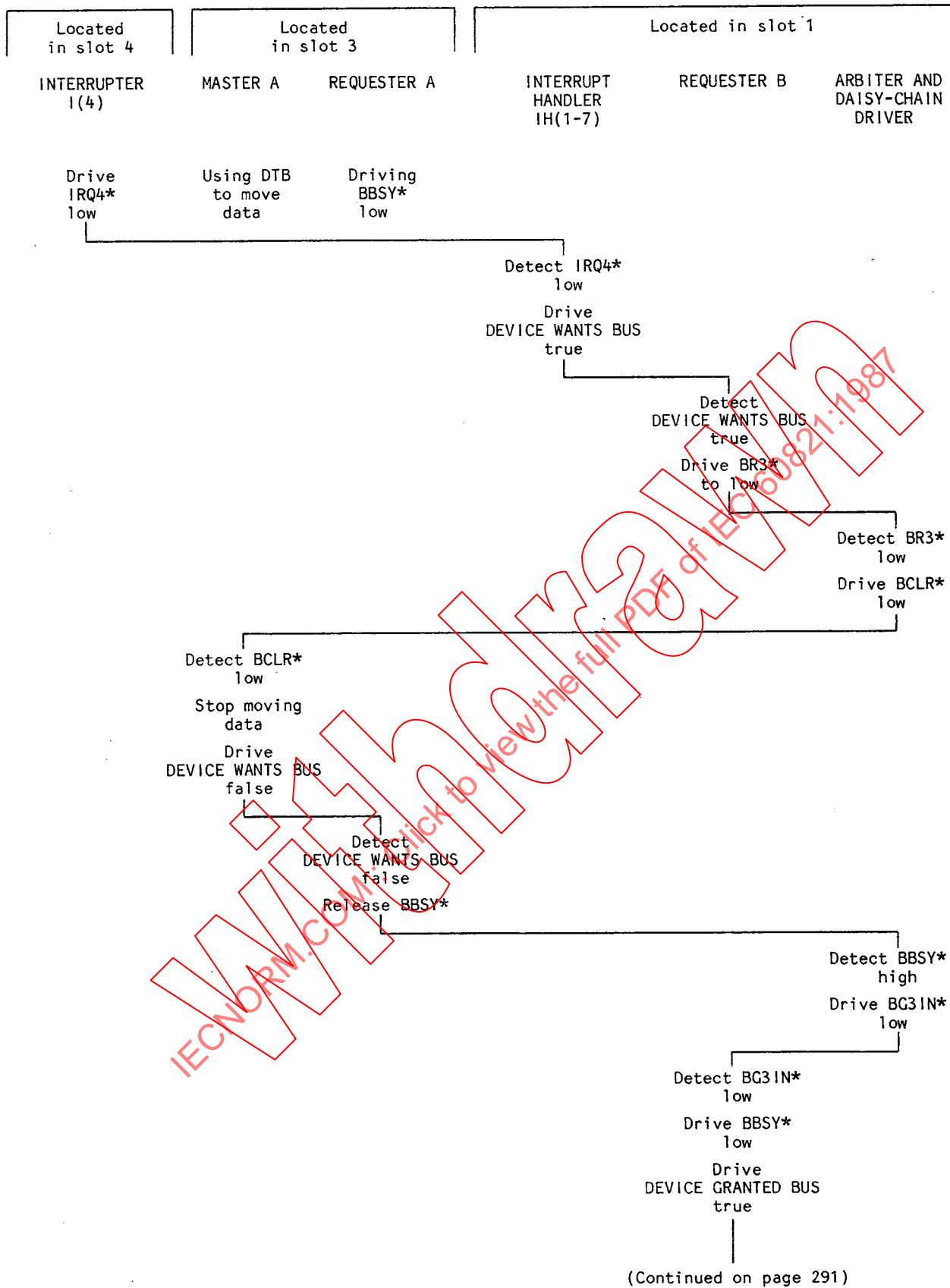


Fig. 4-14. - Typical single handler interrupt system operation flow diagram.

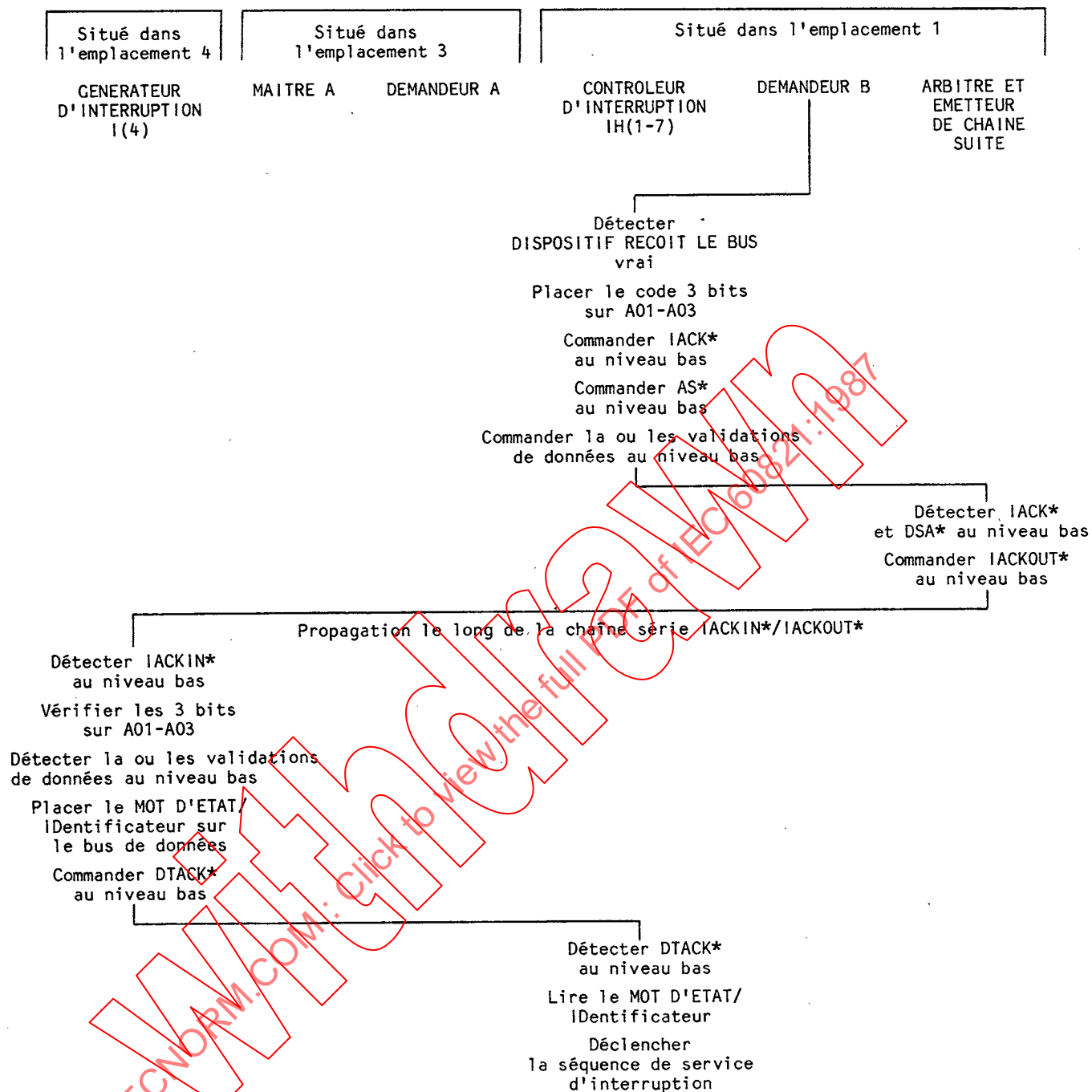


FIGURE 4-14 (fin).

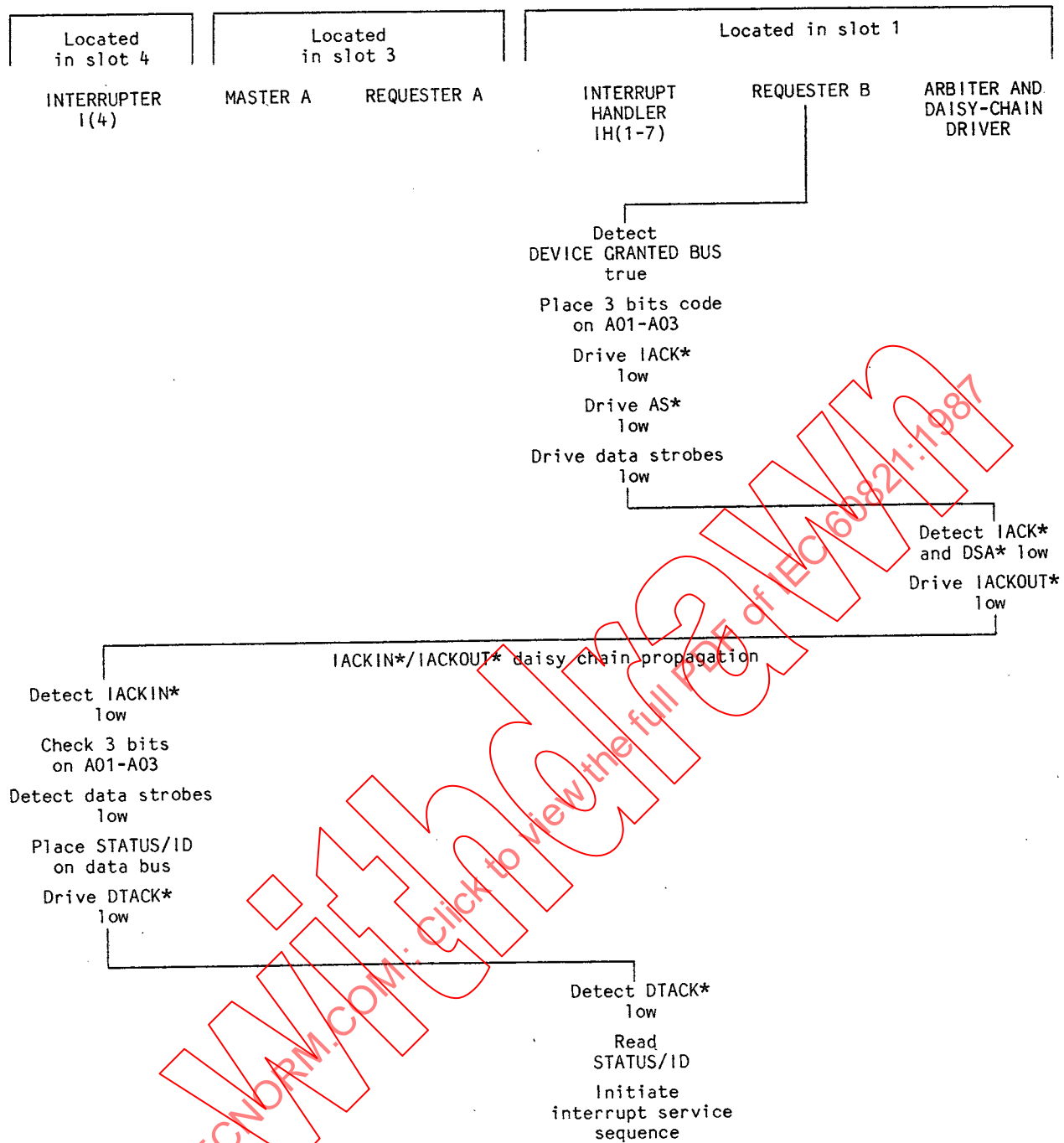


FIGURE 4-14 (concluded).

Tableau 4-7

Code de la reconnaissance d'interruption sur 3 bits

LIGNE D'INTERRUPTION RECONNUE	Utilisation du bus d'adresse pour diffuser le code de reconnaissance d'interruption sur 3 bits		
	A03	A02	A01
IRQ1*	L	L	H
IRQ2*	L	H	L
IRQ3*	L	H	H
IRQ4*	H	L	L
IRQ5*	H	L	H
IRQ6*	H	H	L
IRQ7*	H	H	H

H = niveau haut
L = niveau bas

4.4.4 Exemple: priorité de deux interruptions dans un système à interruptions distribuées

La figure 4-15, page 294, illustre le fonctionnement d'un système à interruptions distribuées à deux CONTROLEURS D'INTERRUPTION. Le CONTROLEUR D'INTERRUPTION A surveille IRQ1*-IRQ4* pendant que le CONTROLEUR D'INTERRUPTION B surveille IRQ5*-IRQ7*. Le CONTROLEUR D'INTERRUPTION A considère IRQ4* comme son interruption la plus prioritaire et le CONTROLEUR D'INTERRUPTION B considère IRQ7* comme son interruption la plus prioritaire. En haut de l'organigramme, le GENERATEUR D'INTERRUPTION C commande IRQ3* au niveau bas et le GENERATEUR D'INTERRUPTION D commande IRQ6* au niveau bas. Les deux CONTROLEURS D'INTERRUPTION détectent le niveau bas sur leurs lignes respectives de demande d'interruption et les deux indiquent simultanément à leurs DEMANDEURS sur leurs cartes qu'ils ont besoin du DTB. Les deux DEMANDEURS commandent BR3* au niveau bas. Lorsqu'il détectent BR3* au niveau bas, l'ARBITRE DTB commande BG3IN* au niveau bas dans l'emplacement 1. Ce signal au niveau bas est propagé en aval sur la chaîne série BG3IN*/BG3OUT* jusqu'à ce qu'il soit détecté par le DEMANDEUR B dans l'emplacement 4. Ce DEMANDEUR signale alors à son CONTROLEUR D'INTERRUPTION B sur sa carte que le DTB est disponible. Le CONTROLEUR D'INTERRUPTION B lit alors le MOT D'ETAT/IDentificateur du GENERATEUR D'INTERRUPTION D.

Table 4-7

3-bit interrupt acknowledge code

INTERRUPT LINE BEING ACKNOWLEDGED	Use of the address bus to broadcast the 3-bit interrupt acknowledge code		
	A03	A02	A01
IRQ1*	L	L	H
IRQ2*	L	H	L
IRQ3*	L	H	H
IRQ4*	H	L	L
IRQ5*	H	L	H
IRQ6*	H	H	L
IRQ7*	H	H	H

H = high level
L = low level

4.4.4 Example: prioritization of two interrupts in a distributed interrupt system

Figure 4-15, page 295, illustrates the operation of a distributed interrupt system with two INTERRUPT HANDLERS. INTERRUPT HANDLER A monitors IRQ1*-IRQ4*, while INTERRUPT HANDLER B monitors IRQ5*-IRQ7*. INTERRUPT HANDLER A treats IRQ4* as its highest priority interrupt, while INTERRUPT HANDLER B treats IRQ7* as its highest priority interrupt. At the top of the diagram, INTERRUPTER C drives IRQ3* low, and INTERRUPTER D drives IRQ6* low. Both INTERRUPT HANDLERS detect their respective interrupt request lines low, and both simultaneously indicate to their on-board REQUESTERS that they need the DTB. Both the REQUESTERS drive BR3* low. Upon detecting BR3* low, the DTB ARBITER drives BG3IN* low at slot 1. This low signal is passed down to BG3IN*/BG3OUT* daisy-chain until it is detected by the REQUESTER B in slot 4. This REQUESTER then signals its on-board INTERRUPT HANDLER B that the DTB is available. INTERRUPT HANDLER B then reads the STATUS/ID from INTERRUPTER D.

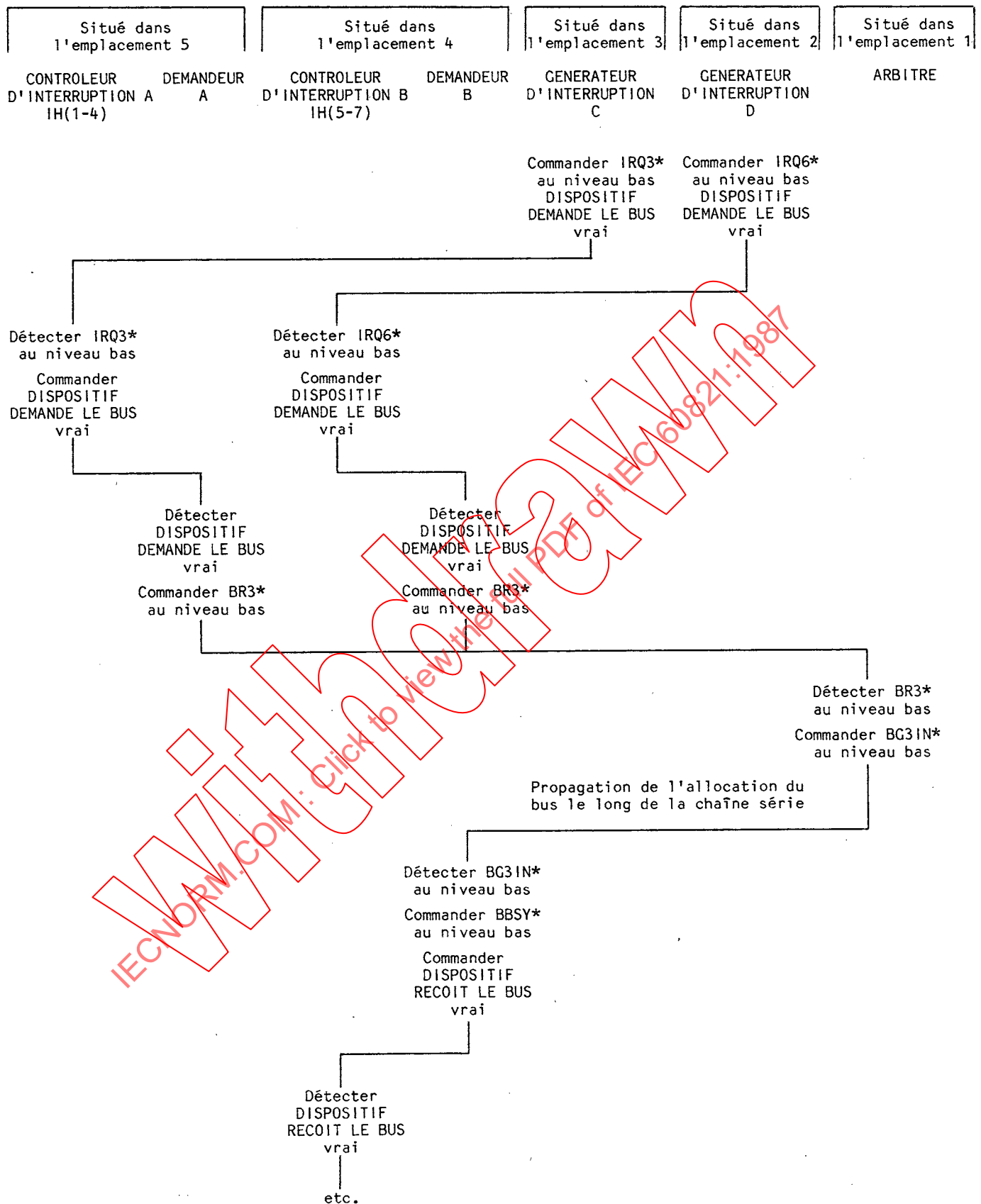


FIG. 4-15. - Organigramme du fonctionnement typique d'un système à interruptions distribuées avec deux CONTROLLEURS D'INTERRUPTION.

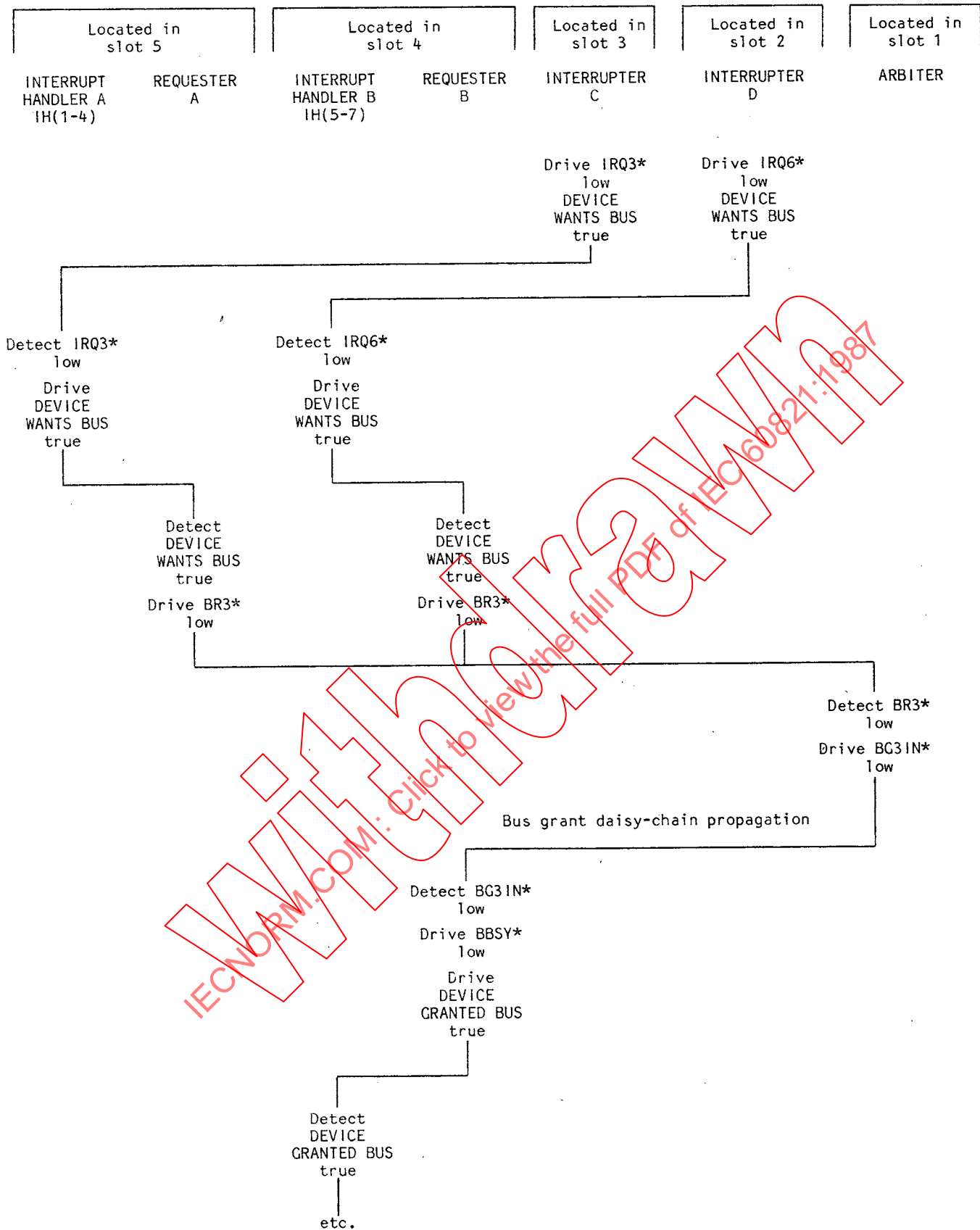


FIG. 4-15. - Typical distributed interrupt system with two INTERRUPT HANDLERS, flow diagram.

4.5 REGLES et OBSERVATIONS concernant le séquençement du bus d'interruption prioritaire

Cette section décrit les REGLES et OBSERVATIONS relatives à la chronologie qui régissent le comportement des CONTROLEURS D'INTERRUPTION, GENERATEURS D'INTERRUPTION et EMETTEURS DE CHAINE SERIE IACK pendant la sélection du GENERATEUR D'INTERRUPTION qui répond (c'est-à-dire le GENERATEUR D'INTERRUPTION qui fournit son MOT D'ETAT/IDentificateur en réponse au cycle de reconnaissance d'interruption). Cette information est donnée sous forme de figures et de tableaux.

Le cycle de reconnaissance d'interruption commence par la sélection du GENERATEUR D'INTERRUPTION répondant. Elle constitue la partie sélection du GENERATEUR D'INTERRUPTION du cycle. Lorsqu'un GENERATEUR D'INTERRUPTION répond, le CONTROLEUR D'INTERRUPTION lit le MOT D'ETAT/IDentificateur. C'est la partie transfert du MOT D'ETAT/IDentificateur du cycle.

Quand le CONTROLEUR D'INTERRUPTION lance un cycle de reconnaissance d'interruption, il peut y avoir entre lui et le GENERATEUR D'INTERRUPTION à reconnaître, d'autres GENERATEURS D'INTERRUPTION qui:

- a) n'ont pas d'interruption en attente,
- b) ou ont une interruption en attente, mais sur une ligne de demande d'interruption différente de celle en cours de reconnaissance.

Bien que ces GENERATEURS D'INTERRUPTION ne renvoient pas un MOT D'ETAT/IDentificateur, ils participent au cycle de reconnaissance d'interruption en propageant le front descendant de leur ligne IACKIN* vers leur ligne IACKOUT*. Pour cette raison, ces GENERATEURS D'INTERRUPTION sont appelés les GENERATEURS D'INTERRUPTION participants.

Le premier GENERATEUR D'INTERRUPTION sur la chaîne série ayant une interruption en attente sur la ligne de demande d'interruption en cours de reconnaissance répond par un MOT D'ETAT/IDentificateur. Pour cette raison, il est appelé le GENERATEUR D'INTERRUPTION répondant.

Tous les autres GENERATEURS D'INTERRUPTION sont appelés les GENERATEURS D'INTERRUPTION non participants.

Les tableaux 2-18 et 4-8 donnent la liste des paramètres de temps et chronogrammes qui spécifient le fonctionnement du CONTROLEUR D'INTERRUPTION et du GENERATEUR D'INTERRUPTION.

Le tableau 4-9 donne la liste des paramètres de temps et chronogrammes qui spécifient le fonctionnement de l'EMETTEUR DE CHAINE SERIE IACK et du GENERATEUR D'INTERRUPTION.

Le tableau 4-10 donne la liste des paramètres de temps et chronogrammes qui spécifient le fonctionnement des GENERATEURS D'INTERRUPTION participants.

Le tableau 4-11 donne la liste des paramètres de temps et chronogrammes qui spécifient le fonctionnement du GENERATEUR D'INTERRUPTION répondant.

4.5 *Priority Interrupt Bus timing RULES and OBSERVATIONS*

This section describes the timing RULES and OBSERVATIONS that govern the behavior of INTERRUPT HANDLERS, INTERRUPTERS, and IACK DAISY-CHAIN DRIVERS during the selection of the responding INTERRUPTER (i.e. the INTERRUPTER that is to provide its STATUS/ID in response to the interrupt acknowledge cycle). This timing information is in the form of figures and tables.

The interrupt acknowledge cycle begins with the selection of the responding INTERRUPTER. This is called the INTERRUPTER selection portion of the cycle. Once an INTERRUPTER responds, the INTERRUPT HANDLER reads the STATUS/ID from it. This is called the STATUS/ID transfer portion of the cycle.

When the INTERRUPT HANDLER initiates an interrupt acknowledge cycle, there might be INTERRUPTERS between it and the INTERRUPTER being acknowledged that either:

- a) do not have an interrupt pending,
- b) or have an interrupt pending, but are on a different interrupt request line than the one being acknowledged.

Although these INTERRUPTERS do not respond with a STATUS/ID, they do participate in the interrupt acknowledge cycle by passing the falling edge from their IACKIN* line on to their IACKOUT* line. For this reason, these INTERRUPTERS are called participating INTERRUPTERS.

The first INTERRUPTER in the daisy-chain that has an interrupt pending on the interrupt request line that is being acknowledged responds with a STATUS/ID. For this reason it is called the responding INTERRUPTER.

All other INTERRUPTERS are called non-participating INTERRUPTERS.

Tables 2-18 and 4-8 list the timing parameters and timing diagrams that specify INTERRUPT HANDLER and INTERRUPTER operation.

Table 4-9 lists the timing parameters and timing diagrams that specify IACK DAISY-CHAIN DRIVER and INTERRUPTER operation.

Table 4-10 lists the timing parameters and timing diagrams that specify participating INTERRUPTER operation.

Table 4-11 lists the timing parameters and timing diagrams that specify responding INTERRUPTER operation.

Le tableau 4-12 définit les mnémoniques utilisés dans les tableaux 4-12 à 4-15.

Les tableaux 4-13 à 4-15 spécifient l'utilisation des lignes de signaux par les modules fonctionnels du bus d'interruption prioritaire.

Les tableaux 4-16 à 4-19 spécifient les paramètres de temps pour les modules fonctionnels du bus d'interruption prioritaire. (Les numéros de référence utilisés dans les tableaux 4-17 à 4-19 correspondent aux numéros des paramètres de temps du tableau 4-16.)

Les figures 4-16 à 4-23, pages 326 à 334, sont des chronogrammes qui spécifient la chronologie pendant les cycles de reconnaissance d'interruption.

La figure 4-24, page 334, spécifie une chronologie supplémentaire inter-cycle pour la chaîne série IACKIN*/IACKOUT*.

Toutes les REGLES et OBSERVATIONS associées aux figures suivantes sont applicables aussi aux CONTROLEURS D'INTERRUPTION, aux GENERATEURS D'INTERRUPTION et aux EMETTEURS DE CHAÎNE SÉRIE IACK.

Les figures 2-22 à 2-24, pages 186 à 188, du chapitre 2 spécifient la chronologie pour les signaux de validation d'adresse et de donnée entre les cycles de transfert de données.

La figure 2-25, page 188, spécifie la chronologie d'un cycle avec dépassement de temps.

La figure 2-26, page 190, spécifie la chronologie pendant un changement de MAÎTRE.

Pour satisfaire à la chronologie spécifiée, les concepteurs de cartes doivent tenir compte des délais de propagation les plus mauvais des émetteurs et récepteurs utilisés sur leur carte de BUS CEI 821. Le délai de propagation des émetteurs dépend de la charge de leur sortie, mais les spécifications des fabricants ne donnent pas toujours assez d'informations pour calculer les délais de propagation avec des charges différentes. Pour aider le concepteur de cartes de BUS CEI 821, des suggestions sont données dans le chapitre 6.

Les OBSERVATIONS spécifient la chronologie des transitions des signaux sur les lignes entrantes. On peut se fier à ces temps pour autant que les REGLES de charge du fond de panier du chapitre 6 ne sont pas transgressées. Les REGLES pour les adaptateurs d'impédance du chapitre 6 garantissent le respect des paramètres de temps pour les lignes de signaux qui sont libérées après avoir été commandées.

Généralement, pour chaque REGLE de séquençement, on trouve une OBSERVATION correspondante. Cependant, le temps qui est garanti dans les observations peut différer du temps spécifié par la REGLE. Par exemple, une étude attentive des chronogrammes montre que le CONTROLEUR D'INTERRUPTION doit permettre un temps d'établissement pour l'adresse de 35 ns, mais qu'on ne garantit que 10 ns au GÉNÉRATEUR D'INTERRUPTION. Cela est dû au fait que les émetteurs d'adresse ne sont pas toujours capables de faire passer le signal du seuil bas au seuil haut sur la ligne de fond de panier avant que la transition ne se soit propagée jusqu'à l'extrémité du fond de panier et

Table 4-12 defines the mnemonics that are used in Tables 4-13 through 4-15.

Tables 4-13 through 4-15 specify the use of bus signal lines by the Priority Interrupt Bus functional modules.

Tables 4-16 through 4-19 specify the timing parameters for the Priority Interrupt Bus functional modules. (The reference numbers used in Tables 4-17 through 4-19 correspond to the timing parameter numbers in Table 4-16.)

Figures 4-16 through 4-23, pages 327 to 335, are timing diagrams that specify the timing during interrupt acknowledge cycles.

Figure 4-24, page 335, specifies additional inter-cycle timing for the IACKIN*/ IACKOUT* daisy-chain.

All of the RULES and OBSERVATIONS associated with the figures listed below also apply to INTERRUPT HANDLERS, INTERRUPTERS, and IACK DAISY-CHAIN DRIVERS.

Figures 2-22 through 2-24, pages 187 to 189, in Chapter 2 specify the timing for the address and data strobes between data transfer cycles.

Figure 2-25, page 189, specifies the timing of a time-out cycle.

Figure 2-26, page 191, specifies the timing during mastership transfer.

In order to meet the specified timing, board designers have to take into account the worst case propagation delays of the bus drivers and receivers used on their IEC 821 BUS boards. The propagation delay of the drivers depends on their output loads, however, manufacturers' specifications do not always give enough information to calculate the propagation delays under various loads. To help the IEC 821 BUS board designer, some suggestions are offered in Chapter 6.

The OBSERVATIONS specify the timing of incoming lines signal transitions. These times can be relied upon as long as the backplane loading RULES in Chapter 6 are not violated. The RULES for the bus terminators in Chapter 6 guarantee that the timing parameters for signal lines that are released after they have been driven, are met.

Typically, for each timing RULE there is a corresponding timing OBSERVATION. However, the time that is guaranteed in the OBSERVATIONS might differ from the time specified by the RULE. For example, a careful inspection of the timing diagrams shows that the INTERRUPT HANDLER is required to provide 35 ns of address set-up time, but the INTERRUPTER is only guaranteed 10 ns. This is because the address drivers are not always able to drive the backplane's signal lines completely through the low to high threshold region, until the transition propagates to the end of the backplane and is reflected back.

ait été réfléchi. Le front descendant du signal de validation d'adresse, cependant, franchit le seuil de 0,8 V sans attendre la réflexion. Ainsi, le temps d'établissement qui en résulte au GENERATEUR D'INTERRUPTION est égal au temps d'établissement du CONTROLEUR D'INTERRUPTION moins deux fois le temps de propagation sur le bus.

Une notation spéciale a été utilisée pour décrire la chronologie des signaux de validation de donnée. Ces deux signaux (DS0* et DS1*) ne changent pas toujours d'état en même temps. Dans les chronogrammes, DSA* représente le premier de ces signaux ayant changé d'état (que ce soit DS0* ou DS1*). DSB* représente le deuxième de ces signaux ayant changé d'état (que ce soit DS1* ou DS0*). La ligne pointillée, lorsque les signaux de validation de donnée sont stables, indique que le premier signal ayant effectué une transition négative peut ne pas être le premier à effectuer la transition positive, c'est-à-dire que DSA* peut représenter DS0* sur son front descendant et DS1* sur son front montant.

Tableau 4-8

*Chronogrammes qui définissent le fonctionnement
du CONTROLEUR D'INTERRUPTION et du
GENERATEUR D'INTERRUPTION*

(Voir tableaux 4-16 et 4-17 pour les paramètres de temps)

Mnémonique	Type de cycle	GENERATEUR D'INTERRUPTION Chronogramme de sélection	MOT D'ETAT/ IDentificateur Chronogramme de transfert
D08(0)	LECTURE DU MOT D'ETAT/IDentificateur OCTET UNIQUE	Figures 2-12, 4-16, pages 168 et 326	Figure 4-20, page 330
D16	LECTURE DU MOT D'ETAT/IDentificateur DOUBLE OCTET	Figures 2-12, 4-16	Figure 4-21, page 332
D32	LECTURE DU MOT D'ETAT/IDentificateur QUADRUPLE OCTET	Figures 2-12, 4-16	Figure 4-21

Tableau 4-9

*Chronogrammes qui définissent le fonctionnement
de l'EMETTEUR DE CHAINE SERIE IACK*

(Voir tableaux 4-16 et 4-19 pour les paramètres de temps)

Type de cycle	GENERATEUR D'INTERRUPTION Chronogramme de sélection
LECTURE DU MOT D'ETAT/IDentificateur OCTET UNIQUE	Figure 4-17, page 326
LECTURE DU MOT D'ETAT/IDentificateur DOUBLE OCTET	Figure 4-17
LECTURE DU MOT D'ETAT/IDentificateur QUADRUPLE OCTET	Figure 4-17

The falling edge of the address strobe, however, typically crosses the 0.8 V threshold without waiting for a reflection. And therefore, the resulting set-up time at the INTERRUPTER is the INTERRUPT HANDLER'S set-up time less two bus propagation times.

A special notation has been used to describe the data strobe timing. The two data strobes (DS0* and DS1*) do not always make their transitions simultaneously. For purposes of these timing diagrams, DSA* represents the first data strobe to make its transition (whether that is DS0* or DS1*). DSB* represents the second data strobe to make its transition (whether that is DS1* or DS0*). The broken line shown while the data strobes are stable indicates that the first data strobe to make a falling transition might not be the first to make its rising transition, i.e. DSA* might represent DS0* on its falling edge and DS1* on its rising edge.

Table 4-8

*Timing diagrams that define INTERRUPT HANDLER
and INTERRUPTER operation*

(See Tables 4-16 and 4-17 for timing parameters)

Mnemonic	Type of cycle	INTERRUPTER selection timing diag.	STATUS/ID transfer timing diag.
D08(0)	SINGLE BYTE STATUS/ID READ	Figures 2-12, 4-16, pages 169 and 327	Figure 4-20, page 331
D16	DOUBLE BYTE STATUS/ID READ	Figures 2-12, 4-16	Figure 4-21, page 333
D32	QUAD BYTE STATUS/ID READ	Figures 2-12, 4-16	Figure 4-21

Table 4-9

*Timing diagrams that define
IACK DAISY-CHAIN DRIVER operation*

(See Tables 4-16 and 4-19 for timing parameters)

Type of cycle	INTERRUPTER selection timing diag.
SINGLE BYTE STATUS/ID READ	Figure 4-17, page 327
DOUBLE BYTE STATUS/ID READ	Figure 4-17
QUAD BYTE STATUS/ID READ	Figure 4-17

Tableau 4-10

*Chronogrammes qui définissent le fonctionnement
du GENERATEUR D'INTERRUPTION qui participe*

(Voir tableaux 4-16 et 4-18 pour les paramètres de temps)

Type de cycle	GENERATEUR D'INTERRUPTION Chronogramme de sélection
LECTURE DU MOT D'ETAT/IDentificateur OCTET UNIQUE	Figure 4-18, page 328
LECTURE DU MOT D'ETAT/IDentificateur DOUBLE OCTET	Figure 4-18
LECTURE DU MOT D'ETAT/IDentificateur QUADRUPLE OCTET	Figure 4-18

Tableau 4-11

*Chronogrammes qui définissent le fonctionnement
du GENERATEUR D'INTERRUPTION qui répond*

(Voir tableaux 4-16 et 4-18 pour les paramètres de temps)

Mnémonique	Type de cycle	GENERATEUR D'INTERRUPTION Chronogramme de sélection	MOT D'ETAT/ IDentificateur Chronogramme de transfert
D08(0)	LECTURE DU MOT D'ETAT/IDentificateur OCTET UNIQUE	Figure 4-19, page 330	Figure 4-22, page 332
D16	LECTURE DU MOT D'ETAT/IDentificateur DOUBLE OCTET	Figure 4-19	Figure 4-23, page 334
D32	LECTURE DU MOT D'ETAT/IDentificateur QUADRUPLE OCTET	Figure 4-19	Figure 4-23

Table 4-10

*Timing diagrams that define
participating INTERRUPTER operation*

(See Tables 4-16 and 4-18 for timing parameters)

Type of cycle	INTERRUPTER Selection timing diag.
SINGLE BYTE STATUS/ID READ	Figure 4-18, page 329
DOUBLE BYTE STATUS/ID READ	Figure 4-18
QUAD BYTE STATUS/ID READ	Figure 4-18

Table 4-11

*Timing diagrams that define
responding INTERRUPTER operation*

(See Tables 4-16 and 4-18 for timing parameters)

Mnemonic	Type of cycle	INTERRUPTER Selection timing diag.	STATUS/ID Transfer timing diag.
D08(0)	SINGLE BYTE STATUS/ID READ	Figure 4-19, page 331	Figure 4-22, page 333
D16	DOUBLE BYTE STATUS/ID READ	Figure 4-19	Figure 4-23, page 335
D32	QUAD BYTE STATUS/ID READ	Figure 4-19	Figure 4-23

Tableau 4-12

*Définitions des mnémoniques utilisés
dans les tableaux 4-13, 4-14 et 4-15*

Mnémonique	Description	Observations
DLBIH	COMMANDE AU NIVEAU BAS PAR LE CONTRÔLEUR D'INTERRUPTION	REGLE 4.10: Le CONTRÔLEUR D'INTERRUPTION DOIT commander les lignes DLBIH au niveau bas
DHBIH	COMMANDE AU NIVEAU HAUT PAR LE CONTRÔLEUR D'INTERRUPTION	REGLE 4.11: Le CONTRÔLEUR D'INTERRUPTION DOIT commander les lignes DHBIH au niveau haut
dhbih?	COMMANDE AU NIVEAU HAUT PAR LE CONTRÔLEUR D'INTERRUPTION?	AUTORISATION 4.7: Le CONTRÔLEUR D'INTERRUPTION PEUT commander les lignes dhbih? au niveau haut. REGLE 4.12: Le CONTRÔLEUR D'INTERRUPTION NE DOIT PAS commander les lignes dhbih? au niveau bas pendant le cycle
DVBI	COMMANDE A L'ETAT VALIDE PAR LE GÉNÉRATEUR D'INTERRUPTION	REGLE 4.13: Le GÉNÉRATEUR D'INTERRUPTION DOIT commander les lignes DVBI à un niveau valide
dhbi?	COMMANDE PAR LE GÉNÉRATEUR D'INTERRUPTION?	AUTORISATION 4.9: Le GÉNÉRATEUR D'INTERRUPTION PEUT commander les lignes dhbi? au niveau haut. REGLE 4.14: Le GÉNÉRATEUR D'INTERRUPTION NE DOIT PAS commander les lignes dhbi? au niveau bas

Tableau 4-13

*Utilisation des lignes d'adresse pendant les cycles
de reconnaissance d'interruption*

Ligne d'interruption en cours d'acquittement	A03	A02	A01	IACK*
IRQ1*	DLBIH	DLBIH	DHBIH	DLBIH
IRQ2*	DLBIH	DHBIH	DLBIH	DLBIH
IRQ3*	DLBIH	DHBIH	DHBIH	DLBIH
IRQ4*	DHBIH	DLBIH	DLBIH	DLBIH
IRQ5*	DHBIH	DLBIH	DHBIH	DLBIH
IRQ6*	DHBIH	DHBIH	DLBIH	DLBIH
IRQ7*	DHBIH	DHBIH	DHBIH	DLBIH

Table 4-12

*Definitions of mnemonics
used in Tables 4-13, 4-14 and 4-15*

Mnemonic	Description	Comments
DLBIH	DRIVEN LOW BY INTERRUPT HANDLER	RULE 4.10: The INTERRUPT HANDLER MUST drive DLBIH lines to a low level
DHBIH	DRIVEN HIGH BY INTERRUPT HANDLER	RULE 4.11: The INTERRUPT HANDLER MUST drive DHBIH lines to a high level
dhbih?	DRIVEN HIGH BY INTERRUPT HANDLER?	PERMISSION 4.7: The INTERRUPT HANDLER MAY drive dhbih? lines high. RULE 4.12: The INTERRUPT HANDLER MUST NOT drive dhbih? lines low during the cycle
DVBI	DRIVEN VALID BY INTERRUPTER	RULE 4.13: The INTERRUPTER MUST drive DVBI lines to a valid level
dhbi?	DRIVEN BY INTERRUPTER?	PERMISSION 4.9: The INTERRUPTER MAY drive dhbi? lines high. RULE 4.14: The INTERRUPTER MUST NOT drive dhbi? lines low

Table 4-13

*Use of addressing lines
during interrupt acknowledge cycles*

Interrupt line being acknowledged	A03	A02	A01	IACK*
IRQ1*	DLBIH	DLBIH	DHBIH	DLBIH
IRQ2*	DLBIH	DHBIH	DLBIH	DLBIH
IRQ3*	DLBIH	DHBIH	DHBIH	DLBIH
IRQ4*	DHBIH	DLBIH	DLBIH	DLBIH
IRQ5*	DHBIH	DLBIH	DHBIH	DLBIH
IRQ6*	DHBIH	DHBIH	DLBIH	DLBIH
IRQ7*	DHBIH	DHBIH	DHBIH	DLBIH

Tableau 4-14

Utilisation des lignes DS1, DS0*, LWORD* et WRITE* pendant les cycles de reconnaissance d'interruption*

Mnémonique	Type de cycle	DS1*	DS0*	LWORD*	WRITE*
D08(0)	Reconnaissance d'interruption à octet unique	dhbih?	DLBIH	dhbih?	dhbih?
D16	Reconnaissance d'interruption à double octet	DLBIH	DLBIH	dhbih?	dhbih?
D32	Reconnaissance d'interruption à quadruple octet	DLBIH	DLBIH	DLBIH	dhbih?

Tableau 4-15

Utilisation des lignes du bus de données pour le transfert du MOT D'ETAT/Identificateur

Mnémonique	Type de cycle	D24-D31	D16-D23	D08-D15	D00-D07
D08(0)	Reconnaissance d'interruption à simple, double ou quadruple octet	dhbi?	dhbi?	dhbi?	DVBI
D16	Reconnaissance d'interruption à double ou quadruple octet	dhbi?	dhbi?	DVBI	DVBI
D32	Reconnaissance d'interruption à quadruple octet	DVBI	DVBI	DVBI	DVBI

Table 4-14

Use of the DS1, DS0*, LWORD*, and WRITE* lines
during interrupt acknowledge cycles*

Mnemonic	Type of cycle	DS1*	DS0*	LWORD*	WRITE*
D08(0)	Single byte interrupt acknowledge	dhbih?	DLBIH	dhbih?	dhbih?
D16	Double byte interrupt acknowledge	DLBIH	DLBIH	dhbih?	dhbih?
D32	Quad byte interrupt acknowledge	DLBIH	DLBIH	DLBIH	dhbih?

Table 4-15

*Use of the data bus lines
to transfer the STATUS/ID*

Mnemonic	Type of cycle	D24-D31	D16-D23	D08-D15	D00-D07
D08(0)	Single, double and quad byte interrupt acknowledge	dhbi?	dhbi?	dhbi?	DVBI
D16	Double and quad byte interrupt acknowledge	dhbi?	dhbi?	DVBI	DVBI
D32	Quad byte interrupt acknowledge	DVBI	DVBI	DVBI	DVBI

Tableau 4-16

Paramètres de temps des chronogrammes
du CONTROLEUR D'INTERRUPTION, du GENERATEUR
D'INTERRUPTION et de l'EMETTEUR DE CHAINE SERIE IACK

NUMERO DU PARAMETRE	CONTROLEUR D'INTERRUPTION Voir tableau 4-17		GENERATEUR D'INTERRUPTION Voir tableau 4-18		EMETTEUR DE CHAINE SERIE IACK Voir tableau 4-19	
	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.
1	0	-	-	-	-	-
2	0	-	-	-	-	-
3	60	-	-	-	-	-
4	35	-	10	-	-	-
5	40	-	30	-	30	-
6	-	-	0	-	-	-
7	-	-	0	-	-	-
9	0	-	0	-	-	-
10	0	-	-	-	-	-
11	40	-	30	-	-	-
12	35	-	10	-	-	-
13	-	10	-	20	-	-
14	0	-	0	-	-	-
16	0	-	0	-	-	-
18	0	-	0	-	-	-
19	40	-	30	-	30	-
20	0	-	0	-	-	-
21	0	-	0	-	-	-
23	10	-	0	-	-	-
24A	0	-	-	-	-	-
24B	0	-	-	-	-	-
25	-	25	-	-	-	-
26	0	-	0	-	-	-
27	-25	-	0	-	-	-
28	30	2T	30	-	-	-
29	0	-	0	-	-	-
30	0	-	0	-	-	-
31	0	-	0	-	-	-
32	-	-	10	-	10	-
34	-	-	30	-	40	-
35	-	-	-	30	-	30
36	-	-	0	-	-	-
37	-	-	0	-	-	-
38A	-	-	0	-	-	-
38B	-	-	0	-	-	-
39	-	-	-	40	-	-
40	-	-	30	-	30	-
41	-	-	0	-	-	-
42	-	-	-	-	30	-
43	-	-	30	-	-	-

Notes:

- 1.- Tous les temps sont en nanosecondes.
- 2.- T = valeur du temps maximal d'occupation du bus, en microsecondes.

Table 4-16

INTERRUPT HANDLER, INTERRUPTER,
and IACK DAISY-CHAIN DRIVER *timing parameters*

PARAMETER NUMBER	INTERRUPT HANDLER See Table 4-17		INTERRUPTER See Table 4-18		IACK DAISY-CHAIN DRIVER See Table 4-19	
	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.
1	0	-	-	-	-	-
2	0	-	-	-	-	-
3	60	-	-	-	-	-
4	35	-	10	-	-	-
5	40	-	30	-	30	-
6	-	-	0	-	-	-
7	-	-	0	-	-	-
9	0	-	0	-	-	-
10	0	-	-	-	-	-
11	40	-	30	-	-	-
12	35	-	10	-	-	-
13	-	10	-	20	-	-
14	0	-	0	-	-	-
16	0	-	0	-	-	-
18	0	-	0	-	-	-
19	40	-	30	-	30	-
20	0	-	0	-	-	-
21	0	-	0	-	-	-
23	10	-	0	-	-	-
24A	0	-	-	-	-	-
24B	0	-	-	-	-	-
25	-	25	-	-	-	-
26	0	-	0	-	-	-
27	-25	-	0	-	-	-
28	30	21	30	-	-	-
29	-	-	0	-	-	-
30	0	-	0	-	-	-
31	0	-	0	-	-	-
32	-	-	10	-	10	-
34	-	-	30	-	40	-
35	-	-	-	30	-	30
36	-	-	0	-	-	-
37	-	-	0	-	-	-
38A	-	-	0	-	-	-
38B	-	-	0	-	-	-
39	-	-	-	40	-	-
40	-	-	30	-	30	-
41	-	-	0	-	-	-
42	-	-	-	-	30	-
43	-	-	30	-	-	-

Notes:

- 1.- All times are in nanoseconds.
- 2.- T = time-out value, in microseconds.

Tableau 4-17

CONTROLEUR D'INTERRUPTION
REGLES et OBSERVATIONS sur la chronologie

Note: La numérotation correspond aux paramètres de temps spécifiés dans le tableau 4-16.

1. REGLE 4.15:

En prenant le contrôle du BUS CEI 821, le CONTROLEUR D'INTERRUPTION NE DOIT commander aucun des signaux IACK*, A01-A03, LWORD*, WRITE*, DS0*, DS1* ou AS*, avant que le MAITRE ou le CONTROLEUR D'INTERRUPTION précédent ait permis à AS* de passer au-dessus du niveau bas.

OBSERVATION 4.12:

Le chapitre 3 décrit comment un DEMANDEUR de CONTROLEUR D'INTERRUPTION reçoit le contrôle du BUS CEI 821.

2. REGLE 4.16:

En prenant le contrôle du BUS CEI 821, le CONTROLEUR D'INTERRUPTION NE DOIT commander aucun des signaux IACK*, A01-A03, LWORD*, WRITE*, DS0*, DS1* ou AS* avant d'avoir reçu le signal DISPOSITIF RECOIT LE BUS à l'état vrai.

OBSERVATION 4.13:

Le chapitre 3 décrit comment un DEMANDEUR de CONTROLEUR D'INTERRUPTION reçoit le contrôle du BUS CEI 821.

3. REGLE 4.17:

En prenant le contrôle du BUS CEI 821, le CONTROLEUR D'INTERRUPTION NE DOIT PAS commander AS* à l'état bas avant ce délai après que le MAITRE ou le CONTROLEUR D'INTERRUPTION précédent eut permis à AS* de passer au-dessus du niveau bas.

OBSERVATION 4.14:

La REGLE 4.17 assure que le paramètre de temps 5 est garanti pour les GENERATEURS D'INTERRUPTION et les ESCLAVES lors d'un changement de maître du DTB.

4. REGLE 4.18:

Le CONTROLEUR D'INTERRUPTION NE DOIT PAS commander AS* à l'état bas avant que IACK* ait été au niveau bas et que LWORD* et A01-A03 aient été valides au minimum pendant ce temps.

5. REGLE 4.19:

Le CONTROLEUR D'INTERRUPTION NE DOIT PAS commander AS* au niveau bas si ce signal n'est pas resté au niveau haut au minimum pendant ce temps, lorsqu'il utilise le DTB pendant deux cycles consécutifs.

9. REGLE 4.20:

Le CONTROLEUR D'INTERRUPTION NE DOIT PAS commander DSA* au niveau bas avant que les deux signaux DTACK* et BERR* soient au niveau haut.

10. REGLE 4.21:

Le CONTROLEUR D'INTERRUPTION NE DOIT PAS commander DSA* au niveau bas avant d'avoir commandé AS* au niveau bas.

Table 4-17

INTERRUPT HANDLER
Timing RULES and OBSERVATIONS

Note: The numbers correspond to the timing parameters specified in Table 4-16.

1. RULE 4.15:

When taking control of the IEC 821 BUS, the INTERRUPT HANDLER MUST NOT drive any of IACK*, A01-A03, LWORD*, WRITE*, DS0*, DS1* or AS* until after the previous MASTER or INTERRUPT HANDLER allows AS* to rise above the low level.

OBSERVATION 4.12:

Chapter 3 describes how an INTERRUPT HANDLER'S REQUESTER is granted use of the IEC 821 BUS.

2. RULE 4.16:

When taking control of the IEC 821 BUS, the INTERRUPT HANDLER MUST NOT drive any of IACK*, A01-A03, LWORD*, WRITE*, DS0*, DS1* or AS* until after it receives DEVICE GRANTED BUS true.

OBSERVATION 4.13:

Chapter 3 describes how an INTERRUPT HANDLER'S REQUESTER is granted use of the IEC 821 BUS.

3. RULE 4.17:

When taking control of the IEC 821 BUS, the INTERRUPT HANDLER MUST NOT drive AS* low until this time after the previous MASTER or INTERRUPT HANDLER allows AS* to rise above the low level.

OBSERVATION 4.14:

RULE 4.17 ensures that timing parameter 5 for INTERRUPTERS and SLAVES is guaranteed when there is an interchange of the DTB mastership.

4. RULE 4.18:

The INTERRUPT HANDLER MUST NOT drive AS* low until IACK* has been low, and LWORD* and A01-A03 have been valid for this minimum time.

5. RULE 4.19:

When using the DTB for two consecutive cycles, the INTERRUPT HANDLER MUST NOT drive AS* low until it has been high for this minimum time.

9. RULE 4.20:

The INTERRUPT HANDLER MUST NOT drive DSA* low until both DTACK* and BERR* are high.

10. RULE 4.21:

The INTERRUPT HANDLER MUST NOT drive DSA* low before it has driven AS* low.

Tableau 4-17 (suite)

11. REGLE 4.22:
Le CONTROLEUR D'INTERRUPTION NE DOIT PAS commander DSA* au niveau bas avant que DS0* et DS1* aient été simultanément au niveau haut au minimum pendant ce temps.
12. REGLE 4.23:
Le CONTROLEUR D'INTERRUPTION NE DOIT PAS commander DSA* au niveau bas avant que WRITE* n'ait été au niveau haut au minimum pendant ce temps.
13. REGLE 4.24:
Pendant des cycles de lecture de double ou quadruple octets de MOT D'ETAT/IDentificateur, le CONTROLEUR D'INTERRUPTION DOIT commander DSB* au niveau bas dans ce délai après avoir commandé DSA* au niveau bas.
- OBSERVATION 4.15:
Le paramètre de temps 13 ne s'applique pas aux lectures d'octet unique de MOT D'ETAT/IDentificateur.
14. REGLE 4.25:
Pendant tous les cycles de reconnaissance d'interruption, le CONTROLEUR D'INTERRUPTION DOIT maintenir valide le code de reconnaissance d'interruption de 3 bits et le niveau approprié sur LWORD* jusqu'à ce qu'il détecte un front descendant sur DTACK* ou BERR*.
16. REGLE 4.26:
Pendant tous les cycles de reconnaissance d'interruption, le CONTROLEUR D'INTERRUPTION doit maintenir IACK* au niveau bas jusqu'à ce qu'il détecte un front descendant sur DTACK* ou BERR*.
18. REGLE 4.27:
Le CONTROLEUR D'INTERRUPTION DOIT maintenir AS* au niveau bas jusqu'à ce qu'il détecte DTACK* ou BERR* au niveau bas.
19. REGLE 4.28:
Le CONTROLEUR D'INTERRUPTION DOIT maintenir AS* au niveau bas au minimum pendant ce temps.
20. REGLE 4.29:
Lorsqu'un CONTROLEUR D'INTERRUPTION a commandé DSA* au niveau bas, il DOIT le maintenir bas jusqu'à ce qu'il détecte DTACK* ou BERR* au niveau bas.
21. REGLE 4.30:
Lorsqu'un CONTROLEUR D'INTERRUPTION a commandé DSB* au niveau bas, il DOIT le maintenir bas jusqu'à ce qu'il détecte DTACK* ou BERR* au niveau bas.
23. REGLE 4.31:
Lorsqu'un CONTROLEUR D'INTERRUPTION a commandé DSA* au niveau bas, il DOIT maintenir un niveau haut sur la ligne WRITE* au minimum pendant ce temps après avoir commandé les deux signaux de validation de donnée au niveau haut.

Table 4-17 (continued)

11. RULE 4.22:
The INTERRUPT HANDLER MUST NOT drive DSA* low until DS0* and DS1* have been simultaneously high for this minimum time.
12. RULE 4.23:
The INTERRUPT HANDLER MUST NOT drive DSA* low until WRITE* has been high for this minimum time.
13. RULE 4.24:
During double byte or quad byte STATUS/ID read cycles, the INTERRUPT HANDLER MUST drive DSB* low within this maximum time after it drives DSA* low.
- OBSERVATION 4.15:
Timing parameter 13 does not apply to single byte STATUS/ID reads.
14. RULE 4.25:
During all interrupt acknowledge cycles, the INTERRUPT HANDLER MUST hold the 3-bit interrupt acknowledge code valid, and maintain the appropriate level on LWORD* until it detects a falling edge on DTACK* or BERR*.
16. RULE 4.26:
During all interrupt acknowledge cycles, the INTERRUPT HANDLER MUST maintain IACK* low until it detects a falling edge on DTACK* or BERR*.
18. RULE 4.27:
The INTERRUPT HANDLER MUST hold AS* low until it detects DTACK* or BERR* low.
19. RULE 4.28:
The INTERRUPT HANDLER MUST hold AS* low for this minimum time.
20. RULE 4.29:
Once an INTERRUPT HANDLER has driven DSA* low, then it MUST maintain it low until it detects DTACK* or BERR* low.
21. RULE 4.30:
Once an INTERRUPT HANDLER has driven DSB* low, then it MUST maintain it low until it detects DTACK* or BERR* low.
23. RULE 4.31:
Once an INTERRUPT HANDLER has driven DSA* low, it MUST maintain a high on the WRITE* line, until this minimum time after both data strobes are high.

Tableau 4-17 (fin)

24A. REGLE 4.32:

SI le CONTROLEUR D'INTERRUPTION commande ou libère AS* au niveau haut après que son DEMANDEUR eut libéré BBSY*,
ALORS il DOIT libérer IACK*, A01-A03, LWORD*, WRITE*, DS0* et DS1* avant de permettre à AS* de passer au-dessus du niveau bas.

24B. REGLE 4.33:

SI le CONTROLEUR D'INTERRUPTION commande ou libère AS* au niveau haut avant que son DEMANDEUR ait libéré BBSY*,
ALORS il DOIT libérer IACK*, A01-A03, LWORD*, WRITE*, DS0* et DS1* avant de changer son signal DISPOSITIF DEMANDE LE BUS de l'état vrai à l'état faux.

25. REGLE 4-34:

SI le CONTROLEUR D'INTERRUPTION commande ou libère AS* au niveau haut après que son DEMANDEUR eut libéré BBSY*,
ALORS il DOIT libérer AS* dans ce délai après lui avoir permis de passer au-dessus du niveau bas:

26. OBSERVATION 4.16:

Le paramètre de temps 26 garantit que le bus de données ne sera pas commandé avant que le CONTROLEUR D'INTERRUPTION ne commande DSA* au niveau bas.

27. OBSERVATION 4.17:

Le CONTROLEUR D'INTERRUPTION est assuré que le bus de données sera valide dans ce délai après le passage du signal DTACK* au niveau bas. Ce temps ne s'applique pas aux cycles où le GENERATEUR D'INTERRUPTION commande BERR* au niveau bas au lieu de commander DTACK*.

28. OBSERVATION 4.18:

Le CONTROLEUR D'INTERRUPTION est assuré que ni le signal DTACK* ni BERR* ne passeront au niveau bas avant ce délai minimal après avoir commandé DSA* au niveau bas. Le LIMITEUR DE TEMPS D'OCCUPATION DU BUS garantit au CONTROLEUR D'INTERRUPTION que si DTACK* n'est pas passé au niveau bas dans un délai compris entre la limite de temps et le double de cette limite, alors le LIMITEUR DE TEMPS D'OCCUPATION DU BUS commande BERR* au niveau bas.

29. OBSERVATION 4.19:

Le CONTROLEUR D'INTERRUPTION est assuré que le bus de données reste valide jusqu'à ce qu'il commande DSA* au niveau haut.

30. OBSERVATION 4.20:

Ce délai garantit que ni DTACK* ni BERR* ne passent au niveau haut avant que le CONTROLEUR D'INTERRUPTION ait commandé les deux signaux DS0* et DS1* au niveau haut.

31. OBSERVATION 4.21:

Le CONTROLEUR D'INTERRUPTION est assuré que le bus de données a été libéré au moment où les signaux DTACK* et BERR* passent au niveau haut.

Table 4-17 (concluded)

- 24A. RULE 4.32:
IF the INTERRUPT HANDLER drives or releases AS* to high after its REQUESTER releases BBSY*,
THEN it MUST release IACK*, A01-A03, LWORD*, WRITE*, DS0*, and DS1* before allowing AS* to rise above the low level.
- 24B. RULE 4.33:
IF the INTERRUPT HANDLER drives or releases AS* to high before its REQUESTER releases BBSY*,
THEN it MUST release IACK*, A01-A03, LWORD*, WRITE*, DS0* and DS1* before changing its DEVICE WANTS BUS signal from true to false.
25. RULE 4.34:
IF the INTERRUPT HANDLER drives or releases AS* to high after its REQUESTER releases BBSY*,
THEN it MUST release AS* within this time after allowing it to rise above the low level.
26. OBSERVATION 4.16:
Timing parameter 26 guarantees that the data bus will not be driven until the INTERRUPT HANDLER drives DSA* low.
27. OBSERVATION 4.17:
The INTERRUPT HANDLER is guaranteed that the data bus will be valid within this time after DTACK* does low. This time does not apply to cycles where the INTERRUPTER drives BERR* low instead of DTACK*.
28. OBSERVATION 4.18:
The INTERRUPT HANDLER is guaranteed that neither DTACK* nor BERR* will go low until this minimum time after it drives DSA* low. The BUS TIMER guarantees the INTERRUPT HANDLER that if DTACK* has not gone low after its time-out period has elapsed, and within twice its time-out period, then the BUS TIMER will drive BERR* low.
29. OBSERVATION 4.19:
The INTERRUPT HANDLER is guaranteed that the data bus remains valid until it drives DSA* high.
30. OBSERVATION 4.20:
This guarantees that neither DTACK* nor BERR* goes high until the INTERRUPT HANDLER drives both DS0* and DS1* high.
31. OBSERVATION 4.21:
The INTERRUPT HANDLER is guaranteed that the data bus has been released by the time DTACK* and BERR* are high.

Tableau 4-18

GENERA TEUR D'INTERRUPTION
REGLES et OBSERVATIONS sur la chronologie

Note: La numérotation correspond aux paramètres de temps spécifiés dans le tableau 4-16.

4. OBSERVATION 4.22:
Les GENERATEURS D'INTERRUPTION sont assurés que IACK*, LWORD* et A01-A03 ont été valides au minimum pendant ce temps lorsqu'ils détectent un front descendant sur AS*.
5. OBSERVATION 4.23:
Tous les GENERATEURS D'INTERRUPTION sont assurés que AS* est au niveau haut au minimum pendant ce temps entre les cycles du DTB.
6. OBSERVATION 4.24:
Le GENERATEUR D'INTERRUPTION qui répond est assuré qu'aucune ligne D00-D31 ne sera commandée par aucun autre module avant que le GENERATEUR D'INTERRUPTION qui répond libère DTACK* et BERR* au niveau haut.
7. OBSERVATION 4.25:
Le GENERATEUR D'INTERRUPTION qui répond est assuré que le bus de données sera libéré par tous les autres modules lorsque DSA* passera au niveau bas.
9. OBSERVATION 4.26:
Le GENERATEUR d'INTERRUPTION qui répond est assuré que ni DS0* ni DS1* ne passeront au niveau bas avant que DTACK* et BERR* du cycle précédent soient passés au niveau haut.
11. OBSERVATION 4.27:
Les GENERATEURS D'INTERRUPTION sont assurés que les deux signaux de validation de donnée sont simultanément au niveau haut entre deux cycles au minimum pendant ce temps.
12. OBSERVATION 4.28:
Les GENERATEURS D'INTERRUPTION sont assurés que WRITE* est au niveau haut au minimum pendant ce temps lorsqu'ils détectent un front descendant sur DSA*.
13. OBSERVATION 4.29:
SI les deux signaux de validation de donnée vont être commandés au niveau bas,
ALORS le GENERATEUR D'INTERRUPTION qui répond est assuré que DSB* passera au niveau bas dans ce délai maximal après DSA*.
Par conséquent:
SI le signal DSB* ne passe pas au niveau bas dans ce délai maximal,
ALORS le GENERATEUR D'INTERRUPTION qui répond en déduit qu'il doit répondre avec un octet unique de MOT D'ETAT/IDentificateur.
14. OBSERVATION 4.30:
Le GENERATEUR D'INTERRUPTION qui répond est assuré que LWORD* et A01-A03 resteront valides jusqu'à ce qu'il commande DTACK* ou BERR* au niveau bas et pourvu qu'il le fasse dans la limite de temps d'occupation du bus.

Table 4-18

INTERRUPTER
Timing RULES and OBSERVATIONS

Note: The numbers correspond to the timing parameters specified in Table 4.16.

4. OBSERVATION 4.22:
INTERRUPTERS are guaranteed that IACK*, LWORD* and A01-A03 have been valid for this minimum time when they detect a falling edge on AS*.
5. OBSERVATION 4.23:
All INTERRUPTERS are guaranteed this minimum high time on AS* between DTB cycles.
6. OBSERVATION 4.24:
The responding INTERRUPTER is guaranteed that none of D00-D31 will be driven by any other module until the responding INTERRUPTER releases DTACK* and BERR* to high.
7. OBSERVATION 4.25:
The responding INTERRUPTER is guaranteed that the data bus will be released by all other modules by the time DSA* goes low.
9. OBSERVATION 4.26:
The responding INTERRUPTER is guaranteed that neither DS0* nor DS1* will go low until DTACK* and BERR* from the previous cycle have gone high.
11. OBSERVATION 4.27:
INTERRUPTERS are guaranteed this minimum time during which both data strobes are simultaneously high between cycles.
12. OBSERVATION 4.28:
INTERRUPTERS are guaranteed that WRITE* has been high for this minimum time when they detect a falling edge on DSA*.
13. OBSERVATION 4.29:
IF both data strobes are going to be driven low,

THEN the responding INTERRUPTER is guaranteed that DSB* will go low within this maximum time after DSA*.
And therefore:
IF DSB* does not go low within this maximum time,

THEN the responding INTERRUPTER assumes that it is to respond with a single byte STATUS/ID.
14. OBSERVATION 4.30:
The responding INTERRUPTER is guaranteed that LWORD* and A01-A03 will remain valid until it drives DTACK* or BERR* low, provided it does so within the bus time-out period.

Tableau 4-18 (suite)

16. OBSERVATION 4.31:

Le GENERATEUR D'INTERRUPTION qui répond est assuré que IACK* restera au niveau bas jusqu'à ce qu'il commande DTACK* et BERR* au niveau bas et pourvu qu'il le fasse dans la limite de temps d'occupation du bus.

18. OBSERVATION 4.32:

Le GENERATEUR D'INTERRUPTION qui répond est assuré que AS* sera maintenu au niveau bas jusqu'à ce qu'il commande DTACK* ou BERR* au niveau bas, pourvu qu'il le fasse dans la limite de temps d'occupation du bus.

19. OBSERVATION 4.33:

Les GENERATEURS D'INTERRUPTION sont assurés que AS* sera maintenu au niveau bas au minimum pendant ce temps.

20. OBSERVATION 4.34:

Le GENERATEUR D'INTERRUPTION qui répond est assuré que lorsque DSA* passera au niveau bas, il restera à ce niveau jusqu'à ce que le GENERATEUR D'INTERRUPTION ait commandé DTACK* ou BERR* au niveau bas, pourvu que le GENERATEUR D'INTERRUPTION le fasse dans la limite de temps d'occupation du bus.

21. OBSERVATION 4.35:

Le GENERATEUR D'INTERRUPTION qui répond est assuré que lorsque DSB* passera au niveau bas, il restera à ce niveau jusqu'à ce que le GENERATEUR D'INTERRUPTION ait commandé DTACK* ou BERR* au niveau bas, pourvu que le GENERATEUR D'INTERRUPTION le fasse dans la limite de temps d'occupation du bus.

23. OBSERVATION 4.36:

Les GENERATEURS D'INTERRUPTION sont assurés que la ligne WRITE* reste au niveau haut jusqu'à ce que les deux signaux de validation de donnée passent au niveau haut.

26. REGLE 4.35:

Le GENERATEUR D'INTERRUPTION qui répond NE DOIT PAS commander le bus de données avant que DSA* soit passé au niveau bas.

27. REGLE 4.36:

Le GENERATEUR D'INTERRUPTION qui répond NE DOIT PAS commander DTACK* au niveau bas avant d'avoir commandé les lignes de données avec un MOT D'ETAT/IDentificateur valide.

OBSERVATION 4.37:

Ce temps ne s'applique pas aux cycles où le GENERATEUR D'INTERRUPTION commande BERR* au niveau bas au lieu de DTACK*.

28. REGLE 4.37:

Le GENERATEUR D'INTERRUPTION qui répond DOIT attendre ce délai minimal après le passage de DSA* au niveau bas avant de commander DTACK* ou BERR* au niveau bas.

29. REGLE 4.38:

Lorsque le GENERATEUR D'INTERRUPTION qui répond a commandé DTACK* au niveau bas, il NE DOIT PAS changer D00-D31 avant que DSA* soit passé au niveau haut.

Table 4-18 (continued)

16. OBSERVATION 4.31:
The responding INTERRUPTER is guaranteed that IACK* will remain low until it drives DTACK* or BERR* low, provided it does so within the bus time-out period.
18. OBSERVATION 4.32:
The responding INTERRUPTER is guaranteed that AS* will remain low until it drives DTACK* or BERR* low, provided that it does so within the bus time-out period.
19. OBSERVATION 4.33:
INTERRUPTERS are guaranteed that the AS* will remain low for this minimum time.
20. OBSERVATION 4.34:
The responding INTERRUPTER is guaranteed that once DSA* goes low, it will remain low until the INTERRUPTER has driven DTACK* or BERR* low, provided that the INTERRUPTER does so within the bus time-out period.
21. OBSERVATION 4.35:
The responding INTERRUPTER is guaranteed that once DSB* goes low, it will remain low until the INTERRUPTER has driven DTACK* or BERR* low, provided that the INTERRUPTER does so within the bus time-out period.
23. OBSERVATION 4.36:
INTERRUPTERS are guaranteed that the WRITE* line remains high until both data strobes are high.
26. RULE 4.35:
The INTERRUPTER MUST NOT drive the data bus until DSA* goes low.
27. RULE 4.36:
The responding INTERRUPTER MUST NOT drive DTACK* low before it drives the data lines with a valid STATUS/ID.
- OBSERVATION 4.37:
This time does not apply to cycles where the INTERRUPTER drives BERR* low instead of DTACK*.
28. RULE 4.37:
The responding INTERRUPTER MUST wait this minimum time after DSA* goes low before driving DTACK* or BERR* low.
29. RULE 4.38:
Once the responding INTERRUPTER has driven DTACK* low, it MUST NOT change D00-D31 until DSA* goes high.

Tableau 4-18 (suite)

30. REGLE 4.39:
Lorsque le GENERATEUR D'INTERRUPTION qui répond a commandé DTACK* ou BERR* au niveau bas, il NE DOIT PAS les libérer avant de détecter à la fois DS0* et DS1* au niveau haut.
31. REGLE 4.40:
Le GENERATEUR D'INTERRUPTION qui répond DOIT libérer toutes les lignes D00-D31 avant de libérer DTACK* et BERR* au niveau haut.
32. OBSERVATION 4.38:
Le GENERATEUR D'INTERRUPTION qui répond est assuré que IACK*, LWORD* et A01-A03 ont été valides au minimum pendant ce temps quand il détecte un front descendant sur DSA*. Ce temps se déduit des paramètres de temps 4 et 10.
34. OBSERVATION 4.39:
Le GENERATEUR D'INTERRUPTION est assuré que AS* est au niveau bas au minimum pendant ce temps quand il détecte un front descendant sur IACKIN*.
35. REGLE 4.41:
Un GENERATEUR D'INTERRUPTION qui participe DOIT commander sa ligne IACKOUT* au niveau haut dans ce délai maximal après le front montant sur AS*.
36. REGLE 4.42:
Le GENERATEUR D'INTERRUPTION NE DOIT PAS commander le bus de données avant que sa ligne IACKIN* ne passe au niveau bas.
37. REGLE 4.43:
SI un GENERATEUR D'INTERRUPTION qui participe commande le bus de données,
ALORS il doit le libérer avant de commander sa ligne IACKOUT* au niveau bas.
- 38A. REGLE 4.44:
Un GENERATEUR D'INTERRUPTION qui participe NE DOIT PAS commander sa ligne IACKOUT* au niveau bas avant d'avoir détecté sa ligne IACKIN* au niveau bas.
- 38B. REGLE 4.45:
Le GENERATEUR D'INTERRUPTION qui répond NE DOIT PAS commander sa ligne DTACK* au niveau bas avant d'avoir détecté sa ligne IACKIN* au niveau bas.
39. OBSERVATION 4.40:
Ce temps garantit que chaque ligne IACKIN* d'un GENERATEUR D'INTERRUPTION passera au niveau haut dans ce délai après le front montant sur AS*. Ce temps se déduit du paramètre de temps 35 où l'EMETTEUR DE CHAINE SERIE IACK et les GENERATEURS D'INTERRUPTION qui participent doivent commander leur ligne IACKOUT* au niveau haut dans ce délai maximal.
40. OBSERVATION 4.41:
Tous les GENERATEURS D'INTERRUPTION sont assurés que leur ligne IACKIN* restera au niveau haut pendant cette durée minimale entre deux cycles consécutifs du DTB.

Table 4-18 (continued)

30. RULE 4.39:
Once the responding INTERRUPTER has driven DTACK* or BERR* to low, it MUST NOT release it until it detects both DS0* and DS1* high.
31. RULE 4.40:
The responding INTERRUPTER MUST release all of D00-D31 before releasing DTACK* and BERR* to high.
32. OBSERVATION 4.38:
The responding INTERRUPTER is guaranteed that IACK*, LWORD* and A01-A03 have been valid for this minimum time when it detects a falling edge on DSA*. This time is derived from timing parameters 4 and 10.
34. OBSERVATION 4.39:
The INTERRUPTER is guaranteed that DSA* has been low for this minimum time, when it detects a falling edge on IACKIN*.
35. RULE 4.41:
A participating INTERRUPTER MUST drive its IACKOUT* high within this maximum time after the rising edge on AS*.
36. RULE 4.42:
The INTERRUPTER MUST NOT drive the data bus until its IACKIN* line goes low.
37. RULE 4.43:
IF a participating INTERRUPTER drives the data bus,
THEN it MUST release it before driving its IACKOUT* line low.
- 38A. RULE 4.44:
A participating INTERRUPTER MUST NOT drive its IACKOUT* line low, until it detects its IACKIN* line low.
- 38B. RULE 4.45:
The responding INTERRUPTER MUST NOT drive its DTACK* line low, until it detects its IACKIN* line low.
39. OBSERVATION 4.40:
This time guarantees that each INTERRUPTER's IACKIN* line will go high within this time after the rising edge on AS*. This time is derived from timing parameter 35, where the IACK DAISY-CHAIN DRIVER and participating INTERRUPTERS are required to drive their IACKOUT* line high within a maximum time.
40. OBSERVATION 4.41:
All INTERRUPTERS are guaranteed that their IACKIN* line will stay high for this minimum time between consecutive DTB cycles.

Tableau 4-18 (fin)

41. OBSERVATION 4.42:

Ce temps garantit que A01-A03 et LWORD* restent valides pendant cette durée minimale après que le GENERATEUR D'INTERRUPTION qui participe eut commandé sa ligne IACKOUT* au niveau bas et pourvu qu'il le fasse dans la limite de temps d'occupation du bus.

43. OBSERVATION 4.43:

Ce temps garantit que AS* reste bas pendant cette durée minimale après que le GENERATEUR D'INTERRUPTION qui participe eut commandé sa ligne IACKOUT* au niveau bas et pourvu qu'il le fasse dans la limite de temps d'occupation du bus.

Tableau 4-19

EMETTEUR DE CHAINE SERIE IACK
REGLES et OBSERVATIONS sur la chronologie

Note: La numérotation correspond aux paramètres de temps spécifiés dans le tableau 4-16.

OBSERVATION 4.44:

Puisque le fond de panier relie IACK* à IACKIN* de l'emplacement 1, ces deux signaux sont équivalents. Par conséquent, toutes les REGLES et OBSERVATIONS qui s'appliquent à l'un, s'appliquent aussi à l'autre.

5. OBSERVATION 4.45:

L'EMETTEUR DE CHAINE SERIE IACK est assuré que AS* sera au niveau haut entre les cycles du DTB, au minimum pendant ce temps.

19. OBSERVATION 4.46:

L'EMETTEUR DE CHAINE SERIE IACK est assuré que AS* restera au niveau bas au minimum pendant ce temps. Cette valeur de temps se déduit des paramètres de temps 8, 16 et 27 du GENERATEUR D'INTERRUPTION.

32. OBSERVATION 4.47:

L'EMETTEUR DE CHAINE SERIE IACK est assuré que IACK* (et le signal IACKIN* de l'emplacement 1) sont valides au minimum pendant ce temps lorsqu'il détecte un front descendant sur DSA*.

34. REGLE 4.46:

SI la ligne IACKIN* est au niveau bas lorsque l'EMETTEUR DE CHAINE SERIE IACK détecte un front descendant sur DSA*,
ALORS il DOIT commander sa ligne IACKOUT* au niveau bas, mais NE DOIT PAS la commander avant ce délai après le front descendant de DSA*.

OBSERVATION 4.48:

L'EMETTEUR DE CHAINE SERIE IACK ne commande pas sa ligne IACKOUT* au niveau bas chaque fois que la ligne DSA* passe au niveau bas. Cette commande a lieu seulement lorsque la ligne IACK* est aussi au niveau bas, indiquant un cycle de reconnaissance d'interruption en cours.

Table 4-18 (concluded)

41. OBSERVATION 4.42:

This time guarantees that A01-A03 and LWORD* remain valid until this time after the participating INTERRUPTER drives its IACKOUT* low, provided it does so within the bus time-out period.

43. OBSERVATION 4.43:

This time guarantees that AS* remains low for this minimum time after the participating INTERRUPTER drives its IACKOUT* low, provided it does so within the bus time-out period.

Table 4-19

IACK DAISY-CHAIN DRIVER
Timing RULES and OBSERVATIONS

Note: The numbers correspond to the timing parameters specified in Table 4-16.

OBSERVATION 4.44:

Since the backplane connects IACK* to the slot 1 IACKIN*, these two signals are equivalent. And therefore, all RULES and OBSERVATIONS that apply to one, are applicable to the other as well.

5. OBSERVATION 4.45:

The IACK DAISY-CHAIN DRIVER is guaranteed this minimum high time on AS* between DTB cycles.

19. OBSERVATION 4.46:

The IACK DAISY-CHAIN DRIVER is guaranteed that the AS* will remain low for this minimum time. This time is derived from timing parameters 8, 16 and 27 of the INTERRUPTER.

32. OBSERVATION 4.47:

The IACK DAISY-CHAIN DRIVER is guaranteed that IACK* (and the slot 1 IACKIN*) has been valid for this minimum time when it detects a falling edge on DSA*.

34. RULE 4.46:

IF the IACKIN* line is low when the IACK DAISY-CHAIN DRIVER detects a falling edge on DSA*,
THEN it MUST drive its IACKOUT* line low, but it MUST NOT do so until this time after the falling edge on DSA*.

OBSERVATION 4.48:

The IACK DAISY-CHAIN DRIVER does not drive its IACKOUT* line low every time DSA* goes low. It only does so when the IACK* line is low as well, indicating that an interrupt acknowledge cycle is in progress.

Tableau 4-19 (fin)

35. REGLE 4.47:

Si l'EMETTEUR DE CHAINE SERIE IACK commande sa ligne IACKOUT* au niveau bas,
ALORS il DOIT commander sa ligne IACKOUT* au niveau haut dans ce délai après le front montant de AS*.

40. REGLE 4.48:

L'EMETTEUR DE CHAINE SERIE IACK NE DOIT PAS commander IACKOUT* au niveau bas avant d'avoir été au niveau haut au minimum pendant ce temps.

42. OBSERVATION 4.49:

SI l'EMETTEUR DE CHAINE SERIE IACK commande sa ligne IACKOUT* au niveau bas dans la limite de temps d'occupation du bus,
ALORS ce temps garantit que IACK* (et IACKIN* de l'emplacement 1) restent valides durant ce temps.

Withdrawing
IEC NORM.COM: Click to view the full PDF of IEC 60821-1987

Table 4-19 (concluded)

35. RULE 4.47:
IF the IACK DAISY-CHAIN DRIVER drives its IACKOUT* line low,
THEN it MUST drive its IACKOUT* high within this time after the
rising edge of AS*.
40. RULE 4.48:
The IACK DAISY-CHAIN DRIVER MUST NOT drive IACKOUT* low until
it has been high for this minimum time.
42. OBSERVATION 4.49:
IF the IACK DAISY-CHAIN DRIVER drives its IACKOUT* line low
within the bus time-out period.
THEN this time guarantees that IACK* (and the slot 1 IACKIN*)
remain valid for this time.

Withdawn
IEC NORM.COM: Click to view the full PDF of IEC 60821:1987

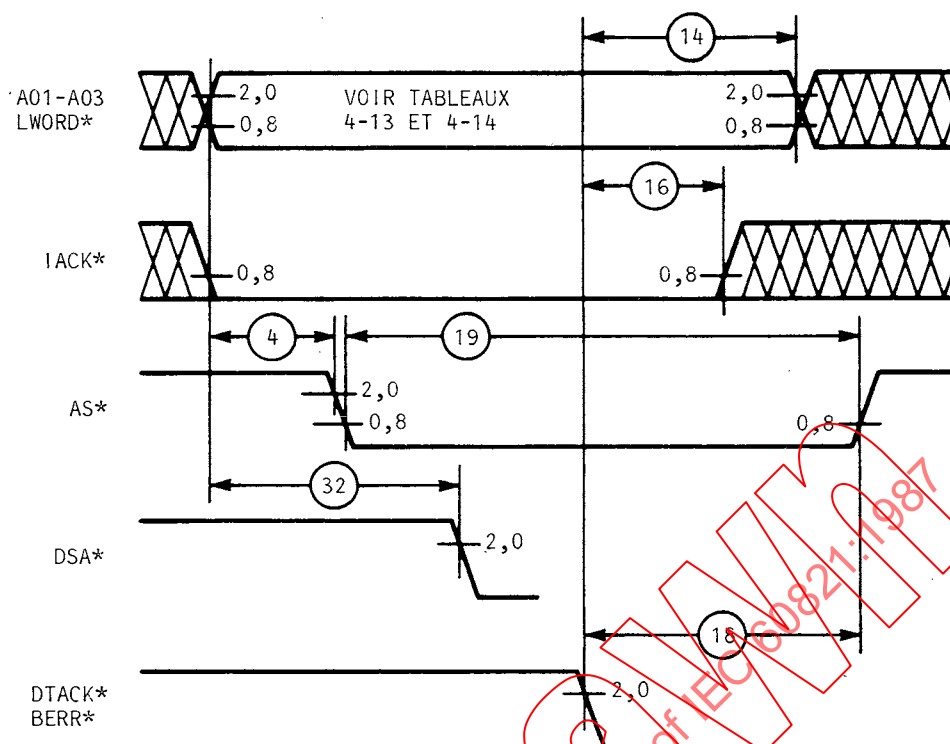


FIG. 4-16. - CONTROLEUR et GENERATEUR D'INTERRUPTION - Chronologie de la sélection du GENERATEUR D'INTERRUPTION CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE, DOUBLE ou QUADRUPLE.

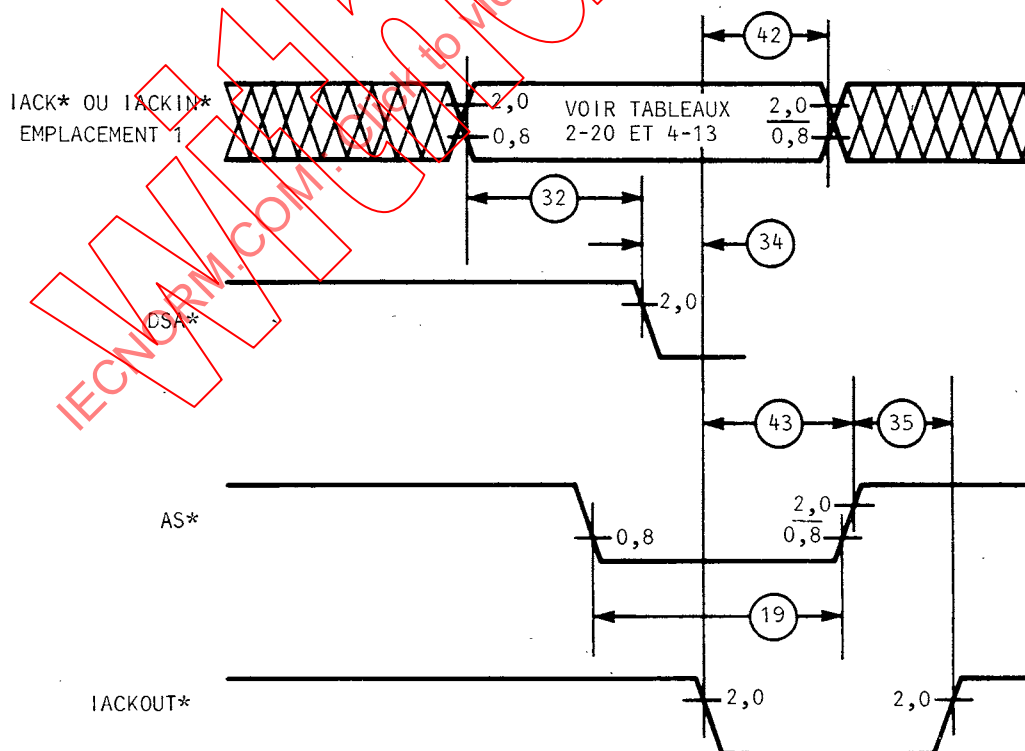


FIG. 4-17. - EMETTEUR DE CHAINE SERIE IACK - Chronologie de la sélection du GENERATEUR D'INTERRUPTION CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE, DOUBLE ou QUADRUPLE.

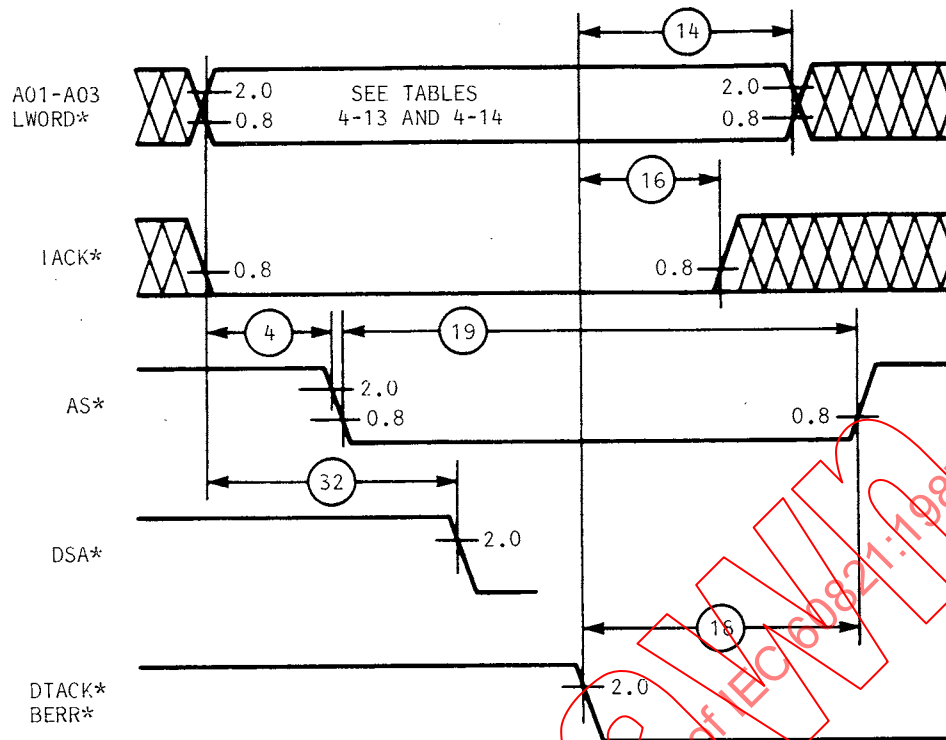


FIG. 4-16. - INTERRUPT HANDLER and INTERRUPTER - INTERRUPTER selection timing
SINGLE, DOUBLE and QUAD BYTE INTERRUPT
ACKNOWLEDGE CYCLE.

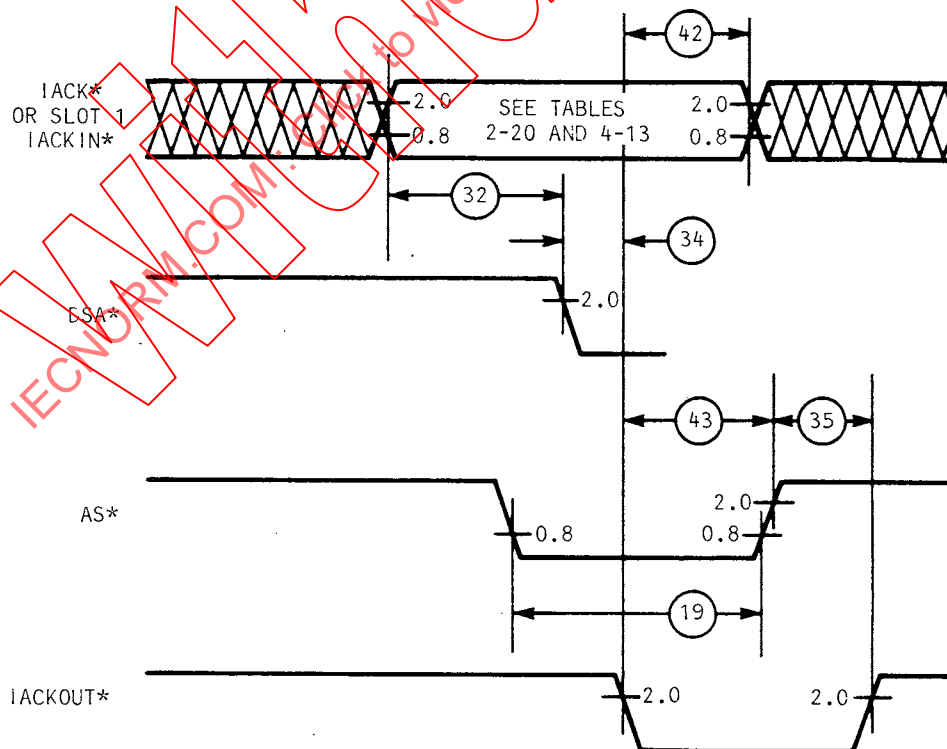


FIG. 4-17. - IACK DAISY-CHAIN DRIVER - INTERRUPTER
selection timing
SINGLE, DOUBLE and QUAD BYTE
INTERRUPT ACKNOWLEDGE CYCLE.

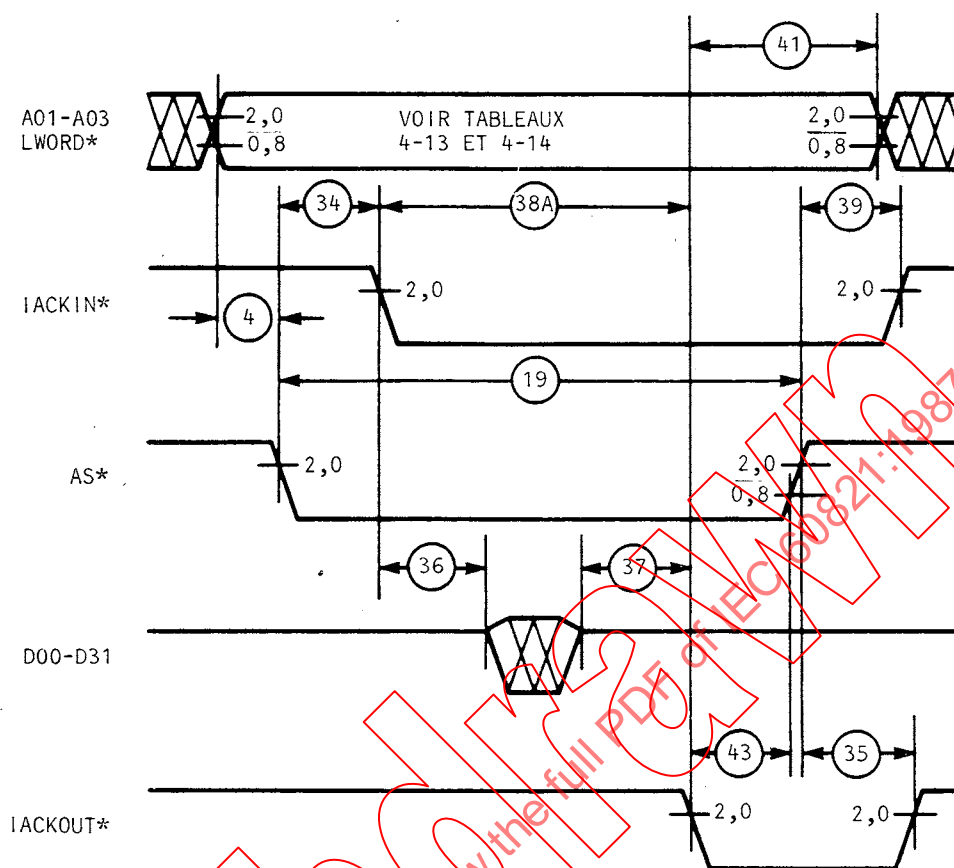


FIG. 4-18. - GENERATEUR D'INTERRUPTION qui participe -
 Chronologie de la sélection du GENERATEUR
 D'INTERRUPTION
 CYCLE DE RECONNAISSANCE D'INTERRUPTION
 à OCTET UNIQUE, DOUBLE ou QUADRUPLE.

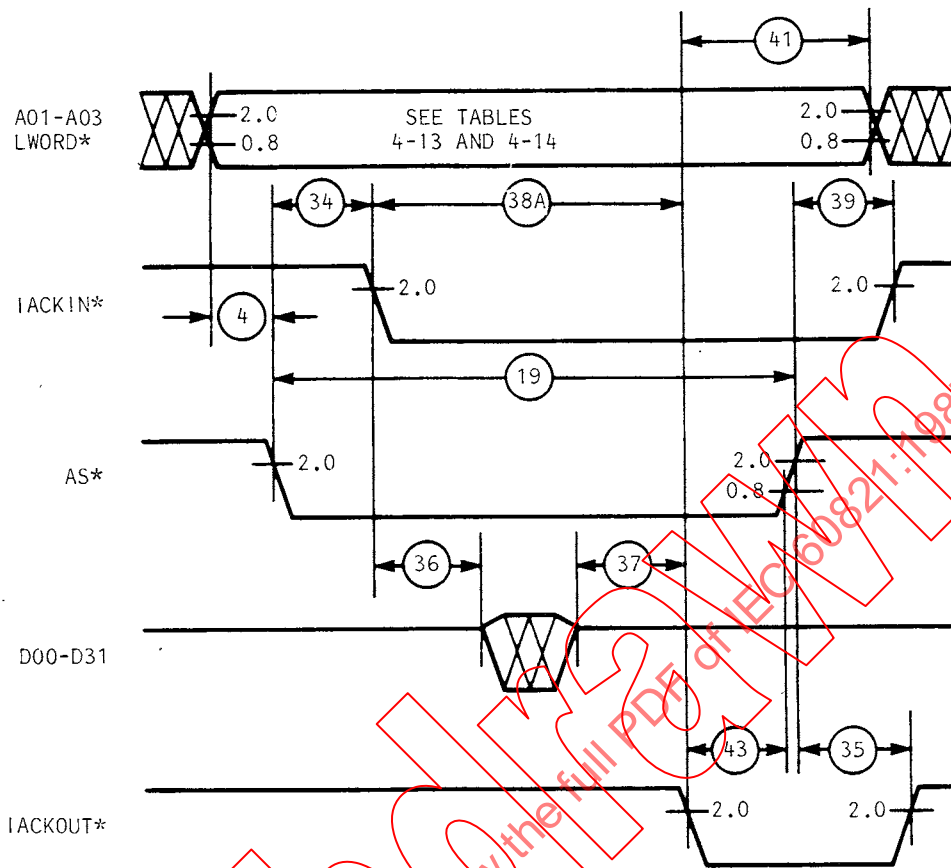


FIG. 4-18. - Participating INTERRUPTER - INTERRUPTER selection timing
SINGLE, DOUBLE and QUAD BYTE
INTERRUPT ACKNOWLEDGE CYCLE.

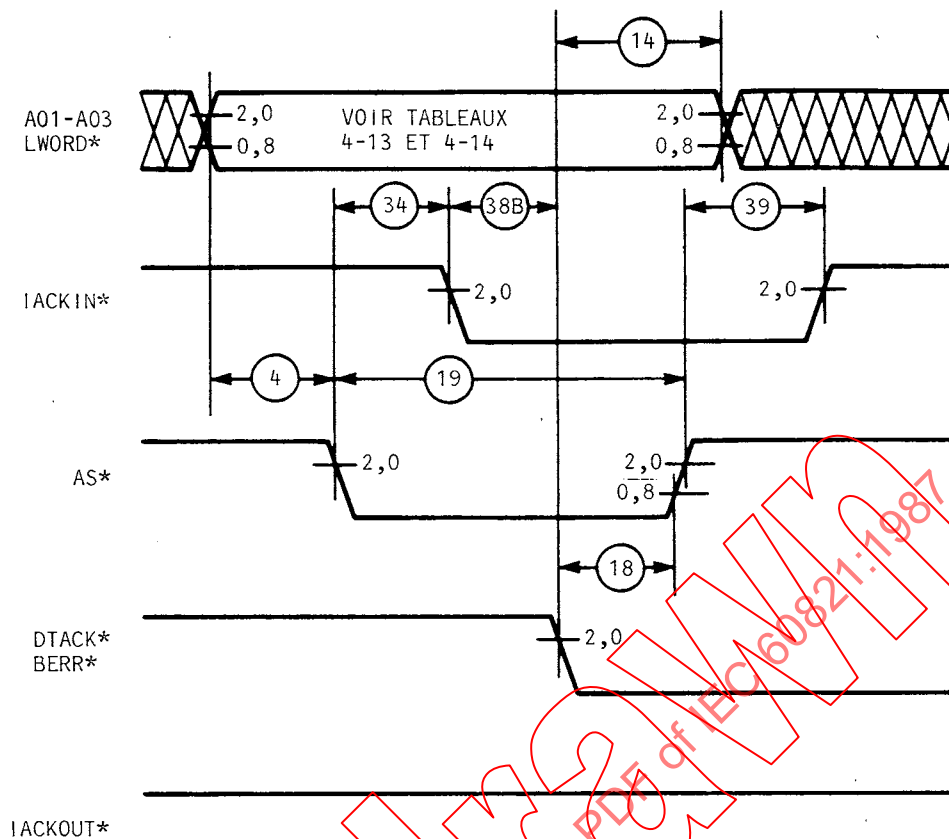


FIG. 4-19. - GENERATEUR D'INTERRUPTION qui répond - Chronologie de la sélection du GENERATEUR D'INTERRUPTION CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE, DOUBLE ou QUADRUPLE.

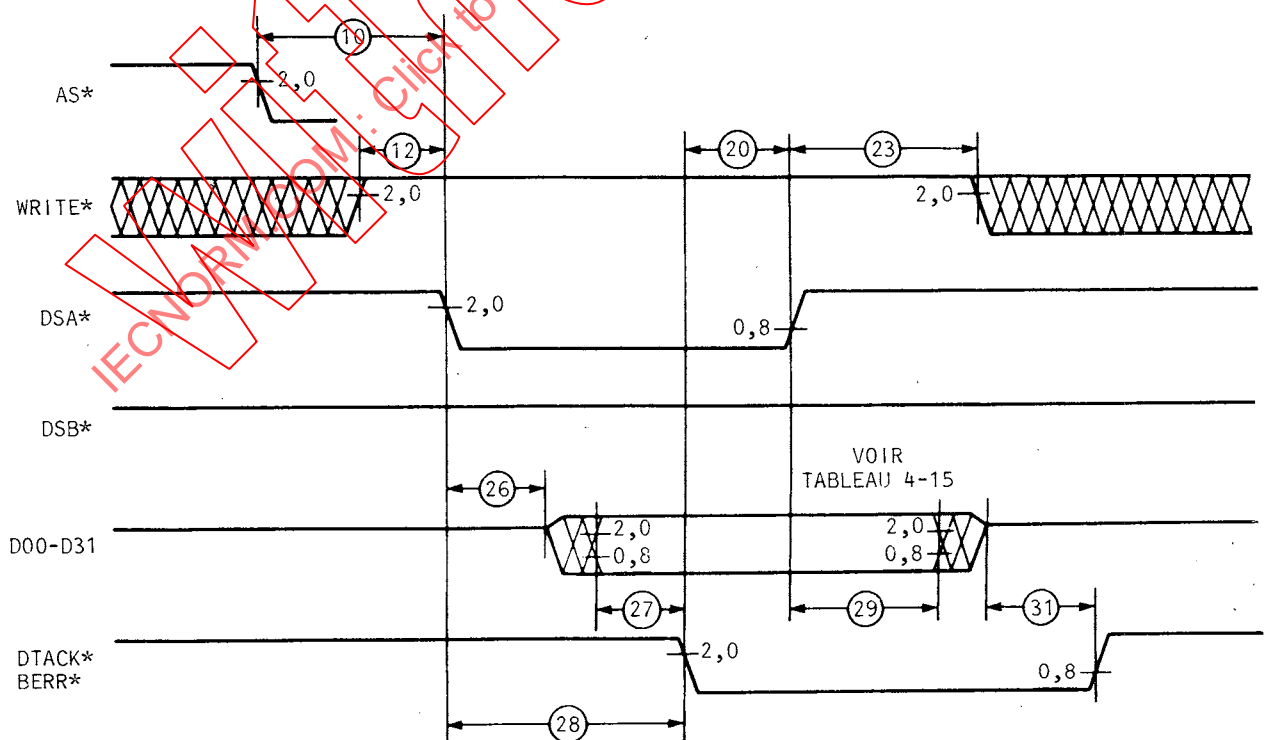
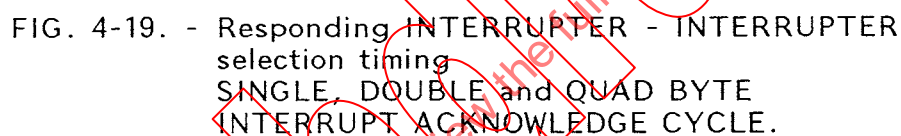


FIG. 4-20. - CONTROLEUR D'INTERRUPTION - Chronologie du transfert du MOT D'ETAT/IDentificateur CYCLE de RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE.



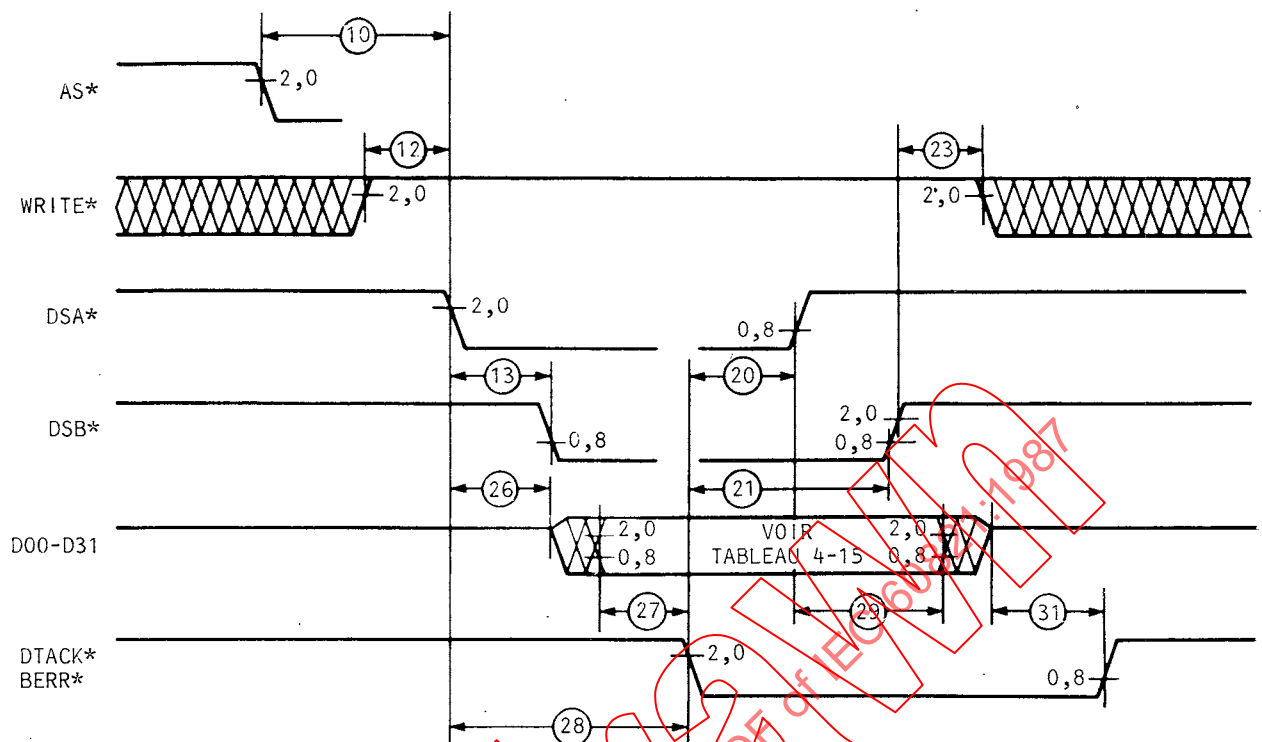


FIG. 4-21. - CONTROLEUR D'INTERRUPTION - Chronologie du transfert du MOT D'ETAT/Identificateur
CYCLE DE RECONNAISSANCE D'INTERRUPTION à DOUBLE OCTET.
CYCLE DE RECONNAISSANCE D'INTERRUPTION à QUADRUPLE OCTET.

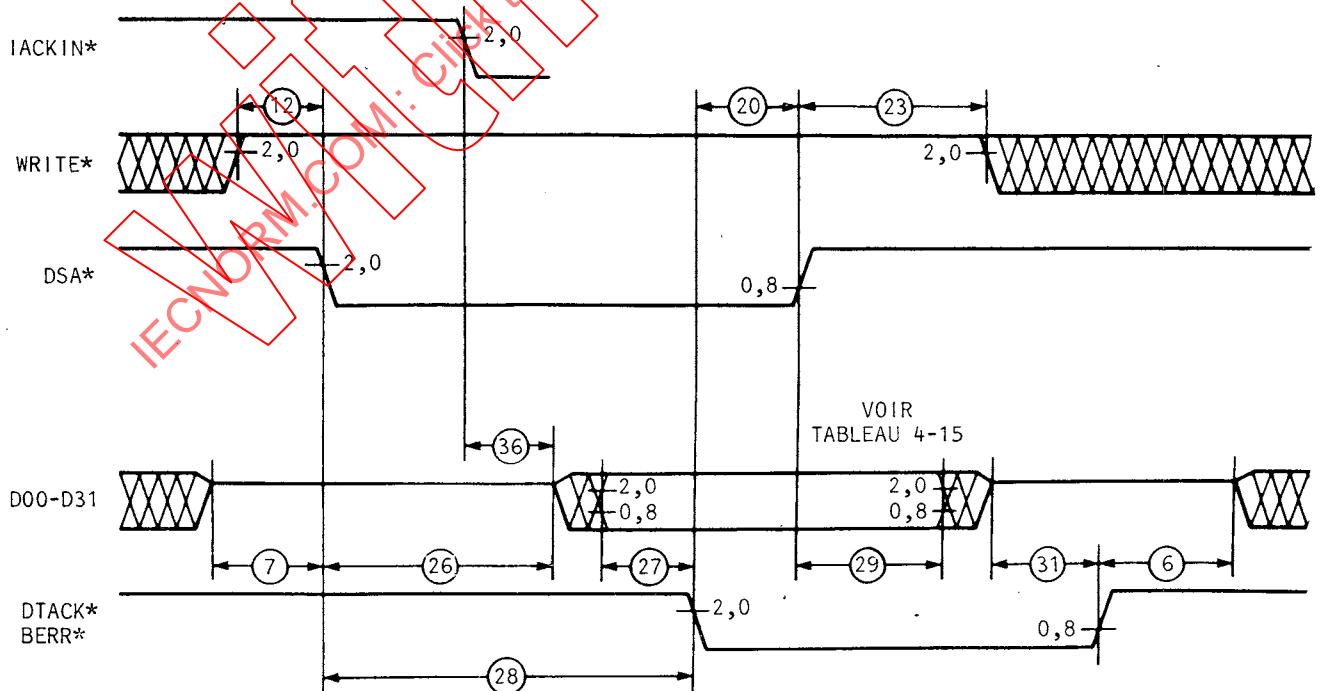


FIG. 4-22. - GENERATEUR D'INTERRUPTION qui répond - Chronologie du transfert du MOT D'ETAT/Identificateur
CYCLE DE RECONNAISSANCE D'INTERRUPTION à OCTET UNIQUE.

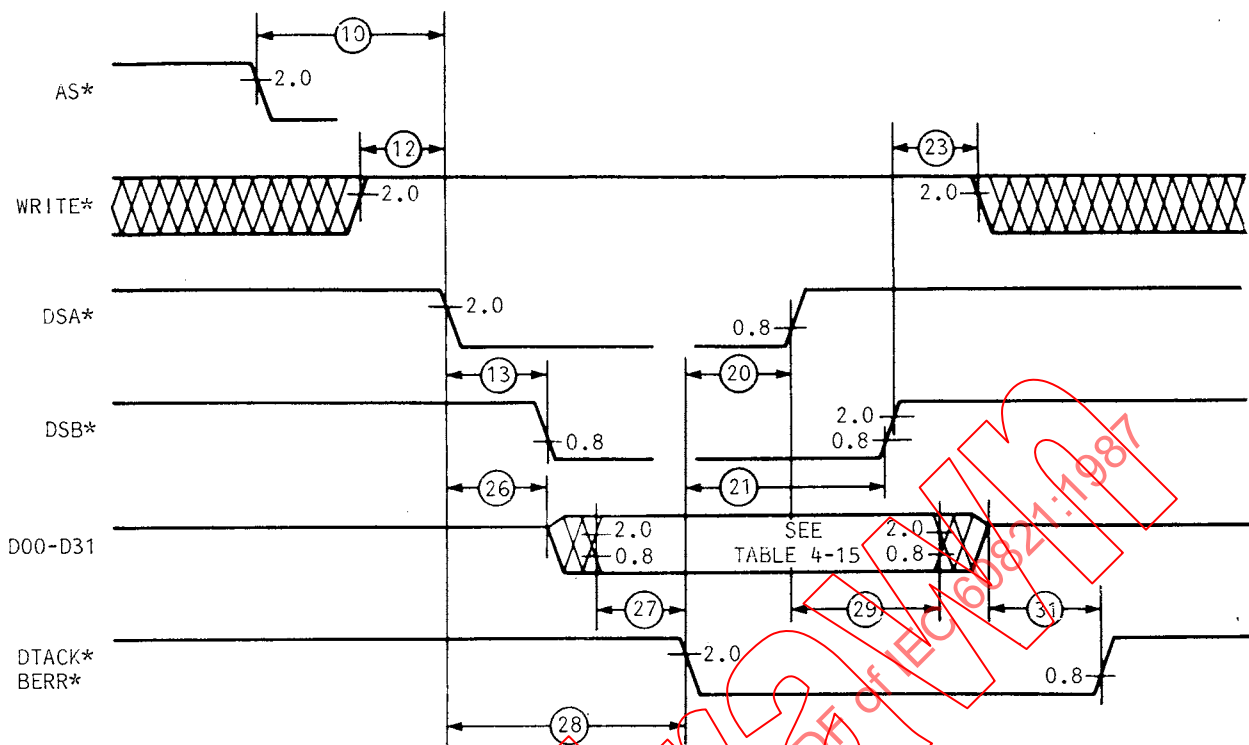


FIG. 4-21. - INTERRUPT HANDLER - STATUS/ID transfer timing
DOUBLE BYTE INTERRUPT ACKNOWLEDGE CYCLE;
QUAD BYTE INTERRUPT ACKNOWLEDGE CYCLE.

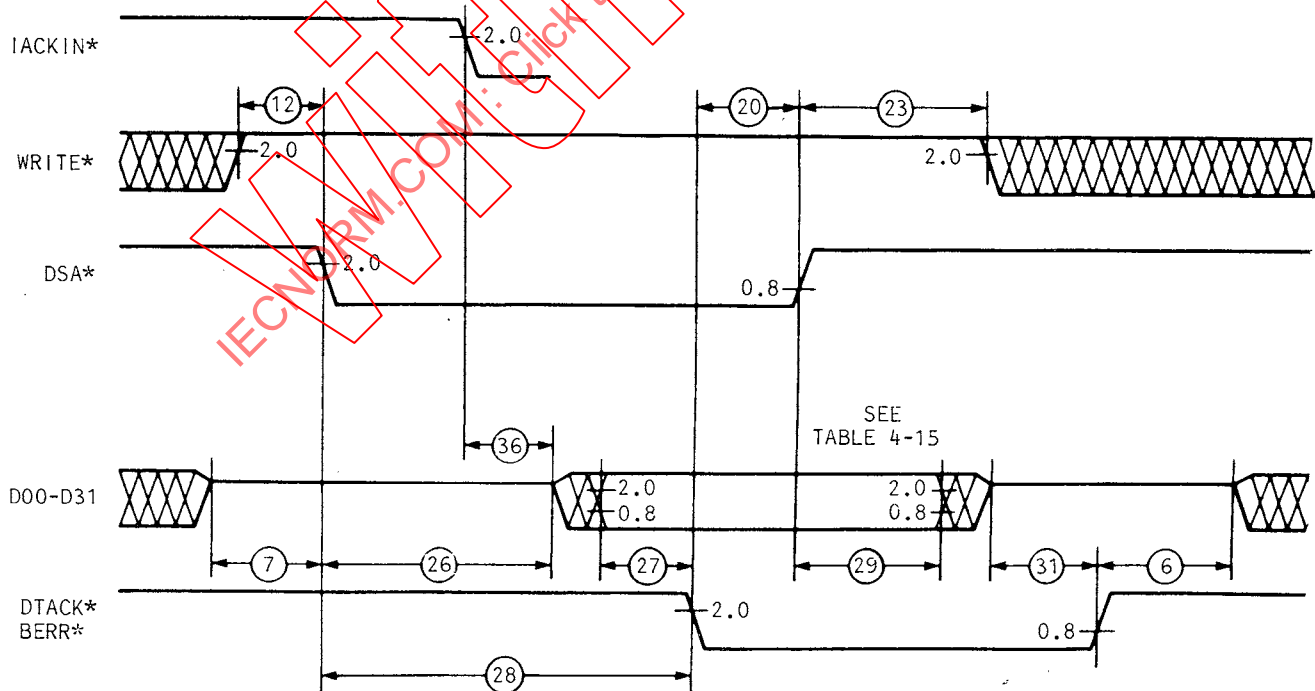


FIG. 4-22. - Responding INTERRUPTER - STATUS/ID transfer timing
SINGLE BYTE INTERRUPT ACKNOWLEDGE CYCLE.

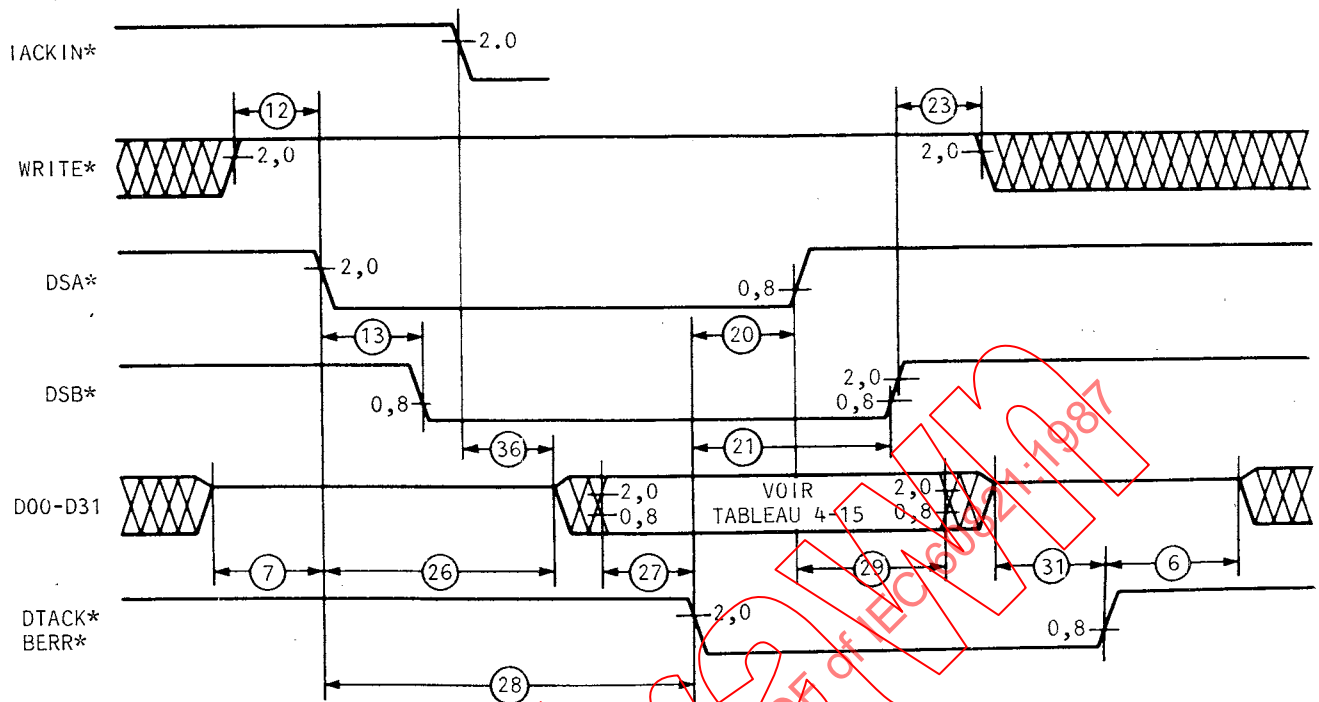


FIG. 4-23. - GENERATEUR D'INTERRUPTION qui répond - Chronologie
du transfert du MOT D'ETAT/IDentificateur
CYCLE DE RECONNAISSANCE D'INTERRUPTION
à DOUBLE OCTET;
CYCLE DE RECONNAISSANCE D'INTERRUPTION
à QUADRUPLE OCTET.

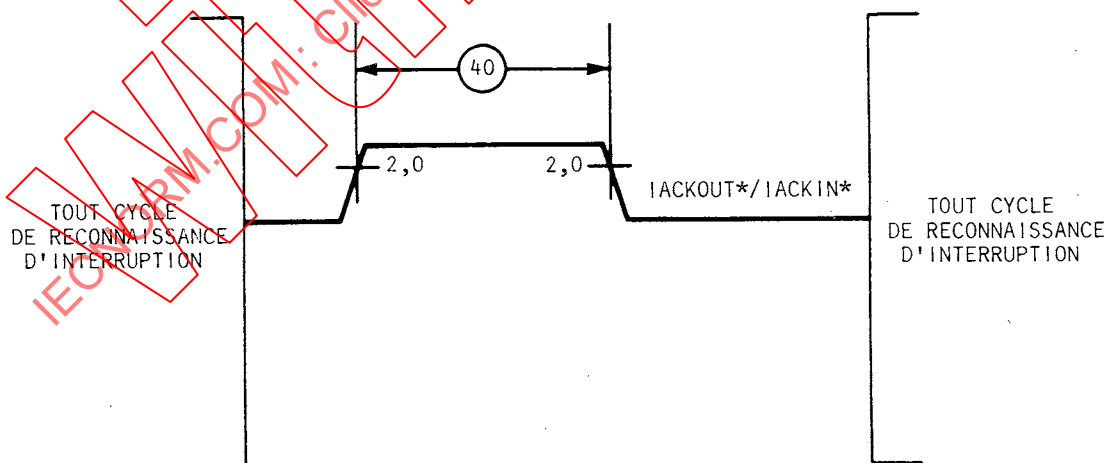


FIG. 4-24. - EMETTEUR DE CHAÎNE SÉRIE IACK, GENERATEUR
D'INTERRUPTION qui répond et
GENERATEUR D'INTERRUPTION qui participe
Chronologie de la chaîne série IACK entre les cycles.

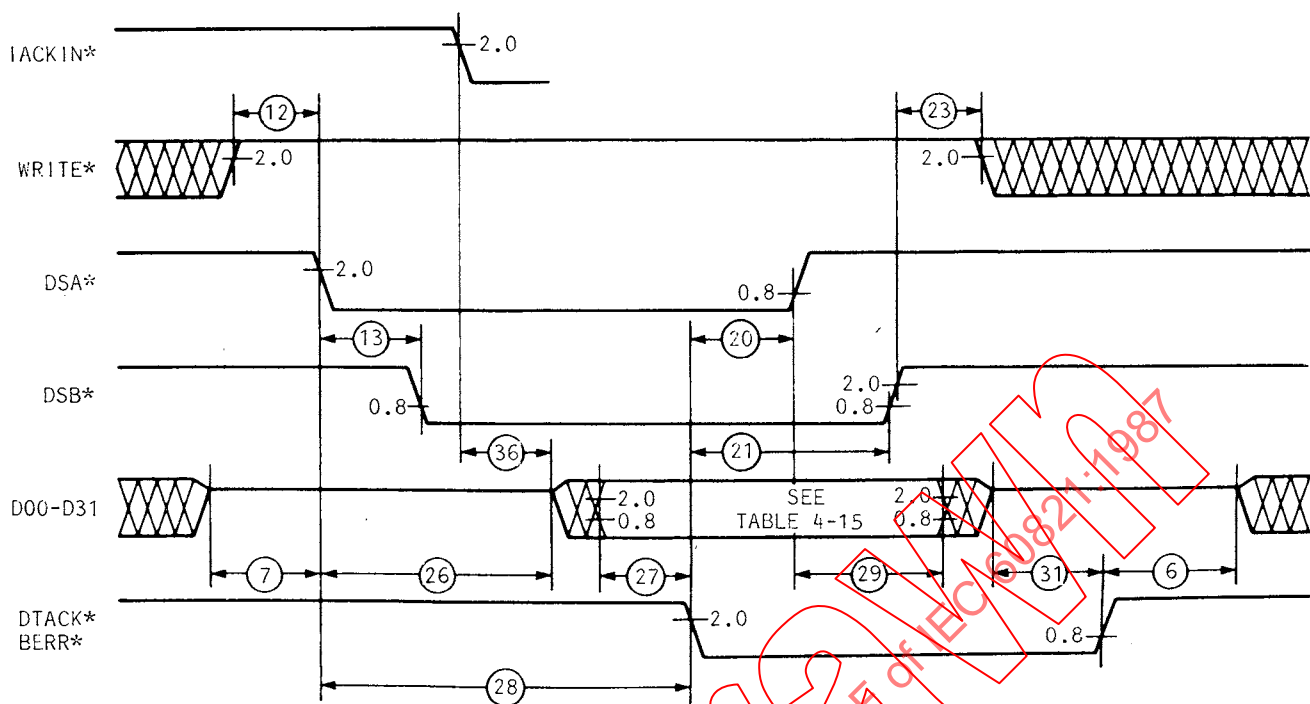


FIG. 4-23. - Responding INTERRUPTER - STATUS/ID transfer timing
DOUBLE BYTE INTERRUPT ACKNOWLEDGE CYCLE;
QUAD BYTE INTERRUPT ACKNOWLEDGE CYCLE.

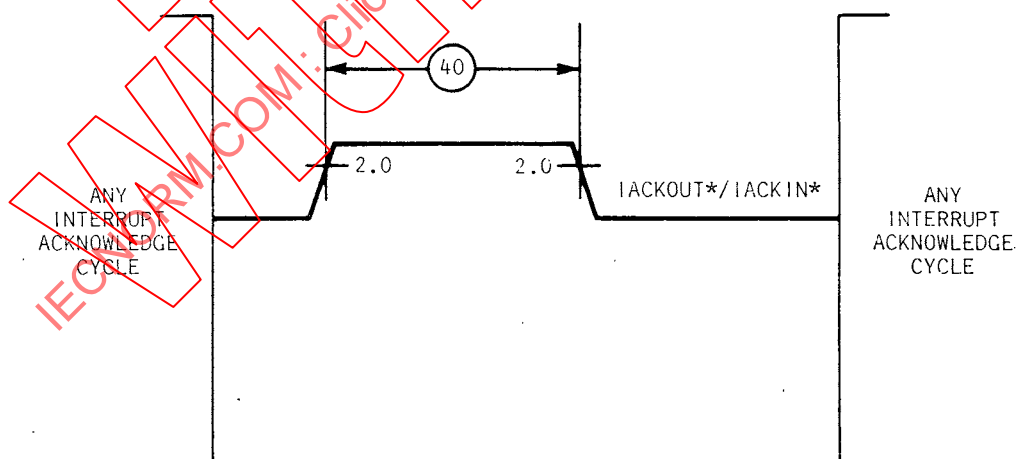


FIG. 4-24. - IACK DAISY-CHAIN DRIVER, responding INTERRUPTER,
and participating INTERRUPTER
IACK daisy-chain inter-cycle timing.

CHAPITRE 5: BUS UTILITAIRE DU BUS CEI 821

5.1 Introduction

Ce chapitre présente et définit les lignes de signal et les modules qui remplissent les fonctions utilitaires du BUS CEI 821. Le bus utilitaire fournit le séquençement périodique, l'initialisation et les moyens de diagnostic du BUS CEI 821 (voir figure 5-1, page 340).

5.2 Signaux du bus utilitaire

Les signaux du bus utilitaire sont listés ci-dessous:

Horloge du système (SYSCLK)
Horloge du bus série (SERCLK)
Données du bus série (SERDAT*)
Défaillance de l'alimentation secteur (ACFAIL*)
Initialisation du système (SYSRESET*)
Défaillance du système (SYSFAIL*)

5.3 Modules du bus utilitaire

5.3.1 L'EMETTEUR DE L'HORLOGE DU SYSTEME

L'horloge du système est un signal indépendant, permanent, à fréquence fixe de 16 MHz, ayant un rapport cyclique (nominal) de 50%. L'émetteur SYSCLK est implanté dans le contrôleur du système situé à l'emplacement 1 (voir chapitre 1). Il fournit une base de temps connue, utilisable pour les temporisations. La figure 5-2, page 342, montre le chronogramme de temps de L'EMETTEUR DE L'HORLOGE DU SYSTEME.

OBSERVATION 5.1:

SYSCLK n'a pas de relation de phase fixe avec d'autres signaux du BUS CEI 821.

5.3.2 L'EMETTEUR DE L'HORLOGE DU BUS SERIE

L'EMETTEUR DE L'HORLOGE DU BUS SERIE est un signal à fréquence fixe de forme d'onde spécifiée utilisé par les modules du BUS CEI 823 situés sur les cartes de BUS CEI 821. La spécification de sa forme d'onde sera donnée dans la norme du BUS CEI 823 (à l'étude). Pour aider les concepteurs, les paramètres de temps valables lors de la publication de cette norme sont donnés dans l'annexe C.

5.3.3 Le CONTROLEUR D'ALIMENTATION

La figure 5-3, page 342, représente le schéma-bloc du module CONTROLEUR D'ALIMENTATION. Ce module détecte les défaillances d'alimentation et les signale en temps voulu au système de BUS CEI 821 pour qu'il puisse effectuer un arrêt correct. Lorsque le système est remis sous tension, le CONTROLEUR D'ALIMENTATION assure que tous les modules sont initialisés.

Le CONTROLEUR D'ALIMENTATION surveille aussi la commande manuelle par bouton-poussoir et initialise le système de BUS CEI 821 lorsque ce bouton est actionné par l'opérateur.

CHAPTER 5: IEC 821 BUS UTILITY BUS

5.1 *Introduction*

This chapter identifies and defines the signal lines and modules which provide utility functions for the IEC 821 BUS. The Utility Bus supplies periodic timing, initialization and diagnostic capability for the IEC 821 BUS (see Figure 5-1, page 341).

5.2 *Utility Bus signal lines*

The Utility Bus signal lines are listed below:

- System Clock (SYSCLK)
- Serial Clock (SERCLK)
- Serial Data (SERDAT*)
- AC Fail (ACFAIL*)
- System Reset (SYSRESET*)
- System Failure (SYSFAIL*)

5.3 *Utility Bus modules*

5.3.1 The SYSTEM CLOCK DRIVER

The system clock is an independent, non-gated, fixed frequency, 16 MHz, 50% (nominal) duty cycle signal. The SYSCLK driver is located on the system controller located in board slot one (see Chapter 1). It provides a known time base that is useful for counting off time delays. Figure 5-2, page 343, shows the SYSTEM CLOCK DRIVER timing diagram.

OBSERVATION 5.1:

SYSCLK has no fixed phase relationships with other IEC 821 BUS timing.

5.3.2 The SERIAL CLOCK DRIVER

The SERIAL CLOCK DRIVER provides a fixed frequency, special waveform signal used by proposed IEC 823 BUS modules that reside on IEC 821 BUS boards. Its waveform will be specified in IEC 823 BUS (under consideration). For the convenience of designers, the timing parameters in effect at the time that this standard has been published are provided in Appendix C.

5.3.3 The POWER MONITOR

Figure 5-3, page 343, is the block diagram for the POWER MONITOR module. This module detects power failures and signals the IEC 821 BUS system in time to effect orderly shut-down. When power is then reapplied to the system the POWER MONITOR ensures that all other IEC 821 BUS modules are initialized.

The POWER MONITOR might also monitor a manually operated push-button and initialize the IEC 821 BUS system whenever that button is depressed by the operator.

Les transitions ACFAIL* et SYSRESET* et le moment où les tensions continues du système sont hors spécification sont liés par des contraintes temporelles données dans les figures 5-4 et 5-5, page 344.

AUTORISATION 5.1:

Les systèmes BUS CEI 821 PEUVENT être construits avec ou sans module CONTROLEUR D'ALIMENTATION.

REGLE 5.1:

Les CONTROLEURS D'ALIMENTATION DOIVENT être conformes aux chronogrammes donnés dans les figures 5-4 et 5-5.

AUTORISATION 5.2:

La ligne SYSRESET* PEUT être commandée au niveau bas par un bouton-poussoir sur n'importe quelle carte de BUS CEI 821 pour initialiser le système. Lorsqu'une carte commande SYSRESET* mais ne commande pas ACFAIL*, les chronogrammes donnés dans les figures 5-4 et 5-5 ne s'appliquent pas.

REGLE 5.2:

Lorsqu'une carte commande SYSRESET* au niveau bas, elle DOIT maintenir SYSRESET* au niveau bas pendant une durée minimale de 200 ms.

The ACFAIL* and SYSRESET* transitions, and the point at which the system d.c. voltages violate specification, have certain timing relationships. These relationships are shown in Figures 5-4 and 5-5, page 345.

PERMISSION 5.1:

IEC 821 BUS systems MAY be built with or without a POWER MONITOR module.

RULE 5.1:

POWER MONITORS MUST comply with the timing given in Figures 5-4 and 5-5.

PERMISSION 5.2:

The SYSRESET* line MAY be driven low by any IEC 821 BUS board to initialize the system from a manual push-button. Where a board drives SYSRESET*, but does not drive ACFAIL*, the timing in Figure 5-4 and 5-5 does not apply.

RULE 5.2:

Whenever any board drives SYSRESET* low, it MUST hold SYSRESET* low for a minimum period of 200 ms.

Withd 2012
IECNORM.COM: Click to view the full PDF of IEC 821:1987

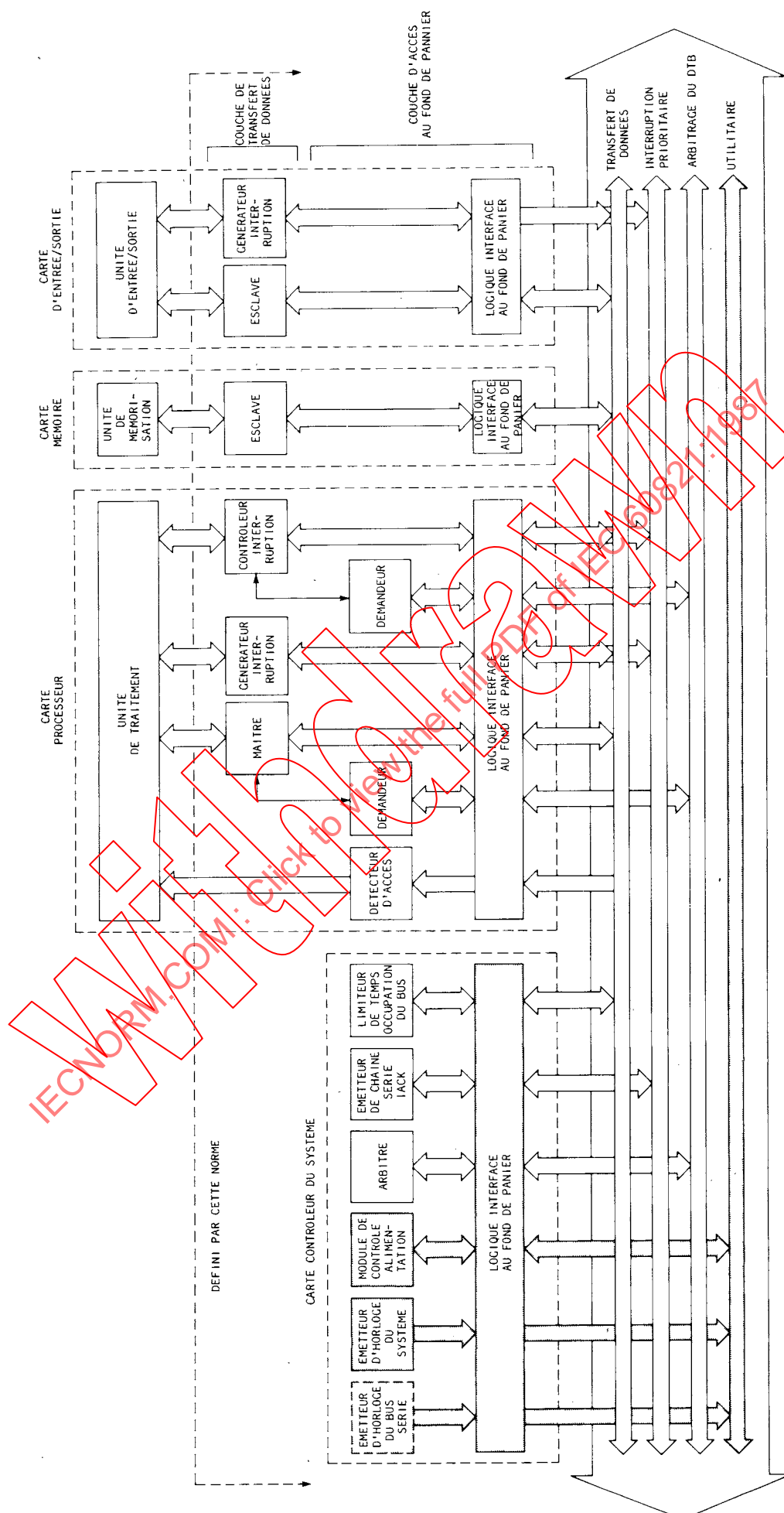


FIG. 5-1. - Schéma-bloc du bus utilitaire.

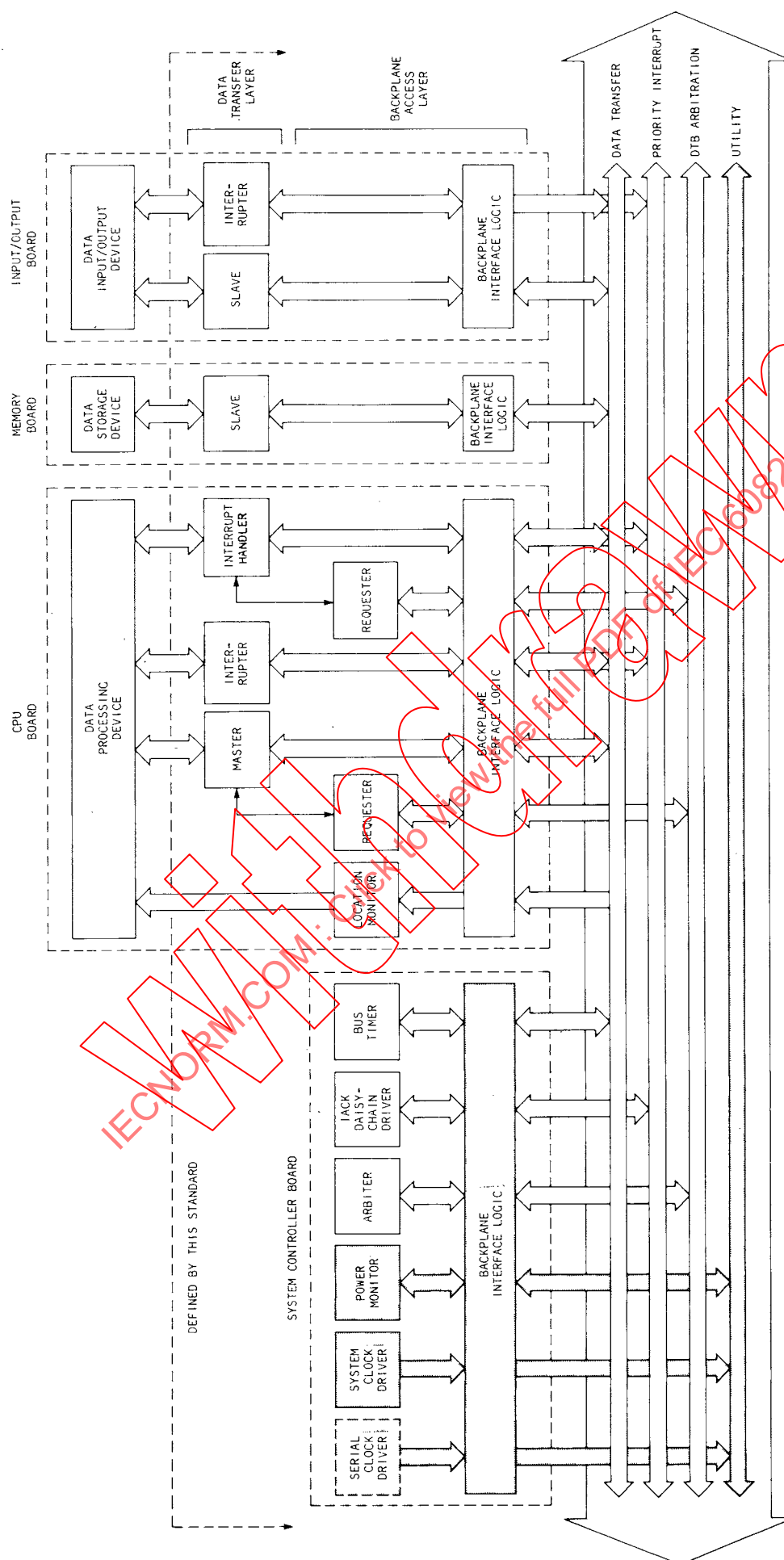


FIG. 5-1. - Utility Bus block diagram.

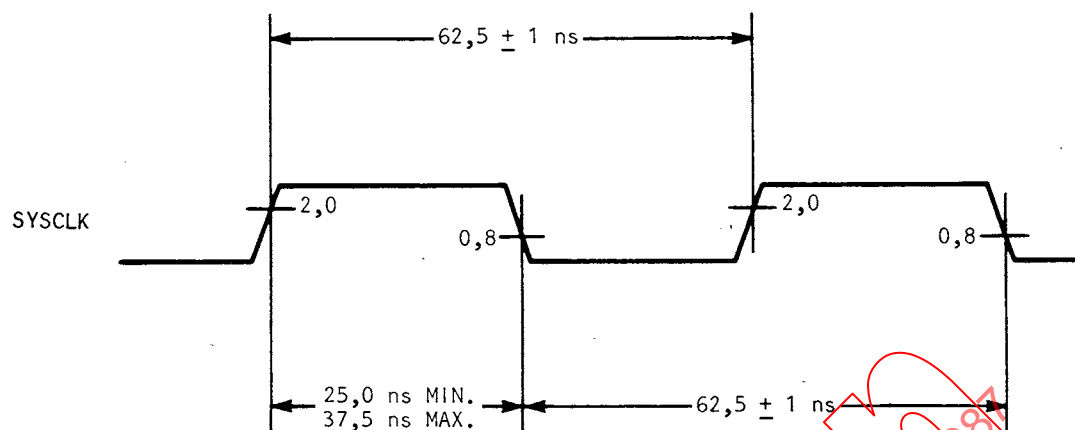


FIG. 5-2. - Chronogramme de l'EMETTEUR DE L'HORLOGE SYSTEME.

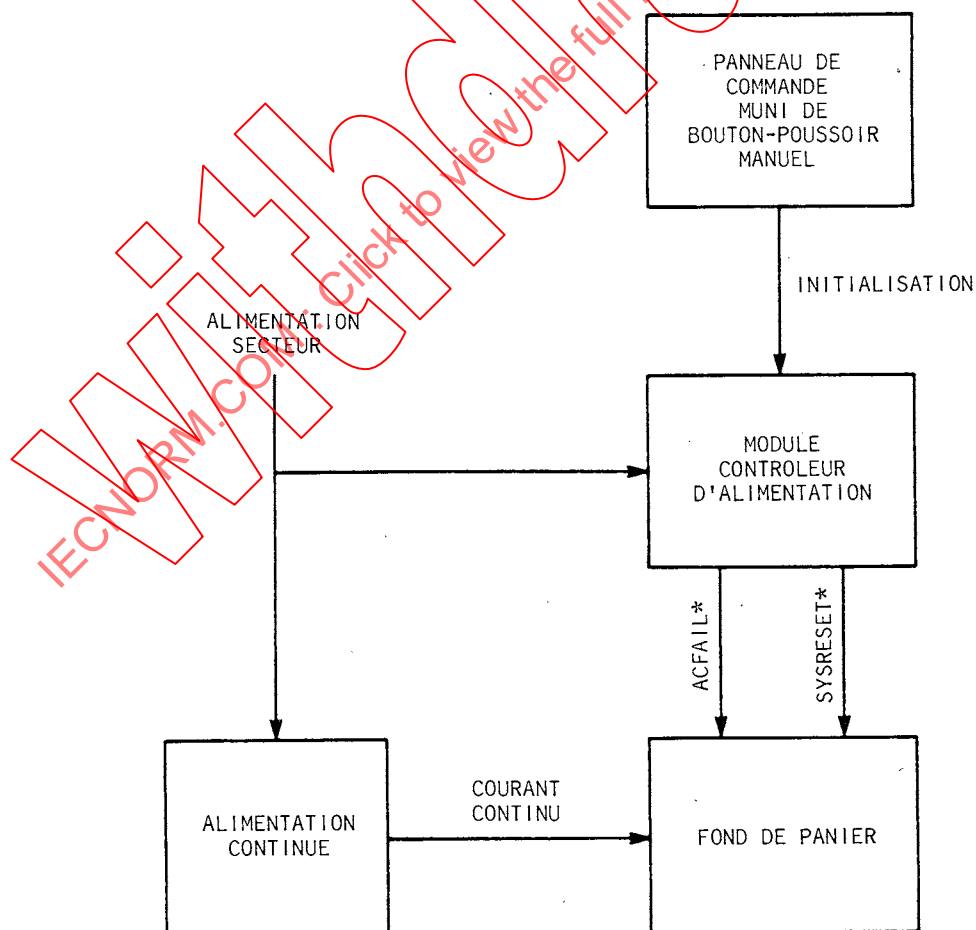


FIG. 5-3. - Schéma-bloc du module CONTROLEUR D'ALIMENTATION.

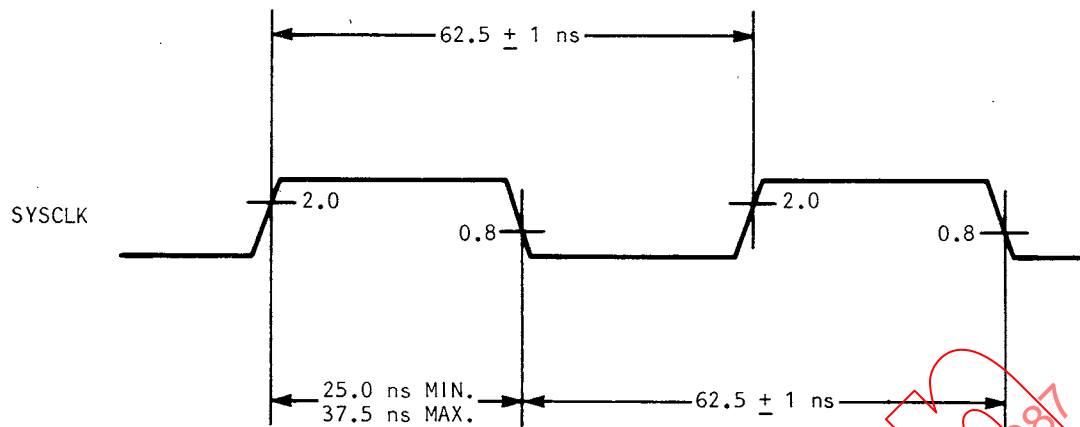


FIG. 5-2. - SYSTEM CLOCK DRIVER timing diagram.

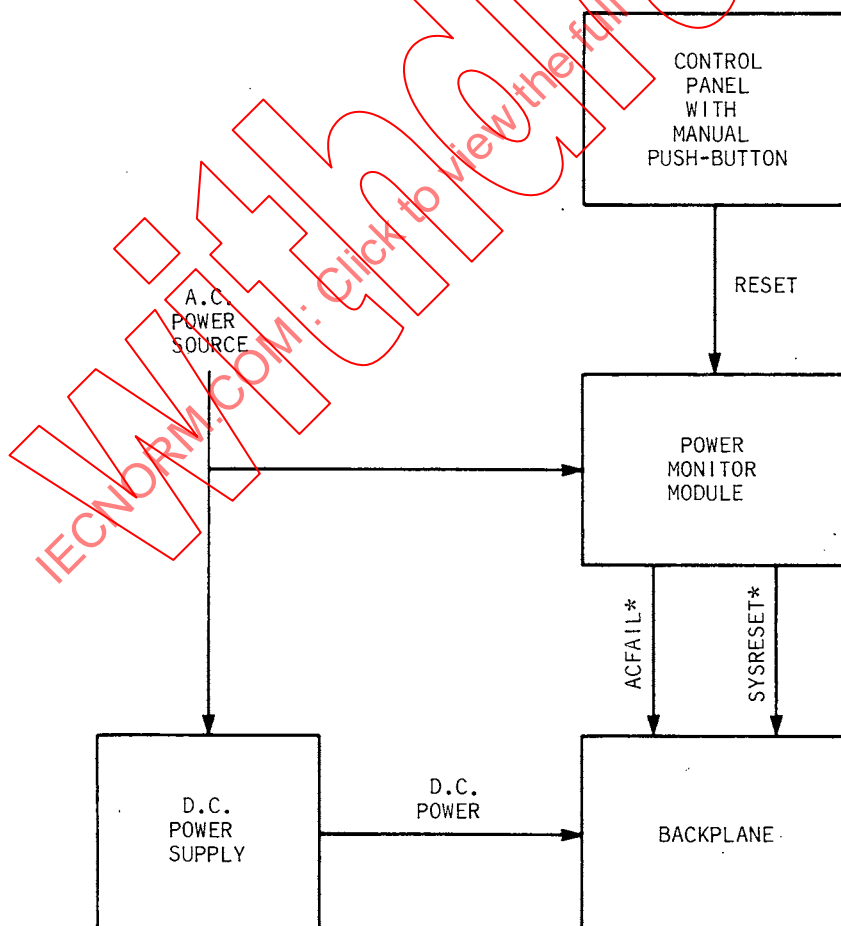


FIG. 5-3. - Block diagram of POWER MONITOR module.

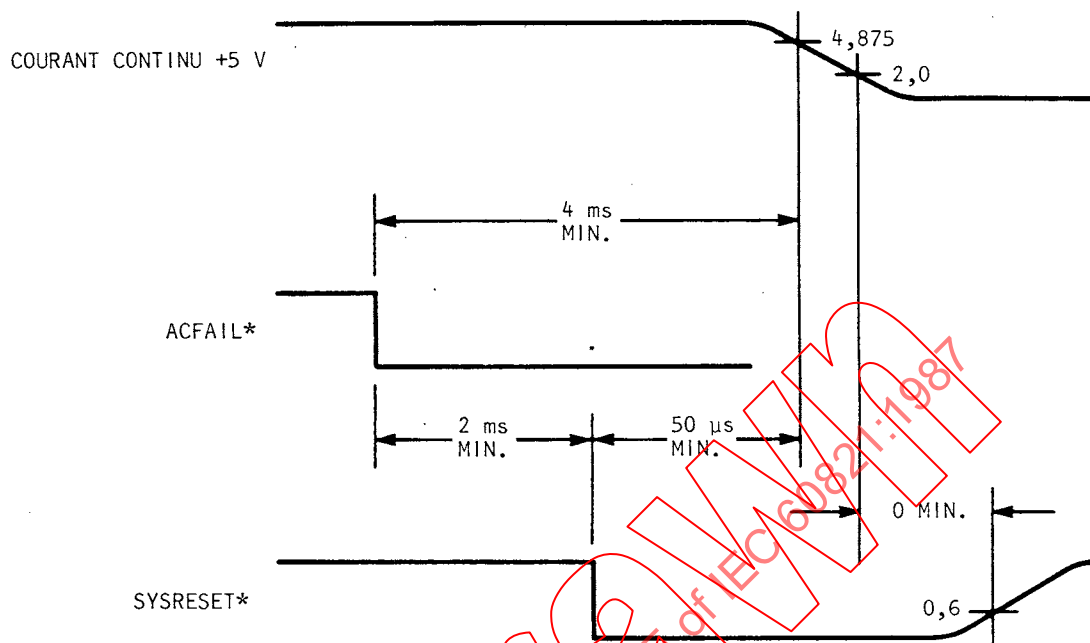


FIG. 5-4. - Chronogramme du CONTROLEUR D'ALIMENTATION lors d'une défaillance d'alimentation.

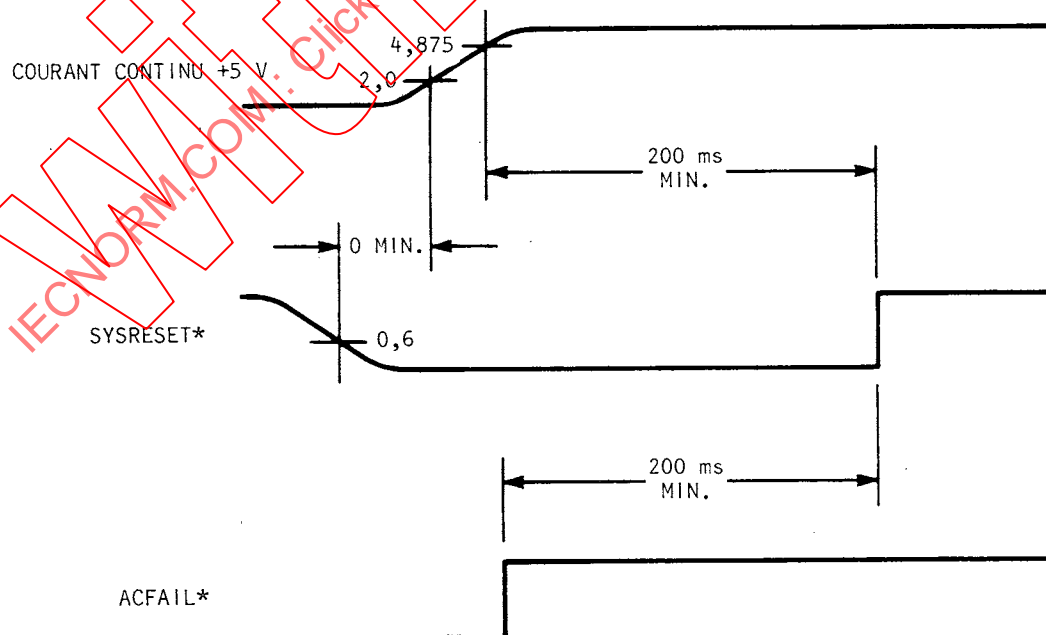


FIG. 5-5. - Chronogramme du CONTROLEUR D'ALIMENTATION lors de la mise sous tension.

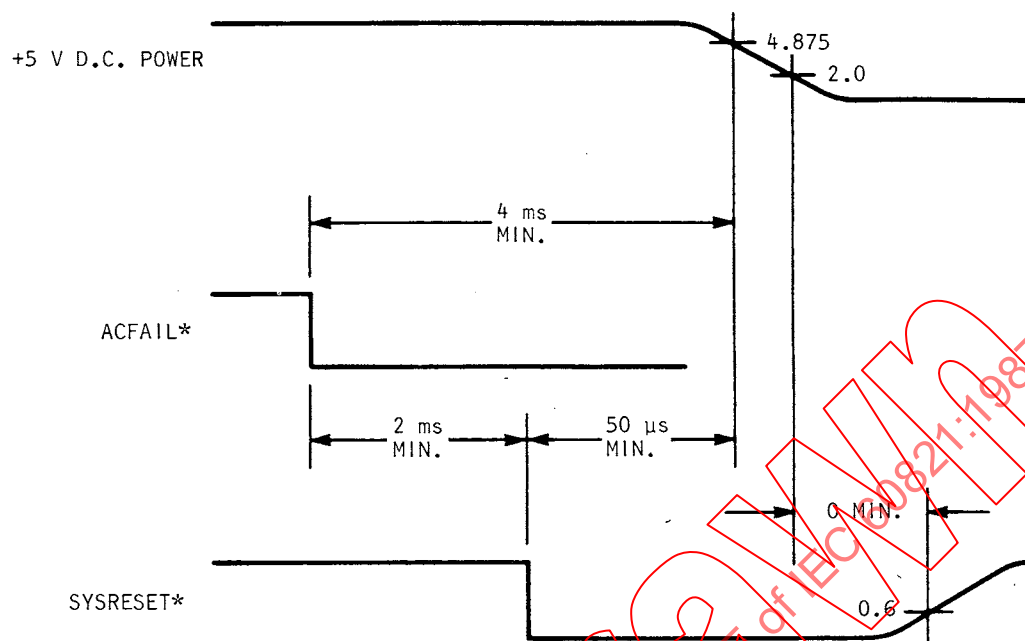


FIG. 5-4. - POWER MONITOR power failure timing.

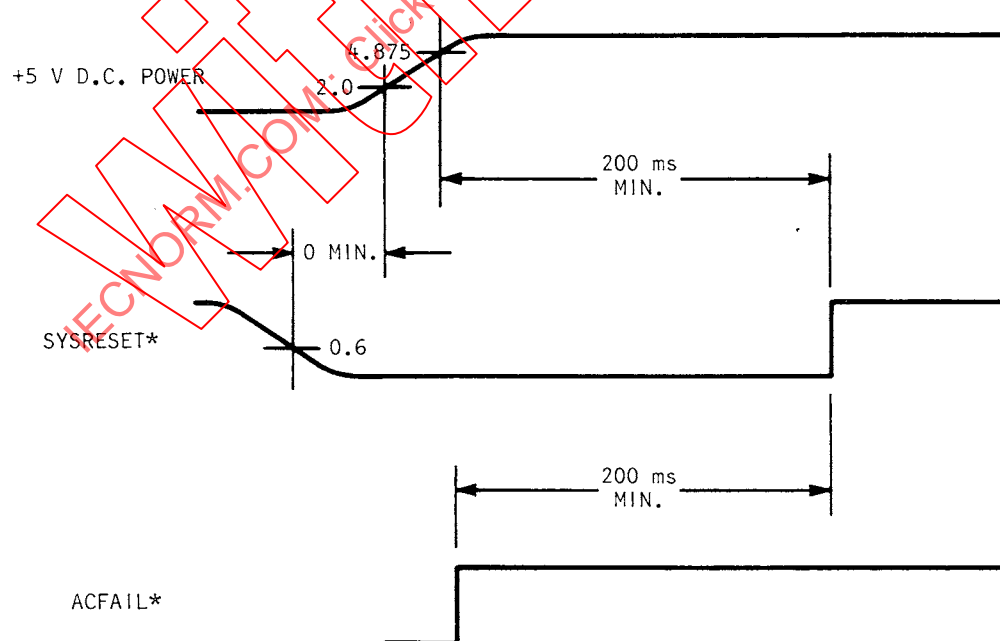


FIG. 5-5. - POWER MONITOR system restart timing.

5.4 Initialisation et diagnostic du système

Le BUS CEI 821 définit des protocoles qui permettent l'arrêt et le démarrage du système de manière ordonnée. Deux lignes de signal sont utilisées pour le séquençement de l'arrêt et du démarrage de l'alimentation: ACFAIL* et SYSRESET*. Une autre ligne de signal SYSFAIL* est utilisée pour la séquence de démarrage.

Le comportement des modules fonctionnels pendant l'opération de coupure de l'alimentation est spécifié comme suit:

RECOMMANDATION 5.1:

Concevoir les MAITRES tels qu'ils ne demandent le bus en aucun cas excepté pour toute opération concernant le traitement de la défaillance de l'alimentation, après que ACFAIL* eut été au niveau bas depuis 200 μ s.

RECOMMANDATION 5.2:

SI les MAITRES ou les CONTROLEURS D'INTERRUPTION ont une demande de bus en attente antérieure à la détection de ACFAIL* au niveau bas,
ALORS ils doivent limiter leur activité non liée au traitement de la défaillance d'alimentation à 200 μ s.

OBSERVATION 5.2:

Les accès au bus nécessaires pour sauvegarder et restituer les données du système à la mémoire du BUS CEI 821 dépendent de l'application et ne sont pas spécifiés ici. (Le système d'exploitation doit assurer que les données sauvegardées pendant l'opération de coupure de l'alimentation soient restituées avant la mise en fonction du système.) Dans le cas de système multiprocesseur, cela peut exiger des communications de processeur à processeur.

La ligne d'initialisation du système (SYSRESET*) à collecteur ouvert est commandée par le module CONTROLEUR D'ALIMENTATION ou par n'importe quelle carte, en réponse à la fermeture d'un bouton-poussoir.

OBSERVATION 5.3:

Si un contact d'initialisation par bouton-poussoir est utilisé, des circuits spéciaux sont nécessaires pour garantir que les rebonds du contact ne vont pas provoquer la non-observation par la carte d'un temps minimal de 200 ms pendant lequel SYSRESET* doit être au niveau bas.

REGLE 5.3:

L'EMETTEUR D'HORLOGE SYSTEME DOIT continuer de fournir le signal de forme d'onde spécifiée SYSCLK, sans tenir compte du niveau de la ligne SYSRESET*.

AUTORISATION 5.4:

Lorsque SYSRESET* descend au niveau bas, toute carte demandant plus de 200 ms pour terminer son initialisation PEUT activer son émetteur SYSRESET* au niveau bas pour maintenir SYSRESET* au niveau bas pour la durée nécessaire.

5.4 System initialization and diagnostics

The IEC 821 BUS provides protocols which allow the system to be shut down and powered up in an orderly manner. Two signal lines are used in the power-down and power-up sequences: ACFAIL* and SYS-RESET*. Another signal line is used in the power-up sequence, SYSFAIL*.

The following specify the behavior of the functional modules during the power-down process.

RECOMMENDATION 5.1:

Design MASTERS so that they do not request the bus for any purpose except power-fail activity after ACFAIL* has been low for 200 μ s.

RECOMMENDATION 5.2:

IF MASTERS or INTERRUPT HANDLERS have a bus request pending prior to detecting ACFAIL* low,

THEN they are to limit their subsequent non-power-fail activity to 200 μ s.

OBSERVATION 5.2:

Bus accesses required to save and restore system data to IEC 821 BUS memory depend upon the application, and are not specified here. (The operating system has to assure that data saved during the shut-down process is restored prior to system operation.) In the case of a multiprocessing system, this might require some processor-to-processor communication.

System Reset (SYSRESET*) is an open-collector line driven by a POWER MONITOR module or by any board in response to a push-button switch closure.

OBSERVATION 5.3:

Special circuitry is needed where push-button reset switches are used, to ensure that switch bounce does not cause the board to violate the 200 ms minimum SYSRESET* low time.

RULE 5.3:

The SYSTEM CLOCK DRIVER MUST continue to provide the specified SYSCLK waveform signal regardless of the state of the SYSRESET* line.

PERMISSION 5.4:

When SYSRESET* goes low, any board that requires more than 200 ms to complete its initialization MAY turn on its SYSRESET* driver low to maintain SYSRESET* low for the required period.

REGLE 5.4:

SI l'alimentation +5 V courant continu est dans les limites spécifiées lorsque SYSRESET* passe au niveau bas,
 ALORS les modules fonctionnels DOIVENT satisfaire aux REGLES données dans le tableau 5-1, dans les délais spécifiés après le passage de SYSRESET* au niveau bas.

REGLE 5.5:

SI SYSRESET* est au niveau bas lorsque l'alimentation +5 V courant continu entre dans les limites spécifiées,
 ALORS les modules fonctionnels DOIVENT satisfaire aux REGLES données dans le tableau 5-1, dans les délais spécifiés après l'entrée de l'alimentation dans ses limites spécifiées.

REGLE 5.6:

Après avoir satisfait aux REGLES du tableau 5-1, les modules fonctionnels NE DOIVENT PAS changer l'état de leurs émetteurs jusqu'à ce que SYSRESET* aille au niveau haut, excepté le cas où l'alimentation +5 V courant continu sort des limites spécifiées.

Tableau 5-1

Commandes émises par les modules pendant le démarrage et l'arrêt de l'alimentation

MODULE	NE DOIVENT PAS	APRES
MAITRES et CONTROLEURS D'INTERRUPTION	Commander AS*, DS0* ou DS1* du niveau haut au niveau bas	5 μ s
MAITRES et CONTROLEURS D'INTERRUPTION	Commander IACK*, LWORD*, AS, DS0*, DS1*, AMO-AM5, A01-A31, WRITE* ou D00-D31	20 μ s
ESCLAVES et GENERATEURS D'INTERRUPTION	Commander D00-D31, DTACK* ou BERR*	30 μ s
GENERATEURS D'INTERRUPTION	Commander IRQ1*-IRQ7*	30 μ s
LIMITEUR DE TEMPS D'OCCUPATION DU BUS	Commander BERR*	30 μ s
ARBITRE	Commander BG0IN*-BG3IN* du niveau haut au niveau bas	5 μ s
ARBITRE	Commander BG0IN*-BG3IN* au niveau bas	30 μ s
DEMANDEURS	Commander BBSY*	30 μ s

REGLE 5-7:

SI l'alimentation +5 V courant continu est dans les limites spécifiées lorsque SYSRESET* passe au niveau bas, et qu'un MAITRE ou un CONTROLEUR D'INTERRUPTION commande AS*, DS0* ou DS1* au niveau bas,
 ALORS il DOIT maintenir ces signaux de validation au niveau bas assez longtemps pour satisfaire aux temps minimaux au niveau bas donnés dans les chapitres 2 et 4.

RULE 5.4:

IF the +5 V d.c. power source is within its specified range when SYSRESET* goes low,
 THEN functional modules MUST satisfy the timing RULES given in Table 5-1 within the specified time after SYSRESET* goes low.

RULE 5.5:

IF SYSRESET* is low when the +5 V d.c. enters its specified range,
 THEN the functional modules MUST satisfy the timing RULES given in Table 5-1 within the specified time after the power source enters its specified range.

RULE 5.6:

After satisfying the RULES in Table 5-1, functional modules MUST NOT change the state of their drivers until SYSRESET* goes high, unless the +5 V d.c. power source exits its specified range.

Table 5-1

Module drive during power-up and power-down sequences

MODULE	MUST NOT	AFTER
MASTERS and INTERRUPT HANDLERS	Drive AS*, DS0*, or DS1* from high to low	5 μ s
MASTERS and INTERRUPT HANDLERS	Drive IACK*, LWORD*, AS*, DS0*, DS1*, AM0-AM5, A01-A31, WRITE*, or D00-D31	20 μ s
SLAVES and INTERRUPTERS	Drive D00-D31, DTACK*, or BERR*	30 μ s
INTERRUPTERS	Drive IRQ1*-IRQ7*	30 μ s
BUS TIMER	Drive BERR*	30 μ s
ARBITER	Drive BG0IN*-BG3IN* from high to low	5 μ s
ARBITER	Drive BG0IN*-BG3IN* low	30 μ s
REQUESTERS	Drive BBSY*	30 μ s

RULE 5.7:

IF the +5 V d.c. power source is within its specified range when SYSRESET* goes low, and a MASTER or INTERRUPT HANDLER is driving AS*, DS0*, or DS1* low,
 THEN it MUST maintain these strobes low long enough to satisfy the minimum low times given in Chapters 2 and 4.

SYSFAIL* est une ligne en collecteur ouvert maintenue au niveau bas lors du démarrage du système jusqu'à la fin des autotests du système (voir figure 5-6). Les considérations suivantes s'appliquent:

SUGGESTION 5.1:

Sur les cartes MAITRES intelligentes, inclure un bit du registre de contrôle accessible localement qui commande la broche SYSFAIL* de la carte au niveau bas à la mise sous tension. Cela permet à l'intelligence locale de la carte de faire un autotest local et de ne relâcher SYSFAIL* que si l'autotest se termine avec succès.

SUGGESTION 5.2:

Concevoir les cartes non intelligentes avec un bit du registre de contrôle accessible globalement et qui est initialisé de manière à commander SYSFAIL* au niveau bas. Cela permet au MAITRE sur le BUS CEI 821 de faire d'abord un test sur la carte non intelligente et ensuite d'écrire un bit du registre pour libérer l'émetteur de SYSFAIL*.

SUGGESTION 5.3:

Lorsqu'un bit du registre de contrôle commandant SYSFAIL* est inclus sur les cartes de BUS CEI 821, prévoir un LED d'état sur le panneau avant de chaque carte pour indiquer l'état de son bit SYSFAIL*. Ainsi, si une défaillance du système est indiquée par la ligne de signal SYSFAIL*, une inspection visuelle aidera à déterminer quelle carte est en défaut.

REGLE 5.8:

SI les bits du registre de contrôle SYSFAIL* sont inclus sur les cartes de BUS CEI 821,
ALORS ils DOIVENT commander SYSFAIL* au niveau bas dans un délai de 50 ms après que SYSRESET* fut passé au niveau bas comme indiqué sur la figure 5-6.

AUTORISATION 5.4:

Une carte de BUS CEI 821 PEUT aussi commander SYSFAIL* au niveau bas à n'importe quel moment du fonctionnement normal pour indiquer qu'il détecte une défaillance.

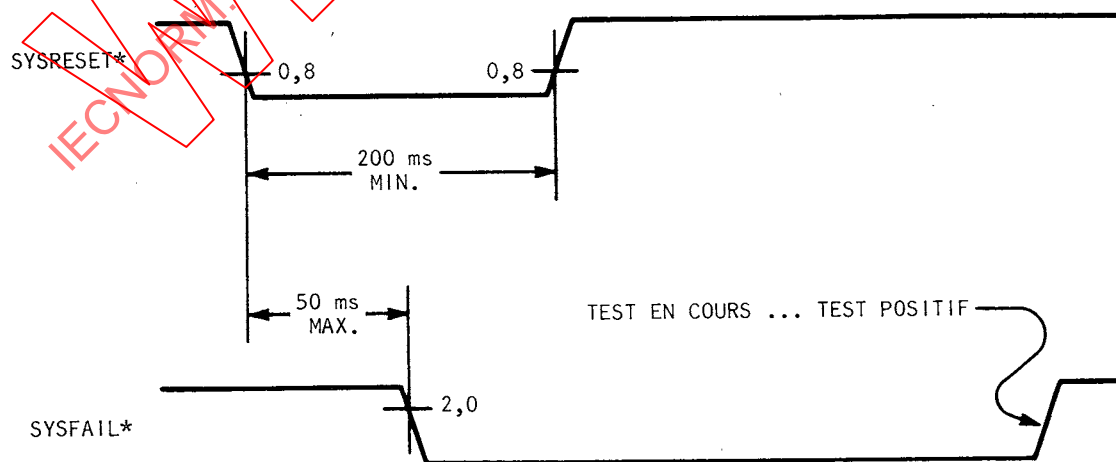


FIG. 5-6. - Chronogramme des signaux SYSRESET* et SYSFAIL*.

SYSFAIL* is an open collector line that is held low when the system is powered-up and remains low until system self-tests are complete (see Figure 5-6). The following applies:

SUGGESTION 5.1:

On intelligent MASTER boards, include a locally accessible control register bit that drives the board's SYSFAIL* pin low when power is first applied to the board. This permits the board's local intelligence to do a local self-test and release SYSFAIL* only if the self-test passes.

SUGGESTION 5.2:

Design non-intelligent boards with a globally accessible control register bit that is initialized to drive SYSFAIL* low. This allows a MASTER on the IEC 821 BUS to first run a test on the non-intelligent board and then write to the register bit to release the board's SYSFAIL* driver.

SUGGESTION 5.3:

Where the SYSFAIL* control register bit is included on IEC 821 BUS boards, provide a status LED on each board's front panel to indicate the status of its SYSFAIL* bit. Then, if a system failure is indicated by the SYSFAIL* signal line, a visual inspection will help determine which board has failed.

RULE 5.8:

IF SYSFAIL* control register bits are included on IEC 821 BUS boards,
THEN they MUST drive SYSFAIL* low within 50 ms after SYSRESET* goes low, as shown in Figure 5-6.

PERMISSION 5.4:

An IEC 821 BUS board MAY also drive SYSFAIL* low at any time during normal operation to indicate that it has detected some kind of failure.

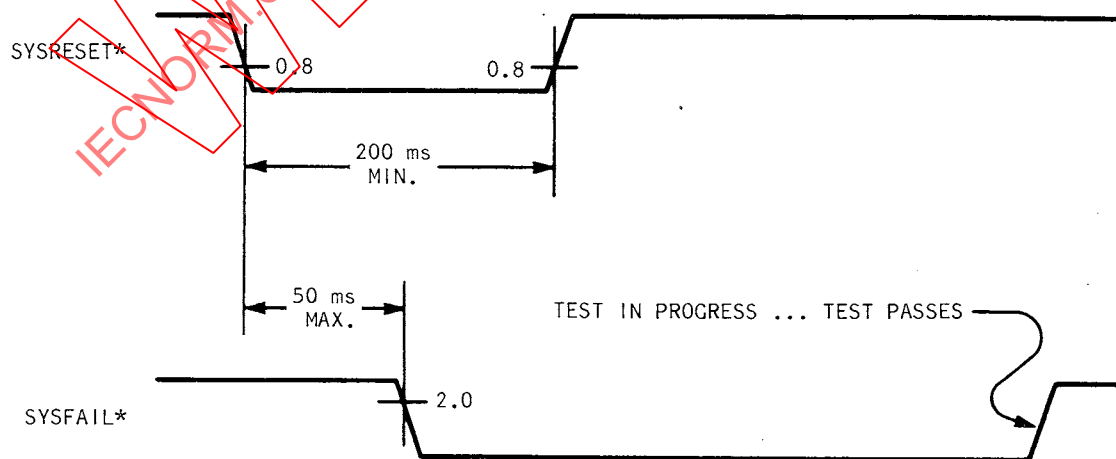


FIG. 5-6. - SYSRESET* and SYSFAIL* timing diagram.

5.5 Broches d'alimentation

La figure 5-7, page 354, donne la courbe du courant dans les broches d'alimentation du BUS CEI 821 en fonction de la température.

OBSERVATION 5.4:

Certaines broches de connecteurs ont une résistance de contact un peu plus élevée que d'autres lorsqu'elles sont enfichées dans le fond de panier. Cela produit un courant déséquilibré dans les broches parallèles. Supposons que deux broches parallèles transportent un courant total de 2 A. Si la résistance de contact sur l'une est de 1 m Ω et sur l'autre de 2 m Ω , une broche ne sera parcourue que par 0,67 A tandis que l'autre sera parcourue par 1,33 A!

REGLE 5.9:

Les broches du connecteur du BUS CEI 821 DOIVENT être capables de transporter au moins les courants indiqués par la courbe continue de la figure 5-7.

OBSERVATION 5.5:

Si une ou plusieurs broches d'alimentation ont une défaillance complète, tout le courant passe par les broches restantes. Par exemple, si la moitié des broches sont défaillantes, les broches restantes supporteront le double du courant normal. Selon l'intensité du courant, ces broches peuvent être endommagées.

SUGGESTION 5.4:

Lors de la conception d'une carte de BUS CEI 821 à consommation élevée, diviser la surface de la carte en zones alimentées par une grille d'alimentation séparée. Ne pas connecter ces grilles entre elles sur la carte de BUS CEI 821. Par contre, les connecter chacune à sa broche d'alimentation du BUS CEI 821.

OBSERVATION 5.6:

SI une carte de BUS CEI 821 double hauteur qui demande plus de puissance que son connecteur P_1 ne peut fournir est enfichée dans un châssis contenant seulement un fond de panier J_1 ,
ALORS ses broches d'alimentation du connecteur P_1 seront surchauffées et pourront être endommagées.

5.6 Ligne RESERVEE

REGLE 5.10:

La norme du BUS CEI 821 indique une ligne de signal qui est RESERVEE. La ligne RESERVEE est continue sur le fond de panier et adaptée. Cette ligne est prévue pour une utilisation future et NE DOIT PAS être utilisée dans les développements de cartes de BUS CEI 821.

5.5 Power pins

Figure 5-7, page 355, gives the current rating for the IEC 821 BUS power pins at various temperatures.

OBSERVATION 5.4:

Some connector pins have a slightly higher contact resistance than others when plugged into the backplane. This produces unbalanced current flow in pins which are paralleled. Suppose that two pins are paralleled and are carrying a total of two amperes of current. If the contact resistance on one is 1 m Ω and the other is 2 m Ω , then one pin will be carrying only 0.67 A while the other carries 1.33 A!

RULE 5.9:

IEC 821 BUS connector pins MUST be capable of at least carrying the currents shown by the solid line in Figure 5-7.

OBSERVATION 5.5:

If one or more power pins fail completely, all of the load current flows through the remaining pins. For example, if half of the pins fail, the remaining pins carry twice the normal current. Depending upon the load current, this might cause damage to these remaining good pins.

SUGGESTION 5.4:

When designing an IEC 821 BUS board with a high current load, divide the board's area into zones which are each powered by a separate power grid. Do not connect these grids to each other on the IEC 821 BUS board. Instead, connect each to its own IEC 821 BUS power pin.

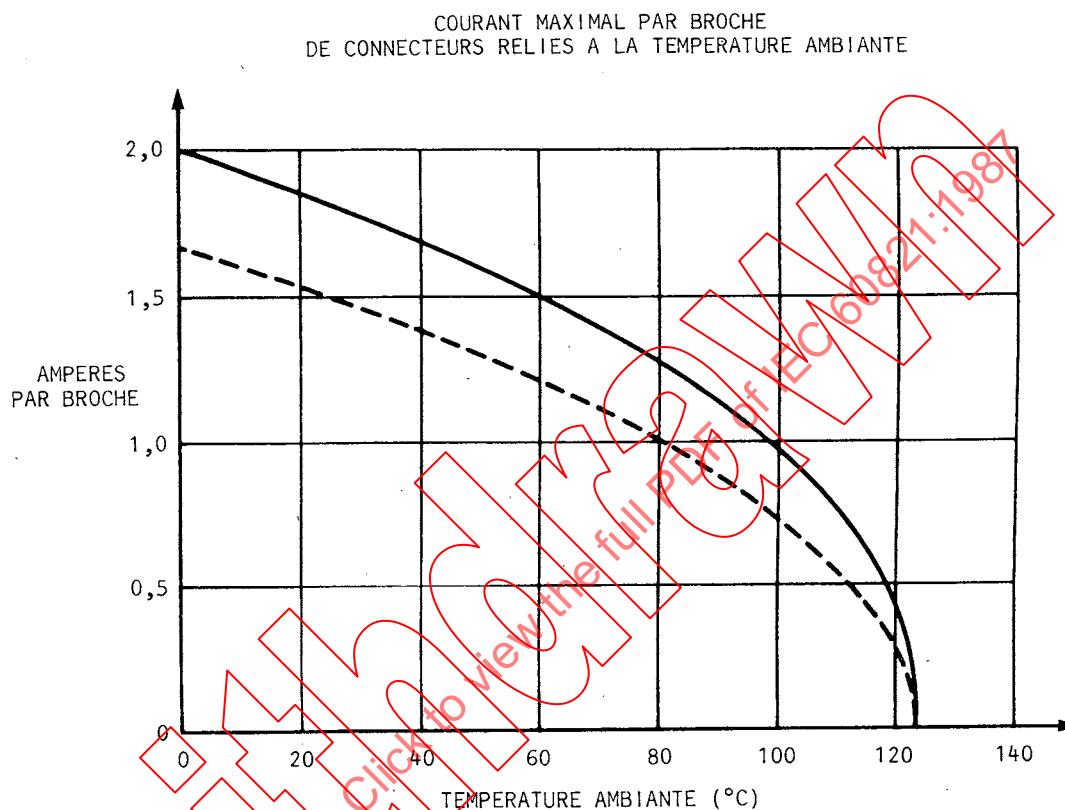
OBSERVATION 5.6:

IF a double height IEC 821 BUS board which draws more power than its P₁ connector can provide is plugged into a subrack which contains only a J₁ backplane,
THEN its P₁ power pins will overheat, and might be damaged.

5.6 RESERVED line

RULE 5-10:

The IEC 821 BUS standard labels one signal line as RESERVED. The RESERVED line is terminated and bussed. This line is set aside for future use and MUST NOT be used in any IEC 821 BUS board designs.

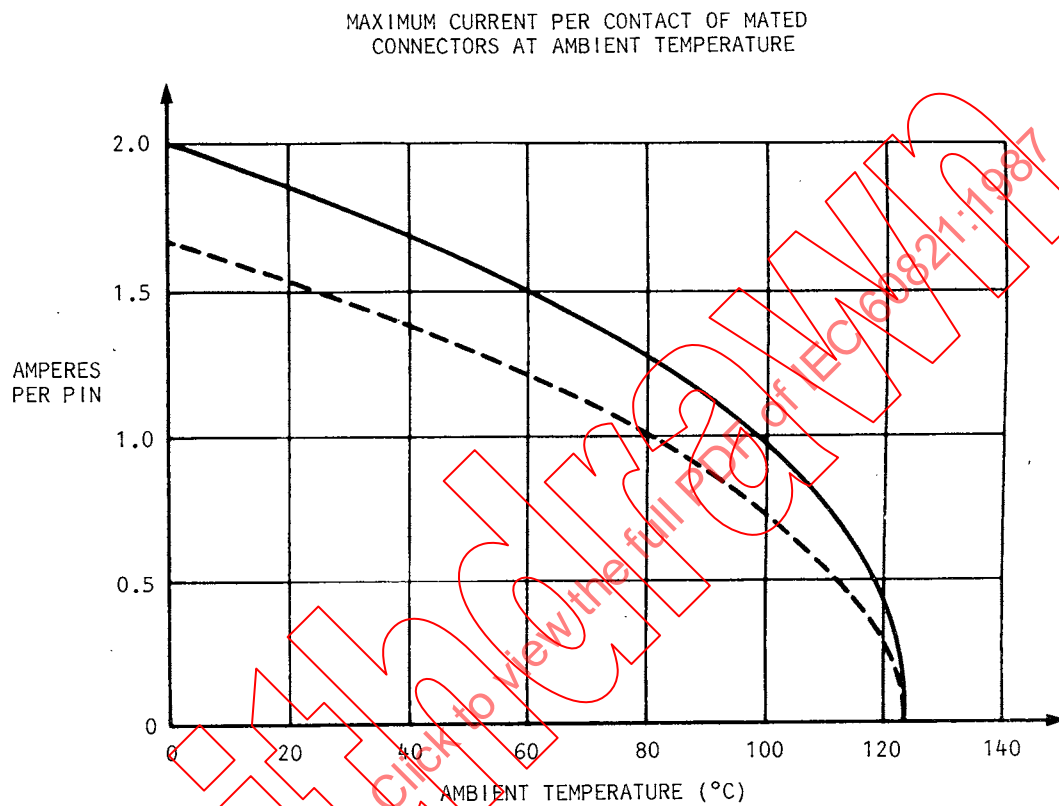


Note:

La courbe pointillée indique la limite supérieure du courant qui peut être fourni de façon fiable par une broche de +5 V courant continu lorsque deux ou plusieurs broches sont connectées à une grille d'alimentation commune de la carte.

La courbe continue indique la limite supérieure du courant qui peut être fourni de façon fiable par une broche de +5 V courant continu lorsque chacune de celle-ci est connectée à une grille séparée d'alimentation séparée de la carte.

FIG. 5-7. - Courant admissible dans les broches d'alimentation.



Note:

The dashed line shows the upper limit for current that can be safely drawn per +5 V d.c. power pin where two or more pins are connected to a common on-board power grid.

The solid line shows the upper limit for current that can safely drawn per +5 V d.c. power pin where each pin is connected to a separate on-board power grid.

FIG. 5-7. - Current rating for power pins.

CHAPITRE 6: SPECIFICATIONS ELECTRIQUES DU BUS CEI 821

6.1 Introduction

La transmission de données entre cartes de BUS CEI 821 telles que processeurs, mémoires et unités d'entrées/sorties a lieu à travers un ou deux fonds de panier suivant la conception. Les REGLES de ce chapitre assurent une chronologie correcte, un bruit minimal et minimisent les problèmes d'interférence sur les lignes de signaux du fond de panier. La conception des fonds de panier du BUS CEI 821 est basée sur les REGLES suivantes:

REGLE 6.1:

Les fonds de panier du BUS CEI 821 NE DOIVENT PAS avoir de pistes de signaux d'une longueur supérieure à 500 mm (19,68 in).

REGLE 6.2:

Les fonds de panier du BUS CEI 821 NE DOIVENT PAS avoir plus de 21 emplacements.

REGLE 6.3:

Pour les lignes qui exigent des adaptations (voir section 6.7), le fond de panier DOIT prévoir ces adaptations d'impédance aux deux extrémités de la ligne de signal.

REGLE 6.4:

Le fond de panier DOIT fournir des pistes d'alimentation pour la distribution de +5 V, +5 V STDBY, +12 V, et -12 V à toutes les broches d'alimentation spécifiées dans la section 7.6.

REGLE 6.5:

Le fond de panier DOIT fournir des connexions à toutes les broches de masse spécifiées dans la section 7.6.

AUTORISATION 6.1:

Les lignes de signaux du BUS CEI 821 sont normalement commandées par des émetteurs bipolaires mais n'importe quelle technologie conforme à cette norme PEUT être utilisée.

6.2 Distribution du courant d'alimentation

Le courant d'alimentation dans un système de BUS CEI 821 est distribué sur le ou les fonds de panier par des tensions continues réglées. Les tensions disponibles sont:

+5 V courant continu C'est la source d'alimentation principale pour la plupart des systèmes de BUS CEI 821 dont la logique TTL, les microprocesseurs MOS et les mémoires nécessitent cette tension.

+/-12 V courant continu Ces tensions sont souvent utilisées pour alimenter les émetteurs RS232C. Elles sont aussi utilisées parfois pour alimenter les dispositifs MOS et analogiques. Il arrive aussi qu'une tension de polarisation de -5 V courant continu ou des tensions ECL de -5,2 V courant continu soient dérivées de la source -12 V courant

CHAPTER 6: IEC 821 BUS ELECTRICAL SPECIFICATIONS

6.1 *Introduction*

The transmission of data between IEC 821 BUS boards such as processors, memories and I/O devices takes place over one or two backplanes, depending on the design. The rules in this chapter ensure proper timing, minimal noise and minimal crosstalk problems on the backplane signal lines. The design of IEC 821 BUS backplanes is governed by the following RULES:

RULE 6.1:

IEC 821 BUS backplanes MUST NOT have any signal conductors longer than 500 mm (19.68 in).

RULE 6.2:

IEC 821 BUS backplanes MUST NOT have more than 21 slots.

RULE 6.3:

For those lines requiring termination (see Section 6.7), the backplane MUST provide some means for terminating them at both ends of the signal line.

RULE 6.4:

The backplane MUST provide power conductors for distribution of +5 V, +5 V STDBY, +12 V, and -12 V to all of the power pins specified in Section 7.6.

RULE 6.5:

The backplane MUST provide ground connections to all of the ground pins specified in Section 7.6.

PERMISSION 6.1:

IEC 821 BUS signal lines are normally driven by bipolar drivers, but any technology which complies with this standard MAY be used.

6.2 *Power distribution*

Power in an IEC 821 BUS system is distributed on the backplane(s) as regulated direct current (d.c.) voltages. The available voltages are:

- + 5 V d.c. This is the main power source for most IEC 821 BUS systems. Most of the system circuitry, including TTL logic, MOS microprocessors, and memories, requires this voltage.
- +/- 12 V d.c. These are often used for powering RS232C drivers. They are also sometimes used for powering MOS and analog devices. In some cases -5 V d.c. bias voltage or -5.2 V d.c. ECL voltages are also derived from the

continu au moyen de régulateurs sur la carte. Habituellement, ces alimentations fournissent au système du BUS CEI 821 un courant inférieur à celui de la source +5 V courant continu.

+5 V courant continu STDBY Cette tension est utilisée pour maintenir la mémoire, les horloges du calendrier, etc., lorsque l'alimentation +5 V courant continu est défectueuse.

6.2.1 Spécifications des tensions courant continu

Le tableau 6-1 résume les spécifications des tensions courant continu. Les spécifications données représentent l'écart maximal admis mesuré sur les broches du connecteur de n'importe laquelle des cartes enfichée dans le fond de panier.

RECOMMANDATION 6.1:

Concevoir et connecter les fonds de panier afin que le point de contrôle de la source d'alimentation soit situé près du centre du fond de panier et aussi près que possible du point où arrive l'alimentation dans le fond de panier.

OBSERVATION 6.1:

En plaçant le point de contrôle près du point d'entrée d'alimentation, on évite aux modules proches du point d'entrée d'alimentation de recevoir un niveau de tension trop élevé.

Tableau 6-1

Spécifications des tensions du bus

MNEMONIQUE	DESCRIPTION	ECART ADMIS (voir OBSERVATIONS)	ONDULATION/BRUIT AU-DESSOUS DE 10 MHz (Crête à crête)
+5 V	+5 V courant continu	+0,25 V/-0,125 V	50 mV
+12 V	Alimentation +12 V courant continu	+0,60 V/-0,36 V	50 mV
-12 V	Alimentation -12 V courant continu	-0,60 V/+0,36 V	50 mV
+5 V STDBY	+5 V courant continu veille	+0,25 V/-0,125 V	50 mV
GND	Masse	REFERENCE	

OBSERVATION 6.2:

Les écarts non symétriques donnés dans le tableau 6-1 garantissent que l'alimentation courant continu reste dans les tolérances demandées par la plupart des circuits intégrés malgré les baisses normales de tension dans le réseau de distribution d'alimentation.

-12 V d.c. source using on-board regulators. These supplies normally don't supply as much power to the IEC 821 BUS system as the +5 V d.c. source does.

+5 V d.c. STDBY

This is used to sustain memory, time-of-day clocks, etc., when the +5 V d.c. power is lost.

6.2.1 D.C. voltage specifications

Table 6-1 summarizes the d.c. voltage specifications. The listed specifications are the maximum allowed variance as measured at the connector pins of any card plugged into the backplane.

RECOMMENDATION 6.1:

Design and connect backplanes so that the power supply sense point is located somewhere near the center of the backplane, and as close as possible to the point where power is introduced into the backplane.

OBSERVATION 6.1:

Placing the power supply sense point near the power input point prevents boards near the power input point from receiving too high a voltage.

Table 6-1

Bus voltage specifications

MNEMONICS	DESCRIPTION	ALLOWED VARIATION (see OBSERVATIONS)	RIPPLE/NOISE BELOW 10 MHz (Peak-to-Peak)
+5 V	+5 V d.c.	+0.25 V/-0.125 V	50 mV
+12 V	+12 V d.c. power	+0.60 V/-0.36 V	50 mV
-12 V	-12 V d.c. power	-0.60 V/+0.36 V	50 mV
+5 V STDBY	+5 V d.c. standby	+0.25 V/-0.125 V	50 mV
GND	Ground	REFERENCE	

OBSERVATION 6.2:

The non-symmetric variation given in Table 6-1 ensures that the d.c. power remains within the tolerance required by most ICs despite the typical voltage drops that occur in the power distribution network.

OBSERVATION 6.3:

La puissance consommée par certains systèmes varie dans une large gamme pendant le fonctionnement normal du système. Par exemple, le rafraîchissement de la mémoire dynamique peut provoquer de grandes variations si une grande partie de la mémoire est rafraîchie en une fois. Dans ce cas le temps de réponse du système de distribution de la tension devient important.

RECOMMANDATION 6.2:

Utiliser un condensateur de découplage sur les cartes de BUS CEI 821 pour minimiser les effets des transitoires d'alimentation.

6.2.2 Caractéristiques électriques des broches et supports des connecteurs

REGLE 6.6:

Le connecteur 96 broches utilisé par le BUS CEI 821 DOIT avoir les caractéristiques suivantes:

- Tension d'isolement: ≥ 100 V courant continu de broche à broche.
- Résistance de contact: ≤ 50 m Ω pour le courant nominal.
- Résistance d'isolement: ≥ 100 M Ω de broche à broche.

6.3 Caractéristiques électriques des signaux

REGLE 6.7:

Les cartes de BUS CEI 821 NE DOIVENT pas commander de ligne de signal du fond de panier à un niveau permanent supérieur au plus haut niveau d'alimentation des broches +5 V ou inférieur au plus bas niveau des broches de masse.

REGLE 6.8:

Les cartes de BUS CEI 821 DOIVENT utiliser des émetteurs et des récepteurs qui correspondent aux caractéristiques suivantes:

- Niveau de sortie bas de l'émetteur en régime permanent $\leq 0,6$ V.
- Niveau d'entrée bas du récepteur en régime permanent $\leq 0,8$ V.
- Niveau de sortie haut de l'émetteur en régime permanent $\geq 2,4$ V.
- Niveau d'entrée haut du récepteur en régime permanent $\geq 2,0$ V.

La figure 6-1 donne une représentation graphique simple de ces niveaux.

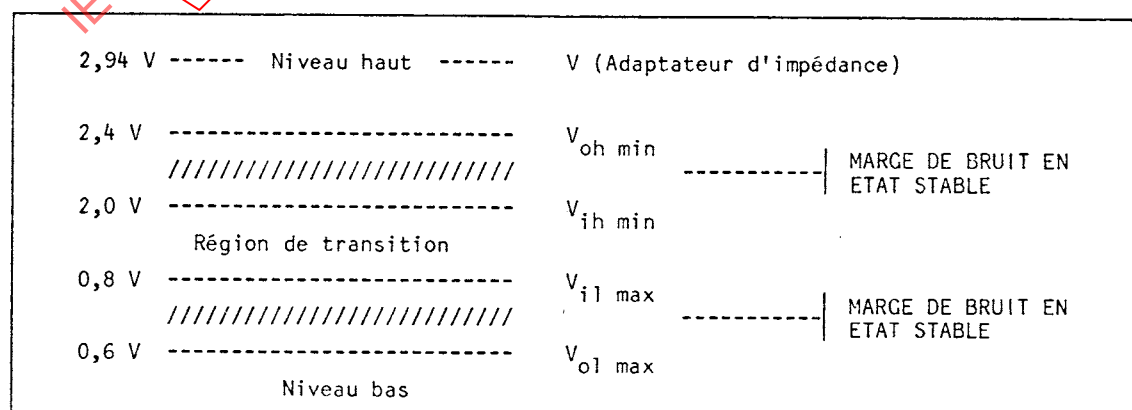


FIG. 6-1. - Niveaux des signaux du BUS CEI 821.

OBSERVATION 6.3:

The power consumed by some systems fluctuates over a wide range during normal system operation. For example, dynamic memory refreshing might cause significant fluctuations if large amounts of memory are refreshed at one time. In this case, the response time of the voltage distribution system becomes important.

RECOMMENDATION 6.2:

Use bypass capacitors on IEC 821 BUS boards to minimize the effects of power transient.

6.2.2 Pin and socket connector electrical ratings**RULE 6.6:**

The 96-pin connector used by the IEC 821 BUS MUST provide the following:

- Voltage rating: ≥ 100 V d.c., isolation pin to pin.
- Contact resistance: ≤ 50 m Ω , at rated current.
- Insulation resistance: ≥ 100 M Ω , pin to pin.

6.3 Electrical signal characteristics**RULE 6.7:**

IEC 821 BUS boards MUST NOT drive any backplane signal line to a higher steady-state voltage than the highest voltage on any of its +5 V power pins, or to a lower steady-state voltage than the lowest voltage on any of its groundpins.

RULE 6.8:

IEC 821 BUS boards MUST use drivers and receivers that meet the following characteristics:

- Steady-state driver low output level ≤ 0.6 V.
- Steady-state receiver low input level ≤ 0.8 V.
- Steady-state driver high output level ≥ 2.4 V.
- Steady-state receiver high input level ≥ 2.0 V.

Figure 6-1 gives a simple graphic representation of these levels.

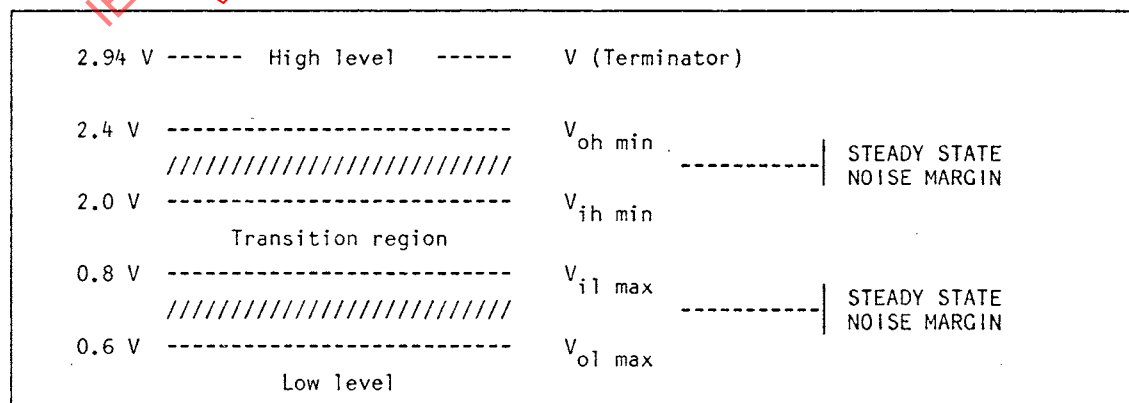


FIG. 6-1. - IEC 821 BUS signal levels.

Les cartes de BUS CEI 821 commandent les lignes du fond de panier par des circuits à trois états, des collecteurs ouverts et des émetteurs "totem-pole". La section 6.4 spécifie les contraintes de commande et de charge pour les différentes lignes de signaux. La section 6.7 résume les types d'émetteurs utilisés pour commander chaque ligne de signal.

REGLE 6.9:

Si l'on mesure les seuils de tension sur une carte de BUS CEI 821 pour vérifier la conformité avec les spécifications de temps, la référence de masse DOIT être prise sur la broche de masse de la carte la plus proche de la broche de signal où s'effectue la mesure, et le signal de tension DOIT être mesuré à la broche du connecteur de la carte.

6.4 Spécifications de commande et de réception du bus

Cette section définit les spécifications des émetteurs et des récepteurs pour toutes les lignes de signaux du BUS CEI 821. Le tableau 6-2 donne tous les signaux et indique les paragraphes qui les décrivent.

Tableau 6-2

Spécifications de commande et de réception du bus

NOM DU SIGNAL	PARAGRAPHE 6.4.2.x
A01-A31	2
ACFAIL*	5
AM0-AM5	2
AS*	1
BBSY*	5
BCLR*	3
BERR*	5
BG0OUT*-BG3OUT*	4
BR0*-BR3*	5
D00-D31	2
DS0*	1
DS1*	1
DTACK*	5
IACK*	2, 5
IACKOUT*	4
IRQ1*-IRQ7*	5
LWORD*	2
SERCLK	3
SYSCLK	3
SYSFAIL*	5
SYSRESET*	5
WRITE*	2

6.4.1 Définitions des circuits de commande du bus

Les circuits de commande type "totem-pole", trois états et à collecteur ouvert sont définis comme suit:

"Totem-pole" - Circuit de commande actif aux deux niveaux qui absorbe du courant au niveau bas et en fournit au niveau haut. Les circuits de commande totem-pole sont utilisés pour les signaux ayant un seul circuit de commande par ligne (par exemple les lignes de chaîne de reconnaissance).

IEC 821 BUS boards drive the backplane lines with three-state, open collector, and totem-pole drivers. Section 6.4 specifies the drive and loading requirements for the various signal lines. Section 6.7 provides a summary, showing which types of drivers are used to drive each signal line.

RULE 6.9:

When making voltage threshold measurements on an IEC 821 BUS board to verify compliance with timing specifications, the ground reference **MUST** be taken from the board's ground pin nearest the signal pin being measured, and the signal voltage **MUST** be measured on the board's connector pin.

6.4 Bus driving and receiving requirements

This section defines the driver and receiver specifications for all IEC 821 BUS signal lines. Table 6-2 lists all of the signals and shows which of the following paragraphs discusses it.

Table 6-2

Bus driving and receiving requirements

SIGNAL NAME	PARAGRAPH 6.4.2.x
A01-A31	2
ACFAIL*	5
AM0-AM5	2
AS*	1
BBSY*	5
BCLR*	3
BERR*	5
BC0OUT*-BC3OUT*	4
BR0*-BR3*	5
D00-D31	2
DS0*	1
DS1*	1
DTACK*	5
IACK*	2, 5
IACKOUT*	4
IRQ1*-IRQ7*	5
LWORD*	2
SERCLK	3
SYSCLK	3
SYSFAIL*	5
SYSRESET*	5
WRITE*	2

6.4.1 Bus driver definitions

Totem-pole, three-state, and open-collector drivers are defined as follows:

Totem-pole - An active driver in both states which sinks current in the low state and sources current in the high state. Totem-pole drivers are used on signals having only a single driver per line (e.g. daisy-chain lines).

Trois états - Identique à un circuit de commande "totem-pole", excepté qu'il peut être mis dans un état haute impédance (circuit de commande déconnecté) en plus des états logiques haut et bas. Les circuits de commande trois états sont utilisés sur les lignes qui peuvent être commandées par plusieurs dispositifs en différents points du bus (par exemple les lignes d'adresse ou de données). Il ne peut y avoir qu'un seul de ces circuits de commande actifs à un instant donné.

Collecteur ouvert - Absorbe du courant au niveau bas mais ne fournit pas de courant significatif au niveau haut. Les résistances d'adaptation d'impédance sur le fond de panier garantissent que la tension de la ligne de signal est au niveau haut chaque fois qu'elle n'est pas commandée. Les circuits de commande à collecteur ouvert sont utilisés sur les lignes de signaux qui peuvent être commandées simultanément par plusieurs dispositifs (par exemple lignes de demande d'interruption et de demande du bus).

6.4.2 *REGLES pour commander et charger toutes les lignes de signaux du BUS CEI 821*

REGLE 6.10:

Toutes les cartes de BUS CEI 821 DOIVENT fournir, sur chaque ligne de signal du BUS CEI 821 qu'elles contrôlent, une limitation qui empêche des dépassements négatifs en dessous de -1,5 V.

OBSERVATION 6.4:

Les dispositifs standards 74LSxxx et 74Fxxx ont des diodes de limitation interne sur leurs entrées qui satisferont la limitation spécifiée par la REGLE 6.10.

REGLE 6.11:

Les récepteurs du BUS CEI 821 DOIVENT garantir la détection d'un niveau logique haut au-dessus du seuil de 2,0 V, comme indiqué sur la figure 6-1, page 360.

REGLE 6.12:

Les récepteurs du BUS CEI 821 DOIVENT garantir la détection d'un niveau logique bas au-dessous du seuil de 0,8 V, comme indiqué sur la figure 6-1.

AUTORISATION 6.2:

Un circuit de commande trois états PEUT être utilisé comme un circuit de commande "totem-pole" si sa sortie est active en permanence.

6.4.2.1 *REGLES pour commander et charger les lignes de signaux trois états à courant élevé*

(AS*, DS0*, DS1*)

REGLE 6.13:

SI une carte de BUS CEI 821 commande les lignes AS*, DS0* ou DS1*,

ALORS les émetteurs sur ces lignes DOIVENT satisfaire aux spécifications suivantes:

Three-state - Similar to a totem-pole driver except that it can go to a high impedance state (drivers turned off) in addition to the low and high logic states. Three-state drivers are used for lines that can be driven by several devices at different points on the bus (e.g. address or data lines). Only one of these drivers can be active at any one time.

Open-collector - Sinks current in the low state but sources no significant current in the high state. Terminating resistors on the backplane ensure that the signal line voltage rises to a high level whenever it is not driven low. Open-collector drivers are used for signal lines which can be driven by several devices simultaneously (e.g. interrupt and bus request lines).

6.4.2 *Driving and loading RULES for all IEC 821 BUS lines*

RULE 6.10:

All IEC 821 BUS boards MUST provide clamping on each IEC 821 BUS signal line that it monitors to prevent negative excursions below -1.5 V.

OBSERVATION 6.4:

Standard 74LSxxx and 74Fxxx devices have internal clamping diodes on their inputs that will satisfy the clamping requirement specified in RULE 6.10.

RULE 6.11:

IEC 821 BUS receivers MUST guarantee detection of a high logic level above a threshold of 2.0 V, as shown in Figure 6-1, page 361.

RULE 6.12:

IEC 821 BUS receivers MUST guarantee detection of a low logic level below a threshold of 0.8 V, as shown in Figure 6-1.

PERMISSION 6.2:

A three-state driver MAY be used as a totem-pole driver if its output is permanently enabled.

6.4.2.1 *Driving and loading RULES for high current three-state lines*

(AS*, DSO*, DS1*)

RULE 6.13:

IF an IEC 821 BUS board drives the lines AS*, DSO*, or DS1*,
THEN its drivers for these lines MUST meet the following specifications:

- Courant absorbé au niveau bas: $IOL \geq 64 \text{ mA}$.
- Tension au niveau bas: $VOL \leq 0,6 \text{ V}$ à $IOL = 64 \text{ mA}$.
- Courant de source au niveau haut: $IOH \geq 3 \text{ mA}$.
- Tension au niveau haut: $VOH \geq 2,4 \text{ V}$ à $IOH = 3 \text{ mA}$.
- Courant minimal de la source avec une broche de la carte à la masse: $IOS \geq 50 \text{ mA}$ à 0 V .
- Courant maximal de la source avec une broche de la carte à la masse: $IOS \leq 225 \text{ mA}$ à 0 V .

REGLE 6.14:

Lorsque les émetteurs sont à l'état haute impédance, les cartes de BUS CEI 821 DOIVENT limiter leur charge sur les lignes AS*, DS0* et DS1* aux valeurs suivantes:

- Courant produit par la carte à $0,6 \text{ V}$, incluant le courant de fuite: $IOZL + IIL \leq 450 \text{ }\mu\text{A}$.
- Courant absorbé par la carte à $2,4 \text{ V}$, incluant le courant de fuite: $IOZH + IIH \leq 100 \text{ }\mu\text{A}$.
- Charge capacitive totale sur le signal incluant la piste du signal: $CT \leq 20 \text{ pF}$.

OBSERVATION 6.5:

Les courants fournis et absorbés indiqués dans les REGLES 6.13 et 6.14 incluent les courants fournis et absorbés par les émetteurs et les récepteurs sur la carte.

SUGGESTION 6.1:

Utiliser les dispositifs 74S241, ou 74F241/244 comme émetteur sur les lignes AS*, DS0* et DS1*.

Utiliser les dispositifs 74LS240, 74LS241 ou 74LS244 comme récepteur sur les lignes AS*, DS0* et DS1*.

6.4.2.2 REGLES pour commander et charger les lignes de signaux trois états standards

(A01-A31, D00-D31, AM0-AM5, IACK*, LWORD*, WRITE*)

REGLE 6.15:

Si une carte de BUS CEI 821 commande les lignes A01-A31, D00-D31, AM0-AM5, IACK*, LWORD* ou WRITE*,

ALORS les émetteurs sur ces lignes DOIVENT satisfaire aux spécifications suivantes:

- Courant absorbé au niveau bas: $IOL \geq 48 \text{ mA}$.
- Tension au niveau bas: $VOL \leq 0,6 \text{ V}$ à $IOL = 48 \text{ mA}$.
- Courant de source au niveau haut: $IOH \geq 3 \text{ mA}$.
- Tension au niveau haut: $VOH \geq 2,4 \text{ V}$ à $IOH = 3 \text{ mA}$.
- Courant minimal de la source avec une broche de la carte à la masse: $IOS \geq 50 \text{ mA}$ à 0 V .
- Courant maximal de la source avec une broche de la carte à la masse: $IOS \leq 225 \text{ mA}$ à 0 V .

REGLE 6.16:

Lorsque les émetteurs sont à l'état haute impédance, les cartes de BUS CEI 821 DOIVENT limiter leur charge sur les lignes A01-A31, D00-D31, AM0-AM5, IACK*, LWORD* et WRITE* aux valeurs suivantes:

- Low state sink current: $IOL \geq 64 \text{ mA}$.
- Low state voltage: $VOL \leq 0.6 \text{ V}$ at $IOL = 64 \text{ mA}$.
- High state source current: $IOH \geq 3 \text{ mA}$.
- High state voltage: $VOH \geq 2.4 \text{ V}$ at $IOH = 3 \text{ mA}$.
- Minimum source current with board pin grounded: $IOS \geq 50 \text{ mA}$ at 0 V .
- Maximum source current with board pin grounded: $IOS \leq 225 \text{ mA}$ at 0 V .

RULE 6.14:

When drivers are turned off, IEC 821 BUS boards MUST limit their loading of the lines AS*, DS0*, and DS1* to the following values:

- Current sourced by board at 0.6 V , including leakage current: $IOZL + IIL \leq 450 \mu\text{A}$.
- Current sunk by board at 2.4 V , including leakage current: $IOZH + IIH \leq 100 \mu\text{A}$.
- Total capacitive load on signal, including signal trace: $CT \leq 20 \text{ pF}$.

OBSERVATION 6.5:

The source and sink currents listed in RULES 6.13 and 6.14 include both driver and receiver currents sourced and sunk on the board.

SUGGESTION 6.1:

Use 74S241 or 74F241/244 devices to drive the lines AS*, DS0*, and DS1*.

Use 74LS240, 74LS241, or 74LS244 devices to receive the lines AS*, DS0*, and DS1*.

6.4.2.2 Driving and loading RULES for standard three-state lines

(A01-A31, D00-D31, AM0-AM5, IACK*, LWORD*, WRITE*)

RULE 6.15:

IF an IEC 821 BUS board drives the lines A01-A31, D00-D31, AM0-AM5, IACK*, LWORD* or WRITE*,
THEN its drivers for these lines MUST meet the following specifications:

- Low state sink current: $IOL \geq 48 \text{ mA}$.
- Low state voltage: $VOL \leq 0.6 \text{ V}$ at $IOL = 48 \text{ mA}$.
- High state source current: $IOH \geq 3 \text{ mA}$.
- High state voltage: $VOH \geq 2.4 \text{ V}$ at $IOH = 3 \text{ mA}$.
- Minimum source current with board pin grounded: $IOS \geq 50 \text{ mA}$ at 0 V .
- Maximum source current with board pin grounded: $IOS \leq 225 \text{ mA}$ at 0 V .

RULE 6.16:

When drivers are turned off, IEC 821 BUS boards MUST limit their loading of the lines A01-A31, D00-D31, AM0-AM5, IACK*, LWORD*, and WRITE* to the following values:

- Courant produit par la carte à 0,6 V incluant le courant de fuite: $IOZL + IIL \leq 700 \mu A$.
- Courant absorbé par la carte à 2,4 V, incluant le courant de fuite: $IOZH + IIH \leq 150 \mu A$.
- Charge capacitive totale sur le signal incluant la piste du signal: $CT \leq 20 pF$.

OBSERVATION 6.6:

Les courants fournis et absorbés spécifiés dans les REGLES 6.15 et 6.16 incluent les courants fournis et absorbés par les émetteurs et les récepteurs sur la carte.

SUGGESTION 6.2:

Utiliser les dispositifs 74ALS645-1, 74F244, 74AS573 ou 74AS580 comme émetteur sur les lignes A01-A31, D00-D31, AM0-AM5, IACK*, LWORD* et WRITE*.

Utiliser les dispositifs 74LS240, 74LS241 ou 74LS244 comme récepteur sur les lignes A01-A31, D00-D31, AM0-AM5, IACK*, LWORD* et WRITE*.

Utiliser les dispositifs 74ALS645-1, 74ALS245A-1, 74ALS646-1 ou 74ALS648-1 pour émettre et recevoir sur les lignes A01-A31, D00-D31, AM0-AM5, IACK*, LWORD* et WRITE*.

6.4.2.3 REGLES pour commander et charger les lignes "totem-pole" à courant élevé

(SERCLK, SYSCLK, BCLR*)

REGLE 6.17:

Les systèmes du BUS CEI 821 DOIVENT n'avoir qu'une seule carte commandant chacune des lignes SERCLK, SYSCLK, ou BCLR*. Les émetteurs sur ces lignes DOIVENT satisfaire aux spécifications suivantes.

- Courant absorbé au niveau bas: $IOL \geq 64 mA$.
- Tension au niveau bas: $VOL \leq 0,6 V$ à $IOL = 64 mA$.
- Courant de source au niveau haut: $IOH \geq 3 mA$.
- Tension au niveau haut: $VOH \geq 2,4 V$ à $IOH = 3 mA$.
- Courant minimal de la source avec une broche de la carte à la masse: $IOS \geq 50 mA$ à 0 V.
- Courant maximal de la source avec une broche de la carte à la masse: $IOS \leq 255 mA$ à 0 V.

REGLE 6.18:

Toutes les cartes de BUS CEI 821 DOIVENT limiter leur charge sur les lignes SERCLK, SYSCLK et BCLR* aux valeurs suivantes:

- Courant produit par la carte à 0,6 V, incluant le courant de fuite: $IOZL + IIL \leq 600 \mu A$.
- Courant absorbé par la carte à 2,4 V incluant le courant de fuite: $IOZH + IIH \leq 50 \mu A$.
- Charge capacitive totale sur le signal, incluant la piste de signal pour le contrôleur de système (avec émetteurs): $CT \leq 20 pF$.
- Charge capacitive totale sur le signal, incluant la piste de signal pour toutes les autres cartes (sans émetteur): $CT \leq 12 pF$.

- Current sourced by board at 0.6 V, including leakage current: $IOZL + IIL \leq 700 \mu A$.
- Current sunk by board at 2.4 V, including leakage current: $IOZH + IIH \leq 150 \mu A$.
- Total capacitive load on signal, including signal trace: $CT \leq 20 \text{ pF}$.

OBSERVATION 6.6:

The source and sink currents specified in RULES 6.15 and 6.16 include both driver and receiver currents sourced and sunk on the board.

SUGGESTION 6.2:

Use 74ALS645-1, 74F244, 74AS573, or 74AS580 devices to drive the lines A01-A31, D00-D31, AM0-AM5, IACK*, LWORD*, and WRITE*.

Use 74LS240, 74LS241, or 74LS244 devices to receive the lines A01-A31, D00-D31, AM0-AM5, IACK*, LWORD*, and WRITE*.

Use 74ALS645-1, 74ALS245A-1, 74ALS646-1, or 74ALS648-1 devices to transceive the lines A01-A31, D00-D31, AM0-AM5, IACK*, LWORD*, and WRITE*.

6.4.2.3 *Driving and loading RULES for high current totem-pole lines*

(SERCLK, SYSCLK, BCLR*)

RULE 6.17:

IEC 821 BUS systems MUST have no more than one board driving each of the lines SERCLK, SYSCLK, or BCLR*. Its drivers for these lines MUST meet the following specifications:

- Low state sink current: $IOL \geq 64 \text{ mA}$.
- Low state voltage: $VOL \leq 0.6 \text{ V}$ at $IOL = 64 \text{ mA}$.
- High state source current: $IOH \geq 3 \text{ mA}$.
- High state voltage: $VOH \geq 2.4 \text{ V}$ at $IOH = 3 \text{ mA}$.
- Minimum source current with board pin grounded: $IOS \geq 50 \text{ mA}$ at 0 V.
- Maximum source current with board pin grounded: $IOS \leq 255 \text{ mA}$ at 0 V.

RULE 6.18:

All IEC 821 BUS boards MUST limit their loading of the lines SERCLK, SYSCLK, and BCLR* to the following values:

- Current sourced by board at 0.6 V including leakage current: $IOZL + IIL \leq 600 \mu A$.
- Current sunk by board at 2.4 V, including leakage current: $IOZH + IIH \leq 50 \mu A$.
- Total capacitive load on signal, including signal trace, for system controllers (which have drivers): $CT \leq 20 \text{ pF}$.
- Total capacitive load on signal, including signal trace, for other boards (which have no drivers): $CT \leq 12 \text{ pF}$.

OBSERVATION 6.7:

Les courants fournis et absorbés spécifiés dans les REGLES 6.17 et 6.18 incluent les courants fournis et absorbés par les émetteurs et les récepteurs sur la carte.

SUGGESTION 6.3:

Utiliser les dispositifs 74S241 ou 74F241/244 comme émetteur sur les lignes SERCLK, SYSCLK et BCLR*.

Utiliser les dispositifs 74LS240, 74LS241 ou 74LS244 comme récepteur sur les lignes SERCLK, SYSCLK et BCLR*.

6.4.2.4 REGLES pour commander et charger les lignes "totem-pole" standards

(BG0OUT*-BG3OUT*/BG0IN*-BG3IN*, IACKOUT*/IACKIN*)

REGLE 6.19:

SI une carte de BUS CEI 821 commande les lignes BG0OUT*-BG3OUT*/BG0IN*-BG3IN* ou IACKOUT*/IACKIN*,

ALORS les émetteurs sur ces lignes DOIVENT satisfaire aux spécifications suivantes:

- Courant absorbé au niveau bas: $I_{OL} \geq 8 \text{ mA.}$
- Tension au niveau bas: $V_{OL} \leq 0,6 \text{ V à } I_{OL} = 8 \text{ mA.}$
- Courant de source au niveau haut: $I_{OH} \geq 400 \text{ } \mu\text{A.}$
- Tension au niveau haut: $V_{OH} \geq 2,7 \text{ V à } I_{OH} = 400 \text{ } \mu\text{A.}$

REGLE 6.20:

Toutes les cartes de BUS CEI 821 DOIVENT limiter leur charge sur les lignes BG0OUT*-BG3OUT*/BG0IN*-BG3IN* et IACKOUT*/IACKIN* aux valeurs suivantes:

- Courant produit par la carte à 0,6 V incluant le courant de fuite: $I_{OZL} + I_{IL} \leq 600 \text{ } \mu\text{A.}$
- Courant absorbé par la carte à 2,4 V, incluant le courant de fuite: $I_{OZH} + I_{IH} \leq 50 \text{ } \mu\text{A.}$
- Charge capacitive totale sur le signal incluant la piste du signal: $CT \leq 20 \text{ pF.}$

OBSERVATION 6.8:

Les courants fournis et absorbés dans les REGLES 6.19 et 6.20 incluent les courants fournis et absorbés par les émetteurs et les récepteurs sur la carte.

SUGGESTION 6.4:

Utiliser n'importe quel dispositif standard qui satisfasse aux spécifications ci-dessus comme émetteur sur les lignes BG0OUT*-BG3OUT*/BG0IN*-BG3IN* et IACKOUT*/IACKIN*.

Utiliser les dispositifs 74LS240, 74LS241 ou 74LS244 comme récepteur sur les lignes BG0OUT*-BG3OUT*/BG0IN*-BG3IN* et IACKOUT*/IACKIN*.

OBSERVATION 6.7:

The source and sink currents specified in RULES 6.17 and 6.18 include both driver and receiver currents sourced and sunk on the board.

SUGGESTION 6.3:

Use 74S241 or 74F241/244 devices to drive the lines SERCLK, SYSCLK, and BCLR*.

Use 74LS240, 74LS241, or 74LS244 devices to receive the lines SERCLK, SYSCLK, and BCLR*.

6.4.2.4 Driving and loading RULES for standard totem-pole lines

(BG0OUT*-BG3OUT*/BG0IN*-BG3IN*, IACKOUT*/IACKIN*)

RULE 6.19:

IF an IEC 821 BUS board drives the lines BG0OUT*-BG3OUT*/-BG0IN*-BG3IN*, or IACKOUT*/IACKIN*,
THEN its drivers for these lines MUST meet the following specifications:

- Low state sink current: $IOL \geq 8 \text{ mA}$.
- Low state voltage: $VOL \leq 0.6 \text{ V}$ at $IOL = 8 \text{ mA}$.
- High state source current: $IOH \geq 400 \text{ } \mu\text{A}$.
- High state voltage: $VOH \geq 2.7 \text{ V}$ at $IOH = 400 \text{ } \mu\text{A}$.

RULE 6.20:

All IEC 821 BUS boards MUST limit their loading of each of the lines BG0OUT*-BG3OUT*/BG0IN*-BG3IN*, and IACKOUT*/IACKIN* to the following values:

- Current sourced by board at 0.6 V, including leakage current: $IOZL + IIL \leq 600 \text{ } \mu\text{A}$.
- Current sunk by board at 2.4 V, including leakage current: $IOZH + IIH \leq 50 \text{ } \mu\text{A}$.
- Total capacitive load on signal, including signal trace: $CT \leq 20 \text{ pF}$.

OBSERVATION 6.8:

The source and sink currents specified in RULES 6.19 and 6.20 include both driver and receiver currents sourced and sunk on the board.

SUGGESTION 6.4:

Use any standard device that meets the specifications above to drive the lines BG0OUT*-BG3OUT*/BG0IN*-BG3IN*, IACKOUT*/IACKIN*.

Use 74LS240, 74LS241, or 74LS244 devices to receive the lines BG0OUT*-BG3OUT*/BG0IN*-BG3IN*, and IACKOUT*/IACKIN*.

6.4.2.5 REGLES pour commander et charger les lignes à collecteur ouvert

(BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL*, IACK*)

REGLE 6.21:

SI une carte de BUS CEI 821 commande les lignes BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL* ou IACK*.

ALORS les émetteurs sur ces lignes DOIVENT satisfaire les spécifications suivantes:

- Courant absorbé au niveau bas: $IOL \geq 48 \text{ mA}$.
- Tension de sortie au niveau bas: $VOL \leq 0,6 \text{ V}$ à $IOL = 48 \text{ mA}$.

REGLE 6.22:

Toutes les cartes de BUS CEI 821 DOIVENT limiter leur charge sur les lignes BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL* et IACK* aux valeurs suivantes:

- Courant produit par la carte à 0,6 V, incluant le courant de fuite: $IOZL + IIL \leq 400 \mu\text{A}$ (DTACK* et BERR*)
 $IOZL + IIL \leq 600 \mu\text{A}$ (tous les autres signaux).
- Courant absorbé par la carte à 2,4 V, incluant le courant de fuite: $IOZH + IIH \leq 50 \mu\text{A}$.
- Charge capacitive totale sur le signal, incluant la piste de signal: $CT \leq 20 \text{ pF}$.

OBSERVATION 6.9:

Les courants absorbés spécifiés dans les REGLES 6.21 et 6.22 incluent les courants fournis et absorbés par les émetteurs et les récepteurs sur la carte.

SUGGESTION 6.5:

Utiliser les dispositifs 74S38 comme émetteur sur les lignes BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, ACFAIL* et IACK*.

Utiliser les dispositifs 74LS240, 74LS241 ou 74LS244 comme récepteur sur les lignes BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL* et IACK*.

SUGGESTION 6.6:

Puisque la plupart des émetteurs TTL ne fonctionnent pas d'une manière fiable lorsque la source d'alimentation +5 V est hors spécification, commander SYSRESET* dans le module CONTROLEUR D'ALIMENTATION avec un émetteur construit à partir d'un transistor discret de faible signal et grand gain.

6.5 Interconnexions des lignes de signaux du fond de panier

Le BUS CEI 821 est un système d'interface à haute performance. Sa conception tient compte des effets de ligne de transmission sur le fond de panier. Les temps d'établissement des adresses et des données spécifiés dans les chapitres 2 et 4 tiennent compte du fait que les émetteurs disponibles aujourd'hui ne commandent pas correctement les

6.4.2.5 Driving and loading RULES for open-collector lines

(BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL*, IACK*)

RULE 6.21:

IF an IEC 821 BUS board drives the lines BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL*, or IACK*.
THEN its drivers for these lines MUST meet the following specifications:

- Low state sink current: $IOL \geq 48 \text{ mA}$.
- Low state voltage: $VOL \leq 0.6 \text{ V}$ at $IOL = 48 \text{ mA}$.

RULE 6.22:

All IEC 821 BUS boards MUST limit their loading of each of the lines BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL*, and IACK* to the following values.

- Current sourced by board at 0.6 V, including leakage current: $IOZL + IIL \leq 400 \mu\text{A}$ (DTACK* and BERR*).
 $IOZL + IIL \leq 600 \mu\text{A}$ (all others).
- Current sunk by board at 2.4 V, including leakage current: $IOZH + IIH \leq 50 \mu\text{A}$.
- Total capacitive load on signal, including signal trace: $CT \leq 20 \text{ pF}$.

OBSERVATION 6.9:

The sink current specified in RULES 6.21 and 6.22 includes both driver and receiver currents sourced and sunk on the board.

SUGGESTION 6.5:

Use 74S38 devices to drive the lines BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, ACFAIL* and IACK*.

Use 74LS240, 74LS241, or 74LS244 devices to receive the lines BR0*-BR3*, BBSY*, IRQ1*-IRQ7*, DTACK*, BERR*, SYSFAIL*, SYSRESET*, ACFAIL* and IACK*.

SUGGESTION 6.6:

Since most TTL drivers do not work reliably when the +5 V power source is out of specification, on a POWER MONITOR module, drive SYSRESET* with a driver built from a discrete high-gain small signal transistor.

6.5 Backplane signal line interconnections

The IEC 821 BUS is a high performance interface system. Its design takes into account transmission line effects on the backplane. The address and data set-up times specified in Chapters 2 and 4 take into account the fact that most drivers available today do not reliably drive backplane signal lines from the low to the high level until there is a

lignes de signaux du fond de panier de leur niveau bas à leur niveau haut sans qu'une réflexion à l'extrémité du bus ne se produise. Bien que ces réflexions soient inévitables, elles ne doivent pas être excessives sinon des oscillations pourraient se produire. Les paragraphes suivants spécifient les caractéristiques du fond de panier pour obtenir les résultats désirés.

6.5.1 Réseaux d'adaptation d'impédance

REGLE 6.23:

Des adaptateurs DOIVENT être utilisés à chaque extrémité de toutes les lignes de signaux du BUS CEI 821, sauf pour les lignes de chaîne série.

OBSERVATION 6.10:

Les adaptateurs dans le BUS CEI 821 ont quatre objectifs:

- Réduire les réflexions des extrémités du fond de panier.
- Fournir une résistance de rappel au niveau haut pour les circuits de commande à collecteur ouvert.
- Restituer les lignes de signaux au niveau haut quand les dispositifs trois états sont hors service.
- Fournir un courant de maintien pour commuter le transistor de drainage du circuit de commande, aidant à commuter les lignes de transmission sur des transitions positives.

L'équivalent Thévenin de l'adaptateur d'impédance est donné sur la figure 6-2, page 376. Le diviseur de tension, également donné, fournit les valeurs de l'adaptateur.

OBSERVATION 6.11:

Si une tolérance maximale de $\pm 5\%$ est admise sur les valeurs de résistances et de tensions de source utilisées dans le réseau de résistances représenté à la figure 6-2 ce réseau satisfera aux tolérances permises par l'équivalent Thévenin.

OBSERVATION 6.12:

Le réseau de résistances indiqué par la figure 6-2 présente l'impédance équivalente Thévenin seulement si la source +5 V est découplée à la masse de façon satisfaisante par un condensateur de découplage.

RECOMMANDATION 6.3:

Prévoir un condensateur de découplage d'une valeur comprise entre $0,01 \mu\text{F}$ à $0,1 \mu\text{F}$ aussi près que possible de la broche V_{cc} de chacun des boîtiers de résistances d'adaptation d'impédance.

AUTORISATION 6.3:

Tout réseau de résistances et source de tension PEUVENT être utilisés pour fournir l'adaptateur aussi longtemps qu'ils répondent à l'équivalent Thévenin indiqué à la figure 6-2.

reflection from the end of the bus. Although these reflections serve a useful purpose, they cannot be excessive or ringing will result. The following paragraphs specify the backplane characteristics that achieve the desired result.

6.5.1 Termination networks

RULE 6.23:

Termination networks **MUST** be used on each end of all IEC 821 BUS signal lines except the daisy-chain lines.

OBSERVATION 6.10:

The terminations in the IEC 821 BUS serve four purposes:

- They reduce reflections from the ends of the backplane.
- They provide a high state pull-up for open-collector drivers.
- They restore the signal lines to the high level when three-state devices are disabled.
- They provide a standing current for the driver sink transistor to switch off, causing the signal line to rise more swiftly on positive transitions.

The Thevenin equivalent of the termination is shown in Figure 6-2, page 377. The voltage divider also shown provides this termination value.

OBSERVATION 6.11:

If a maximum tolerance of $\pm 5\%$ is maintained on the resistor values and source voltage used in the resistor network shown in Figure 6-2, the circuit shown will meet the tolerances shown for the Thevenin equivalent.

OBSERVATION 6.12:

The resistor network shown in Figure 6-2 presents its Thevenin equivalent impedance only when its +5 V source is adequately decoupled to ground by a bypass capacitor.

RECOMMENDATION 6.3:

Provide a bypass capacitor with a value in the range of $0.01 \mu\text{F}$ to $0.1 \mu\text{F}$ as close as possible to the V_{cc} pin of each resistor termination package.

PERMISSION 6.3:

Any resistor network and voltage source **MAY** be used to provide the termination, as long as they provide the Thevenin equivalent shown in Figure 6-2.

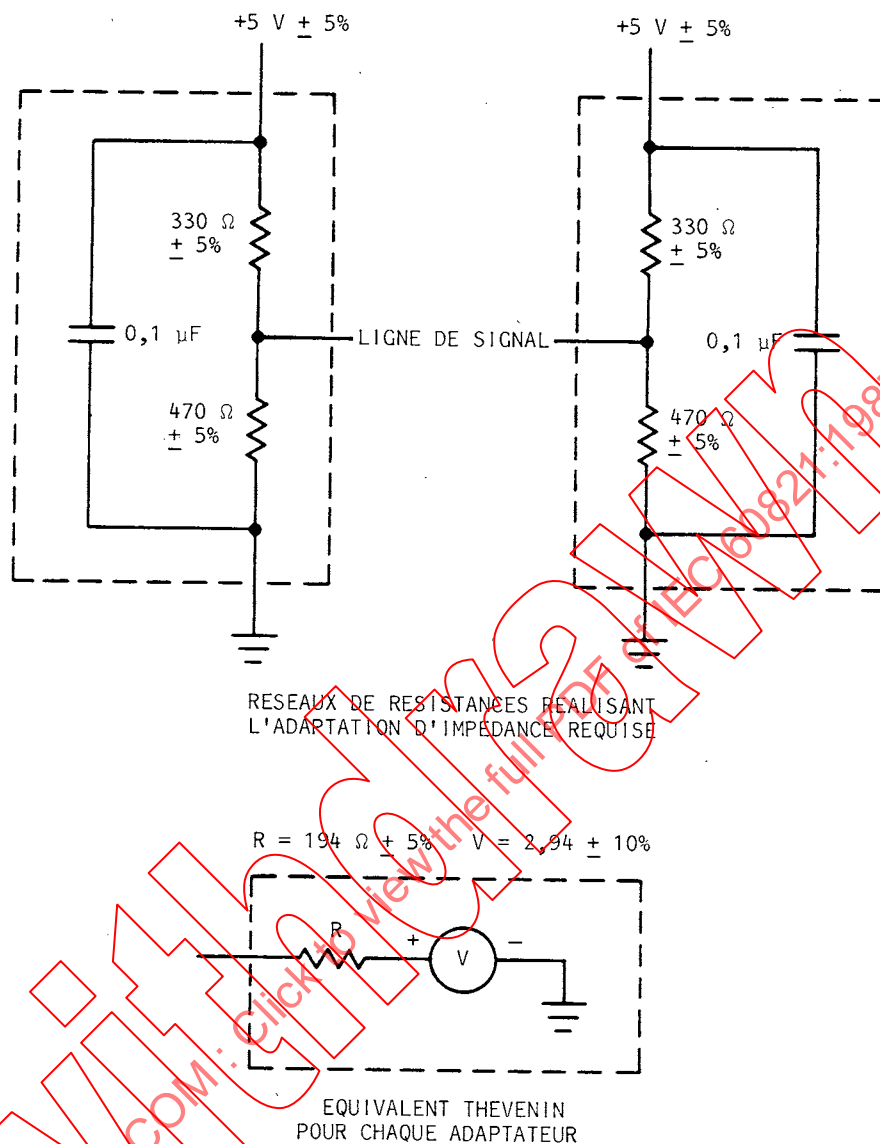
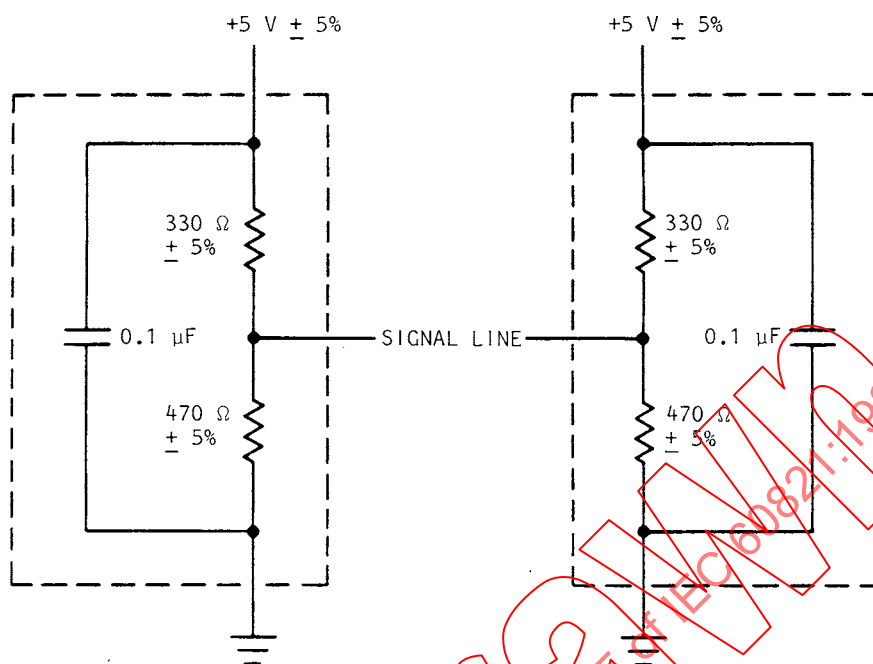


FIG. 6-2. - Adaptation d'impédance standard du bus.

6.5.2 Impédance caractéristique

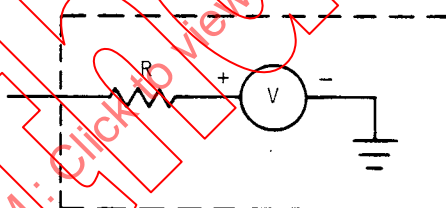
Chaque piste sur le fond de panier a une impédance caractéristique associée Z_0 . Cette impédance caractéristique est importante parce que des discontinuités de Z_0 (dues à l'effet capacitif et aux charges du bus) et des désadaptations entre Z_0 et les adaptations d'impédance peuvent causer des distorsions des signaux.

La figure 6-3, page 378, indique la section d'une piste de microruban normale pour une ligne de signal d'un fond de panier à multicouches. Z_0 est fonction de la largeur et de l'épaisseur de la ligne, de l'épaisseur du diélectrique et de sa constante diélectrique relative. La figure 6-4, page 380, indique l'impédance caractéristique en fonction de la largeur de la ligne de microruban pour une carte en fibre de verre époxy d'épaisseur courante.



RESISTOR NETWORKS THAT PROVIDE
THE REQUIRED TERMINATIONS

$$R = 194 \, \Omega \pm 5\% \quad V = 2.94 \pm 10\%$$



THEVENIN EQUIVALENT
FOR EACH TERMINATOR

FIG. 6-2. - Standard bus termination.

6.5.2 Characteristic impedance

Each signal line in the backplane has an associated characteristic impedance Z_0 . This characteristic impedance is important because discontinuities in Z_0 (due to capacitive effect and loads on the bus) and mismatches between Z_0 and the terminations can cause distortions of signal waveforms.

Figure 6.3, page 379, shows a microstrip signal line cross section which is the normal configuration for a multilayer backplane signal line. The Z_0 is a function of the width and thickness of the line, the thickness of the dielectric and its relative dielectric constant. Figure 6-4, page 381, shows characteristic impedance versus microstrip line width for common thickness of fiberglass-epoxy board.