ISO/IEC 24739-3

Edition 1.0 2010-03

INTERNATIONAL STANDARD

Information technology – AT attachment with packet interface-7 – Part 3: Serial transport protocols and physical interconnect (ATA/ATAPI-7 V3)

ISO/IEC 24739-3:2010(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

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

About IEC publications

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

- Catalogue of IEC publications: www.iec.ch/searchpub
The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv
If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

ISO/IEC 24739-3

Edition 1.0 2010-03

INTERNATIONAL STANDARD

Information technology – AT attachment with packet interface-7 – Part 3: Serial transport protocols and physical interconnect (ATA/ATAPI-7 V3)

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE XE

ICS 35.200

ISBN 2-8318-1083-6

CONTENTS

INFORMATION TECHNOLOGY – AT ATTACHMENT WITH PACKET INTERFACE-7 –

Part 3: Serial transport protocols and physical interconnect (ATA/ATAPI-7 V3)

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

10) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 24739-3 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

ISO/IEC 24739-3 is to be used in conjunction with ISO/IEC 24739-1 and ISO/IEC 24739-2.

The list of all currently available parts of the ISO/IEC 24739 series, under the general title *Information technology – AT attachment with packet interface-7,* can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

INTRODUCTION

The ISO/IEC 24739 series specifies the AT Attachment Interface between host systems and storage devices. It provides a common attachment interface for systems manufacturers, system integrators, software suppliers, and suppliers of intelligent storage devices.

ISO/IEC 24739-1 defines the register delivered commands used by devices implementing the standard. ISO/IEC 24739-2 defines the connectors and cables for physical interconnection between host and storage device, the electrical and logical characteristics of the interconnecting signals, and the protocols for the transporting of commands, data, and status over the interface for the parallel interface. ISO/IEC 24739-3 defines the connectors and cables for physical interconnection between host and storage device, the electrical and logical characteristics of the interconnecting signals, and the protocols for the transporting of commands, data, and status over the interface for the serial interface. Figure 1 shows the relationship of these documents. For devices implementing the PACKET command feature set, additional command layer standards are listed in Table 1 and described in Clause 2.

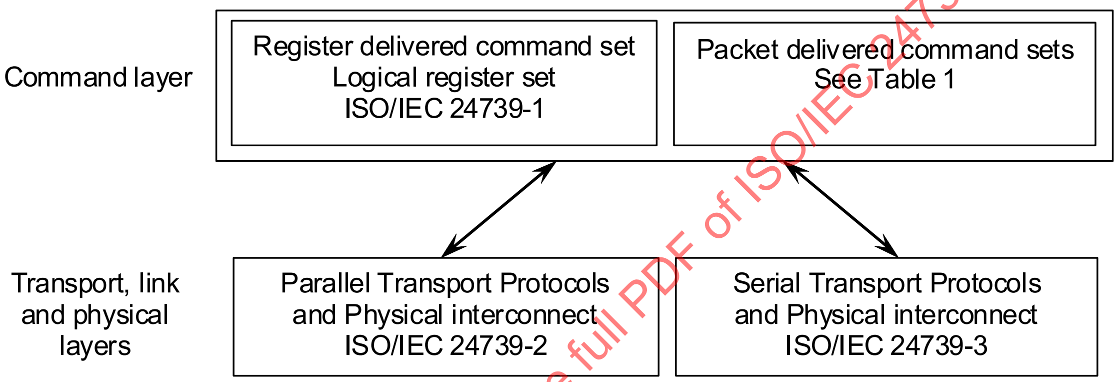


Figure 1 – ATA document relationships

Table 1 – PACKET delivered command sets

Standard
SCSI Primary Commands (SPC)
SCSI Primary Commands-2 (SPC-2)
SCSI Primary Commands-3 (SPC-3)
SCSI Block Commands (SBC-2)
SCSI Stream Commands (SSC)
Multimedia Commands (MMC)
Multimedia Commands-2 (MMC-2)
Multimedia Commands-3 (MMC-3)
Multimedia Commands-4 (MMC-4)
ATAPI for Removable Media
ATA Packet Interface (ATAPI) for Streaming Tape QIC-157 revision D

This standard maintains compatibility with the AT Attachment with Packet Interface - 6 standard (ATA/ATAPI-6) and while providing additional functions, is not intended to require changes to presently installed devices or existing software.

INFORMATION TECHNOLOGY – AT ATTACHMENT WITH PACKET INTERFACE-7 –

Part 3: Serial transport protocols and physical interconnect (ATA/ATAPI-7 V3)

1 Scope

This part of ISO/IEC 24739 specifies the connectors and cables for physical interconnection between host and storage device, the electrical and logical characteristics of the interconnecting signals, and the protocols for the transporting of commands, data, and status over the interface for the serial interface.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

The provisions of the referenced specifications other than ISO/IEC, IEC, ISO and ITU documents, as identified in this Clause, are valid within the context of this International Standard. The reference to a specification within this International Standard does not give it any further status within ISO/IEC. In particular, it does not give the referenced specification the status of an International Standard.

ISO/IEC 14776-331, *Information technology – Small computer system interface (SCSI) – Part 331: Stream Commands (SSC)*[1]

ISO 7779:1999, *Acoustics – Measurement of airborne noise emitted by information technology and telecommunications equipment*

ANSI INCITS 317-1998 (R2003), *AT Attachment with Packet Interface Extension (ATA/ ATAPI-4)*

3 Terms and definitions, abbreviations and conventions

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
ASCII character
8-bit value that is encoded using the ASCII character set

3.1.2
acoustics
measurement of airborne noise emitted by information technology and telecommunications equipment
[ISO 7779:1999(E)]

3.1.3
ATA
AT attachment
physical, electrical, transport and command protocols for the internal attachment of storage devices to host systems

1 See also ANSI INCITS 335-2000 (PACKET command feature set commands)

3.1.4
ATA/ATAPI-4 device
device that complies with ANSI INCITS 317-1998, AT Attachment Interface with Packet Interface Extensions-4

3.1.5
ATA/ATAPI-5 device
device that complies with ANSI INCITS 340-2000, the AT Attachment with Packet Interface-5

3.1.6
ATA/ATAPI-6 device
device that complies with ANSI INCITS 361-2002, the AT Attachment with Packet Interface-6

3.1.7
ATA/ATAPI-7 device
device that complies with the ISO/IEC 24739 series

3.1.8
ATAPI device
AT attachment packet interface device
device implementing the packet command feature set

3.1.9
allocation unit
AU
minimum number of logically contiguous sectors on the media as used in the streaming feature set

NOTE An Allocation Unit may be accessed with one or more requests.

3.1.10
audio-video
AV
audio-video applications use data that is related to video images and/or audio; the distinguishing characteristic of this type of data is that accuracy is of lower priority than timely transfer of the data

3.1.11
backchannel
receive channel when transmitting an FIS

3.1.12
bit error rate
BER
statistical probability of a transmitted encoded bit being erroneously received in a communication system

3.1.13
bus release
for devices implementing overlap, the term bus release is the act of clearing both DRQ and BSY to zero before the action requested by the command is completed; this allows the host to select the other device or deliver another queued command

3.1.14
byte count
value placed in the byte count register by the device to indicate the number of bytes to be transferred during this DRQ assertion when executing a PACKET PIO data transfer command

3.1.15
byte count limit
value placed in the byte count register by the host as input to a PACKET PIO data transfer command to specify the maximum byte count that may be transferred during a single DRQ assertion

3.1.16
CompactFlash™ Association
CFA
association which created the specification for compact flash memory that uses the ATA interface

3.1.17
check condition
for devices implementing the PACKET command feature set, this indicates an error or exception condition has occurred

3.1.18
cylinder-head-sector
CHS
obsolete method of addressing the data on the device by cylinder number, head number and sector number

3.1.19
code violation
in a serial interface implementation, a code violation is an error that occurs in the decoding of an encoded character (see Clause 15)

3.1.20
command aborted
command completion with ABRT set to one in the error register and ERR set to one in the status register

3.1.21
command acceptance
a command is considered accepted whenever the currently selected device has the BSY bit cleared to zero in the status register and the host writes to the command register

NOTE An exception exists for the DEVICE RESET command (see ISO/IEC 24739-1:2009, Clause 6). In a serial implementation, command acceptance is a positive acknowledgment of a host to device register FIS.

3.1.22
command block registers
interface registers used for delivering commands to the device or posting status from the device

NOTE In a serial implementation, the command block registers are FIS payload fields.

3.1.23
command completion
completion by the device of the action requested by the command or the termination of the command with an error, the placing of the appropriate error bits in the error register, the placing of the appropriate status bits in the status register, the clearing of both BSY and DRQ to zero and Interrupt Pending

3.1.24
command packet
data structure transmitted to the device during the execution of a PACKET command that includes the command and command parameters

3.1.25
command released
when a device supports overlap or queuing, a command is considered released when a bus release occurs before command completion

3.1.26
control block registers
in a parallel implementation, interface registers used for device control and to post alternate status; in a serial interface implementation, the logical field of a FIS corresponding to the device register bits of a parallel implementation

3.1.27
control character
in a serial interface implementation, an encoded character that represents a non-data byte (see Clause 15)

3.1.28
cyclical redundancy check
CRC
means used to check the validity of certain data transfers

3.1.29
cylinder high register
name used for the LBA high register in previous ATA/ATAPI standards

3.1.30
cylinder low register
name used for the LBA mid register in previous ATA/ATAPI standards

3.1.31
data character
in a serial interface implementation, an encoded character that represents a data byte (see Clause 15)

3.1.32
data-in
protocol that moves data from the device to the host

NOTE These transfers are initiated by READ commands.

3.1.33
data-out
protocol that moves data from the host to the device

NOTE These transfers are initiated by WRITE commands.

3.1.34
delayed LBA
any sector for which the performance specified by the streaming performance parameters log is not valid

3.1.35
device
storage peripheral

NOTE Traditionally, a device on the interface has been a hard disk drive, but any form of storage device may be placed on the interface provided the device adheres to this standard.

3.1.36
device selection
in a parallel implementation, a device is selected when the DEV bit of the device register is equal to the device number assigned to the device by means of a device 0/device 1 jumper or switch, or use of the CSEL signal

NOTE In a serial implementation the device ignores the DEV bit, the host adapter may use this bit to emulate device selection.

3.1.37
disparity
difference between the number of ones and the number of zeroes in an encoded character (see Clause 15)

3.1.38
DMA (direct memory access) data transfer
means of data transfer between device and host memory without host processor intervention

3.1.39
don't care
term to indicate that a value is irrelevant for the particular function described

3.1.40
driver
active circuit inside a device or host that sources or sinks current to assert or negate a signal on the bus

3.1.41
DRQ data block
unit of data words transferred during a single assertion of DRQ when using PIO data transfer

3.1.42
elasticity buffer
in a serial interface implementation, a portion of the receiver where character slipping and/or character alignment is performed

3.1.43
encoded character
in a serial interface implementation, the output of the 8b/10b encoder (see Clause 15)

3.1.44
first party DMA access
method by which a device accesses host memory

NOTE First party DMA differs from DMA in that the device sends a DMA setup FIS to select host memory regions; whereas for DMA the host configures the DMA controller.

3.1.45
frame information structure
FIS
data structure; it is the payload of a frame and does not include the SOF primitive, CRC and EOF primitive

3.1.46
frame
unit of information exchanged between the host adapter and a device

NOTE A frame consists of an SOF primitive, a Frame Information Structure, a CRC calculated over the contents of the FIS and an EOF primitive.

3.1.47
forced unit access
FUA
requires that user data be transferred to or from the device media before command completion even if caching is enabled

3.1.48
Gen1 DWORD time
the time it takes to transmit a 40 bit encoded value at 1,5 Gbit/s

3.1.49
host
computer system executing the software BIOS and/or operating system device driver controlling the device and the adapter hardware for the ATA interface to the device

3.1.50
host adapter
implementation of the host transport, link and physical layers

3.1.51
interrupt pending
in a parallel implementation, an internal state of a device;

in this state, the device asserts INTRQ if nIEN is cleared to zero and the device is selected (see ISO/IEC 24739-2:2010, Clause 8);
in a serial implementation, the Interrupt Pending state is an internal state of the host adapter; this state is entered by reception of a FIS with the I field set to one (see Clause 16)

3.1.52
logical block address (LBA)
addressing of data on the device by the linear mapping of sectors

3.1.53
linear feedback shift register (LFSR)
(see Clause 15)

3.1.54
link
the link layer manages the phy layer to achieve the delivery and reception of frames (see Clause 15)

3.1.55
logical sector
a uniquely addressable set of 256 words (512 bytes)

3.1.56
native max address
the highest address a device accepts in the factory default condition, that is, the highest address that is accepted by the SET MAX ADDRESS command

3.1.57
overlap
a protocol that allows devices that require extended command time to perform a bus release so that commands may be executed by the other device (if present) on the bus

3.1.58
packet delivered command
command that is delivered to the device using the PACKET command via a command packet that contains the command and the command parameters; see also register delivered command

3.1.59
PHY
physical layer electronics, see Clause 14

3.1.60
physical sector
group of contiguous logical sectors that are read from or written to the device media in a single operation

3.1.61
programmed input/output data transfer
PIO data transer
transfers performed by the host processor utilizing accesses to the data register

3.1.62
primitive
in a serial interface implementation, a single DWORD of information that consists of a control character in byte 0 followed by three additional data characters in byte 1 through 3

3.1.63
queued
command queuing allows the host to issue concurrent commands to the same device

NOTE Only commands included in the Overlapped feature set may be queued. In this standard, the queue contains all commands for which command acceptance has occurred, but command completion has not occurred.

3.1.64
read command
command that causes the device to transfer data from the device to the host (e.g., READ SECTOR(S), READ DMA, etc.)

3.1.65
register
register may be a physical hardware register or a logical field

3.1.66
register delivered command
command that is delivered to the device by placing the command and all of the parameters for the command in the device command block registers

NOTE See also packet delivered command.

3.1.67
register transfers
host reading and writing any device register except the data register; register transfers are 8 bits wide

3.1.68
released
in a parallel interface implementation, indicates that a signal is not being driven

NOTE For drivers capable of assuming a high-impedance state, this means that the driver is in the high-impedance state. For open-collector drivers, the driver is not asserted.

3.1.69
sector
uniquely addressable set of 256 words (512 bytes)

3.1.70
sector number register
LBA low register in previous ATA/ATAPI standards

3.1.71
shadow command block
in a serial interface implementation, a set of virtual fields in the host adapter that map the command block registers defined at the command layer to the fields within the FIS content

3.1.72
shadow control block
in a serial interface implementation, a set of virtual fields in the host adapter that map the control block registers defined at the command layer to the fields within the FIS content

3.1.73
signature
unique set of values placed in the command block registers by the device to allow the host to distinguish devices implementing the PACKET command feature set from those devices not implementing the PACKET command feature set

3.1.74
self-monitoring, analysis and reporting technology
SMART
for prediction of device degradation and/or faults

3.1.75
transport
transport layer manages the lower layers (link and phy) as well as constructing and parsing FISs, see Clause 16

3.1.76
ultra DMA burst
the period from an assertion of DMACK− to the subsequent negation of DMACK− when an ultra DMA transfer mode has been enabled by the host

3.1.77
unaligned write
write command that does not start at the first logical sector of a physical sector or does not end at the last logical sector of a physical sector

3.1.78
unit attention condition
state that a device implementing the PACKET command feature set maintains while the device has asynchronous status information to report to the host

3.1.79
unrecoverable error
when the device sets either the ERR bit or the DF bit to one in the status register at command completion

3.1.80
vendor specific (VS)
bits, bytes, fields and code values that are reserved for vendor specific purposes

NOTE 1 These bits, bytes, fields and code values are not described in this standard and they may vary among vendors. This term is also applied to levels of functionality whose definition is left to the vendor.

NOTE 2 Industry practice could result in conversion of a Vendor Specific bit, byte, field or code value into a defined standard value in a future standard.

3.1.81
write command
command that causes the device to transfer data from the host to the device (e.g., WRITE SECTOR(S), WRITE DMA, etc.)

3.1.82
world wide name (WWN)
64-bit worldwide unique name based upon a company's IEEE identifier, (see IDENTIFY DEVICE Words 108 to 111 in ISO/IEC 24739-1:2009, Clause 6.)

3.2 Abbreviations

ABRT	Abort
a/r	As Required
BSY	Busy
C/D	Command/Data
CFA	Compact Flash Association
DASP	Drive Active / Slave Present
DD	Device Data
DEV	Device
DF	Device fault
DF/SE	Device Fault / Stream Error
DIOW	Device Input Output Write
DMA	Direct Memory Access
DMACK	DMA aknowledge
DMARQ	DMA request
DRDY	Device ready
DRQ	Date Request
ECC	Error Correcting Code

ENB	Enable
ERR	Errror
ERR/CHK	Error check
FIFO	First in / First Out
HOB	High Order Bit
ID	Identification
INTRQ	Interupt Request
I/O	Input / Output
LBA	Logical Block Address
na	Not Aplicable
nIEN	Negative Interupt Enable
NM	No media
NOP	No Operation
obs	Obsolete
OVL	Overlap
PDIAG	Passed Diagnostics
PIO	programmed input/output
REL	Release
R/W	Read / Write
SERV	Service
SRST	Soft Reset

3.3 Conventions

3.3.1 General

Lowercase is used for words having the normal English meaning. Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in Clause 3 or in the text where they first appear.

The names of abbreviations, commands, fields and acronyms used as signal names are in all uppercase (e.g., IDENTIFY DEVICE). Fields containing only one bit are usually referred to as the "name" bit instead of the "name" field. (See 3.3.7 for the naming convention used for naming bits.)

Names of device registers begin with a capital letter (e.g., LBA Mid register).

The expression “word n” or “bit n” shall be interpreted as indicating the content of word n or bit n.

3.3.2 Precedence

If there is a conflict between text, figures and tables, the precedence shall be tables, figures, then text.

3.3.3 Lists

Ordered lists, those lists describing a sequence, are of the form:

a)
b)
c)

Unordered list are of the form:

1)
2)
3)

3.3.4 Keywords

Several keywords are used to differentiate between different levels of requirements and optionality.

expected
A keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented.

mandatory
A keyword indicating items to be implemented as defined by this standard.

may
Indicates flexibility of choice with no implied preference.

obsolete
Indicates that the designated bits, bytes, words, fields and code values that may have been defined in previous standards are not defined in this standard and shall not be reclaimed for other uses in future standards. However, some degree of functionality may be required for items designated as "obsolete" to provide for backward compatibility.

Obsolete commands should not be used by the host. Commands defined as obsolete may be commands aborted by devices conforming to this standard. However, if a device does not command abort an obsolete command, the minimum that is required by the device in response to the command is command completion.

optional
Describes features that are not required by this standard. However, if any optional feature defined by the standard is implemented, the feature shall be implemented in the way defined by the standard.

prohibited
indicates that an item shall not be implemented by an implementation

reserved
indicates reserved bits, bytes, words, fields, and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word, or field shall be cleared to zero, or in accordance with a future extension to this standard. The recipient shall not check reserved bits, bytes, words, or fields. Receipt of reserved code values in defined fields shall be treated as a command parameter error and reported by returning command aborted

retired
indicates that the designated bits, bytes, words, fields, and code values that had been defined in previous standards are not defined in this standard and may be reclaimed for other uses in future standards. If retired bits, bytes, words, fields, or code values are used before they are reclaimed, they shall have the meaning or functionality as described in previous standards

shall
indicates a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard

should
indicates flexibility of choice with a strongly preferred alternative. Equivalent to the phrase "it is recommended"

3.3.5 Numbering

Numbers that are not immediately followed by a lowercase "b" or "h" are decimal values. Numbers that are immediately followed by a lowercase "b" (e.g., 01b) are binary values. Numbers that are immediately followed by a lowercase "h" (e.g., 3Ah) are hexadecimal values.

3.3.6 Signal conventions

Signal names are shown in all uppercase letters.

All signals are either high active or low active signals. A dash character (-) at the end of a signal name indicates the signal is a low active signal. A low active signal is true when the signal is below V_{iL}, and is false when the signal is above V_{iH}. No dash at the end of a signal name indicates the signal is a high active signal. A high active signal is true when the signal is above V_{iH}, and is false when the signal is below V_{iL}.

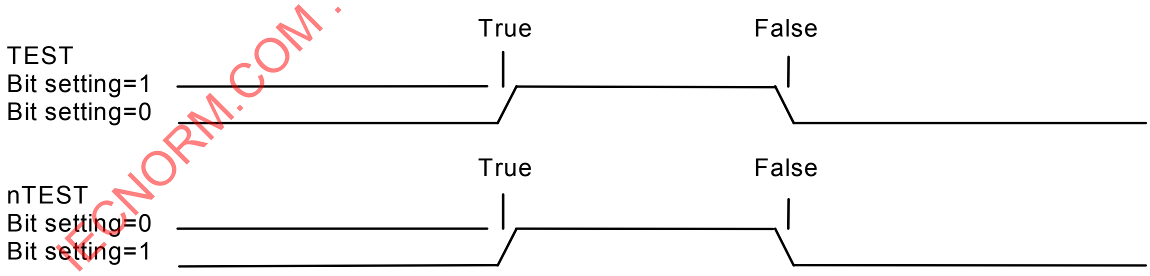
Asserted means that the signal is driven by an active circuit to the true state. Negated means that the signal is driven by an active circuit to the false state. Released means that the signal is not actively driven to any state (see ISO/IEC 24739-2:2010, Clause 7). Some signals have bias circuitry that pull the signal to either a true state or false state when no signal driver is actively asserting or negating the signal.

Control signals that may be used for more than one mutually exclusive functions are identified with their function names separated by a colon (e.g., DIOW-:STOP).

SIGNAL(n:m) denotes a set of signals, for example, DD(15:0).

3.3.7 Bit conventions

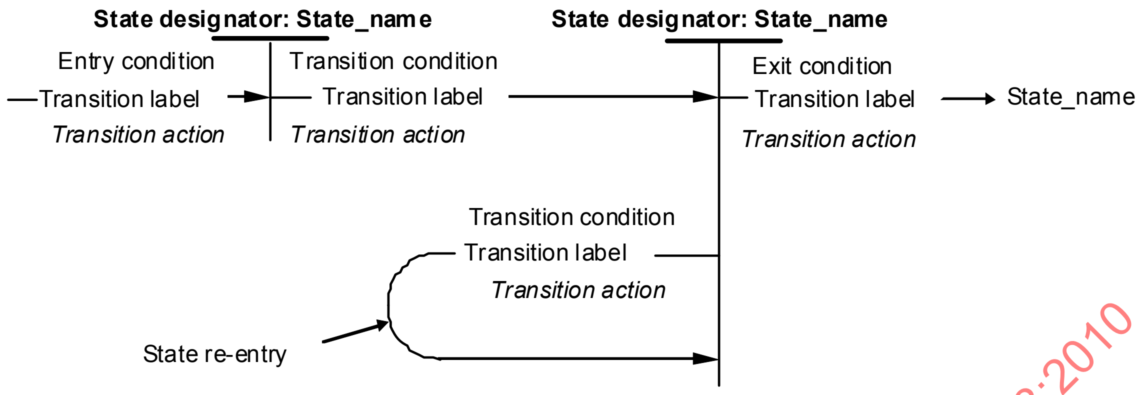
Bit names are shown in all uppercase letters except where a lowercase n precedes a bit name. If there is no preceding n, then when BIT is set to one the meaning of the bit is true, and when BIT is cleared to zero the meaning of the bit is false. If there is a preceding n, then when nBIT is cleared to zero the meaning of the bit is true and when nBIT is set to one the meaning of the bit is false.



Bit (n:m) denotes a set of bits, for example, bits (7:0).

3.3.8 State diagram conventions

State diagrams are as shown in Figure 2.



BSY	DRQ	REL	SERV	C/D	I/O	INTRQ	DMARQ	PDIAG-	DASP-
v	v	v	v	v	v	V	V	V	V

Figure 2 – State diagram convention

Each state is identified by a state designator and a state name. The state designator is unique among all states in all state diagrams in this document. The state designator consists of a set of letters that are capitalized in the title of the figure containing the state diagram followed by a unique number. The state name is a brief description of the primary action taken during the state, and the same state name may appear in other state diagrams. If the same primary function occurs in other states in the same state diagram, they are designated with a unique letter at the end of the name. Additional actions may be taken while in a state and these actions are described in the state description text.

In device command protocol state diagrams, the state of bits and signals that change state during the execution of this state diagram are shown under the state designator:state_name, and a table is included that shows the state of all bits and signals throughout the state diagram as follows:

v = bit value changes.
1 = bit set to one.
0 = bit cleared to zero.
x = bit is don't care.
V = signal changes.
A = signal is asserted.
N = signal is negated.
R = signal is released.
X = signal is don't care.

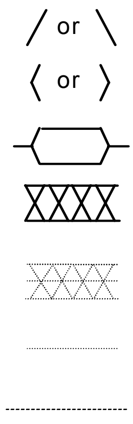
Each transition is identified by a transition label and a transition condition. The transition label consists of the state designator of the state from which the transition is being made followed by the state designator of the state to which the transition is being made. In some cases, the transition to enter or exit a state diagram may come from or go to a number of state diagrams, depending on the command being executed. In this case, the state designator is labeled xx. The transition condition is a brief description of the event or condition that causes the transition to occur and may include a transition action, indicated in italics, that is taken when the transition occurs. This action is described fully in the transition description text.

Upon entry to a state, all actions to be executed in that state are executed. If a state is re-entered from itself, all actions to be executed in the state are executed again.

Transitions from state to state shall be instantaneous.

3.3.9 Timing conventions

Certain symbols are used in the timing diagrams. These symbols and their respective definitions are listed below.

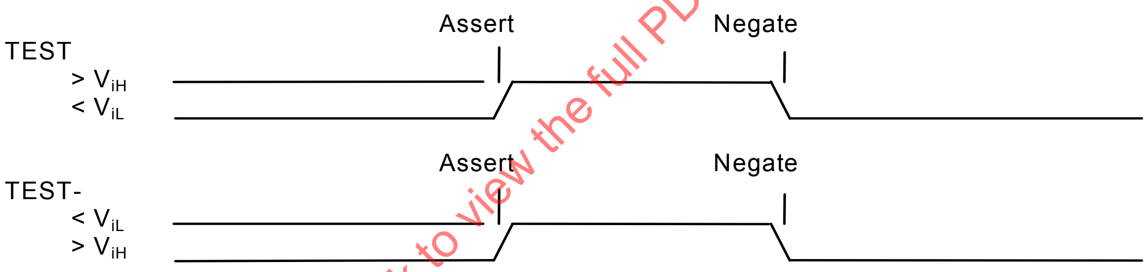


- signal transition (asserted or negated)
- data transition (asserted or negated)
- data valid
- undefined but not necessarily released
- asserted, negated or released
- released
- the “other” condition if a signal is shown with no change

All signals are shown with the asserted condition facing to the top of the page. The negated condition is shown towards the bottom of the page relative to the asserted condition.

The interface uses a mixture of negative and positive signals for control and data. The terms asserted and negated are used for consistency and are independent of electrical characteristics.

In all timing diagrams, the lower line indicates negated, and the upper line indicates asserted. The following illustrates the representation of a signal named TEST going from negated to asserted and back to negated, based on the polarity of the signal.



3.3.10 Byte ordering for data transfers

Data is transferred in blocks using either PIO or DMA protocols. PIO data transfers occur when the BSY bit is cleared to zero and the DRQ bit is set to one. These transfers are usually 16-bit but CFA devices may implement 8-bit PIO transfers. Data is transferred in blocks of one or more bytes known as a DRQ block. DMA data transfers occur when the host asserts DMACK- in response to the device asserting DMARQ. DMA transfers are always 16-bit. Each assertion of DMACK- by the host defines a DMA data burst. A DMA data burst is two or more bytes.

Assuming a DRQ block or a DMA burst of data contains "n" bytes of information, the bytes are labeled Byte(0) through Byte(n-1), where Byte(0) is first byte of the block, and Byte(n-1) is the last byte of the block. Table 2 shows the order the bytes shall be presented when such a block of data is transferred on the interface using 16-bit PIO and DMA transfers. Table 3 shows the order the bytes shall be presented in when such a block or burst of data is transferred on the interface using 8-bit PIO.

Table 2 – 16-bit Transfer Byte order

	DD 15	DD 14	DD 13	DD 12	DD 11	DD 10	DD 9	DD 8	DD 7	DD 6	DD 5	DD 4	DD 3	DD 2	DD 1	DD 0
First transfer	Byte (1)								Byte (0)							
Second transfer	Byte (3)								Byte (2)							
....																
Last transfer	Byte (*n*-1)								Byte (*n*-2)							

Table 3 – 8-bit Transfer Byte order

	DD 7	DD 6	DD 5	DD 4	DD 3	DD 2	DD 1	DD 0
First transfer	Byte (0)							
Second transfer	Byte (1)							
....								
Last transfer	Byte (*n*-1)							

NOTE The above description is for data on the interface. Host systems and/or host adapters may cause the order of data as seen in the memory of the host to be different.

Some parameters are defined as a string of ASCII characters. ASCII data fields shall contain only code values 20h through 7Eh. For the string “Copyright”, the character “C” is the first byte, the character “o” is the second byte, etc. When such fields are transferred, the order of transmission is:
the 1st character (“C”) is on DD(15:8) of the first word,
the 2nd character (“o”) is on DD(7:0) of the first word,
the 3rd character (“p”) is on DD(15:8) of the second word,
the 4th character (“y”) is on DD(7:0) of the second word,
the 5th character (“r”) is on DD(15:8) of the third word,
the 6th character (“i”) is on DD(7:0) of the third word,
the 7th character (“g”) is on DD(15:8) of the fourth word,
the 8th character (“h”) is on DD(7:0) of the fourth word,
the 9th character (“t”) is on DD(15:8) of the fifth word,
the 10th character (“space”) is on DD(7:0) of the fifth word,
etc.

Word (n:m) denotes a set of words, for example, words (103:100).

Byte, word and DWORD relationships.

Figure 3 illustrates the relationship between bytes, words and DWORDs.

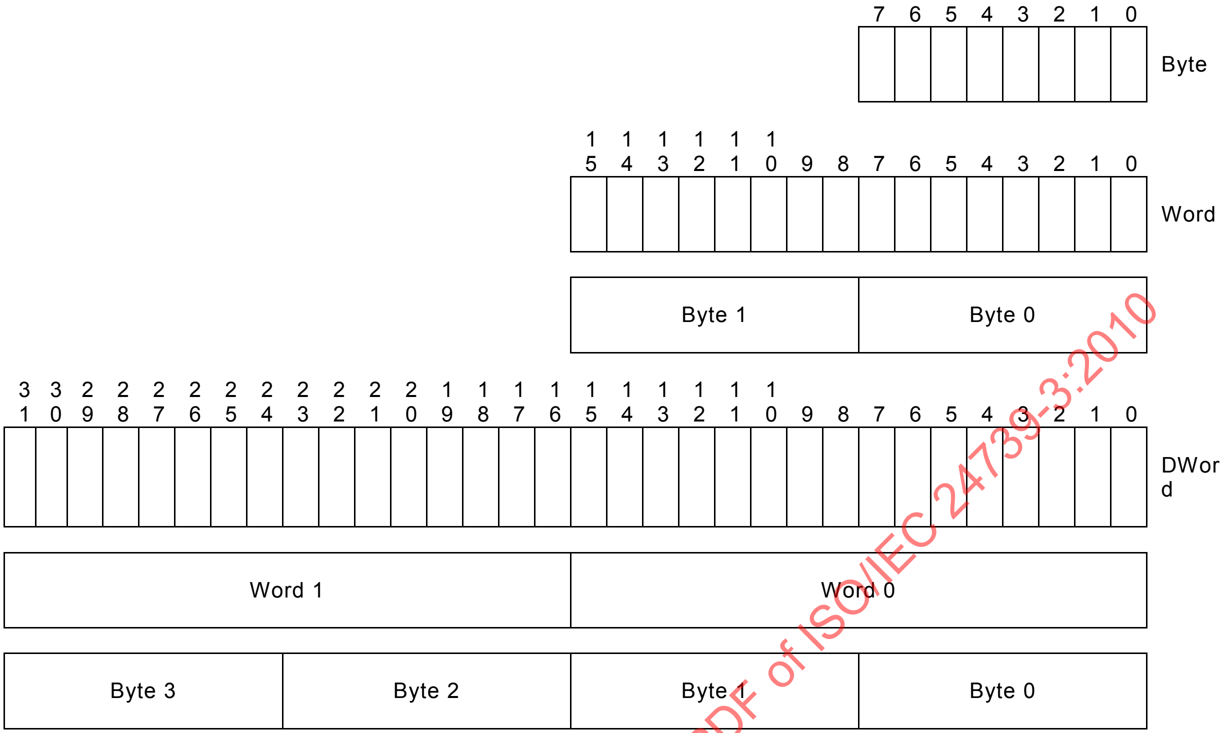


Figure 3 – Byte, word and DWORD relationships

4 General operational requirements

Refer to Clause 4 of ISO/IEC 24739-1:2009 for general operational requirements.

5 I/O register descriptions

Refer to Clause 5 of ISO/IEC 24739-1:2009 for I/O register. In addition, see also Clauses 10 and 11 of ISO/IEC 24739-2.

6 Command descriptions

Refer to Clause 6 of ISO/IEC 24739-1:2009 for command descriptions.

7 Parallel interface physical and electrical requirements

Refer to Clause 7 of ISO/IEC 24739-2:2009 for parallel interface physical and electrical requirements.

8 Parallel interface signal assignments and descriptions

Refer to Clause 8 of ISO/IEC 24739-2:2009 for parallel interface signal assignments and descriptions.

9 Parallel interface general operating requirements of the physical, data link, and transport layers

Refer to Clause 9 of ISO/IEC 24739-2:2009 for parallel interface general operating requirements of the physical, data link, and transport layers.

10 Parallel interface register addressing

Refer to Clause 10 and Clause 11 of ISO/IEC 24739-2 for parallel interface register addressing.

11 Parallel interface transport protocols

Refer to Clause 11 of ISO/IEC 24739-2:2010 for parallel interface transport protocols.

12 Parallel interface timing

Refer to Clause 12 of ISO/IEC 24739-2:2010 for parellel interface timing.

13 Serial interface general overview

13.1 Overview

The serial implementation of ATA is a high-speed serial replacement for the parallel implementation of ATA attachment of mass storage devices. The serial interface employed is a high-speed differential layer that utilizes Gigabit technology and 8b/10b encoding.

Figure 4 illustrates how two devices are connected to a Parallel ATA host adapter. This method allows up to two devices to be connected to a single port using a Device 0/Device 1 communication technique. Each device is connected via a ribbon cable that "daisy chains" the devices.

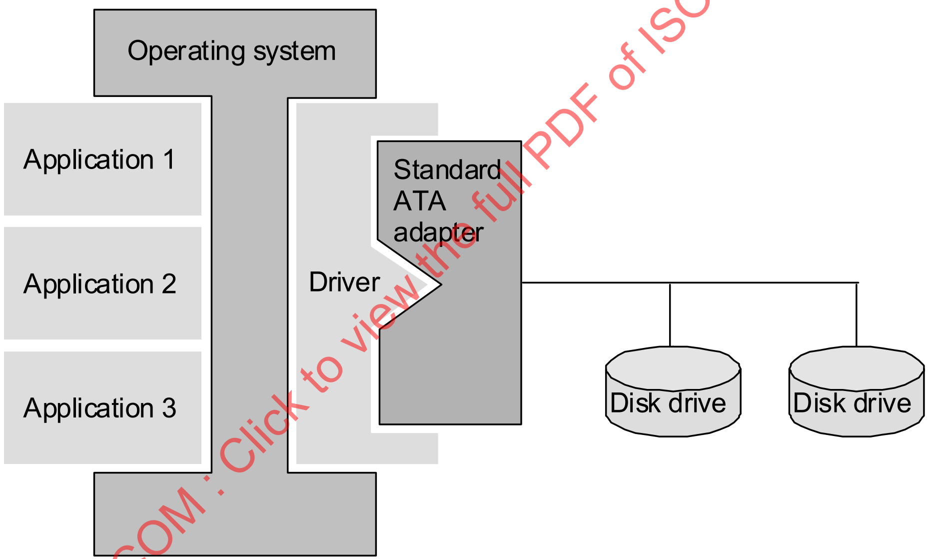


Figure 4 – Standard ATA device connectivity

Figure 5 illustrates how the same two devices are connected using a serial implementation of an ATA host adapter. In this diagram the dark grey portion is functionally similar to the dark grey portion of the previous diagram. ATA host software accesses the serial implementation of ATA subsystem in the same manner and functions in the same way as previous parallel implementation definitions. In this case, however, the software views the two devices as if they were both Device 0 on two separate ports. The right hand portion of the host adapter is of a new design that converts the operations of the software into a serial data/control stream. The serial interface structure connects each of the two drives with their own respective cables in a point-to-point fashion.

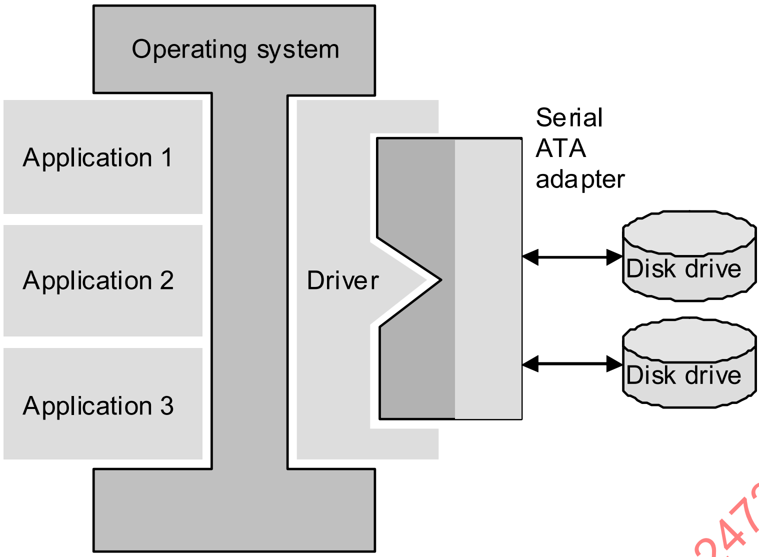


Figure 5 – The serial implementation of ATA connectivity

13.2 Sub-module operation

The Transport control state machine and the Link state machine are the two core sub-modules that control overall operation. The Link state machine controls the operation(s) related to the serial line and the Transport control state machine controls the operation(s) relating to the host platform. The two state machines coordinate their actions and utilize resources to transfer data between a host computer and attached mass storage device. The host Link state machine communicates via the serial line to a corresponding Link state machine located in the device. The host Transport machine also likewise communicates with a corresponding device Transport state machine. The two Link state machines ensure that control sequences between the two Transport control state machines are properly exchanged. Figure 6 shows how the machines communicate their various needed parameters in the traditional layered model. Each layer communicates with its counterpart directly or indirectly.

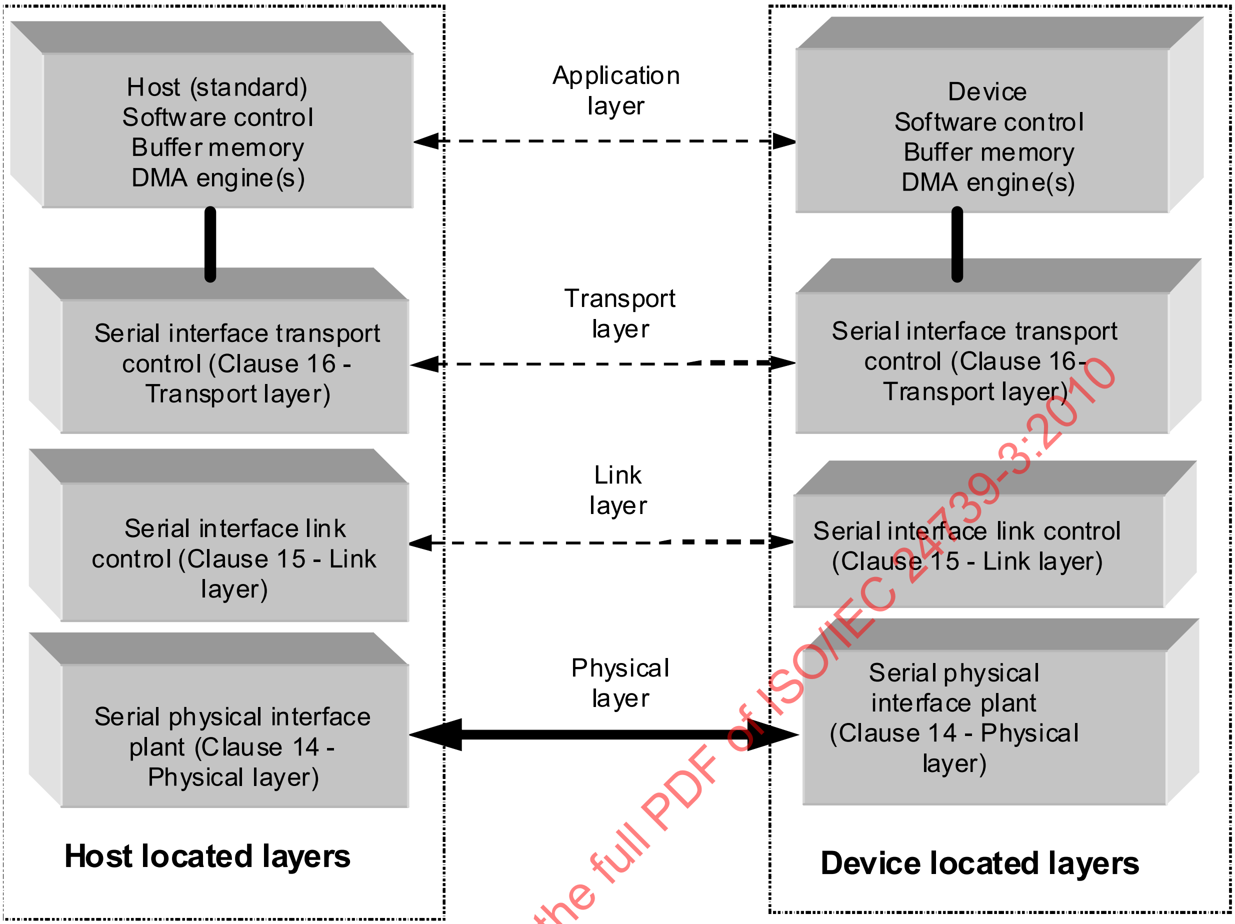


Figure 6 – Communication layers

The host interacts with the Transport control state machine through a register interface that is equivalent to that presented by a parallel implementation of an ATA host adapter. This allows host software to follow existing standards and conventions when accessing the register interface and follows standard command protocol conventions. The Transport control state machine breaks down these operations into a sequence of actions that are exchanged with the Link state machine.

13.3 Parallel ATA emulation

This subclause is optional.

13.3.1 General

See Clause 18 for state diagrams.

See 18.2 for additional information about nIEN.

Emulation of parallel implementations of ATA device behavior as perceived by the host BIOS or software driver, is a cooperative effort between the device and a serial interface host adapter hardware. The behavior of Command and Control Block registers, PIO and DMA data transfers, resets and interrupts are all emulated.

The host adapter contains a set of registers that Shadow the contents of the traditional device registers, referred to as the Shadow Command Block and Shadow Control Block. The Command Block registers are used for sending commands to the device or posting status from the device. These registers include the LBA High, LBA Mid, Device, Sector Count, Command, Status, Features, Error and Data registers. The Control Block registers are used for device control and to post alternate status. These registers include the Device Control and Alternate Status registers.

All serial implementations of ATA devices behave like Device 0 devices. Devices shall ignore the DEV bit in the Device field of received Register FISes, and it is the responsibility of the host adapter to gate transmission of Register FISes to devices, as appropriate, based on the value of the DEV bit.

After a reset or power-on, the host bus adapter emulates the behavior of a traditional ATA system during device discovery. Immediately after reset, the host adapter shall place the value 0x7Fh in its Shadow Status register and Shadow Alternate Status Register and shall place the value 0xFFh in all the other Shadow Command Block registers (0xFFFFh in the Data register). In this state the host bus adapter shall not accept writes to the Shadow Command Block and Shadow Control Block. When the host Phy detects presence of an attached device, see 20.2.2.1 and 19.2.2, the host bus adapter shall set bit 7 in the Shadow Status register yielding the value 0xFFh or 0x80h, and the host bus adapter shall allow writes to the Shadow Command Block and Shadow Control Block. If a device is present, the Phy shall take no longer than 10 ms to indicate that it has detected the presence of a device and has set bit 7 in the Shadow Status register. The Serial implementation of ATA time limit of 10 ms is different than the parallel implementation of ATA. (See 11 of ISO/IEC 24739-2). When the attached device establishes communication with the host bus adapter, it shall send a register FIS to the host, resulting in the Shadow Command Block and Shadow Control Block being updated with values appropriate for the attached device.

The host adapter may present a Device 0-only emulation to host software, that is, each device is a Device 0, and each Device 0 is accessed at a different set of host bus addresses. The host adapter may optionally present a Device 0/Device 1 emulation to host software, that is, two devices on two separate serial ports are represented to host software as a Device 0 and a Device 1 accessed at the same set of host bus addresses.

13.3.2 Software reset

Issuing a software reset is performed by toggling the SRST bit in the Shadow Device Control register. The toggle period is no shorter than a minimum timeout (see Clause 11 of ISO/IEC 24739-2). As a result of the SRST bit changing in the Shadow Device Control register, host adapters shall issue at least two Register FISes to the device (one with the SRST bit set and a subsequent one with the SRST bit cleared). See 16.2.3.2 for a detailed definition of Register FIS. Although host software is required to toggle the SRST bit no faster than specified in parallel implementations of ATA, serial devices shall not rely on the inter-arrival time of received Register FISes also meeting this timing. Because of flow control, frame handshaking and other protocol interlocks, serial devices may receive the resulting Register FISes back-to-back.

Due to flow control, protocol interlocks, power management state, or other transmission latencies, the subsequent Register frame transmission clearing the SRST bit during a software reset may be triggered prior to the previous Register FIS transmission having been completed. Host adapters are required to allow host software to toggle the SRST with the minimum timing specified for the parallel implementation of ATA, even if frame transmission latencies result in the first Register FIS transmission taking longer than specified in the parallel implementation. Host adapters are required to ensure transmission of the two resulting Register FISes to the device regardless of the transmission latency of each individual FIS.

13.3.3 Device 0-only emulation

A serial implementation of an ATA host adapter behaves the same as if a parallel implementation Device 0 only device were attached with no Device 1 present. It is the responsibility of the host adapter to properly interact with host software and present the correct behavior for this type of configuration. All serial implementations of ATA devices, therefore, need not be aware of Device 0/Device 1 issues and ignore parallel implementation I/O register information that deals with a secondary device. When the DEV bit in the Device register is set to one, selecting the non-existent Device 1, the host adapter shall respond to register reads and writes as specified for a Device 0 with no Device 1 present, as defined in the parallel implementation. This includes not setting the BSY bit in the Shadow Status register when Device 1 is selected, as described in the parallel implementation. When Device 0 is selected, the host adapter shall execute the serial implementation protocols for managing the Shadow Command Block and Shadow Control Block contents as defined in Clause 17.

When Device 0 is selected and the Command register is written in the Shadow Command Block, the host adapter sets the BSY bit in its Shadow Status register. The host adapter then transmits a frame to the device containing the new Control and Command Block register contents. When the Device Control register is written in the Shadow Control Block with a change of state of the SRST bit, the host adapter sets the BSY bit in its Shadow Status register and transmits a frame to the device containing the new register contents. Transmission of register contents when the Device Control register is written with any value that is not a change of state of the SRST bit shall not set the BSY bit in the Shadow Status register, and transmission of a frame to the device containing new register contents is optional. Similarly, the host adapter sets the BSY bit in its Shadow Status register to one when a hard reset (COMRESET) is requested or the SRST bit is set to one in the Device Control register, see Figure 95.

The device updates the contents of the host adapter Shadow Command Block and Shadow Control Block by transmitting a register frame to the host. This allows the device to set the proper ending status in the host adapter at the completion of a command or control request. Specific support is added to ensure proper timing of the DRQ and BSY bits in the Status register for PIO transfers.

Finally the host adapter provides an Interrupt Pending flag in the host adapter. This flag is set by the host adapter when the device sends a serial bus frame including the request to set the Interrupt Pending flag. The host adapter asserts the interrupt to the host processor any time the Interrupt Pending flag is set, the DEV bit is cleared to zero in the Shadow Device register, and the nIEN bit in the Shadow Device Control register is cleared to zero. The host adapter clears the Interrupt Pending flag any time a COMRESET is requested, the SRST bit in the Shadow Device Control register is set to one, the Shadow Command register is written and DEV is cleared to zero, or the Shadow Status register is read and DEV is cleared to zero. This allows the emulation of the host interrupt and its proper timing.

13.3.4 Device 0/Device 1 emulation (optional)

13.3.4.1 General

All devices behave as if they are Device 0 devices. However, the host adapter may optionally implement emulation of the behavior of a Device 0/Device 1 configuration by pairing two serial ports, and managing their associated Shadow Command Block and Shadow Control Block accordingly, as though they were a Device 0 and Device 1 at the same set of host bus addresses.

A host adapter that emulates Device 0/Device 1 behavior shall manage the two sets of Shadow Command Block and Shadow Control Block (one set for each of the two devices) based on the value of the DEV bit in the Shadow Device register. Based on the value of the DEV bit, the host adapter shall direct accesses to the Shadow Command Block and Shadow Control Block to the appropriate set of Shadow Command Block and Shadow Control Block in the correct device. It is the responsibility of the host adapter to ensure that communication with one or both of the attached devices is handled properly, and that information gets routed to the devices correctly. Each device shall process any communication with the host adapter as if it is targeted for the device regardless of the value of the DEV bit.

If a host adapter is emulating Device 0/Device 1 behavior, and there is no device attached to the cable designated as the Device 1 cable, the host adapter shall emulate Device 0 behavior with no Device 1 present as described in the parallel implementation.

When device 1 is selected and device 0 is responding for device 1 (see Table 44 in ISO/IEC 24739-2:2009), a host adapter with Device 0/Device 1 emulation generates the non-packet device response for both packet and non-packet devices.

13.3.4.2 Software reset

Host adapters that emulate Device 0/Device 1 behavior shall emulate parallel implementation behavior for software reset. Based on the Phy initialization status, the host adapter knows whether a device is attached to each of the two ports used in a Device 0/Device 1 emulation configuration. Device Control Register writes, that have the SRST bit set to one, shall result in the associated Shadow Control Block being written for each port to which a device is attached. The frame transmission protocol for each associated port executed, results in a Register FIS being transmitted to each attached device. Similarly, the subsequent write to the Device Control register that clears the SRST bit shall result in a Register FIS being sent to each attached device. The host adapter shall then await a response from each attached device (or timeout), and shall merge the contents of the Error and Status registers for the attached devices, in accordance with the parallel implementation, to produce the Error and Status register values visible to host software.

13.3.4.3 EXECUTE DEVICE DIAGNOSTICS

Host adapters that emulate Device 0/Device 1 behavior shall emulate the parallel implementation behavior for EXECUTE DEVICE DIAGNOSTICS (see ISO/IEC 24739-1:2009, Clause 6 and ISO/IEC 24739-2:2010, Clause 11). The host adapter shall detect the EXECUTE DEVICE DIAGNOSTIC command being written to the Command register. Host adapter detection of the EXECUTE DEVICE DIAGNOSTICS command shall result in the associated Shadow Command Block and Shadow Control Block being written for each port to which a device is attached. A Register FIS is transmitted to each attached device. The host adapter shall then await a response from each attached device (or timeout), and shall merge the contents of the Error and Status registers for the attached devices, in accordance with the parallel implementation to produce the Error and Status register values visible to host software.

13.3.4.4 Restrictions and limitations

Superset capabilities that are unique to the Serial implementation of ATA and not supported by the parallel implementation of ATA are not required to be supported in Device 0/Device 1 emulation. Device 0/Device 1 emulation is recommended only in configurations where legacy software drivers are used and the number of attached devices exceeds the number of interfaces the legacy software supports.

14 Serial interface physical layer

14.1 Overview

14.1.1 General

This clause describes the physical layer of the serial implementation of ATA. Unless otherwise described, the information is normative. The information that is provided and marked informative is provided to help the reader better understand the normative clauses and should be taken as examples only. Exact implementations may vary.

14.1.2 List of services

The following services are provided.

1) Transmit a 1.5 Gb/s differential NRZ serial stream at specified voltage levels.
2) Provide matched termination at the transmitter.
3) Serialize a 10, 20, 40, or other width parallel input from the Link for transmission.
4) Receive a 1.5 Gb/s differential NRZ serial stream.
5) Provide a matched termination at the receiver.
6) Extract data (and, optionally, clock) from the serial stream.
7) Deserialize the serial stream.
8) Detect the K28.5 comma character and provide a bit and word aligned 10, 20, 40, or other width parallel output.
9) Provide specified OOB signal detection and transmission.
10) Perform proper power-on sequencing and speed negotiation.
11) Provide interface status to Link layer.

12) Host/device present.
13) Host/device absent.
14) Host/device present but failed to negotiate communications.
15) Optionally support power management modes.
16) Optionally perform transmitter and receiver impedance calibration.
17) Handle the input data rate frequency variation due to a spread spectrum transmitter clock.
18) Accommodate request to go into Far-End retimed loopback test mode of operation when commanded.

14.2 Connectors specifications

14.2.1 Overview

This clause covers the serial implementation of ATA connectors and cable assemblies. It defines the

1) connector mating interfaces,
2) connector location on the device,
3) electrical, mechanical and reliability requirements of the connectors and cable assemblies,
4) connector and cable testing procedures.

It does not define how the connector and cable assembly are implemented, such as

1) the mounting feature of the connectors,
2) the cabling and cable terminations,
3) the methods on how the PCB connects to other components of the system.

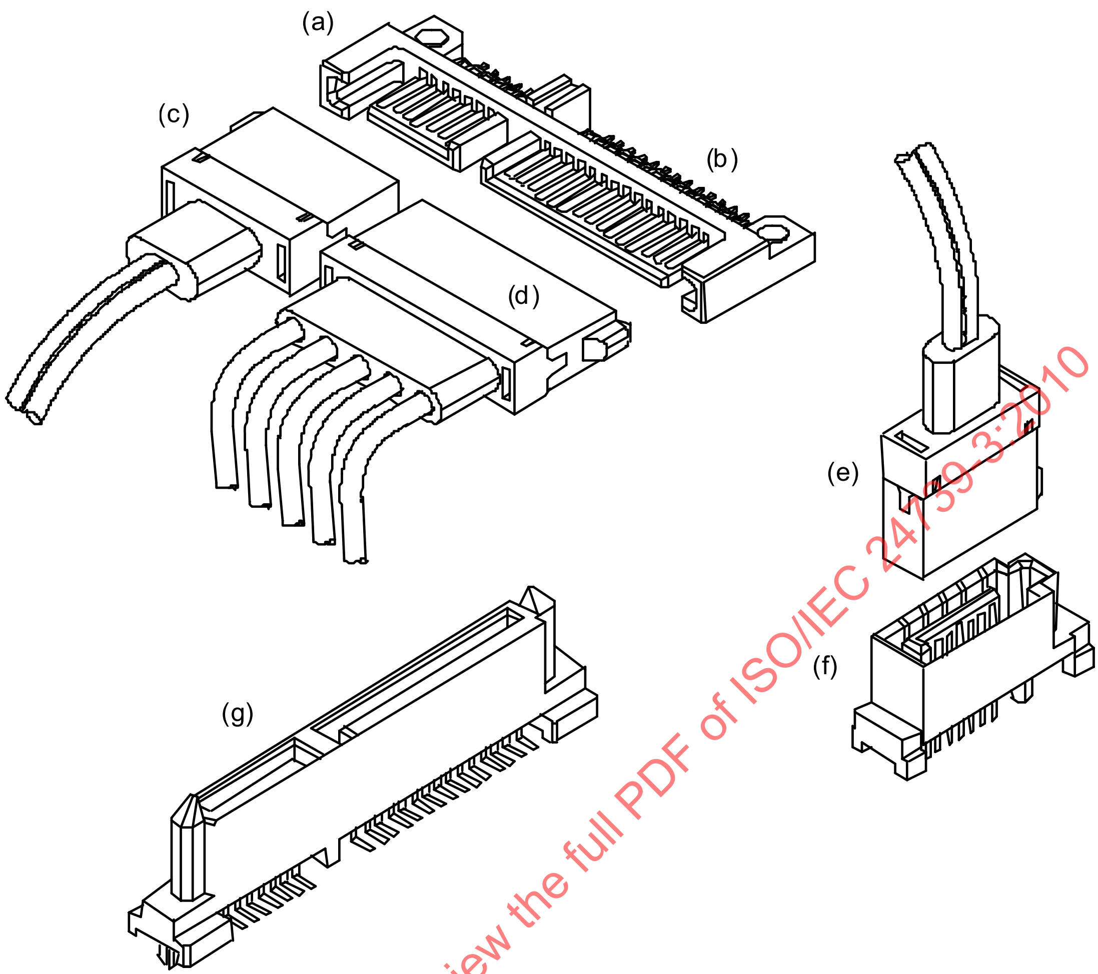
14.2.2 General descriptions

A serial implementation of an ATA device may be either directly connected to a host or connected to a host through a cable.

For direct connection, the device plug connector, shown as (a) and (b) in Figure 7, is inserted directly into a host receptacle connector, illustrated as (g) in Figure 7. The device plug connector and the host receptacle connector incorporate features that enable the direct connection to be hot pluggable and blind mateable.

For connection via cable, the device signal plug connector, shown as (a) in Figure 7, mates with the signal cable receptacle connector on one end of the cable, illustrated as (c) in Figure 7. The signal cable receptacle connector on the other end of the cable is inserted into a host signal plug connector, shown as (f) in Figure 7. The signal cable wire consists of two twinax sections in a common outer sheath.

There is also a separate power cable for the cabled connection. A Serial ATA power cable includes a power cable receptacle connector, shown as (d) in Figure 7, on one end and may be directly connected to the host power supply on the other end or may include a power cable receptacle on the other end. The power cable receptacle connector on one end of the power cable mates with the device power plug connector, shown as (b) in Figure 7. The host end of the power cable is not covered in this standard.



Components

(a) device signal plug segment or connector;

(b) device power plug segment or connector;

(c) signal cable receptacle connector, to be mated with (a);

(d) power cable receptacle connector, to be mated with (b);

(e) signal cable receptacle connector, to be mated with (f), the host signal plug connector;

(g) host receptacle connector mating directly with device plug connector (a) and (b).

Figure 7 – Serial implementation connector examples

14.2.3 Connector drawings

14.2.3.1 Device plug connector

Figure 8, Figure 9 and Table 4 show the interface dimensions for the device plug connector with both signal and power segments. All dimensions are in milimeters.

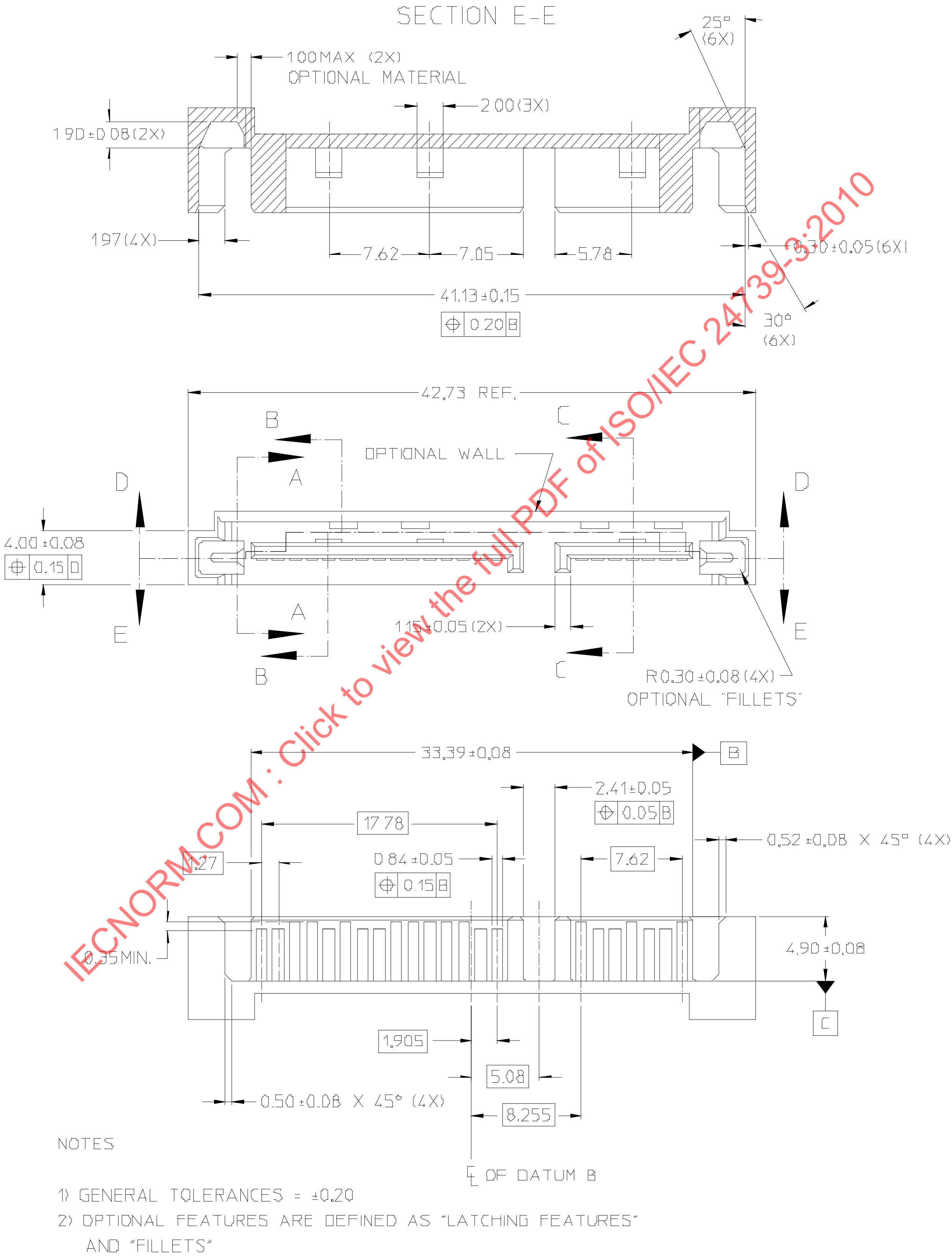
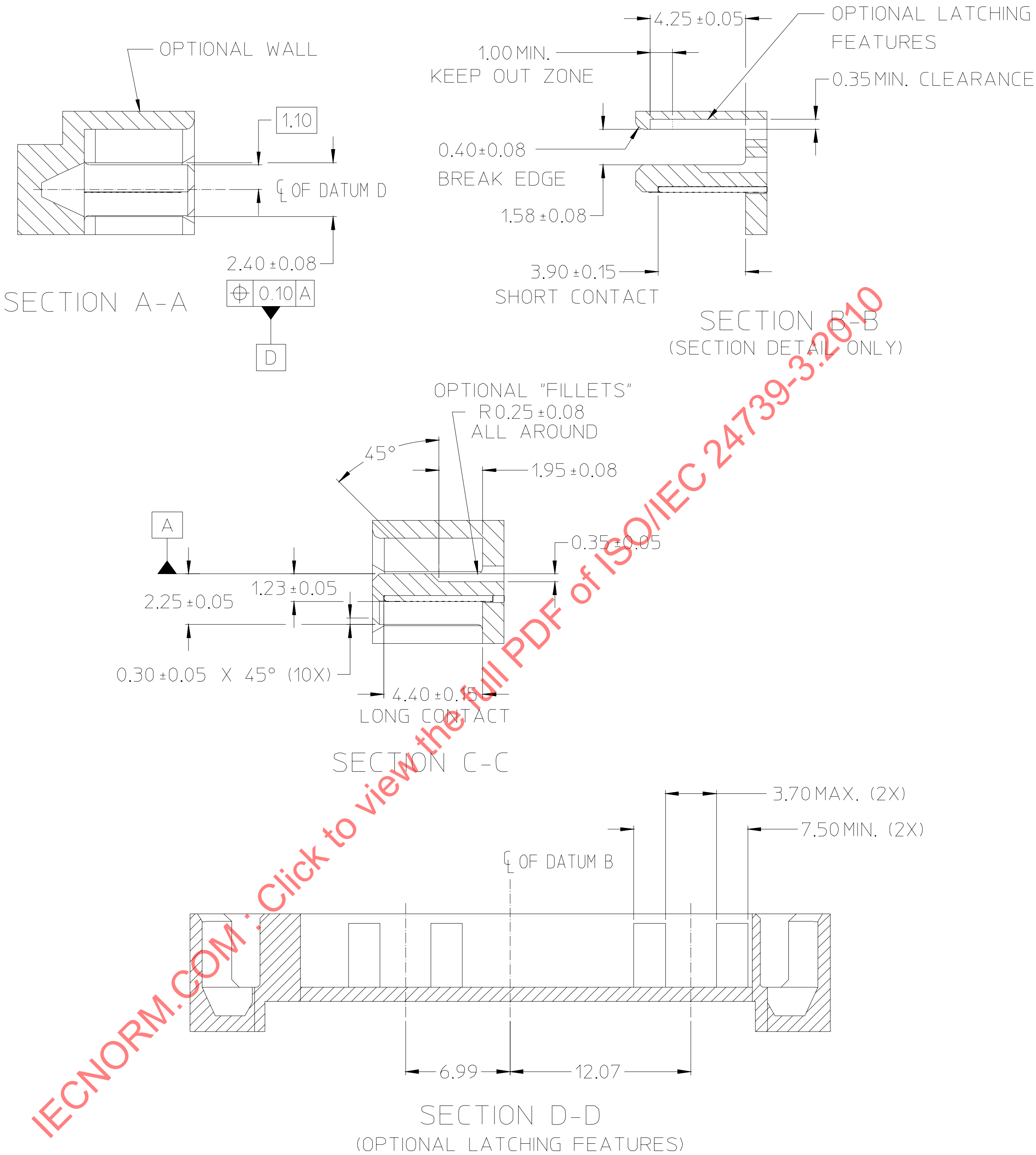


Figure 8 – Device plug connector part 1 of 2



NOTES:

1) GENERAL TOLERANCES = ±0.20
2) OPTIONAL FEATURES ARE DEFINED AS "LATCHING FEATURES" AND "FILLETS"

Figure 9 – Device plug connector part 2 of 2

14.2.3.2 Signal cable receptacle connector

Figure 10 shows the interface dimensions for the signal cable receptacle connector. There are two identical receptacles at the two ends of the Serial ATA cable assembly. The cable receptacle mates with either the signal segment of the device plug connector on the device, or the host plug connector on the host. All dimensions are in milimeters.

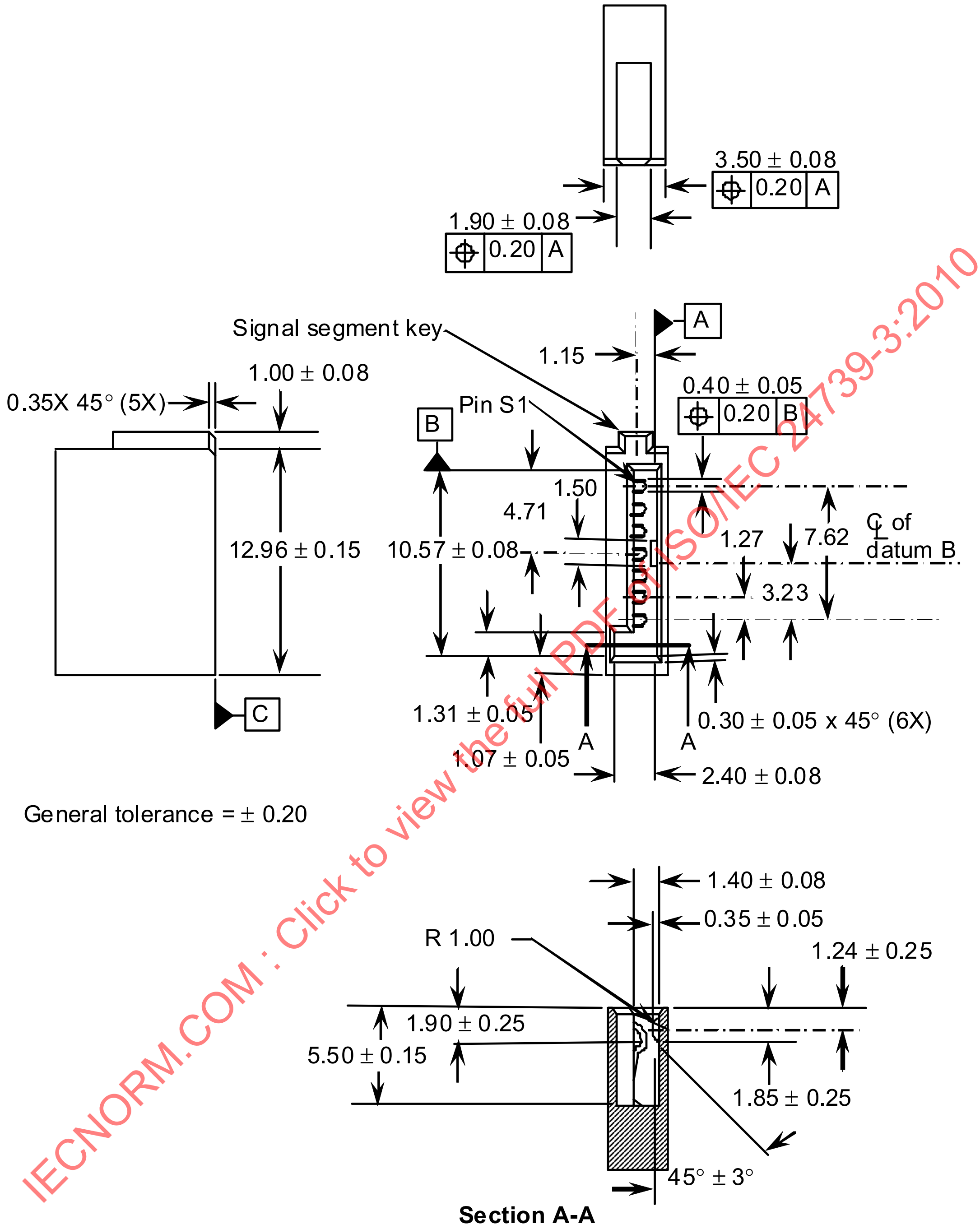


Figure 10 – Non-Latching Signal Cable receptacle connector interface dimensions

Figure 11 shows the additional dimensions for the optional Latching Signal Cable Receptacle. All dimensions are in milimeters.

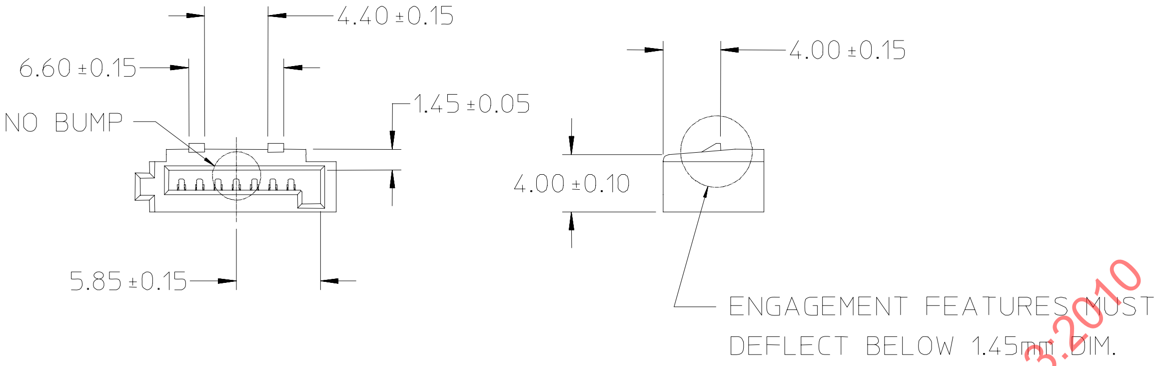


Figure 11 – Optional Latching Signal Cable Receptacle connector interface dimensions

14.2.3.3 Signal host plug connector

The signal host plug connector is to be mated with one end of the Serial ATA cable assembly. So the pinout of the host plug connector is the mirror image of the signal cable receptacle. Figure 12 shows the host plug connector interface definition.

For applications where multiple Serial ATA ports or connectors are stacked together on the host, there is a clearance or spacing requirement to prevent the cable assemblies from interfering with each other. Figure 21 shows the recommended clearance or spacing. All dimensions are in milimeters.

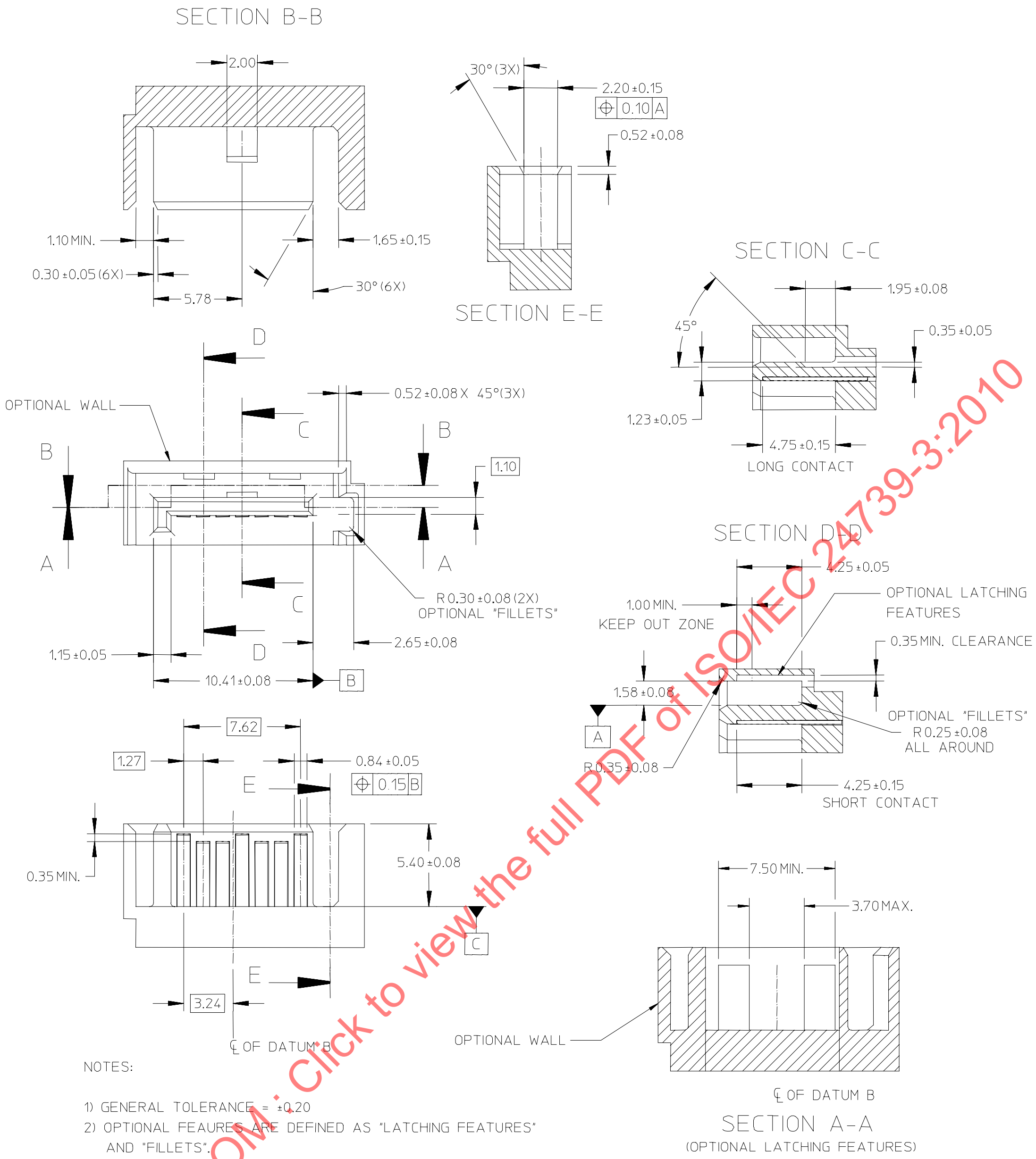


Figure 12 – Host plug connector interface dimension

14.2.3.4 Host receptacle connector

The host receptacle connector is to be blind-mated directly with the device plug connector. The interface dimensions for the host receptacle connector are shown in Figure 13. Note that dimension B allows two values: 8.15 mm and 14.15 mm. There are two levels of contacts in the host receptacle connector. The advancing ground contacts P4 and P12 mate first with the corresponding ground pins on the device plug connector, followed by the engaging of the pre-charged power pins. An appropriate external retention mechanism independent of the connector is required to keep the host PCB and the device in place and is not specified by this standard. The host receptacle connector is not designed with any retention mechanism. All dimensions are in milimeters.

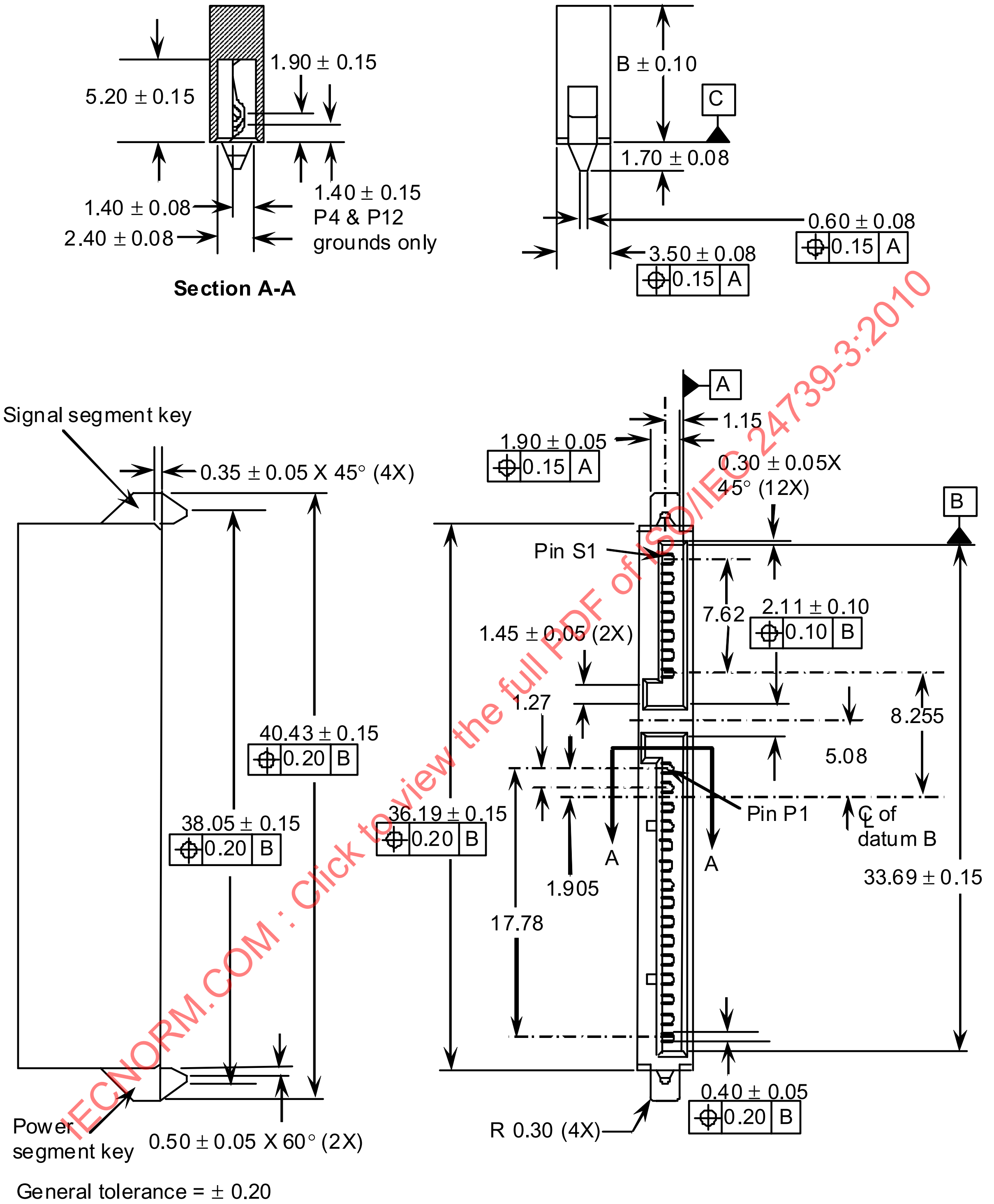


Figure 13 – Host receptacle connector interface dimensions

14.2.3.5 Power cable receptacle connector

The power cable receptacle connector mates with the power segment of the device plug, bringing power to the device. Figure 14 shows the interface dimensions of the power receptacle connector. The pinout of the connector is the mirror image of the power segment of the device plug shown in Table 4. All dimensions are in milimeters.

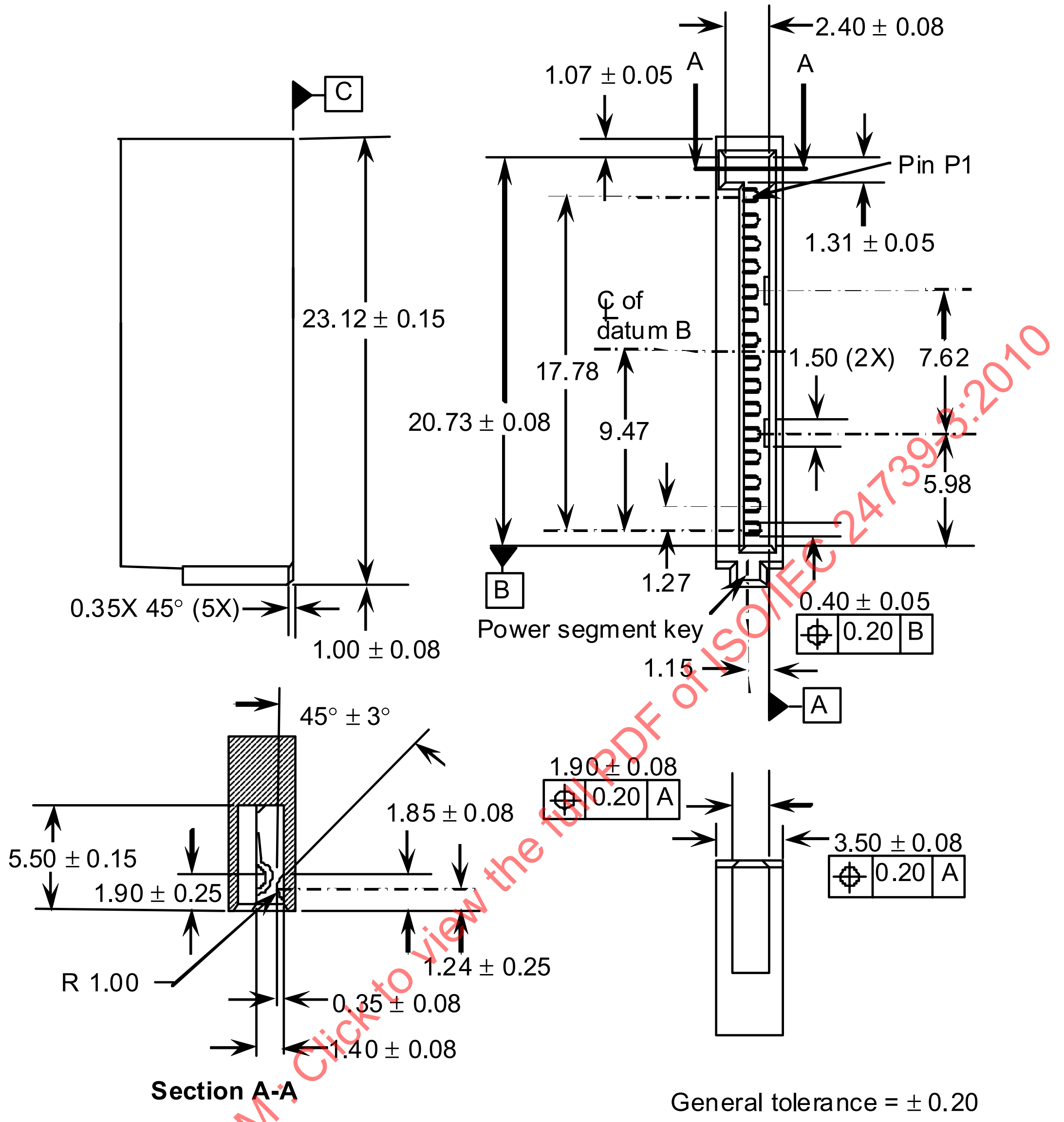


Figure 14 – Non-Latching Power cable receptacle connector interface dimensions

Figure 15 shows the dimensions for the optional latching feature for the power cable receptacle. All dimensions are in milimeters.

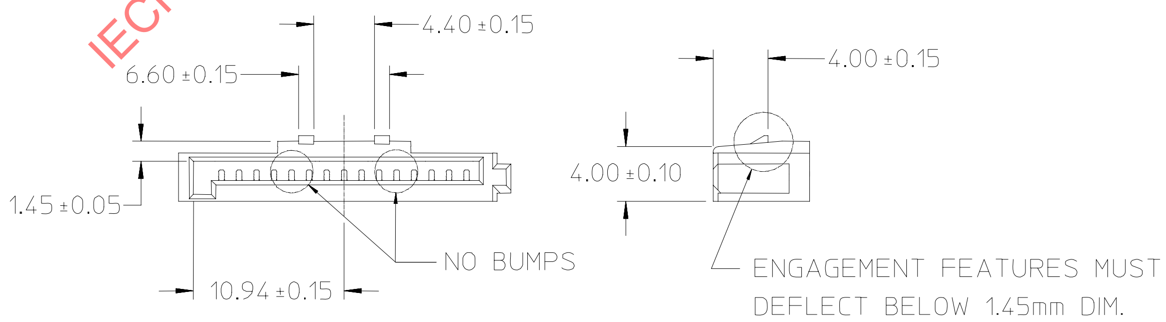


Figure 15 – Optional Latching Power Cable Receptacle

14.2.4 Connector pinouts

There are a total of 7 pins in the signal segment and 15 pins in the power segment. The pin definitions are shown in Table 4.

Table 4 – Device plug connector pin definition

Signal segment key			
Signal segment	S1	Gnd	Second mate
	S2	A+	Differential signal pair A from Phy
	S3	A-	
	S4	Gnd	Second mate
	S5	B-	Differential signal pair B from Phy
	S6	B+	
	S7	Gnd	Second mate
Signal segment "L"			
Central connector polarizer			
Power segment "L"			
Power segment	P1	V33	3.3 V power
	P2	V33	3.3 V power
	P3	V33	3.3 V power, pre-charge, second mate
	P4	Gnd	First mate
	P5	Gnd	Second mate
	P6	Gnd	Second mate
	P7	V5	5 V power, pre-charge, second mate
	P8	V5	5 V power
	P9	V5	5 V power
	P10	Gnd	Second mate
	P11	Reserved	1) The pin corresponding to P11 in the backplane receptacle connector is also reserved 2) The corresponding pin to be mated with P11 in the power cable receptacle connector shall be grounded
	P12	Gnd	First mate
	P13	V12	12 V power, pre-charge, 2nd mate
	P14	V12	12 V power
	P15	V12	12 V power
Power segment key			

NOTE 1 The comments on the mating sequence in Table 3 apply to the case of backplane blind-mate connector only. In this case, the mating sequences are: (1) the ground pins P4 and P12; (2) the pre-charge power pins and the other ground pins; and (3) the signal pins and the rest of the power pins.

NOTE 2 There are three power pins for each voltage. One pin from each voltage is used for pre-charge in the backplane blind-mate situation.

NOTE 3 V33 pins shall be connected together in the device.

NOTE 4 V5 pins shall be connected together in the device.

NOTE 5 V12 pins shall be connected together in the device.

14.2.5 Backplane connector configuration and blind-mating tolerance

The maximum blind-mate misalignment tolerances are ±1.50 mm and ±1.00 mm, respectively, for two perpendicular axes illustrated in Figure 16. Any skew angle of the plug, with respect to the receptacle, reduces the blind-mate tolerances. All dimensions are in milimeters.

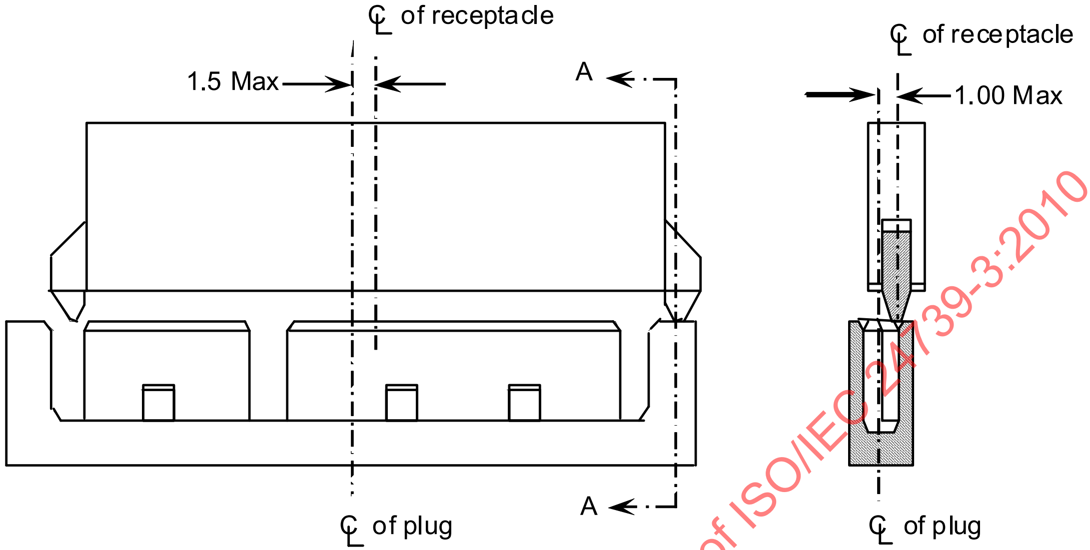


Figure 16 – Connector pair blind-mate misalignment tolerance

The device-to-backplane mating configuration is shown in Figure 17. Note that two values (8.45 and 14.45 mm) are allowed for dimension A. All dimensions are in milimeters.

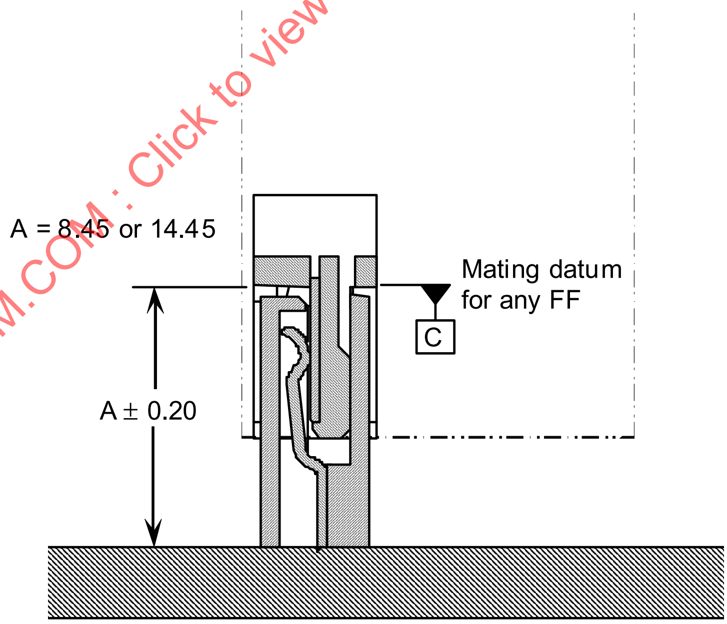


Figure 17 – Device-backplane mating configuration

14.2.6 Connector locations

The device connector location is defined to facilitate blind mating. Figure 18 and Figure 19 define the connector locations on 95 mm (3.5”) and 65 mm (2.5”) devices, respectively. All dimensions are in milimeters.

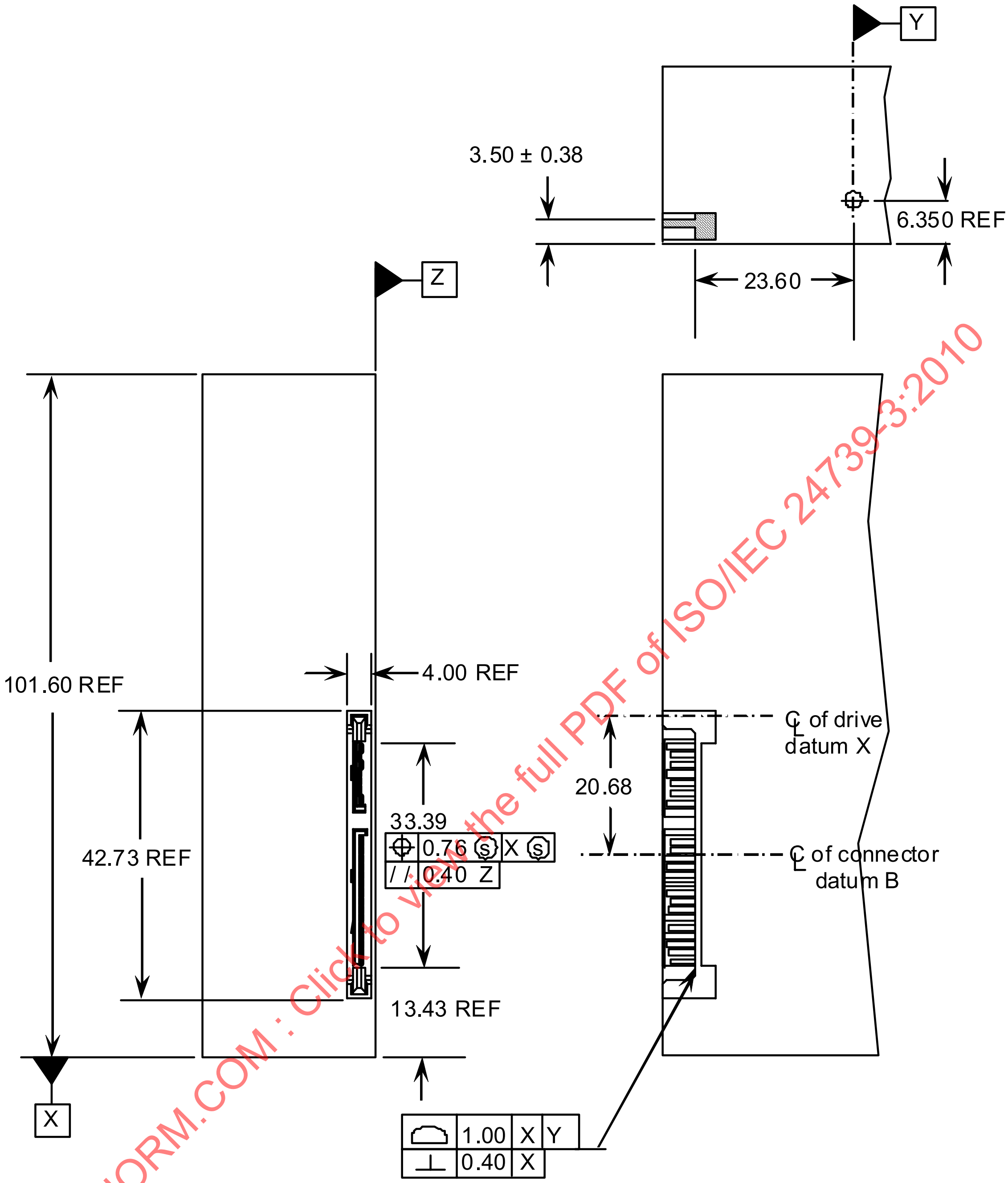


Figure 18 – Device plug connector location on 95 mm (3.5") device

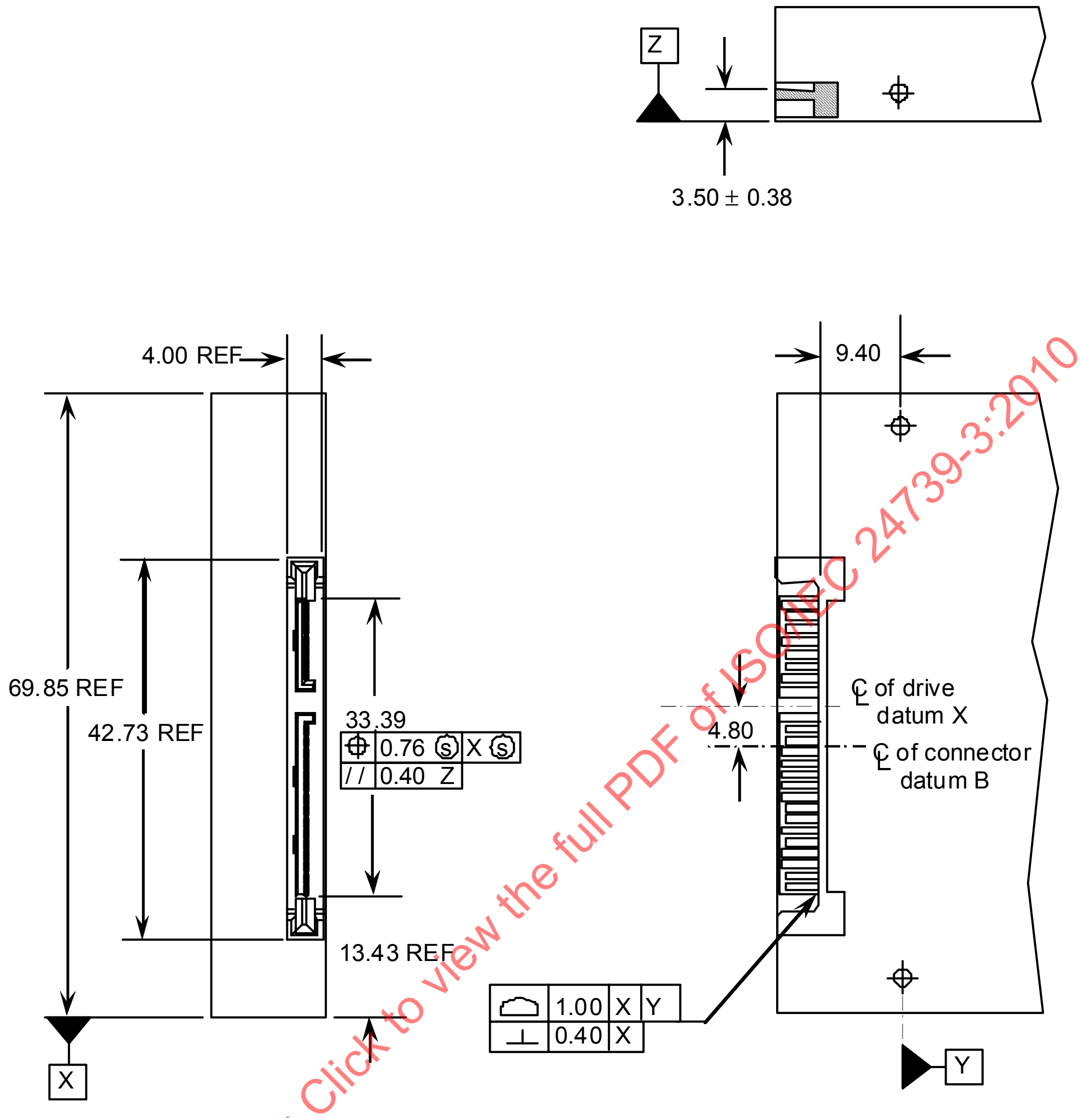


Figure 19 – Device plug connector location on 65 mm (2.5") device

Recommended host plug connector clearance is shown in Figure 20. All dimensions are in milimeters.

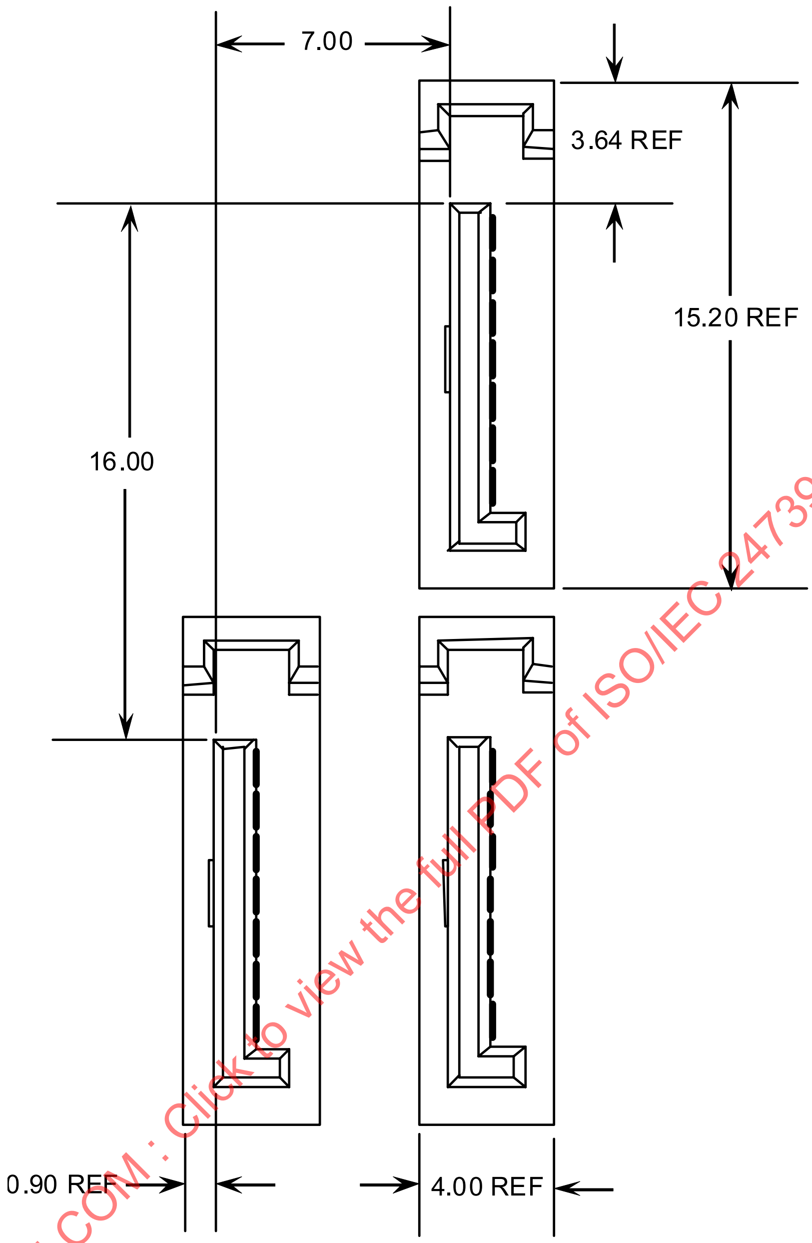


Figure 20 – Recommended host plug spacing for Non-Latching Connectors

Figure 21 shows the recommended host plug connector clearance and orientation for optional latching connectors. All dimensions are in milimeters.

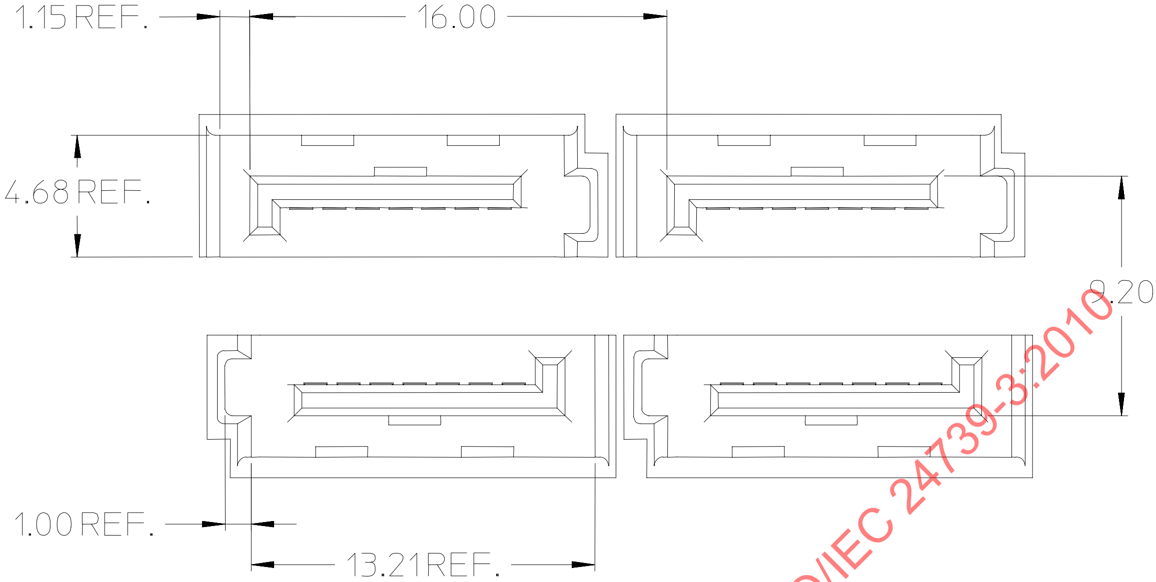


Figure 21 – Recommended host plug connector clearance and Orientation for Optional Latching Connectors

14.2.7 Connector conformance requirements

14.2.7.1 General

Unless otherwise specified, all measurements shall be performed within the following lab conditions:

- mated;
- temperature: 15°C to 35°C;
- relative humidity: 20 % to 80 %;
- atmospheric pressure: 650 mm to 800 mm of Hg.

If an EIA (Electronic Industry Association) test is specified without a letter suffix in the test procedures, the latest approved version of that test shall be used.

14.2.7.2 Signal

The test board shall consist of differential traces (100 Ω ± 5 Ω) over a ground plane (single-ended 50 Ω ± 2.5 Ω).

Open or shorted traces with the same length as the input signal traces shall be provided to measure the system input risetime and to synchronize pulses. Traces for crosstalk measurements diverge from each other. Provisions for attenuation reference measurement shall also be provided.

Unless otherwise specified, the requirements in Table 5 are for the entire signal path from the host mated pair connector to the device plug mated pair connector, but not including PCB traces.

A cable assembly shall meet Table 5 electrical signalling parameters and requirements when tested with the above specified test fixture or equivalent.

Table 5 – Signal integrity requirements and test procedures

Parameter	Procedure	Requirements
Mated connector impedance	1) Minimize skew (see Note 1). 2) Set the Time Domain Reflectometer (TDR) pulsers in differential mode with a positive going pulse (*V*+) and a negative going pulse (*V*-). Define a reflected differential trace: $V_{diff} = V+ - V-$. 3) With the TDR connected to the risetime reference trace, verify an input risetime of 70 ps (measured 20 % to 80 % V_p). Filtering may be used to slow the system down (see Note 2). 4) Connect the TDR to the sample measurement traces. Calibrate the instrument and system (see Note 3). 5) Measure and record the maximum and minimum values of the near end connector impedance.	100 Ω ± 15 %
Cable absolute impedance	1) Minimize skew (see Note 1). 2) Set the Time Domain Reflectometer (TDR) pulsers in differential mode with a positive going pulse (*V*+) and a negative going pulse (*V*-). Define a reflected differential trace: $V_{diff} = V+ - V-$. 3) With the TDR connected to the risetime reference trace, verify an input risetime of 70 ps (measured 20 % to 80 % V_p). Filtering may be used to slow the system down (see Note 2). 4) Connect the TDR to the sample measurement traces. Calibrate the instrument (see Note 3). 5) Measure and record maximum and minimum cable impedance values in the first 500 ps of cable response following any vestige of the connector response.	100 Ω ± 10 %
Cable pair matching	1) Set the Time Domain Reflectometer (TDR) to differential mode. 2) With the TDR connected to the risetime reference traces verify an input risetime of 70 ps (measured 20 % to 80 % V_p). Filtering may be used to slow the system down (see Note 2). 3) Connect the TDR to the sample measurement traces. Calibrate the instrument and system (see Note 3). 4) Measure and record the singleended cable impedance of each cable within a pair Measure and record maximum and minimum cable impedance values in the first 500 ps of cable response following any vestige of the connector response. 5) The parameter then equals $Line1_{imp} - Line2_{imp}$.	± 5 Ω
Common mode impedance	1) Set two TDR pulsers to produce a differential signal. 2) Minimize skew (see Note 1). 3) With the TDR connected to the risetime reference trace verify an input risetime of 70 ps (measured 20 % to 80 % V_p). Filtering may be used to slow the system down (see Note 2). 4) Calibrate the TDR (see Note 3). 5) Set both TDR pulsers to produce positive going pulses. 6) Measure the even mode impedance of the first pulser. Divide this by 2 to get the common mode impedance. 7) Do the same for the other pulser. Both values shall meet the requirement.	25 Ω to 40 Ω
Insertion loss	1) Produce a differential signal with the signal source (see Note 4). 2) Assure that skew between the pairs is minimized. (see Note 1). 3) Measure and store the insertion loss (IL) of the fixturing, using the IL reference traces provided on the board, over a frequency range of 10 MHz to 4 500 MHz. 4) Measure and record the IL of the sample, which includes fixturing IL, over a frequency range of 10 MHz to 4 500 MHz. 5) The IL of the sample is then the results of procedure 4 minus the results of procedure 3.	6 dB maximum

Parameter	Procedure	Requirements
Crosstalk: NEXT	1) Produce a differential signal with the signal source (see Note 1). 2) Connect the source to the risetime reference traces. Assure that skew between the pairs is minimized. (see Note 1). 3) Terminate the far ends of the reference trace with loads of characteristic impedance. 4) Measure and record the system and fixturing crosstalk. This is the noise floor. 5) Terminate the far ends of the drive and listen lines with loads of characteristic impedance. 6) Connect the source to the drive pair and the receiver to the near-end of the listen pair. 7) Measure the NEXT over a frequency range of 10 MHz to 4 500 MHz. 8) Verify that the sample crosstalk is out of the noise floor.	−26 dB
Rise time	1) Minimize skew (see Note 1). 2) Set the Time Domain Reflectometer (TDR) pulsers in differential mode with a positive going pulse (*V*+) and a negative going pulse (*V*-). Define a reflected differential trace on the receive channels as: V_{diff} = *V*+ −*V*-. 3) With the TDR connected to the risetime reference trace measure and record the input risetime. Verify that the input risetime is between 25 ps to 35 ps (measured 20 % to 80 % V_p) (see Note 2). 4) Remove the reflected trace definition. 5) Connect the TDR to the sample measurement traces. 6) Define a differential trace on the receive channels as: V_{diff} = *V*+ − *V*-. 7) Measure (measured 20 % to 80 % V_p) and record the output risetime.	85 ps
Inter-symbol Interference	K - 28.5 signal source running at 1.5 Gbit/s. The average position of zero crossing should not move more than the specified value.	50 ps maximum
Intra-Pair Skew	1) Set one of the Time Domain Reflectometer (TDR) pulsers in differential mode with a positive going pulse (*V*+) and a negative going pulse (*V*-). 2) With the TDR connected to the risetime reference trace verify an input risetime of 70 ps (measured 20% to 80% V_p). Filtering may be used to slow the system down (see Note 2). 3) Measure propagation delay (50% of V_p) of each line in a pair single-endedly. The skew equals the difference between each single ended propagation delay.	10 ps maximum

NOTE 1 Time domain measurement equipment allows for delay adjustment of the pulses so launch times can be synchronized. Frequency domain equipment requires the use of phase matched fixturing. The fixturing skew should be verified to be <1 ps on a TDR.

NOTE 2 The system risetime is to be set via equipment filtering techniques. The filter risetime is significantly close to the stimulus risetime. Therefore the filter programmed equals the square root of ($t_{r(observed)}$)squared − ($t_{r(stimulus)}$) squared. After filtering, verify the risetime is achieved using the risetime reference traces on the PCB fixture.

NOTE 3 Calibrate the system by substituting either precision 50 Ω loads or precision air lines (also terminated in 50 Ω loads) for the test fixture. This places the calibration plane directly at the input interface of the test fixture.

NOTE 4 A network analyzer is preferred. If greater dynamic range is required a signal generator/ spectrum analyzer may be used. Differential measurements require the use of a four port network analyzer although baluns or hybrid couplers may be used.

14.2.7.3 Housing and contact electrical requirements

Table 6 is the connector housing and contact electrical requirements.

Table 6 – Housing and contact electrical parameters, test procedures and requirements

Parameter	Procedure	Requirement
Insulation resistance	IEC 60512-3-1 After 500 VDC for 1 min, measure the insulation resistance between the adjacent contacts of mated and unmated connector assemblies.	1 000 MΩ minimum
Dielectric withstanding voltage	IEC 60512-2-2 Method B Test between adjacent contacts of mated and unmated connector assemblies.	The dielectric shall withstand 500 VAC for 1 min at sea level.
Low level contact resistance (LLCR)	IEC 60512-2-1 Subject mated contacts assembled in housing to 20 mV maximum open circuit at 100 mA maximum.	• Initially 30 mΩ maximum. • Resistance increase 15 mΩ maximum after stress
Contact current rating (Power segment)	• Mount the connector to a test PCB • Wire power pins P1, P2, P8 and P9 in parallel for power • Wire ground pins P4, P5, P6, P10 and P12 in parallel for return • Supply 6 A total DC current to the power pins in parallel, returning from the parallel ground pins (P4, P5, P6, P10 and P12) • Record temperature rise when thermal equilibrium is reached	1.5 A per pin minimum. The temperature rise above ambient shall not exceed 30 °C at any point in the connector when contact positions are powered. The ambient condition is still air at 25 °C.

14.2.7.4 Mechanical and environmental requirements

Table 7 lists the mechanical parameters and requirements, while Table 8 lists the environmental and reliability tests and requirements.

Table 7 – Mechanical test procedures and requirements

Test description	Procedure	Requirement
Visual and dimensional inspections	IEC 60512-1-1 Visual, dimensional and functional per applicable quality inspection plan.	Meets product drawing requirements.
Cable pull-out	IEC 60512-17-3 Condition A Subject a Serial ATA cable assembly to a 40 N axial load for a minimum of 1 min while clamping one end of the cable plug.	No physical damage
Cable flexing	For round cable: EIA 364-41 Condition I Dimension x = 3.7 × cable diameter, 100 cycles in each of two planes. For flat cable: EIA 364-41 Condition II 250 cycles using either Method 1 or 2.	No physical damage. No discontinuity over 1 µs during flexing.
Insertion force	IEC 60512-13-2 Measure the force necessary to mate the connector assemblies at a max. rate of 12.5 mm (0.492") per minute.	45 N maximum
Removal force	IEC 60512-13-2 Measure the force necessary to unmate the connector assemblies at maximum rate of 12.5 mm (0.492") per minute.	10 N minimum
Durability	IEC 60512-9-1 50 cycles for internal cabled application; 500 cycles for backplane/blindmate application. Test done at a maximum rate of 200 cycles per hour.	No physical damage. Meet requirements of additional tests as specified in the test sequence in 14.2.7.6.

Table 8 – Environmental parameters, test procedures, and requirements

Parameter	Procedure	Requirement
Physical shock	IEC 60512-6-3 Condition H Subject mated connectors to 30 g's half-sine shock pulses of 11 msec duration. Three shocks in each direction applied along three mutually perpendicular planes for a total of 18 shocks, see Note 2.	No discontinuities of 1 µs or longer duration. No physical damage.
Random vibration	IEC 60512-6-4 Condition V Test letter A Subject mated connectors to 5.35 g's RMS. 30 minutes in each of three mutually perpendicular planes, see Note 2.	No discontinuities of 1 µs longer duration.
Humidity	IEC 60512-11-3 Method II Test Condition A. Subject mated connectors to 96 hours at 40° C with 90% to 95% RH.	See Note 1
Temperature life	IEC 60512-9-2 Test Condition III Method A. Subject mated connectors to temperature life at +85°C for 500 hours.	See Note 1.
Thermal shock	IEC 60512-11-4 Test Condition I. Subject mated connectors to 10 cycles between -55°C and +85°C.	See Note 1.
Mixed Flowing Gas	IEC 60512-11-7, Class 2A Half of the samples are exposed unmated for seven days, then mated for remaining seven days. Other half of the samples are mated during entire testing.	See Note 1.
NOTE 1 Must meet IEC 60512-1-1 Visual Examination requirements, show no physical damage and shall meet requirements of additional tests as specified in the test sequence in 14.2.7.6. NOTE 2 Shock and vibration test fixture is to be determined by each user with connector vendors.		

An additional requirement is listed in Table 9.

Table 9 – Additional requirement

Parameter	Procedure	Requirement
Flammability	IEC 60512-20-2	Material certification or certificate of compliance required with each lot to satisfy IEC 60512-20-2, or better.

This standard does not attempt to define the connector and cable assembly reliability requirements that are considered application-specific. It is up to users and their connector suppliers to determine if additional requirements are needed to satisfy the application needs. For example, a user who requires a SMT connector may want to include additional requirements for SMT connector reliability.

14.2.7.5 Sample selection

Samples shall be prepared in accordance with applicable manufacturers' instructions and shall be selected at random from current production. Each test group shall provide at least 100 data points for a good statistical representation of the test result. For a connector with greater than 20 pins, a test group shall consist of a minimum of five connector pairs. From these connector pairs, a minimum of 20 contact pairs per mated connector shall be selected and identified. For connectors with less than 20 pins, choose the number of connectors sufficient to provide 100 data points.

14.2.7.6 Test sequence

Table 10 shows the connector test sequences for five groups of tests.

Table 10 – Connector test sequences

Test group →	**A**	**B**	**C**	**D**	**E**
Test or examination ↓					
Examination of the connector(s)	1, 5	1, 9	1, 8	1, 8	1, 7
Low-Level Contact Resistance (LLCR)	2, 4	3, 7	2, 4, 6		4, 6
Insulation resistance				2, 6	
Dielectric withstanding voltage				3, 7	
Current rating			7		
Insertion force		2			
Removal force		8			
Durability	3	4 (see note)			2 (see note)
Physical shock		6			
Vibration		5			
Humidity				5	
Temperature life			3		
Reseating (manually unplug/plug three times)			5		5
Mixed Flowing Gas					3
Thermal shock				4	
NOTE Preconditioning, 20 cycles for the 50-durability cycle requirement, 50 cycles for the 500-durability cycle requirement. The insertion and removal cycle is at the maximum rate of 200 cycles per hour.					

For example, in Test Group A, one would perform the following tests:

1) examination of the connector(s);
2) LLCR;
3) durability;
4) LLCR;
5) examination of the connector(s).

14.3 Cable assemblies

Figure 22 illustrates how signals and grounds are assigned in direct connect and cabled configurations.

The Serial ATA cable consists of four conductors in two differential pairs. If necessary, the cable may also include drain wires to be terminated to the ground pins in the Serial ATA cable receptacle connectors. The conductor size may be 0.25 mm to 0.41 mm diameter (30 AWG to 26 AWG). The cable maximum length is one meter.

This standard does not specify how to construct a standard cable for a serial implementation of ATA. Any cable that meets the electrical requirements is acceptable. The connector and cable vendors have the flexibility to choose cable constructions and termination methods based on performance and cost considerations. An example cable construction is given in Clause J.2 for informative purposes only.

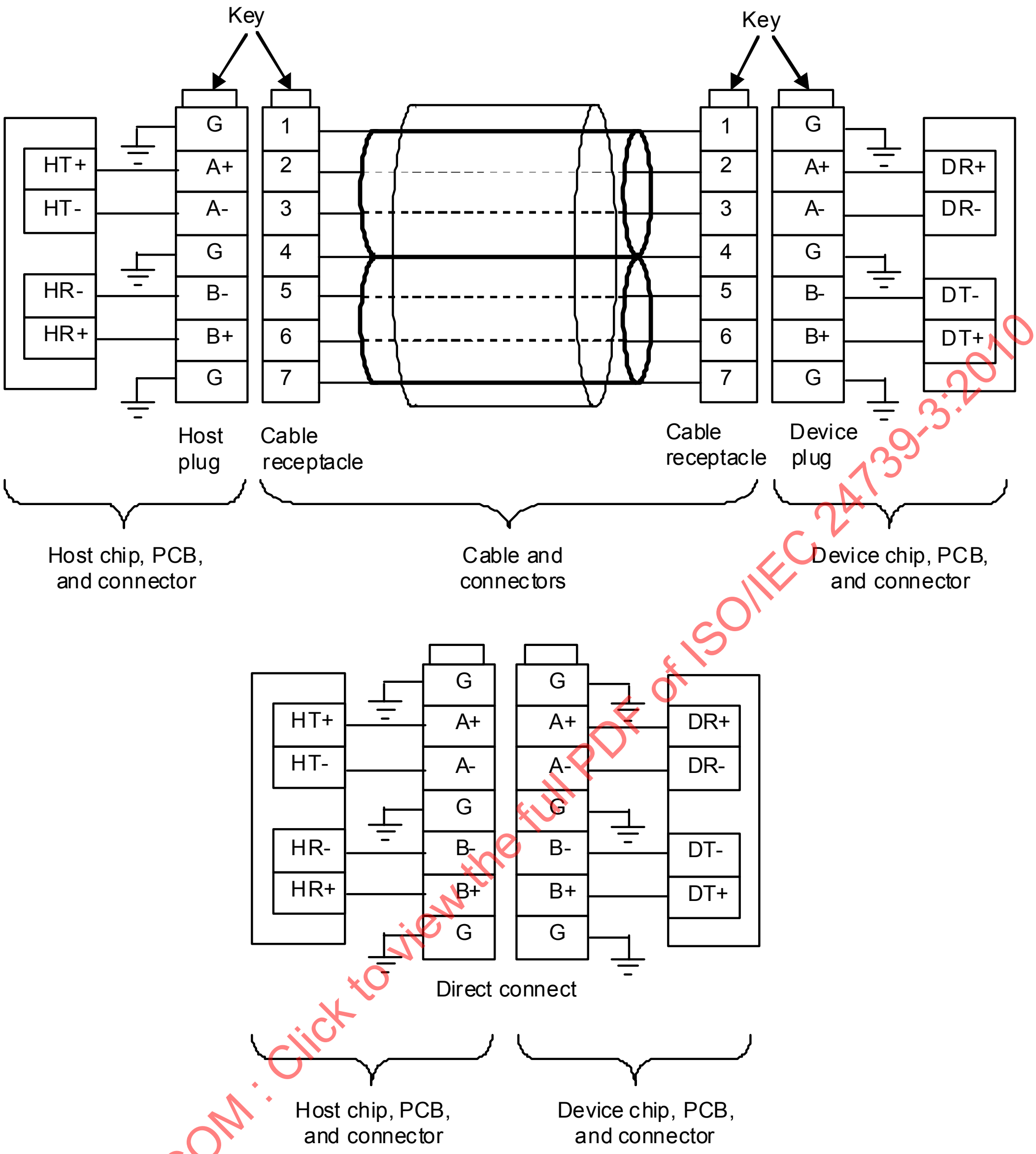


Figure 22 – Signals and grounds assigned in direct connect and cabled

The power receptacle connector is terminated onto 1 mm^2 (18 AWG) conductors that are connected to the system power supply or other power sources. Five 1 mm^2 (18 AWG) conductors may be used, with three conductors terminated to the nine power pins for the three voltages, while the remaining two conductors connect to the six ground pins.

14.4 Phy (physical layer electronics)

14.4.1 Physical plant as a system

14.4.1.1 General

A number of components of the system make up the system's performance, not just by mutually exclusively summing up the low-level error parameters detected at the bottom of the protocol stack can be made, but also including the retry algorithms.

There are two basic classes of errors that will affect the bit-error rate performance that have to be considered: bit-errors and burst-errors. The following clause addresses the methodology, for in-system bit-error-rate computations.

For in-system testing, the criteria for testing to a specified maximum frame error rate will be addressed, using a set of reference frames, defined by a specific set of ordered test patterns within the frame.

14.4.1.2 Test bit patterns and sequence characteristics

Test bit sequences are those bit sequences that are transmitted onto a serial link, so as to test the serial implementation of ATA interface's jitter compliance, as well as its derived communications-link performance.

Jitter is classified as Random Jitter (RJ), and Deterministic Jitter (DJ). RJ is Gaussian in nature, adds on a power basis and is measured as an RMS value. DJ is also referred to as systematic jitter, and usually is introduced into the serial link due to duty cycle distortion, power supply noise, substrate noise, or inter-symbol interference.

We will focus on various types of bit sequence patterns that emphasize low/high transition density patterns, as well as low/high-frequency patterns.

1) Low transition density patterns are those patterns containing long runs of ones and zeroes, intended to create inter-symbol interference by varying the excursion times at either extreme of the differential signalling levels.
2) High transition density patterns are those patterns containing short runs of ones and zeroes, also intended to create inter-symbol interference.
3) Bit patterns that contain low-frequency spectral components are a good test of the input high-pass filter circuitry, more specifically, introduced amplitude signal distortion, due to a marginal design. These bit patterns are a better test than those bit patterns having high-frequency spectral content.
4) Simultaneous switching output patterns are achieved by transmitting alternating 1's complement bit patterns (10-bits) for recovery at the receiver. These patterns create worst case power supply, or chip substrate, noise, and are achieved by selecting bit test pattern sequences that maximize current extremes at the recovered bit pattern parallel interface. These patterns induce noise into substrate supply and are a good test of the receiver circuitry.
5) The intent of random bit patterns is to provide those patterns containing sufficiently broad spectral content and minimal peaking, that may be used for both component, and system level architecture measurement of jitter output and bit-error-rate performance. These patterns are also intended to be the common baseline pattern stimulus, for system/component vendor comparative testing, attributing the Transmit jitter output measurement to the component performance and not to the spectral profile of the data pattern used.

These patterns may be used for testing of the serial implementation of ATA interfaces. The patterns are classified in two categories:

1) non-compliant patterns;
2) compliant patterns.

Non-compliant patterns are those patterns that are used for baseline jitter measurements and assessment of signal quality, given specified stimulus. These patterns do not comply to the required FIS formats, but are just a repeated selected set of 8b/10b characters.

Compliant patterns are those specified patterns that do contain the leading SOF primitive, the specified pattern as data content, and trailing CRC and EOF primitives. There is no suppression of the dual-consecutive ALIGN primitive during stimulus with this class of pattern.

The test patterns illustrated in the following clauses are indicated to start with negative running disparity for illustrative purposes only in order to convey the encoded 10b patterns transmitted on the wire for each sequence.

14.4.1.3 Low transition density bit pattern sequences

Low transition density bit patterns, as shown in Figure 23 below, are those patterns containing long runs of ones and zeroes. These patterns create high-frequency jitter due to inter-symbol interference, emphasized further when part of the composite pattern described in a later clause.

–

D17.7(-)	D30.7(+)	D7.1(+)	D14.7(+)	D30.7(-)	D7.6(-)	D30.3(-)	D30.3(+)
F1h	FEh	27h	EEh	FEh	C7h	7Eh	7Eh
100011 0111	100001 1110	000111 1001	011100 1000	011110 0001	111000 0110	011110 0011	100001 1100

D30.3(-)	D30.3(+)	D30.3(-)	D30.3(+)	D30.3(-)	D30.3(+)	D30.3(-)	D30.3(+)
7Eh	7Eh	7Eh	7Eh	7Eh	7Eh	7Eh	7Eh
011110 0011	100001 1100	011110 0011	100001 1100	011110 0011	100001 1100	011110 0011	100001 1100

Byte group (D30.3(+), D30.3(-)) repeat n times (n>12)

D30.3(-)	D30.3(+)	D30.3(-)	D30.3(+)	D3.7(-)	D28.7(+)	D3.7(-)	D28.7(+)
7Eh	7Eh	7Eh	7Eh	E3h	FCh	E3h	FCh
011110 0011	100001 1100	011110 0011	100001 1100	110001 1110	001110 0001	110001 1110	001110 0001

Figure 23 – Low transition density pattern

14.4.1.4 High transition density bit pattern sequences

High transition density patterns are those patterns containing only short runs of ones and zeroes. Because these patterns have significant inter-symbol interference, they may be used to generate high-frequency jitter. For this case there are two sub-classes of high-transition density patterns (see Figure 24):

1) half-rate high transition density bit pattern sequence;
2) quarter-rate high transition density bit pattern sequence.

An combination of the two sub-classes of high-transition density bit patterns will be used to represent high transition density test pattern.

–

D21.5(-)	D21.5(-)	D21.5(-)	D21.5(-)	D21.5(-)	D21.5(-)	D21.5(-)	D21.5(-)
B5h	B5h	B5h	B5h	B5h	B5h	B5h	B5h
101010 1010	101010 1010	101010 1010	101010 1010	101010 1010	101010 1010	101010 1010	101010 1010

Repeat *n* times (*n*>12)

D24.3(-)	D24.3(+)	D24.3(-)	D24.3(+)	D24.3(-)	D24.3(+)	D24.3(-)	D24.3(+)
78h	78h	78h	78h	78h	78h	78h	78h
110011 0011	001100 1100	110011 0011	001100 1100	110011 0011	001100 1100	110011 0011	001100 1100

Byte group (D24.3(+), D24.3(-)) repeat *n* times (*n*>12)

D10.2(-)	D10.2(-)	D10.2(-)	D10.2(-)	D10.2(-)	D10.2(-)	D10.2(-)	D10.2(-)
4Ah	4Ah	4Ah	4Ah	4Ah	4Ah	4Ah	4Ah
010101 0101	010101 0101	010101 0101	010101 0101	010101 0101	010101 0101	010101 0101	010101 0101

Repeat n times (*n*>12)

D25.6(-)	D6.1(-)	D25.6(-)	D6.1(-)	D25.6(-)	D6.1(-)	D25.6(-)	D6.1(-)
D9h	26h	D9h	26h	D9h	26h	D9h	26h
100110 0110	011001 1001	100110 0110	011001 1001	100110 0110	011001 1001	100110 0110	011001 1001

Repeat n times (*n*>12)

Figure 24 – Half-rate / quarter-rate high transition density pattern

14.4.1.5 Low-frequency spectral content bit pattern sequences

Bit patterns that contain low-frequency spectral components as shown in Figure 25 are a good test of the input high pass filter circuitry, as far as introduced signal distortion, due to a marginal design.

–

D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010

Repeat *n* times (*n*>12)

D11.7(-) EBh 110100 1110	D20.2(+) 54h 001011 0101	D20.2(+) 54h 001011 0101	D20.2(+) 54h 001011 0101	D20.2(+) 54h 001011 0101	D20.2(+) 54h 001011 0101	D20.2(+) 54h 001011 0101	D20.2(+) 54h 001011 0101

Byte group (D30.3(+), D30.3(-)) repeat *n* times (*n*>12)

D20.2(+) 54h 001011 0101	D20.2(+) 54h 001011 0101	D20.2(+) 54h 001011 0101	D20.7(+) F4h 001011 0001	D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010	D11.5(-) ABh 110100 1010

Figure 25 – Low-frequency spectral content pattern

14.4.1.6 Simultaneous switching outputs bit pattern sequences

Simultaneous switching outputs bit pattern sequences as shown in Figure 26 induce noise into substrate supply, and is a good test of the receiver circuitry. This is achieved by transmitting alternating 1's complement bit patterns (10-bits) for recovery at the receiver.

–

D31.3(-) 7Fh 101011 0011	D31.3(+) 7Fh 010100 1100	D31.3(-) 7Fh 101011 0011	D31.3(+) 7Fh 010100 1100	D31.3(-) 7Fh 101011 0011	D31.3(+) 7Fh 010100 1100	D31.3(-) 7Fh 101011 0011	D31.3(+) 7Fh 010100 1100

Repeat n times (*n*>512)

Figure 26 – Simultaneous switching outputs patterns

14.4.1.7 Composite bit pattern sequences

For the measurement of jitter, patterns should combine low-frequency, low transition density and high transition density patterns. All these combinations, but the Low Frequency Spectral Content class can be performed for relatively short test time intervals, for jitter and performance measurements.

The lower frequency pattern needs to be tested for longer interval periods to be able to observe the lower frequency jitter effects on the interface.

By including a composite set of the cited patterns, the resultant composite pattern as shown in Figure 27 stresses the interface components within the link with low and high-frequency jitter, tests for component and various amplitude distortions due to marginal receiver input circuitry or interface components.

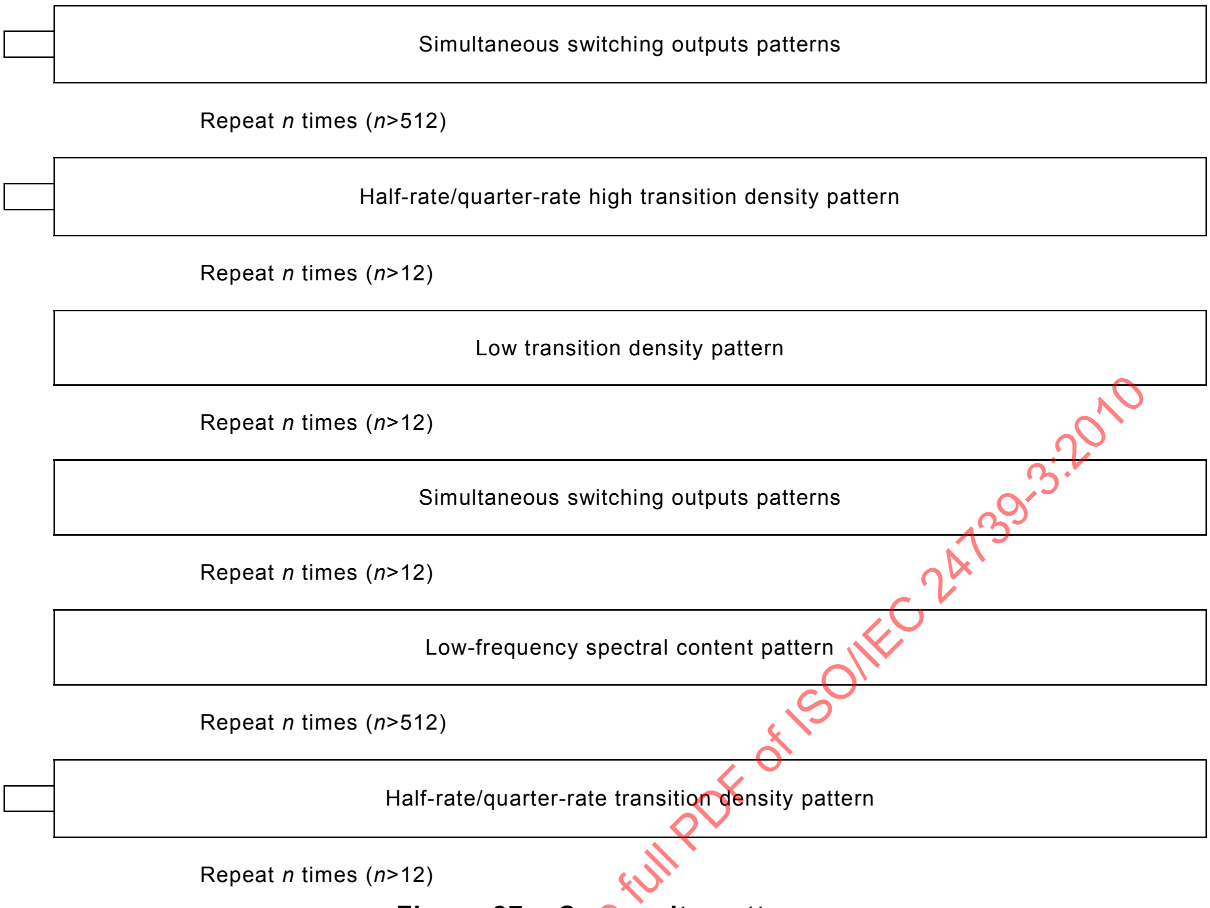


Figure 27 – Composite patterns

14.4.2 Bit error rate testing

NOTE This subclause is given for information only.

14.4.2.1 General

In order to get a fair assessment of bit-error-rate performance, bit-errors, as well as burst errors, should be considered separately.

Note that a single bit error has a high probability of causing a byte-wise error, or an 8b/10b code violation error, due to the 8b/10b encoding, thus a single bit error translates to 8-bits or 10-bits of error. Each of these occurrences are classified as "byte errors", and these form the basis for the bit-error-rate calculations.

A missing or an extra bit detected by the receiver translates into a series of errors that span across multiple byte boundaries until bit realignment via ALIGN primitives (see 14.6), worst case. This series of errors are defined as burst errors.

A third type of byte-wise error exists where, an entire byte is not received, this type of error is not classified differently, as it will cause a burst error whose span may be limited by higher-layer protocol transmission conventions, at the cost of additional overhead ALIGN primitive sequences.

A single error may result in several related errors occurring closely together which in turn may result in multiple bit-error counts. A character might have a single bit error in it that causes a code-violation error. A disparity error might occur on a following character, caused by the same single error. To prevent multiple error counts from a single cause, the following concept of an error burst is introduced.

- An “error burst” is defined as a time period 1.5 s ± 0.5 s long, during which one or more invalid transmission words are recognized. This time may be exceeded due to infrequent unusual conditions.
- Only one error in an error burst shall be counted toward the error-burst-rate threshold.

14.4.2.2 Error-burst-rate-thresholding measurement

The error-burst-rate thresholding process is designed to detect an increased error rate before performance degradation becomes serious. Measurement of error-burst-rate thresholding is accomplished by counting the number of error bursts that occur in a 5-min period, when tested with the compliant frame patterns, defined by 14.4.1.3, but with the N parameters extended so as to achieve the maximum frame length. When the count equals a specific number, the threshold is exceeded. This specific number is called the “threshold error count.”

An error-burst-rate threshold is detected when 15 error bursts occur in a 5-min period. The required accuracy of the 5-min period mentioned above is ±0.3 s. Reaching the threshold error count of 15 in a 5-min period is a positive indication of the interface failing the error rate requirements and may be used to terminate testing. However, not reaching the threshold should not be taken as indication that the interface is passing without also confirming that the frame error rate requirements are satisfied using statistically sound measurement techniques. Being under the threshold error count after a 5-min period should therefore not be used as a measure of system compliance.

The error-burst-rate counting process will be restarted when PHY Ready state is entered, and when an amount of time has elapsed after a error-burst-rate threshold is detected. After an error-burst-rate threshold is detected, at least 15 additional error bursts will occur before the next error-burst-rate threshold is detected. In addition, the error-burst counting process may be restarted whenever the 5-min time period has expired even though an error-burst-rate threshold is not reached.

14.4.2.3 Bit-error-rate measurements

Bit error rates if measured and computed, byte-wise, should be no greater than 10^{-12} bit-errors when tested with the reference test patterns, cited in 14.4.1.2

Note that the Frame-Error Rate measurements constitute the basis for the applicable test requirements for this standard.

14.4.3 Frame error rate testing

14.4.3.1 General

Frame error rate testing is the measure of link performance using all the intermediate circuit blocks in the chain from low-level Physical layer, Link Layer, through Transport Layer. Error detection is at the frame level using the CRC (Cyclic Redundancy Check) error detection mechanism and respective reporting to the higher layer levels.

14.4.3.2 Frame error-rate patterns

14.4.3.2.1 General

Frame error rate patterns contain the elements of the test bit patterns and sequence of patterns, so as to thoroughly be able to test the serial interface in system, using the higher level CRC error detection/reporting from the lower protocol level layers to the Application level layer.

The frame patterns shall be comprised of the set of the Composite Patterns, cited in 14.4.1.7, but with the N parameters extended so as to achieve the maximum frame length. Refer to 14.7.2 for a description of Loopback Testing configurations.

14.4.3.2.2 Loopback test

Test patterns cited in this clause used as stimulus that may be used to verify the serial interface compliance and signal integrity, using the following test models.

1) Non-compliant test patterns for jitter measurements, physical connection media tests, and electrical parameter testing.
2) Compliant test patterns for frame error rate testing and in-system tests.

14.4.4 Test requirements, non-compliant patterns

Electrical parameters of 14.4.9 shall be verified using the following set of test patterns, using the BIST FIS, Far-End Transmit Mode, described in 16.2.8.

1) Low transition density bit patterns, as per 14.4.1.3.
2) High transition density bit patterns as per 14.4.1.4
3) Low-frequency spectral component bit patterns as per 14.4.1.5.
4) Simultaneous switching outputs bit patterns as per 14.4.1.6.

14.4.5 Test requirements, compliant frame patterns

The frame error rates specified in 14.4.10 shall be tested for compliance when subjected to any implementation-determined worst-case compliant patterns, as well as the following set of compliant patterns.

1) Compliant low transition density bit patterns, as per 14.4.1.3.
2) Compliant high transition density bit patterns as per 14.4.1.4.
3) Compliant low-frequency spectral component bit patterns as per 14.4.1.5.
4) Compliant simultaneous switching outputs bit patterns as per 14.4.1.6.
5) Compliant composite patterns as per 14.4.1.7.

Where the qualifying prefix term "compliant" signifies transmission of the cited pattern encapsulated in the data portion of the Frame Information Structure and used in a the serial implementation of ATA operational transmission context.

Note that the cited patterns should appear on the wire, and the N parameters of the reference patterns shall be extended to achieve the maximum frame length. These compliant patterns as shown in Figure 28 contain the necessary SOF leading primitive, the calculated CRC and the trailing EOF primitive, as shown in the figure below.

SOF	Specified test pattern	CRC	EOF

Figure 28 – Compliant test patterns

14.4.6 Test requirements, loopback

14.4.6.1 General

As specified in 14.7.2, only one of the three types of Loopback test schemes form part of this standard: Far-End Retimed Loopback. Both Far-end and Near-End Analog Loopback schemes are classified as Vendor Specific, thus will not have specified test requirements.

14.4.6.2 Test requirements for loopback, far-end retimed

Generation of loopback test patterns is optional and vendor specific. As this loopback scheme needs a specific action from the far-end connected interface, this mode is entered by way of the BIST FIS described in 16.2.8.

This loopback test is intended for a relatively quick assessment of interface integrity, accommodating an in-system test capability.

In this Far-End Retimed Loopback mode, the interface shall operate without a CRC error for a sustained minimum period of 1 000 ms at Gen1 speed, using the following set of compliant reference frame patterns if the test pattern generation is supported by the initiating device.

1) Compliant composite reference frame patterns as specified in 14.4.3.2.
2) Compliant low transition density bit patterns, as per 14.4.1.3.
3) Compliant high transition density bit patterns as per 14.4.1.4.
4) Compliant low-frequency spectral component bit patterns as per 14.4.1.5.
5) Compliant simultaneous switching outputs bit patterns as per 14.4.1.6.
6) Compliant composite patterns as per 14.4.1.7.

Where the qualifying prefix term "compliant" signifies transmission of the cited pattern encapsulated in the data portion of a valid Frame Information Structure and used in a the serial implementation of ATA operational transmission context.

The Far-End Interface shall remain in this Far-End Retimed Loopback, until receipt of the COMRESET/COMINIT OOB Signalling sequence.

14.4.6.3 Test requirements for loopback, far-end analog (vendor specific)

The test requirements for this Loopback scheme are Vendor Specific.

14.4.6.4 Test requirements for loopback, near-end analog (vendor specific)

The test requirements for this Loopback scheme are Vendor Specific.

14.4.7 Test Method for Data Rate Frequency Variation, SSC Profile

HFTP and MFTP clock patterns are used to best verify the SSC profile, as well as the ppm extremes, the ISI effects due to the media will be minimized during these tests. Using the BIST FIS, invoke the Transmit-Only option with valid 8B10B encoded patterns that are periodic, such as sustained D10.2, or D24.3. These “clock” patterns may be processed, using cycle-averaging techniques to verify the SSC profile, for ppm-extremes, modulation-rate and ensure that the SSC-modulation peak-jitter levels never exceed the specified jitter maximums.

Cycle-averaging is one method used that removes the high-frequency jitter components, of the captured/displayed waveform so as to focus analysis on the low-frequency effects. A set number of cycles is averaged for and these averages of UI-periods are plotted versus time. For analysis purposes, the waveform of the SATA channel shall be displayed as UI-period versus Time, where the time axis shall be scaled such that the entire SSC modulation period is displayed.

Separate numerical analysis shall be provided to indicate the average UI-periods, at the peaks of the modulation profile. These average-UI periods shall not exceed the ppm-extremes, per Table 11.

These methods are used to evaluate the SSC profile, for profile-shape, distortions, profile-rate and respective peak-frequencies. The “clock patterns” are unclouded by the effects of pattern-related ISI.

14.4.8 Block diagram

The following block diagram, see Figure 29, is provided as a reference for the following clauses of this document. Although informative in nature, the functions of the blocks described herein provide the basis upon which the normative specifications apply. The individual blocks provided are representative of an approach this design and are provided as an example of one possible implementation.

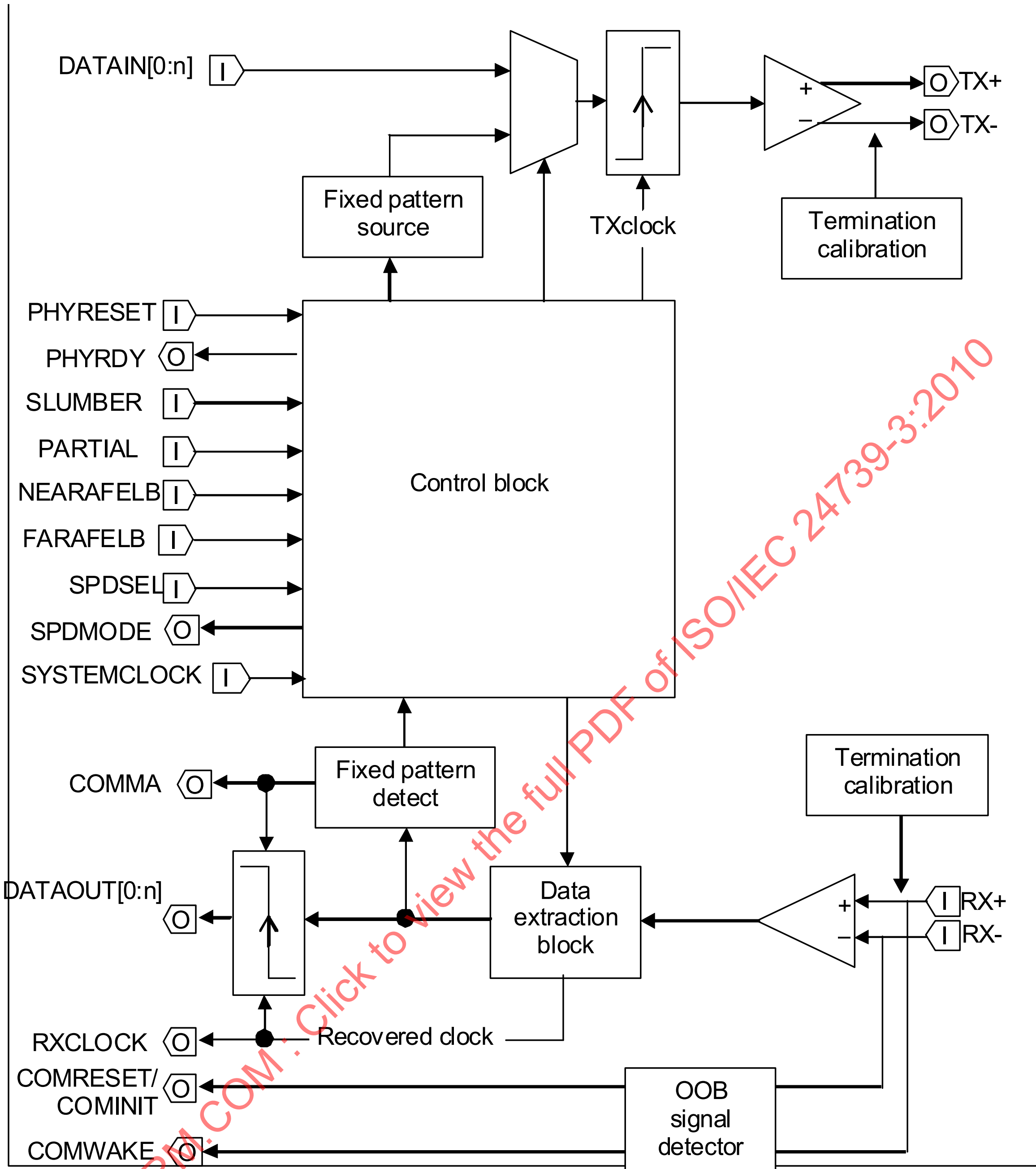


Figure 29 – Physical plant overall block diagram

Analog front end	This block is the basic interface to the transmission line. This block consists of the high speed differential drivers and receivers as well as the Out of Band signalling circuitry.
Control block	This block is a collection of logic circuitry that controls the overall functionality of the Physical plant circuitry.
Fixed pattern source	This block provides the support circuitry that generates the patterns as needed to implement the ALIGN primitive activity.
Fixed pattern detect	This block provides the support circuitry to allow proper processing of the ALIGN primitives.
Data extraction block	This block provides the support circuitry to separate the clock and data from the high speed input stream.
TX clock	This signal is internal to the Physical plant and is a reference signal that regulates the frequency at which the serial stream is sent via the high speed signal path.
TX+ / TX-	These signals are the outbound high speed differential signals that are connected to the serial ATA cable.

RX+ / RX-	These signals are the inbound high speed differential signals that are connected to the serial ATA cable.
DATAIN	Data sent from the Link layer to the Physical layer for serialization and transmission.
PHYRESET	This input signal causes the PHY to initialize to a known state and start generating the COMRESET Out of Band signal across the interface.
PHYRDY	Signal indicating Phy has successfully established communications. The Phy is maintaining synchronization with the incoming signal to its receiver and is transmitting a valid signal on its transmitter.
SLUMBER	Causes the Physical layer to transition to the Slumber power management state.
PARTIAL	Causes the Physical layer to transition to the Partial power management state.
NEARAFELB	Causes the Phy to loop back the serial data stream from its transmitter to its receiver.
FARAFELB	Causes the Phy to loop back the serial data stream from its receiver to its transmitter.
SPDSEL	Causes the control logic to automatically negotiate for a usable interface speed or sets a particular interface speed. The actual functionality of this input is vendor specific and varies from manufacturer to manufacturer.
SPDMODE	Output signal that reflects the current interface speed setting. The actual functionality of this signal is vendor specific and varies from manufacturer to manufacturer.
SYSTEMCLOCK	This input is the reference clock source for much of the control circuit and is the basis from which the transmitting interface speed is established.
COMMA	This signal indicates that a K28.5 character was detected in the inbound high speed data stream.
DATAOUT	Data received and deserialized by the Phy and passed to the Link layer.
RX CLOCK / Recovered clock -	This signal is derived from the high speed input data signal and determines when parallel data has been properly formed at the DATAOUT pins and is available for transfer to outside circuitry.
OOB signal detector	This block decodes Out of Band signal from the high speed input signal path.
COMRESET / COMINIT	Host: Signal from the out of band detector that indicates the COMINIT out of band signal is being detected. Device: Signal from the out of band detector that indicates the COMRESET out of band signal is being detected.
COMWAKE	Signal from the out of band detector that indicates the COMWAKE out of band signal is being detected.

14.4.9 Electrical specifications

The serial implementation of the ATA physical layer electrical requirements are depicted in Table 11.

The electrical performance is specified at the mated connector pair and includes the effects of the PCB.

The electrical portion of the physical layer includes the driver, receiver, PCB and mated connector pair. Unless otherwise specified, all measurements shall be taken through the mated connector pair. Driver and receiver designs compensate for the effects of the path to/from the I/O connector. The receiver schematic shown in Figure 32 or the transmitter schematic shown in Figure 31 shall be used to design a test fixture.

Table 11 – Physical Layer Electrical Requirements

	Nom.	Min.	Max.	Units	Comments
T,UI		666.43	670.12	ps	Operating data period (nominal value architecture specific)
t_{rise}	0.3	0.15	0.41	UI	20 % to 80 % at transmitter
t_{fall}	0.3	0.15	0.41	UI	80 % to 20 % at transmitter
$V_{CM,DC}$	250	200	450	mV	Common mode DC level measured at receiver connector. This spec only applies to direct-connect designs or designs that hold the common-mode level. AC coupled designs may allow the common mode to float. See $V_{CM,AC\ coupled}$ requirements.
$V_{CM,AC\ coupled}$ TX		0	2.0	V	Open circuit DC voltage level of each signal in the TX pair at the IC side of the coupling capacitor in an AC coupled PHY. This requirement shall be met during all possible power and electrical conditions of the PHY including power off and power ramping. Transmitter common mode DC levels outside this range may be used provided that the following is met: The common mode voltage transients measured at the TX pins of the connector into an open-circuit load during all power states and transitions shall not exceed a +2.0 V or -2.0 V change from the CM value at the beginning of each transient. Test conditions shall include system power supply ramping at the fastest possible power ramp.
$V_{CM,AC\ coupled}$ RX		0	2.0	V	Open circuit DC voltage level of each signal in the RX pair at the IC side of the coupling capacitor in an AC coupled PHY. Shall be met during all possible power and electrical conditions of the PHY including power off and power ramping. Receiver common mode DC levels outside this range may be used provided that the following is met: The common mode voltage transients measured at the RX pins of the connector into an open-circuit load during all power states and transitions shall not exceed a +2.0 V or -2.0 V change from the CM value at the beginning of each transient. Test conditions shall include system power supply ramping at the fastest possible power ramp
F_{CM}		2	200	MHz	All receivers shall be able to tolerate sinusoidal common-mode noise components inside this frequency range with an amplitude of $V_{CM,AC}$.
$T_{settle,CM}$			10	ns	Maximum time for common-mode transients to settle to within 10 % of DC value during transitions to and from the idle bus condition.
V_{diff},TX	500	400	600	mVp-p	±250 mV differential nominal. Measured at Serial ATA connector on transmit side
V_{diff},RX	400	325	600	mVp-p	±200 mV differential nominal. Measured at Serial ATA connector on receive side
TX pair differential impedance	100	85	115	Ω	As seen by a differential TDR with 100 ps (max) edge looking into connector (20 % to 80 %). Measured with TDR in differential mode.
RX pair differential impedance	100	85	115	Ω	As seen by a differential TDR with 100 ps (max.) edge looking into connector (20 % to 80 %). Measured with TDR in differential mode.

	Nom.	Min.	Max.	Units	Comments
TX single-ended impedance		40		Ω	As seen by TDR with 100 ps (max) edge looking into connector (20 % to 80 %). TDR set to produce simultaneous positive pulses on both signals of the TX pair. Single-ended impedance is the resulting (even mode) impedance of each signal. Both signals shall meet the single ended impedance requirement. Shall be met during all possible power and electrical conditions of the PHY including power off and power ramping.
RX single-ended impedance		40		Ω	As seen by TDR with 100 ps (max.) edge looking into connector (20 % to 80 %). TDR set to produce simultaneous positive pulses on both signals of the RX pair. Single-ended impedance is the resulting (even mode) impedance of each signal. Both signals shall meet the single ended impedance requirement. Shall be met during all possible power and electrical conditions of the PHY including power off and power ramping.
CACcoupling			12	nF	Coupling capacitance value for AC coupled TX and RX pairs.
TX DC clock frequency skew		-350	+350	ppm	Specifies the allowed ppm tolerance for TX DC frequency variations around the nominal 1.500 GHz. Excludes the +0/-5 000 ppm SSC downspread AC modulation per 14.5.3.
TX AC clock frequency skew		-5 000	+0	ppm	Specifies the allowed ppm extremes for the SSC AC modulation, subject to the “Downspread SSC” triangular modulation (30kHz to 33kHz) profile per 14.5.3. Note: Total TX Frequency variation around nominal 1.500G, includes [TX DC] + [TX AC] ppm variations.
TX differential skew			20	ps	(Nominal value architecture specific)
Squelch detector threshold	100	50	200	mVp-p	Minimum differential signal amplitude
COMRESET/COMINIT detector off threshold		175	525	ns	Detector shall reject all bursts with spacings outside this specification.
COMRESET/COMINIT detector on threshold	320	304	336	ns	Detector shall detect all bursts with spacings meeting this period.
COMRESET/COMINIT transmit spacing	320.0	310.4	329.6	ns	As measured from 100 mV differential crosspoints of last and first edges of bursts.
COMWAKE detector off threshold		55	175	ns	Detector shall reject all bursts with spacings outside this specification.
COMWAKE detector on threshold	106.7	101.3	112	ns	Detector shall detect all burst spacings meeting this period.
COMWAKE transmit spacing	106.7	103.5	109.9	ns	As measured from 100 mV differential crosspoints from last to first edges of bursts.
UIOOB		646.67	686.67	ps	Operating data period during OOB burst transmission.

14.4.10 Frame error-rate measurements

Frame Error Rates of a system shall be measured and computed, to be no greater than (8.196×10^{8}) frame errors when tested with any given 8b/10b pattern, but shall include the Frame-Error-Rate reference patterns cited in 14.4.3

The cited patterns appear on the wire, and the N parameters of the reference patterns shall be repeated to achieve the maximum frame length, as specified in 14.4.3.2

14.4.11 Receiver Differential voltage

When subjected to the high-transition density patterns of 14.4.1.4, the Differential voltage measured at the Receiver shall comply to the electrical specifications of 14.4.9.

14.4.12 Receiver Common-mode voltage

References to peak-to-peak voltages are cited in 14.4.9.

The Receiver shall operate to within the bit error rates cited in 14.4.2, when subjected to a sinusoidal interfering signal with peak-to-peak voltage, see Table 11, and swept from the frequency range extremes, at a sweep rate period no shorter than 33.33 µs.

The Receiver shall operate to within the frame error rates cited in 14.4.3, when subjected to a sinusoidal interfering signal with peak-to-peak voltage and swept from the frequency range extremes, at a sweep rate period no shorter than 33.33 µs.

14.4.13 Transmitter Differential voltage

When subjected to the high-transition density patterns of 14.4.1.4, the Differential voltage measured at the Transmitter shall comply to the respective electrical specifications of 14.4.9.

14.4.14 Transmitter Common-mode voltage

The Transmitter shall comply to the electrical specifications of 14.4.9, when subjected to a sinusoidal interfering signal with peak-to-peak voltage, and swept from the frequency range extremes, at a sweep rate period no shorter than 33.33 µs.

14.4.15 Rise/fall times

Output rise times and fall times are measured between 20 % and 80 % of the signal, see Signal Rise and Fall Times, Figure 30. Rise and fall time requirements apply to differential transitions, for both In-Band and Out-Of-Band signaling.

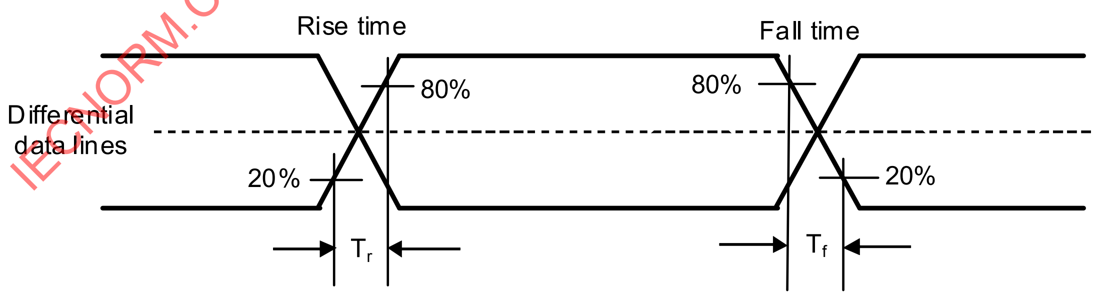


Figure 30 – Signal rise and fall times

The rise and fall times for transmitter differential buffer lines are measured with the load shown in Figure 31, at the transmitter mated connector pair, and shall comply with the Electrical Specifications 14.4.9 as well as be matched within ±10 % of each other to minimize RFI emissions and signal skew.

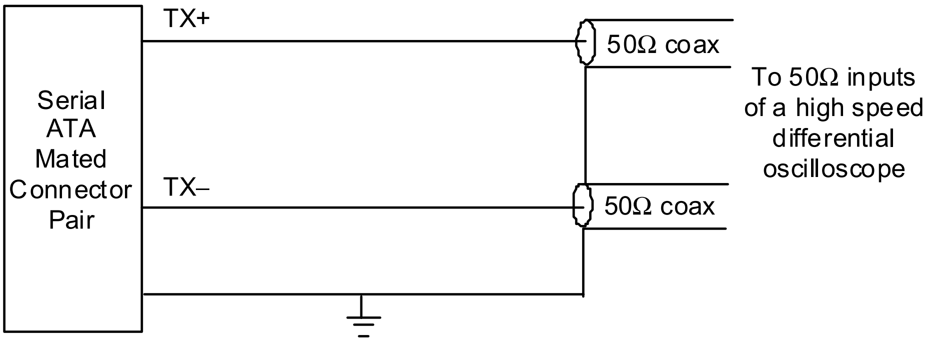


Figure 31 – Transmit test fixture

When testing the serial interface with the specified bit pattern sequences of 14.4.1, use the Receiver Test fixture as shown in Figure 32.

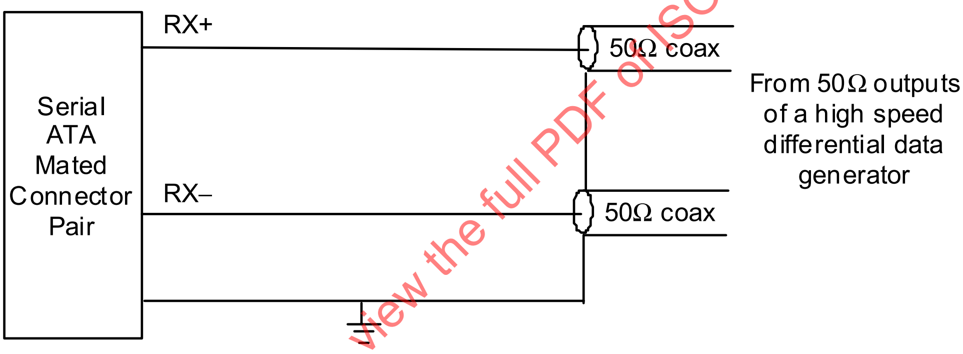


Figure 32 – Receive test fixture

14.5 Electrical features

14.5.1 Definitions

RJ Random jitter (peak-to-peak). Assumed to be Gaussian and equal to 14 times the 1σ r.m.s. value given the 10^{-12} BER requirement.

DJ Deterministic Jitter (peak-to-peak). All jitter sources that do not have tails on their probability distribution function (i.e. values outside the bounds have probability zero). Four kinds of deterministic jitter are identified: duty cycle distortion, data dependent (ISI), sinusoidal and uncorrelated (to the data) bounded. DJ is characterized by its bounded, peak-to-peak value.

TJ Total jitter (peak-to-peak). Peak-to-peak measured jitter including DJ and RJ.

ISI Inter-symbol interference. Data-dependent deterministic jitter caused by the time differences required for the signal to arrive at the receiver threshold when starting from different places in bit sequences (symbols). For example media attenuates the peak amplitude of the bit sequence [0,1,0,1...], more than it attenuates the peak amplitude of the bit sequence [0,0,0,0,1,1,1,1...], thus the time required to reach the receiver threshold with the [0,1,0,1...] sequence is less than required from the [0,0,0,0,1,1,1,1...] sequence. The run length of 4 produces a higher amplitude which takes more time to overcome when changing bit values and therefore produces a time difference compared to the run length of 1 bit sequence. When different run lengths are mixed in the same transmission the different bit sequences (symbols) therefore interfere with each other. ISI is expected whenever any bit sequence has frequency components that are propagated at different rates by the transmission media. This translates into high-high-frequency, data-dependent, jitter.

UI Unit Interval. Equal to the time required to transmit one bit (666.667 ps for Gen1).

DC Strictly, the non-AC component of a signal. In this specification, DC means all frequency components below f_{dc} = 100 kHz.

Differential signal A signal derived by taking the difference between two conductors. In this spec a differential signal is comprised of a positive conductor and a negative conductor. The differential signal is the voltage on the positive conductor minus the voltage on the negative conductor (i.e. TX+ − TX-).

Burst A short pulse of data starting from and ending with the idle condition on the interface. These are used for low-level signalling when the high-speed communication channel is not established.

SSC Spead Spectrum Clocking. The technique of modulating the operating frequency of a circuit slightly to spread its radiated emissions over a range of frequencies rather than just one tone. This reduction in the maximum emission for a given frequency helps meet FCC requirements.

14.5.2 Differential voltage/timing (EYE) diagram

14.5.2.1 General

The EYE diagram is more of a qualitative measurement than a specification. Any low-frequency (trackable) modulation is tracked by the oscilloscope to prevent measurement error caused by benign EYE closure. It is also unrealistic to try to capture sufficient edges to guarantee the 14 sigma RJ requirement in order to achieve an effective 10^{-12} BER. Nonetheless, this method is useful and easy to set up in the lab.

If t3-t1 is set equal to DJ+6×RJ_{1sigma}, then less than one in every 750 edges cross the illegal region for a well designed transmitter. By controlling the number of sweeps displayed, a quick health check may be performed with minimal setup. Note that this criteria is informative, and these conditions are not sufficient to guarantee compliance, but are to be used as part of the design guidelines.

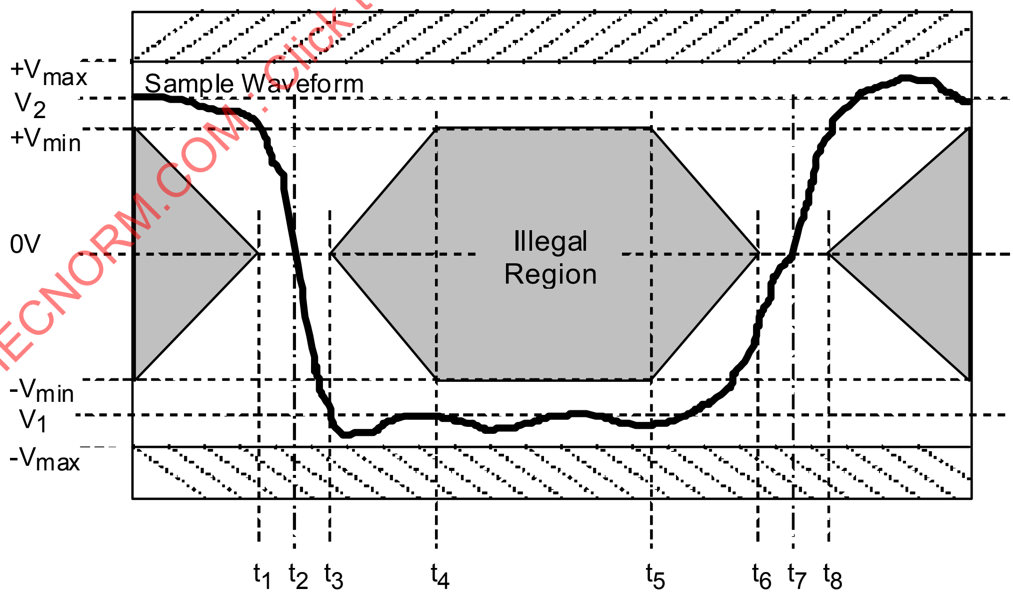


Figure 33 – Voltage / timing margin base diagram

Table 12 – Voltage / timing margin definition

Name	Definition	Notes
t_{jitter}	$t_3 - t_1$	$t_3 - t_1 = t_8 - t_6$
T	$t_7 - t_2$	$t_2 - t_1 = t_3 - t_2$ $t_7 - t_6 = t_8 - t_7$
V_{diff}	$V_2 - V_1$	

14.5.2.2 Jitter output/tolerance mask

The spectral jitter shall comply to the requirements as indicated in the Jitter Output/Tolerance Graph, shown in Figure 34 with magnitudes defined in Table 13. A_x are the maximum peak-to-peak transmitter output and the minimum peak-to-peak receiver tolerance requirements as measured from data edge to any following data edge up to $n_x \times$UI later (where x is 0,1 or 2). See Annex I.

Transmitter output data edge to data edge timing variation from t_o to t_y shall not exceed the value computed by the following:

- y is an integer from 1 to n_x,
- t_y is the time between the data transition at t_0 and a data transition y∗UI bit periods later.

This measurement may be made with an oscilloscope having a histogram function or with a Timing Interval Analyzer (TIA). For further information on jitter measurements, see 14.4. A receiver shall meet the error rate requirement under the maximum allowed jitter conditions found in Figure 33, Table 12 and Table 13.

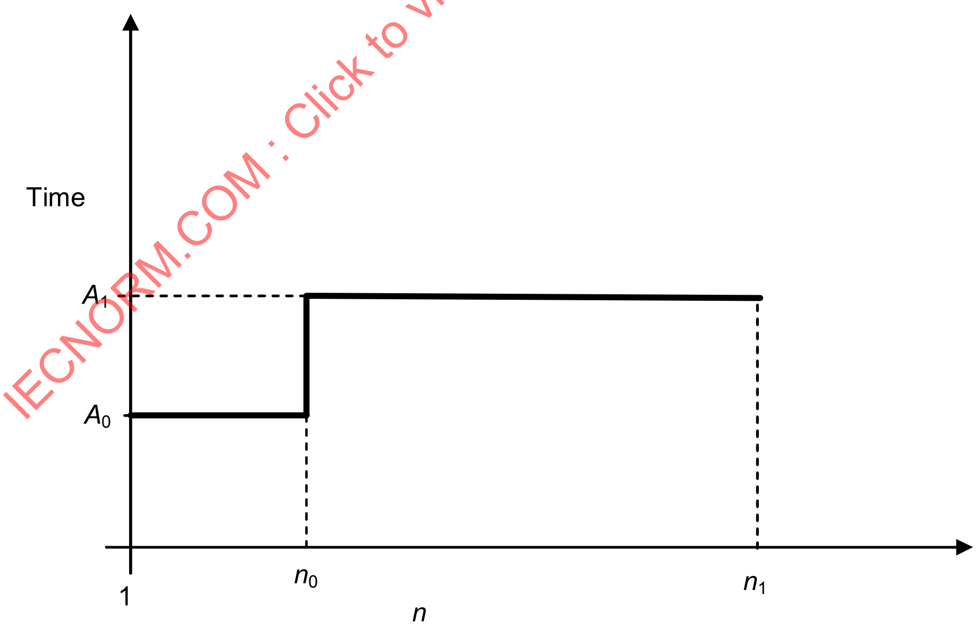


Figure 34 – Jitter output/tolerance mask

14.5.2.3 Sampling differential noise budget

Sampling jitter specifications relate to the relationship between the sampling clock and the data. Any phase error that results in the sample being improperly read (i.e. prior bit or following bit sampled) results in a bit error.

These error components have been broken out into Deterministic Jitter (DJ) and Total Jitter (TJ) where appropriate. DJ is the peak-to-peak phase variation in the 0 V_{diff} crossing point of the data stream that is fixed given any specific set of conditions. TJ is defined as DJ + Random Jitter (RJ). RJ is defined as 14 times the rms (1 sigma) value of the jitter that is Gaussian (normal).

The serial interface jitter characteristics should comply to within the jitter budget allocations tabulated in Table 13.

Table 13 – Sampling differential noise budget

Description	Driver output (see Note 1)		Driver PCB connector		Receiver PCB connector		Receiver Input (see Note 2)		Notes
	DJ	TJ	DJ	TJ	DJ	TJ	DJ	TJ	
$A_{0,p\text{-}p}$ (UI)	0.15	0.33	0.175	0.355	0.25	0.43	0.275	0.455	3,4
n_0	5	5	5	5	5	5	5	5	3,4
$A_{1,p\text{-}p}$ (UI)	0.2	0.45	0.22	0.47	0.35	0.6	0.37	0.62	3,4
n_1	250	250	250	250	250	250	250	250	3,5

NOTE 1 The driver output is the maximum jitter that a driver may exhibit to guarantee operation.

NOTE 2 This field is the maximum jitter that a receiver shall tolerate to guarantee operation.

NOTE 3 Does not include UI error due to frequency skew (XTAL or SSC related).

NOTE 4 Primarily determined by non-tracking architecture requirements.

NOTE 5 Primarily determined by tracking architecture requirements.

14.5.2.4 Jitter output

14.5.2.4.1 General

Jitter output tests are intended to determine the jitter amplitudes of the random jitter and deterministic jitter components.

14.5.2.4.2 Jitter measurements

The jitter specifications are based in a data transition to data transition Timing Interval Analyzer (TIA) format. If such instrumentation is not available, equivalent measurements may be made with an oscilloscope. The oscilloscope is set up to trigger on the data and an EYE diagram is examined some integer number of unit intervals (UI) later. The histogram function may be used to monitor the distribution of the zero crossing to extract jitter information. Figure 35 shows an example where the oscilloscope is set up to examine an edge 16 UI away from the trigger point.

The DJ and RJ (or TJ) should be less than the maximum A_x indicated in Figure 35 for $n_x \geq 16$. For example, assume n_0 is five and n_1 is 250. In this case, the DJ and RJ for t_{16} is less than A_1 since 250 > 16 but need not meet the A_0 requirement since 5 < 16. This method is employed because of the ease of measurement and to enable the specification to be relaxed at lower frequencies. If DJ is large enough, it can be fairly accurately measured from the histogram as the distance between the two outside peaks (there may be more than the three peaks shown in the example). If DJ and RJ are too intermixed for measurement, then other methods are used. One such method is to transmit a clock pattern (fixed frequency components) to determine the RJ (DJ is zero for fixed frequencies), transmit a random pattern to measure the TJ, and derive the DJ from these. Another alternative is to wait until >>10^{12} edges are included in the histogram and measure the Total Jitter (TJ).

Since t_0 is triggered from the serial signal rather than a reference clock the resulting measurements represent a combination of the jitter at t_0 and t_n.

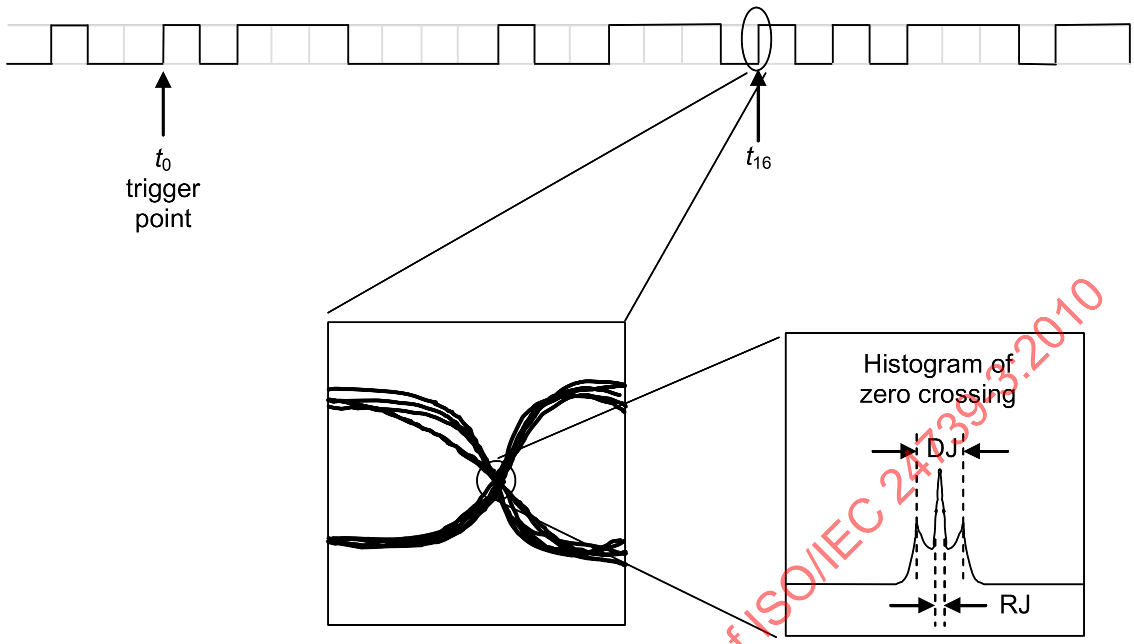


Figure 35 – Jitter measurement example

14.5.3 Spread spectrum clocking (SSC)

Spread Spectrum functionality is defined as follows.

1) All transmitter timings (including jitter, skew, min-max clock period, output rise/fall time) shall meet the existing non-spread spectrum specifications when spread spectrum is on.
2) Because the minimum clock period cannot be violated, the transmitter shall adjust the spread technique to not allow for modulation above the nominal frequency. This technique is often called “down-spreading”. An example triangular frequency modulation profile is shown in Figure 36. The modulation profile in a modulation period can be expressed as:

$$f = \begin{cases} (1-\delta)\, f_{nom} + 2\, f_{nom} \cdot \delta \cdot f_{nom} \cdot t & \text{when } 0 < t < \dfrac{1}{2f_m} \\ . & \\ (1+\delta)\, f_{nom} - 2\, f_{nom} \cdot \delta \cdot f_{nom} \cdot t & \text{when } \dfrac{1}{2f_m} < t < \dfrac{1}{f_m} \end{cases}$$

where

f_{nom} is the nominal frequency in the non-SSC mode,

f_m is the modulation frequency,

δ is the modulation amount, and

t is time.

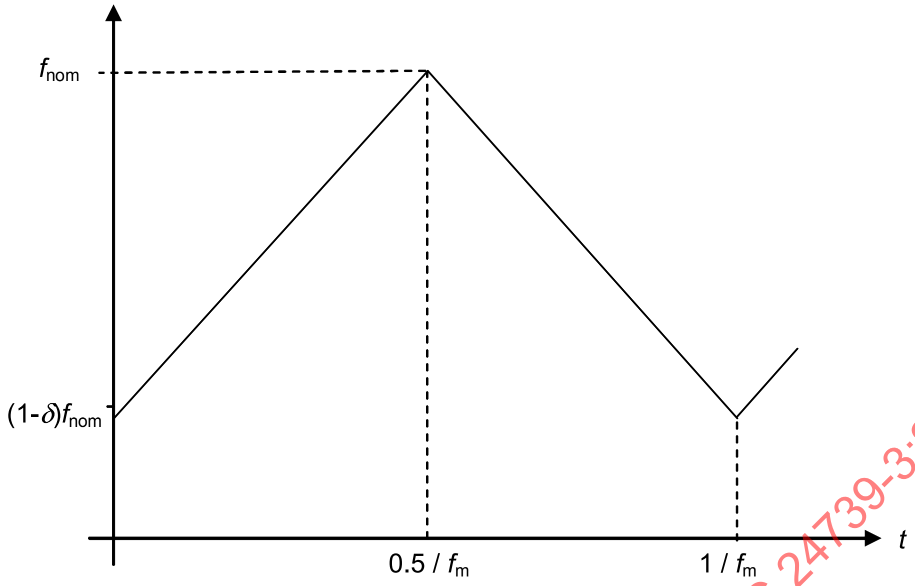


Figure 36 – Triangular frequency modulation profile

3) For triangular modulation, the clock frequency deviation (δ) is required to be no more than 0.5 % “down-spread” from the corresponding nominal frequency, i.e., +0 %/ to 0.5 %. The absolute spread amount at the fundamental frequency is shown in Figure 37, as the width of its spectral distribution (between the −3 dB roll-off). The ratio of this width to the fundamental frequency cannot exceed 0.5 %. This parameter can be measured in the frequency domain using a spectrum analyzer.

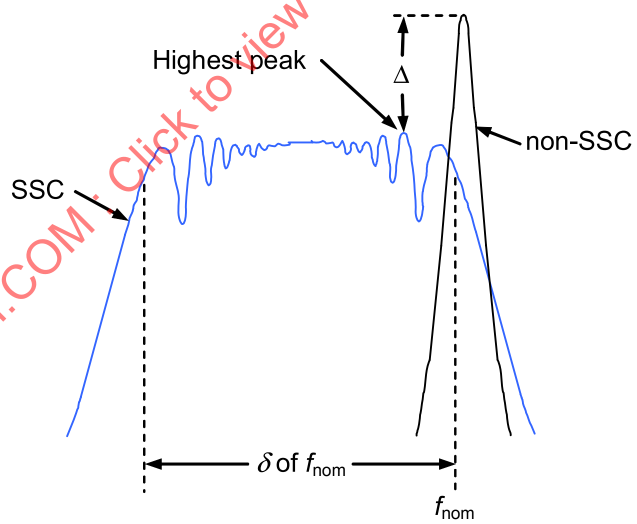


Figure 37 – Spectral fundamental frequency comparison

4) To achieved sufficient system-level EMI reduction, it is desired that SSC reduce the spectral peaks in the non-SSC mode by the amount specified in Table 14. The peak reduction Δ is defined, as shown in Figure 37, as the difference between the spectral peaks in SSC and non-SSC modes at the specified measurement frequency.

Table 14 – Desired peak amplitude reduction by SSC

Clock frequency	Peak reduction Δ	Measurement frequency
66 MHz	7 dB	466 MHz (7th harmonics)
75 MHz	7 dB	525 MHz (7th harmonics)

NOTE 1 The spectral peak reduction is not necessarily the same as the system EMI reduction. However, this relative measurement gives the component-level indication of SSC's EMI reduction capability at the system level.

NOTE 2 It is recommended that a spectrum analyzer be used for this measurement. The spectrum analyzer should have measurement capability out to 1 GHz. The measured SSC clock needs to be fed into the spectrum analyzer via a high-impedance probe compatible with the spectrum analyzer. The output clock should be loaded with 20 pF capacitance. The resolution bandwidth of the spectrum analyzer needs to be set at 120 kHz to comply with FCC EMI measurement requirements. The video band needs to be set at higher than 300 kHz for appropriate display. 100 kHz may be used as the resolution bandwidth in case of measurement equipment limitation. The display should be set with maximum hold. The corresponding harmonic peak readings should be recorded in both the non-SSC and the SSC modes, and be compared to determine the magnitude of the spectral peak reduction.

5) The modulation frequency of SSC is required to be in the range of 30 kHz to 33 kHz to avoid audio band demodulation and to minimize system timing skew.

14.5.4 Common-mode biasing

The serial interface supports both direct coupled and AC-coupled solutions, there are four scenarios to be considered when applying the DC bias to TX and RX designs. Only DC-coupled designs need sustain the specified 250 mV common-mode level to ensure interoperability.

A DC-coupled receiver (with no blocking capacitors) shall bias the common-mode level of its inputs to 250 mV. A DC-coupled transmitter shall transmit with the nominally specified 250 mV common-mode level.

An AC-coupled receiver (see Table 11) is not required to sustain the cable side of the its blocking capacitors. Similarly, an AC-coupled transmitter (see Table 11) is not required to sustain the common-mode level on the cable side of its blocking capacitors.

14.5.5 Matching

The host adapter shall provide impedance matching circuits to ensure termination for both its TX and RX as per the electrical parameters of Table 11.

Device peripherals shall provide impedance terminations, as per the specified parameters in Table 11, and may adapt their termination impedance to that of the host.

The host adapter, since it is given the first opportunity to calibrate during the power on sequence, cannot assume that the far end of the cable is calibrated yet. For this reason, the host adapter utilizes a separate reference to perform calibration. In cabled systems the cable provides the optimal impedance reference for calibration.

Using Time Domain Reflectometry (TDR) techniques, the host may launch a step waveform from its transmitter, so as to get a measure of the impedance of the transmitter, with respect to the cable and adjust its impedance settings as necessary.

In a mobile system environment, where the cable is small or non-existent, the host adapter makes use of a separate reference (such as an accurate off-chip resistor) for the calibration phase.

The device, on the other hand, may assume that the termination on the far side (host side) of the cable is fully calibrated, and may make use of this as the reference. Using the host termination as the calibration reference allows the devices operating in both the desktop and the mobile system environment to use the same hardware.

Signals generated for the impedance calibration process shall not duplicate the OOB signals, COMWAKE, COMINIT or COMRESET. Signals generated for the impedance calibration process shall not exceed the normal operating voltage levels, cited in 14.4.9. See the power management clause for suggested times to perform calibration during power-on.

14.5.6 Out of band signalling

14.5.6.1 General

There are three Out Of Band (OOB) signals used/detected by the Phy, COMRESET, COMINIT and COMWAKE. COMINIT, COMRESET and COMWAKE OOB signalling shall be achieved by transmission of a burst of ALIGN primitives each burst being 160 UI_{OOB}. Each burst is followed by idle periods (at common-mode levels), having durations as depicted in Figure 38 and Table 15.

During OOB signaling transmissions, the differential and common mode levels of the signal lines shall comply with the same electrical specifications as for in-band data transmission, specified in 14.4.9. In Figure 38 below, COMRESET, COMINIT and COMWAKE are shown. OOB signals shall be observed by detecting the temporal spacing between adjacent bursts of activity on the differential pair. It is not required for a receiver to check the duration of an OOB burst.

Any spacing less than or greater than the COMWAKE detector off threshold in Table 11 shall deassert the COMWAKE detector output. The COMWAKE OOB signalling is used to bring the Phy out of a power-down state (PARTIAL or SLUMBER) as described in 14.5.6.3.10. The interface shall be held inactive for at least the maximum COMWAKE detector off threshold in Table 11 after the last burst to ensure far-end detector detects the deassertion properly. The device shall hold the interface inactive no more than the maximum COMWAKE detector off threshold + 2 Gen1 DWORDs (approximately 228.3 ns) at the end of a COMWAKE to prevent susceptibility to crosstalk.

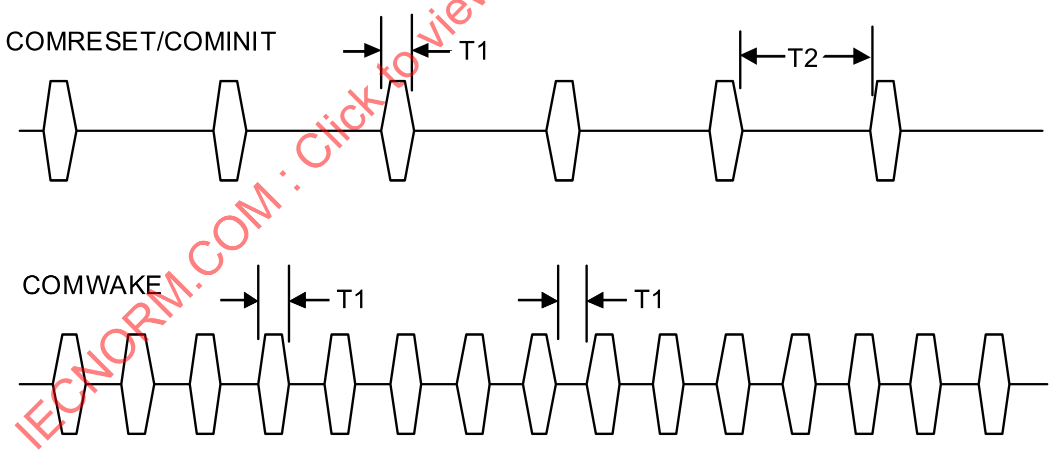


Figure 38 – Out of band signals

Table 15 – Out of band signal times

Time	Value
T1	160 UI_{oob} (106.7 ns nominal)
T2	480 UI_{oob} (320.0 ns nominal)

14.5.6.2 Idle bus status

During the idle bus condition, the differential signal diminishes to zero while the common mode level remains.

Common-mode transients shall not exceed the maximum amplitude levels ($V_{CM,AC}$) cited in 14.4.9, and shall settle to within 25 mV of V_{CM}, DC within $T_{settle,CM}$, cited in 14.4.9. Annex I shows several transmitter examples and how the transition to and from the idle state may be implemented.

14.5.6.3 Power-up and COMRESET sequences

14.5.6.3.1 General

The following state diagrams specify the behavior of the host and device PHY from power-on or COMRESET to the establishment of an active communication channel.

In those states where the Phy relies on detection of received ALIGN primitives, see 15.4.4.

14.5.6.3.2 Host power-up and COMRESET state machine

Reception of a COMINIT signal shall cause the host to reinitialize communications with the device and shall unconditionally force the Host Phy state machine to transition to the HP3:HR_AwaitNoCOMINIT state regardless of other conditions. Reception of COMINIT is effectively an additional transition into the HP3:HR_AwaitNoCOMINIT state that appears in every Host Phy state. For the sake of brevity, this implied transition has been omitted from all the states.

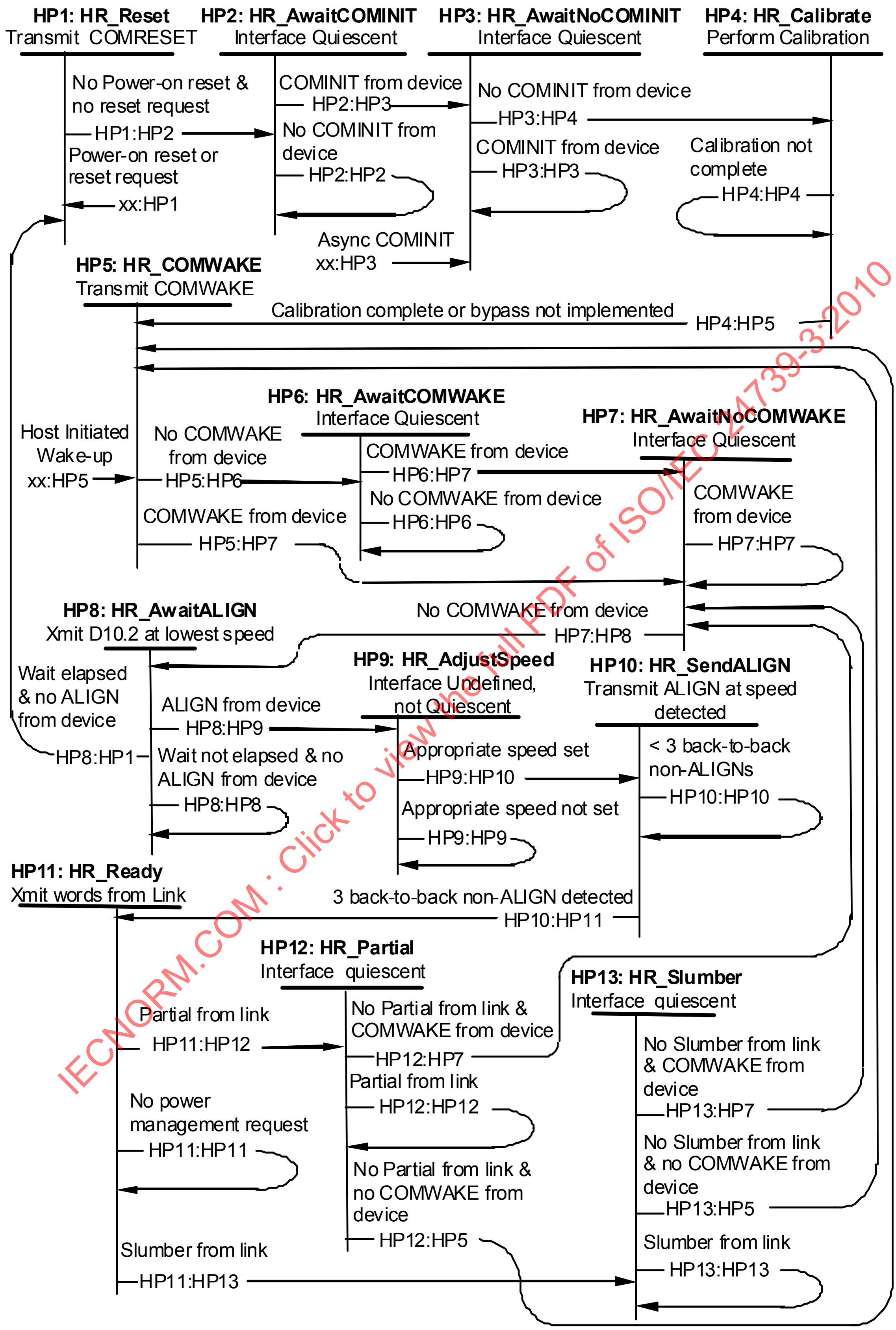


Figure 39 – Host phy initialization state machine (States HP1-HP13)

HP1: HR_Reset state: This state is entered asynchronously when power-on reset is asserted or when a host adapter reset request occurs.

The COMRESET sequence may be transmitted for the duration of this state, may be transmitted during this state and cease transmition after departure from this state, or may be transmitted upon departure from this state. The COMRESET sequence shall be transmitted for 6 bursts or a multiple of 6 bursts, see 14.5.6.3.4.1.

Transition HP1:HP2: When power-on reset is negated and no host adapter reset is requested, the Physical layer shall make a transition to the HP2: HR_AwaitCOMINIT state.

Transition xx:HP1: When power-on reset is asserted or a host adapter reset is requested, the Physical layer shall make a transition to the HP1: HR_Reset state.

HP2: HR_AwaitCOMINIT state: This state is entered when power-on reset is negated and host adapter reset is negated.

When in this state the Physical layer waits for COMINIT to be asserted.

Transition HP2:HP2: When COMINIT is not asserted, the Physical layer shall make a transition to the HP2: HR_AwaitCOMINIT state.

Transition HP2:HP3: When COMINIT is asserted, the Physical layer shall make a transition to the HP3: HR_AwaitNoCOMINIT state.

HP3: HR_AwaitNoCOMINIT state: This state is entered when the COMINIT signal is asserted while the Physical layer is in the HR2: HR_Await COMINIT state. It is also entered asychronously at any time in response to the receipt of the COMINIT sequence unless power-on reset is asserted or a host adapter reset request is asserted (in which case HP1 is entered).

When in this state the Physical layer waits for COMINIT from the device to be negated.

Transition HP3:HP3: When the COMINIT signal is asserted, the Physical layer shall make a transition to the HP3: HR_AwaitNoCOMINIT state.

Transition HP3:HP4: When the COMINIT signal negation is detected, the Physical layer shall make a transition to the HP4: HR_Calibrate state.

HP4: HR_Calibrate state: This state is entered when the COMINIT signal has been negated by the device. Calibration is optional. If calibration is bypassed or not implemented this state proceeds to state HP5.

When in this state the Physical layer performs calibration if implemented and enabled.

Transition HP4:HP4: If calibration has not completed, the Physical layer shall make a transition to the HP4: HR_Calibration state.

Transition HP4:HP5: When calibration has completed, the Physical layer shall make a transition to the HP5: HR_COMWAKE state.

HP5: HR_COMWAKE state: This state is entered when the COMINIT signal has been negated and optional calibration is not supported or when optional calibration has been completed or upon a host initiated wake-up.

When in this state the Physical layer shall transmit the COMWAKE sequence.

Transition HP5:HP6: After transmitting the COMWAKE sequence and when the COMWAKE signal is negated , the Physical layer shall make a transition to the HP6: HR_AwaitCOMWAKE state.

Transition HP5:HP7: After transmitting the COMWAKE sequence and the COMWAKE signal is asserted, the Physical layer shall make a transition to the HP7: HR_AwaitNoCOMWAKE state.

HP6: HR_AwaitCOMWAKE state: This state is entered when the COMWAKE sequence has been transmitted and the COMWAKE signal is negated.

When in this state the Physical layer shall wait for the COMWAKE signal to be asserted.

Transition HP6:HP6: If the COMWAKE signal is not asserted, the Physical layer shall make a transition to the HP6: HR_AwaitCOMWAKE state.

Transition HP6:HP7: When the COMWAKE signal is asserted, the Physical layer shall make a transition to the HP7: HR_AwaitNoCOMWAKE state.

HP7: HR_AwaitNoCOMWAKE state: This state is entered when the COMWAKE signal has been asserted.

When in this state the Physical layer shall wait for the COMWAKE signal to be negated.

Transition HP7:HP7: When the COMWAKE signal is asserted, the Physical layer shall make a transition to the HP7: HR_AwaitNoCOMWAKE state.

Transition HP7:HP8: When the COMWAKE signal is negated, the Physical layer shall make a transition to the HP8: HR_AwaitAlign state.

HP8: HR_AwaitAlign state: This state is entered when the COMWAKE signal is negated.

When in this state the Physical layer shall start transmiting D10.2 characters at the lowest rate no later than 533ns (20 nominal Gen1 DWORD Times) after COMWAKE is negated by the device (see 14.5.6).

Transition HP8:HP9: When the ALIGN primitive is detected from the device, the Physical layer shall make a transition to the HP9: HR_Adjust Speed state. Host designers should be aware that the device is allowed 53.3 ns (2 nominal Gen1 DWORD Times) after terminating the COMWAKE sequence (by holding the idle condition for more than the COMWAKE detector off threshold maximum, see Table 11) to start sending characters. Until this occurs, the bus is in idle condition and may be susceptible to crosstalk from other sources. Care should be taken so that crosstalk during this window does not result in a false detection of an ALIGN primitive. For example, a compliant host may detect the negation of the COMWAKE sequence in as little as COMWAKE detector on threshold maximum (see Table 11), such a host should wait at least 116.3 ns (the COMWAKE detector off threshold maximum, plus 2 nominal Gen1 DWORD times, plus COMWAKE detector on threshold maximum (116.3 ns = 175 + 53.3 - 112) after detecting the release of the COMWAKE signal to start looking for ALIGNs.

Transition HP8:HP1: If the ALIGN primitive is not detected from the device and 873.8 μs (32768 nominal Gen1 DWORD Times) have elapsed, the Physical layer shall make a transition to the HP1: HR_Reset state. The host shall retry the power-on sequence indefinitely unless explicitly turned off by the application layer. The host Phy state machine may use the transition to HR_Reset as a method of speed negotiation.

Transition HP8:HP8: When the COMWAKE signal is not asserted and less than 873.8 us (32 768 nominal Gen1 DWORD times) have elapsed, the Physical layer shall make a transition to the HP8: HR_AwaitAlign state.

HP9: HR_AdjustSpeed state: This state is entered when the ALIGN primitive has been detected. The interface is undefined, but not quiescent.

When in this state the Physical layer shall complete the transition to the appropriate speed. Some implementations may undergo a transient condition where invalid signals are transmitted during the change in their internal transmission/reception speed. The host may transmit invalid signals for a period of up to 53.3 ns (2 nominal Gen1 DWORD times) during the speed transition. Transmit jitter and unit interval timing requirements may not be met during this period but shall be met for all other bits transmitted in this state. A phase shift may occur across the speed transition time.

Transition HP9:HP10: When the appropriate speed transition has completed, the Physical layer shall make a transition to the HP10: HR_SendALIGN state.

Transition HP9:HP9: If the appropriate speed transition has not completed, the Physical layer shall make a transition to the HP9: HR_AdjustSpeed state.

HP10: HR_SendALIGN state: This state is entered when the appropriate speed is set.

When in this state the Physical layer shall transmit the ALIGN primitive.

Transition HP10:HP11: When three back-to-back non-ALIGN primitives have been received from the device, the Physical layer shall make a transition to the HP11: HR_Ready state. Non-ALIGN primitives can be detected by the presence of the k28.3 control character in the byte 0 position.

Transition HP10:HP10: If three back-to-back non-ALIGN primitives have not been received from the device, the Physical layer shall make a transition to the HP10: HR_SendALIGN state. The host retries indefinitely unless explicitly turned off by the application layer.

HP11: HR_Ready state: This state is entered when non-ALIGN primitives have been received from the device.

When in this state the Physical layer shall transmit the words provided by the Link layer. PHYRDY shall be asserted when in this state, and the Phy is maintaining synchronization with the incoming signal to its receiver and is transmitting a valid signal on its transmitter.

Transition HP11:HP12: When the Partial signal is asserted from the Link layer, the Physical layer shall make a transition to the HP12: HR_Partial state.

Transition HP11:HP13: When the Slumber signal is asserted from the Link layer, the Physical layer shall make a transition to the HP13: HR_Slumber state.

Transition HP11:HP11: When neither the Partial nor Slumber signal is asserted from the Link layer, the Physical layer shall make a transition to the HP11: HR_Ready state.

HP12: HR_Partial state: This state is entered when the Partial signal is asserted from the Link layer.

When in this state the Physical layer shall enter the Partial power mode state. The interface is quiescent in this state.

Transition HP12:HP5: When the Partial signal is negated from the Link layer and the COMWAKE signal is not asserted, the Physical layer shall make a transition to the HP5: HR_COMWAKE state. The host Phy shall remember if the COMWAKE signal was detected during Partial to determine if the wakeup request originated from the host or the Phy. (See state DP4 and DP6 in 14.5.6.3.3.)

Transition HP12:HP7: When the Partial signal is negated from the Link layer and the COMWAKE signal is asserted, the Physical layer shall make a transition to the HP7: HR_AwaitNoCOMWAKE state. The host Phy shall remember if the COMWAKE signal was detected during Partial to determine if the wakeup request originated from the host or the Phy.

Transition HP12:HP12: When the Partial signal is asserted from the Link layer, the Physical layer shall make a transition to the HP12: HR_Partial state.

HP13: HR_Slumber state: This state is entered when the Slumber signal is asserted from the Link layer.

When in this state the Physical layer shall enter the Slumber power mode state. The interface is quiescent in this state.

Transition HP13:HP5: When the Slumber signal is negated from the Link layer and no COMWAKE sequence is received from the device, the Physical layer shall make a transition to the HP5: HR_COMWAKE state. The host Phy shall remember if the COMWAKE signal was detected during Slumber to determine if the wakeup request originated from the host or the Phy. (See state DP4 and DP6 in 14.5.6.3.3.) The host Phy may take this transition only after it has recovered from slumber mode and the Phy is prepared to initiate communications. If the Phy has not yet recovered from the slumber mode it shall remain in this state.

Transition HP13:HP7: When the Slumber signal is negated from the Link layer and a COMWAKE sequence is received from the device, the Physical layer shall make a transition to the HP7: HR_AwaitNoCOMWAKE state. The host Phy shall remember if the COMWAKE signal was detected during Slumber to determine if the wakeup request originated from the host or the Phy. (See state DP4 and DP6 in 14.5.6.3.3.) The host Phy may take this transition only after it has recovered from slumber mode and the Phy is prepared to initiate communications. If the Phy has not yet recovered from the slumber mode it shall remain in this state.

Transition HP13:HP13: When the Slumber signal is asserted from the Link layer, the Physical layer shall make a transition to the HP13: HR_Slumber state.

14.5.6.3.3 Device power-up and COMRESET state machine

Reception of a COMRESET signal shall be treated by the device as a hard reset signal and shall unconditionally force the Device Phy state machine to transition to the DP1:DR_Reset initial state regardless of other conditions. Reception of the COMRESET signal is effectively an additional transition into the DP1:DR_Reset state that appears in every Device Phy state. For the sake of brevity, this implied transition has been omitted from all the states.

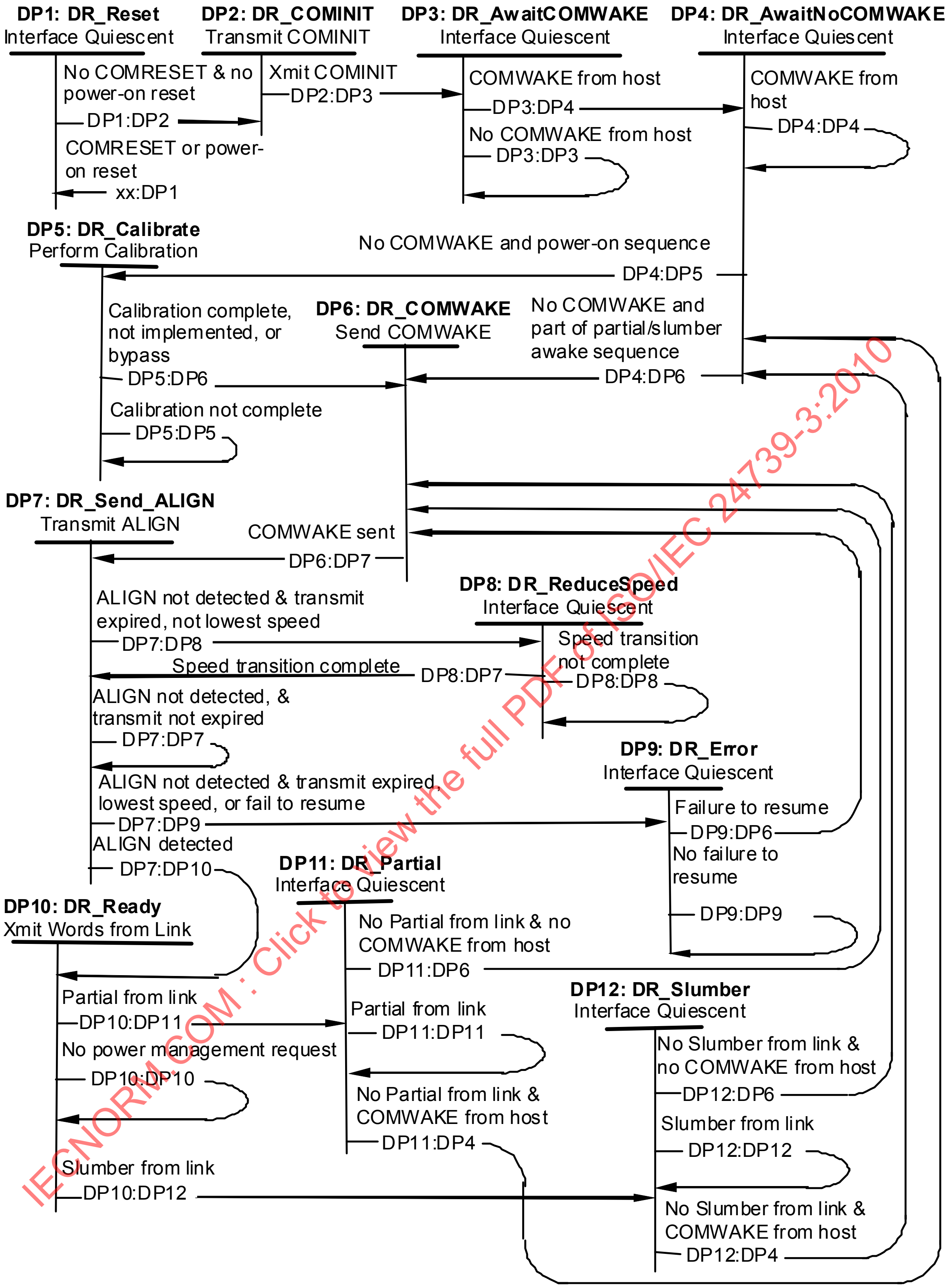


Figure 40 – Device phy initialization state machine (States DP1-DP12)

An additional transition into the DP1:DR_Reset state that appears in every state in the Device Phy Initialization state, see Figure 40. For the sake of brevity, this implied transition has been omitted from all the states.

DP1: DR_Reset state: This state is entered asynchronously at power-on or when the COMRESET sequence is detected from the host.

Transition DP1:DP2: When the COMRESET signal is not asserted and power-on reset is negated, the Physical layer shall make a transition to the DP2: HR_COMINIT state.

Transition xx:DP1: When the COMRESET signal is asserted or power-on reset is asserted, the Physical layer shall make a transition to the DP1: DR_Reset state.

DP2: DR_COMINIT state: This state is entered when the COMRESET signal is not asserted and power-on reset is negated.

When in this state the Physical layer transmits the COMINIT sequence. The COMINIT sequence shall be transmitted for a 6 burst duration. The COMINIT sequence shall be transmitted for 6 bursts or a multiple of 6 bursts.

Transition DP2:DP3: When the COMINIT sequence has been transmitted, the Physical layer shall make a transition to the DP3: DR_AwaitCOMWAKE state.

DP3: DR_AwaitCOMWAKE state: This state is entered when the COMINIT sequence has been transmitted.

When in this state the Physical layer waits for the COMWAKE signal to be asserted.

Transition DP3:DP3: If the COMWAKE signal has not been asserted, the Physical layer shall make a transition to the DP3: DR_AwaitCOMWAKE state.

Transition DP3:DP4: When the COMINIT signal has been asserted, the Physical layer shall make a transition to the DP4: HR_AwaitNoCOMWAKE state.

DP4: DR_AwaitNoCOMWAKE state: This state is entered when the COMWAKE signal has been asserted.

When in this state the Physical layer waits for the COMWAKE signal to be negated.

Transition DP4:DP4: If the COMWAKE signal is asserted, the Physical layer shall make a transition to the DP4: DR_AwaitNoCOMWAKE state.

Transition DP4:DP5: When the COMWAKE signal has been negated and the power-on reset sequence is being executed, the Physical layer shall make a transition to the DP5: Calibrate state. The device shall remember if it was sent to partial or slumber mode for proper wakeup action. Calibration is optional. If bypassed or not implemented, proceed directly to DP6: DR_COMWAKE.

Transition DP4:DP6: When the COMWAKE signal has been negated and the return from partial or slumber sequence is being executed, the Physical layer shall make a transition to the DP6: DR_COMWAKE state. The device shall remember if it was sent to partial or slumber mode for proper wakeup action.

DP5: DR_Calibrate state: This state is entered when the COMWAKE signal has been negated and optional calibration is supported.

When in this state the Physical layer shall perform calibration. Calibration is optional. If bypassed or not implemented, proceed directly to DP6: DR_COMWAKE.

Transition DP5:DP5: If calibration has not completed, the Physical layer shall make a transition to the DP5: DR_Calibration state.

Transition DP5:DP6: When calibration has completed, the Physical layer shall make a transition to the DP6: DR_COMWAKE state.

DP6: DR_COMWAKE state: This state is entered when the COMWAKE signal has been negated and optional calibration is not supported or when optional calibration has been completed.

When in this state the Physical layer shall transmit the COMWAKE sequence.

Transition DP6:DP7: After transmission of the COMWAKE sequence, the Physical layer shall make an unconditional transition to the DP7: DR_SendALIGN state.

DP7: DR_SendALIGN state: This state is entered when the COMWAKE signal has been transmitted by the Physical layer.

When in this state the Physical layer shall transmit the ALIGN primitive. The ALIGN primitive shall be transmitted at the fastest supported speed first. ALIGNs shall be transmitted only at valid frequencies (if PLL not locked, send D10.2). After the COMWAKE signal is released as specified in 14.5.6, the device shall ensure the interface is active (transmitting D10.2 if PLL not locked, or ALIGNs, not quiescent). The device shall not leave the bus idle for more than 53.3 ns (2 nominal Gen1 DWORD Times) longer than the required COMWAKE detector off threshold maximum (see Table 11) to negate COMWAKE.

Transition DP7:DP7: If the ALIGN primitive is not detected from the host and ALIGN primitives have been transmitted by the Phy for less than 54.6 µs (2048 nominal Gen1 ALIGN primitives) the Physical layer shall make a transition to the DP7: DR_SendALIGN state.

Transition DP7:DP8: If the ALIGN primitive is not detected from the host and ALIGN primitives have been transmitted by the Phy for 54.6 µs (2048 nominal Gen1 ALIGN primitives) at speed other than the lowest, the Physical layer shall make a transition to the DP8: DR_ReduceSpeed state. The device shall not leave the bus idle for more than 53.3 ns (2 nominal Gen1 DWORD Times) longer than the required 175 ns to negate COMWAKE. If this is part of device initiated recovery from the Slumber or Partial power management state, the device Phy shall resume at the speed previously negotiated and shall not reduce its speed in response to failure to establish communications (see transition DP7:DP9).

Transition DP7:DP9: If the ALIGN primitive is not detected from the host and ALIGN primitives have been transmitted by the Phy for 54.6 µs (2048 nominal Gen1 ALIGN primitives) at the lowest speed, the Physical layer shall make a transition to the DP9: DR_Error state. The device shall not leave the bus idle for more than 53.3 ns (2 nominal Gen1 DWORD Times) longer than the required COMWAKE detector off threshold maximum (see Table 11) to negate COMWAKE. If this is part of device initiated recovery from the Slumber or Partial power management state, the device Phy shall resume at the speed previously negotiated and shall not reduce its speed in response to failure to establish communications.

Transition DP7:DP10: When the ALIGN primitive is detected from the host (device locked to incoming data), the Physical layer shall make a transition to the DP10: DR_Ready state. Device designers should be aware that the host is allowed 533 ns (20 nominal Gen1 DWORD Times) after detecting the negation of COMWAKE to start sending D10.2 characters. Until this occurs, the bus is in idle condition and may be susceptible to crosstalk from other sources. Care should be taken so that crosstalk during this window does not result in a false detection of an ALIGN primitive. Devices may extend this timeout up to an additional 54.6 µs (2048 nominal Gen1 DWORD Times), for a maximum total of 109.2 µs, as necessary to allow their receiver time to lock to the host ALIGN.

DP8: DR_ReduceSpeed state: This state is entered when the ALIGN primitive is not detected from the host and ALIGN primitives have been transmitted by the Phy for 54.6 µs (2 048 nominal Gen1 ALIGN primitives) at speed other than the lowest.

When in this state the Physical layer shall transition to a slower speed.

Transition DP8:DP7: When the appropriate speed transition has completed, the Physical layer shall make a transition to the DP7: DR_SendALIGN state. Transition to a new speed is defined as being complete when the device is accurately transmitting a valid signal within the defined signalling tolerances for that speed.

Transition DP8:DP8: If the appropriate speed transition has not completed, the Physical layer shall make a transition to the DP8: DR_ReduceSpeed state.

DP9: DR_Error state: This state is entered when the ALIGN primitive is not detected from the host and ALIGN primitives have been transmitted by the Phy for 54.6 µs (2 048 nominal Gen1 ALIGN primitives) at the lowest speed, or the previous negotiated speed failed to resume from a power state.

Transition DP9:DP9: When the error is not due to failure to resume, the Physical layer shall make a transition to the DP9: DR_Error state.

Transition DP9:DP6: If the error is due to failure to resume, the Physical layer shall make a transition to the DP6: DR_COMWAKE state.

DP10: DR_Ready state: This state is entered when ALIGN primitives have been received from the host.

When in this state the Physical layer shall transmit the words provided by the Link layer. PHYRDY shall be asserted when in this state and the Phy is maintaining synchronization with the incoming signal to its receiver and is transmitting a valid signal on its transmitter.

Transition DP10:DP11: When the Partial signal is asserted from the Link layer, the Physical layer shall make a transition to the DP11: DR_Partial state.

Transition DP10:DP12: When the Slumber signal is asserted from the Link layer, the Physical layer shall make a transition to the DP12: DR_Slumber state.

Transition DP10:DP10: When neither the Partial nor Slumber signal is asserted from the Link layer, the Physical layer shall make a transition to the DP10: DR_Ready state.

DP11: DR_Partial state: This state is entered when the Partial signal is asserted from the Link layer.

When in this state the Physical layer shall enter the Partial power mode state.

Transition DP11:DP6: When the Partial signal is negated from the device Link layer to initiate a resume from partial, and no COMWAKE is detected from the host, the Physical layer shall make a transition to the DP6: DR_COMWAKE state.

Transition DP11:DP4: When the Partial signal is negated from the device Link layer to initiate a resume from Partial and COMWAKE is detected from the host to initiate a resume from Partial, the Physical layer shall make a transition to the DP4: DR_AwaitNoCOMWAKE state.

Transition DP11:DP11: When the Partial signal is asserted from the Link layer, the Physical layer shall make a transition to the DP11: DR_Partial state.

DP12: DR_Slumber state: This state is entered when the Slumber signal is detected from the Link layer.

When in this state the Physical layer shall enter the Slumber power mode state.

Transition DP12:DP6: When the Slumber signal is negated from the device Link layer to initiate a resume from Partial and no COMWAKE signal is asserted, the Physical layer shall make a transition to the DP6: DR_COMWAKE state.

Transition DP12:DP4: When the Slumber signal is negated from the device Link layer to initiate a resume from Partial and COMWAKE is asserted, the Physical layer shall make a transition to the DP4: DR_AwaitNoCOMWAKE state.

Transition DP12:DP12: When the Slumber signal is asserted from the Link layer, the Physical layer shall make a transition to the DP12: DR_Slumber state.

14.5.6.3.4 Power-up and COMRESET timing

14.5.6.3.4.1 COMRESET

COMRESET always originates from the host controller, and forces a hard reset in the device. It is indicated by transmitting bursts (see 14.5.6) separated by an idle bus condition.

The OOB COMRESET signal shall consist of no less than six bursts, and a multiple of six bursts, including inter-burst temporal spacing. The COMRESET signal shall be

1) sustained/continued uninterrupted as long as the hard reset is asserted or
2) started during the system hard reset and ended some time after the deassertion of system hard reset or
3) transmitted immediately following the deassertion of the system hard reset signal.

The host controller shall ignore any signal received from the device from the assertion of the hard reset signal until the COMRESET signal is transmitted.

Each burst shall be 160 Gen1 UI_{OOB}'s long (approximately 106.7 ns) and each inter-burst idle state shall be 480 Gen1 UI_{OOB}'s long (approximately 320 ns). A COMRESET detector will look for four consecutive bursts with 320 ns spacing (nominal).

Any spacing less than the COMRESET/COMINIT detector off threshold minimum time (see Table 11) or greater than the COMRESET/COMINIT detector off threshold maximum time shall negate the COMRESET detector output. The COMRESET interface signal to the Phy layer will initiate the Reset sequence shown in Figure 41 below. The interface shall be held inactive for at least the COMRESET/COMINIT detector off threshold maximum time after the last burst to ensure far-end detector detects the deassertion properly.

Figure 41 is provided for clarity and is informative.

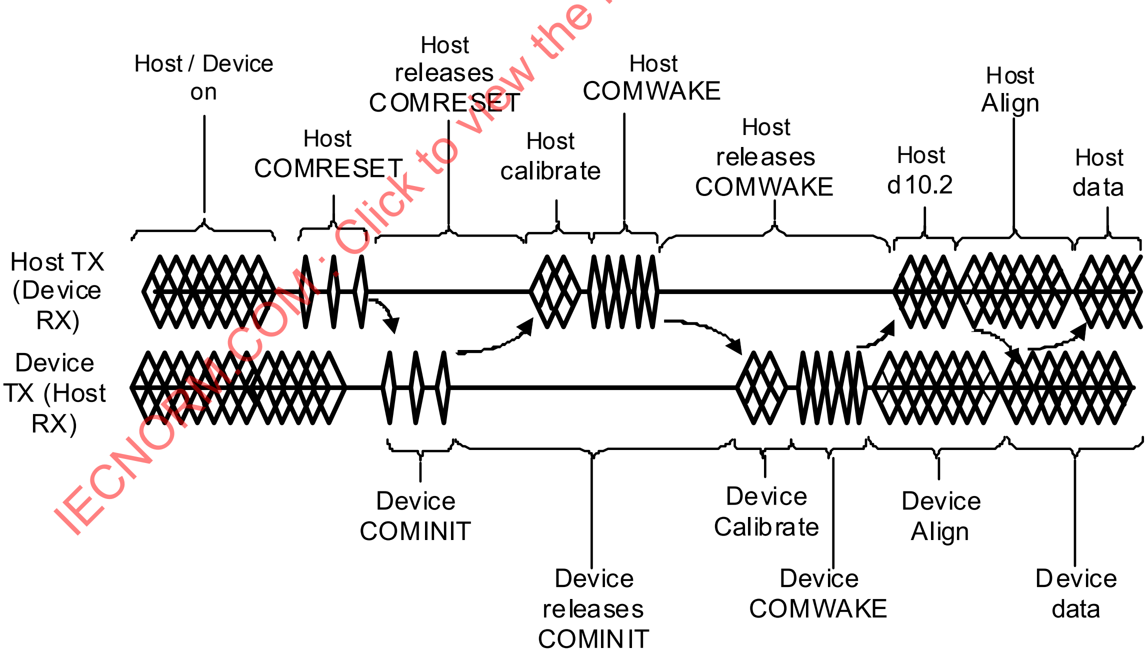


Figure 41 – COMRESET sequence

The following list describes the COMRESET sequence diagram.

a) Host/device are powered and operating normally with some form of active communication.
b) Some condition in the host causes the host to issue COMRESET sequence.
c) Once the condition causing the COMRESET sequence is released, the host signal puts the bus in a quiescent condition.
d) When the device detects the release of COMRESET sequence, it responds with a COMINIT sequence. The device may initiate communications at any time by issuing a COMINIT sequence.
e) Host calibrates and issues a COMWAKE sequence.
f) Device responds - The device detects the COMWAKE sequence on its RX pair and calibrates its transmitter (optional). Following calibration, the device sends a six burst COMWAKE sequence and then sends a continuous stream of the ALIGN sequence starting at the device's highest supported speed. After ALIGN DWORDs have been sent for 54.6 µs (2048 nominal Gen1 DWORD times) without a response from the host as determined by detection of ALIGN primitives received from the host, the device assumes that the host cannot communicate at that speed. If additional speeds are available, the device tries the next lower supported speed by sending ALIGN DWORDs at that rate for 54.6 µs (2 048 nominal Gen1 DWORD times). This step is repeated for as many legacy speeds as are supported. Once the lowest speed has been reached without response from the host, the device will enter an error state.
g) Host locks - After detecting the COMWAKE sequence, the host starts transmitting D10.2 characters (see 14.6) at its lowest supported rate. Meanwhile, the host receiver locks to the ALIGN sequence and, when ready, returns the ALIGN sequence to the device at the same speed as received. A host shall be designed such that it can acquire lock in 54.6 µs (2 048 nominal Gen1 DWORD DWORD times) at any given speed. The host should allow for at least 873.8 µs (32 768 nominal Gen1 DWORD times) after detecting the release of COMWAKE to receive the first ALIGN. This will ensure interoperability with multi-generational and synchronous devices. If no is ALIGN is received within 873.8 µs (32 768 nominal Gen1 DWORD times), the host restarts the power-on sequence - repeating indefinitely until told to stop by the application layer.
h) Device locks - The device locks to the ALIGN sequence and, when ready, sends the SYNC primitive indicating it is ready to start normal operation.
i) Upon receipt of three back-to-back non-ALIGN primitives, the communication link is established and normal operation may begin.

14.5.6.3.5 COMINIT sequence

The COMINIT sequence, see Figure 42, originates from the drive and requests a communication initialization. It is executed by entering the Device phy initialization state diagram (see Figure 40) at the DP2: DR_COMINIT state. The following timing diagram is provided for clarity and is informative.

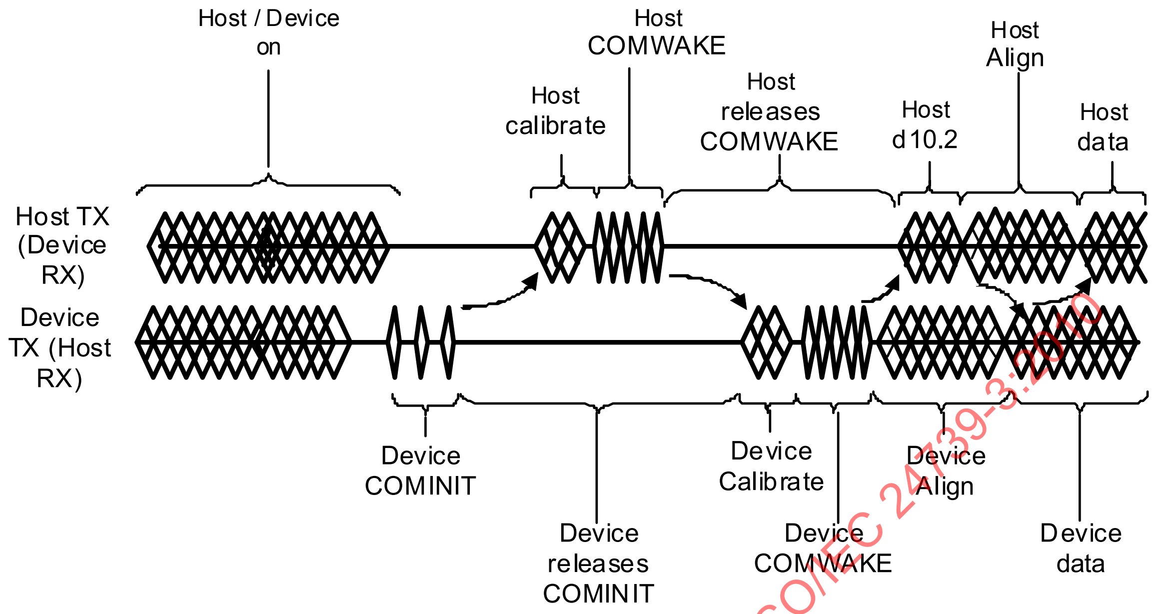


Figure 42 – COMINIT sequence

The following list describes the COMINIT sequence diagram.

a) Host/device are powered and operating normally with some form of active communication.

b) Some condition in the device causes the device to issues a COMINIT

c) Host calibrates and issues a COMWAKE.

d) Device responds - The device detects the COMWAKE sequence on its RX pair and calibrates its transmitter (optional). Following calibration, the device sends a six burst COMWAKE sequence and then sends a continuous stream of the ALIGN sequence starting at the device's highest supported speed. After ALIGN DWORDs have been sent for 54.6 µs (2 048 nominal Gen1 DWORD times) without a response from the host as determined by detection of ALIGN primitives received from the host, the device assumes that the host cannot communicate at that speed. If additional speeds are available, the device tries the next lower supported speed by sending ALIGN DWORDs at that rate for 54.6 µs (2 048 nominal Gen1 DWORD times). This step is repeated for as many legacy speeds as are supported. Once the lowest speed has been reached without response from the host, the device will enter an error state.

e) Host locks - After detecting the COMWAKE sequence, the host starts transmitting D10.2 characters (see 14.6) at its lowest supported rate. Meanwhile, the host receiver locks to the ALIGN sequence and, when ready, returns the ALIGN sequence to the device at the same speed as received. A host shall be designed such that it can acquire lock in 54.6 µs (2 048 nominal Gen1 DWORD times) at any given speed. The host should allow for at least 873.8 µs (32 768 nominal Gen1 DWORD times) after detecting the release of COMWAKE to receive the first ALIGN. This will ensure interoperability with multi-generational and synchronous devices. If no is ALIGN is received within 873.8 µs (32 768 nominal Gen1 DWORD times), the host restarts the power-on sequence - repeating indefinitely until told to stop by the application layer.

f) Device locks - The device locks to the ALIGN sequence and, when ready, sends the SYNC primitive indicating it is ready to start normal operation.

g) Upon receipt of three back-to-back non-ALIGN primitives, the communication link is established and normal operation may begin.

14.5.6.3.6 COMWAKE

COMWAKE can originate from either the host controller or the device. It is signalled by transmitting six bursts separated by an idle bus condition.

The OOB COMWAKE signaling shall consist of no less than six bursts, including inter-burst temporal spacing.

Each burst shall be 160 Gen1 UI_{OOB}'s (approximately 106.7 ns) long and each inter-burst idle state shall be 160 Gen1 UI_{OOB}'s (approximately 106.7 ns) long. A COMWAKE detector will look for four consecutive bursts with a COMWAKE transmit spacing time (approximately 106.7 ns nominal, see Table 11).

Any spacing less than COMWAKE detector off threshold min or greater than COMWAKE detector threshold max (see Table 11) shall negate the COMWAKE detector output. The COMWAKE OOB signaling is used to bring the Phy out of a power-down state (PARTIAL or SLUMBER) as described in 14.5.6.3.7. The interface shall be held inactive for at least the COMWAKE detector off threshold maximum time (see Table 11). after the last burst to ensure far-end detector detects the deassertion properly. The device may hold the interface inactive no more than 228.3 ns (the COMWAKE detector threshold maximum + 2 nominal Gen1 DWORD times) at the end of a COMWAKE to prevent susceptibility to crosstalk.

14.5.6.3.7 Interface power states

14.5.6.3.7.1 General

In the serial implementation of ATA, Interface Power States are controlled by the device and host adapter as defined in the Host phy initialization and Device phy initialization state diagrams, see Figure 39 and Figure 40, and the Link Power Mode State Diagram, see Figure 61. The Serial Interface Power States are defined in Table 16.

Table 16 – Interface power states

READY	The PHY logic and main PLL are both on and active. The interface is synchronized and capable of receiving and sending data.
Partial	The PHY logic is powered, but is in a reduced power state. Both signal lines on the interface are at a neutral logic state (common mode voltage). The exit latency from this state shall be no longer than 10 µs.
Slumber	The PHY logic is powered but is in a reduced power state. Both signal lines on the interface are at the neutral logic state (common mode voltage). The exit latency from this state shall be no longer than 10 ms.

14.5.6.3.7.2 Idle bus condition

During power management states (Partial and Slumber), the electrical interface shall maintain the proper common-mode levels, as cited in 14.4.9, with zero differential on both signal pairs (all four conductors at $V_{CM,DC}$) for all interface scenarios, except for the case where both the device and the host-controller are AC-coupled and the conductor pairs are allowed to float.

All transmitter designs shall ensure that transition to and from the idle bus condition do not result in a disturbance in the differential baseline on the conductors. To accomplish this, an AC-coupled transmitter shall hold its outputs at zero differential with the same common-mode level as normal operation when in the partial power management mode. When operating in the slumber power management mode, the common mode level of the AC coupled transmitter is allowed to float (while maintaining zero differential) as long as it remains within the limits cited in 14.4.9.

TX outputs shall not be held at a logical zero or one state during the idle bus condition since this results in a baseline shift when communications are resumed.

14.5.6.3.8 Power-on sequence timing diagram

The following timing diagrams, see Figure 43, Figure 44 and Figure 45 as well as descriptions are provided for clarity and are informative. The state diagrams provided in 14.5.6.3 comprise the normative behavior specification.

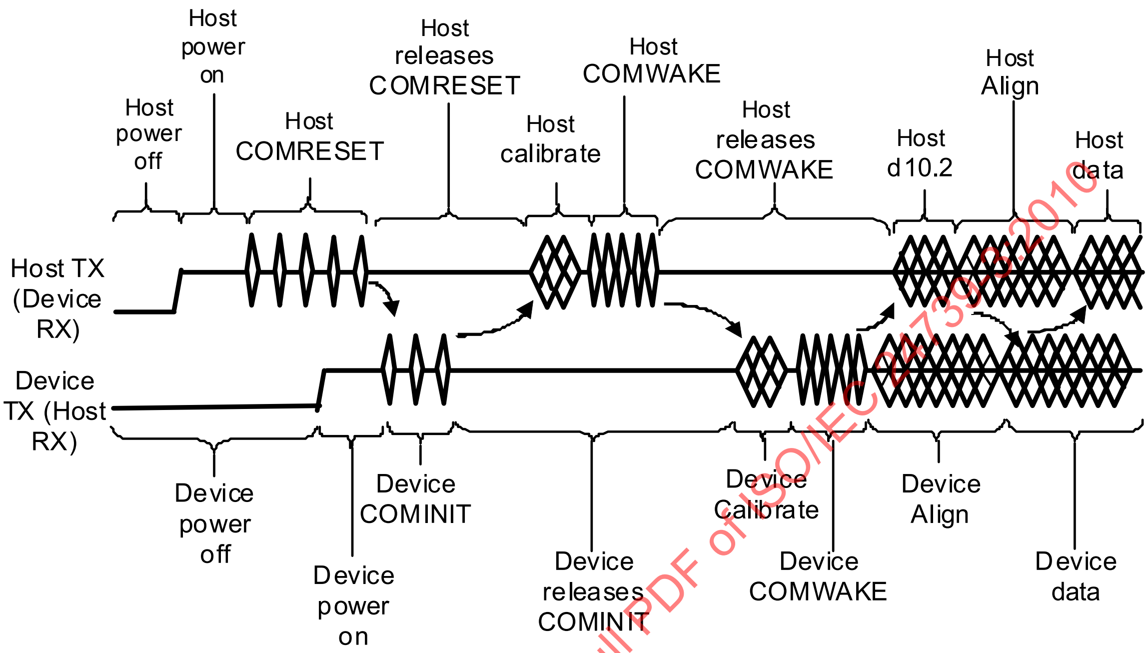


Figure 43 – Power-on sequence

The following list describes the power-on sequence diagram.

a) Host/device power-off = Host and device power-off.

b) Power is applied = Host side signal conditioning pulls TX and RX pairs to neutral state (common mode voltage).

c) Host issues COMRESET sequence.

d) Once the power-on reset is released, the host puts the bus in a quiescent condition.

e) When the device detects the release of COMRESET sequence, it responds with a COMINIT sequence. This is also the entry point if the device is late starting. The device may initiate communications at any time by issuing a COMINIT.

f) Host calibrates and issues a COMWAKE sequence.

g) Device responds - The device detects the COMWAKE sequence on its RX pair and calibrates its transmitter (optional). Following calibration, the device sends a six burst COMWAKE sequence and then sends a continuous stream of the ALIGN sequence starting at the device's highest supported speed. After ALIGN DWORDs have been sent for 54.6 µs (2 048 nominal Gen1 DWORD times) without a response from the host as determined by detection of ALIGN primitives received from the host, the device assumes that the host cannot communicate at that speed. If additional speeds are available, the device tries the next lower supported speed by sending ALIGN DWORDs at that rate for 54.6 µs (2 048 nominal Gen1 DWORD times). This step is repeated for as many legacy speeds as are supported. Once the lowest speed has been reached without response from the host, the device will enter an error state.

h) Host locks - After detecting the COMWAKE sequence, the host starts transmitting D10.2 characters (see 14.6) at its lowest supported rate. Meanwhile, the host receiver locks to the ALIGN sequence and, when ready, returns the ALIGN sequence to the device at the same speed as received. A host shall be designed such that it can acquire lock in 54.6 µs (2 048 nominal Gen1 DWORD times) at any given speed. The host should allow for at least 873.8 µs (32 768 Gen1 DWORD times) after detecting the release of COMWAKE to receive the first ALIGN. This will ensure interoperability with multi-generational and synchronous devices. If no ALIGN is received within 873.8 µs (32 768 nominal Gen1 DWORD times), the host restarts the power-on sequence, repeating indefinitely until told to stop by the application layer.

i) Device locks - The device locks to the ALIGN sequence and, when ready, sends the SYNC primitive indicating it is ready to start normal operation.

j) Upon receipt of three back-to-back non-ALIGN primitives, the communication link is established and normal operation may begin.

14.5.6.3.9 READY to Partial/Slumber

READY to Partial/Slumber is initiated by the transmission of the PMREQ_P or PMREQ_S and PMACK primitives as described in 15.4.9.

14.5.6.3.10 Partial/Slumber to READY

The host initiates a wakeup from the partial or slumber states by entering the initialization sequence at the HP5: HR_COMWAKE state in the Host phy initializationstate machine (see Figure 39). Calibration and speed negotiation is bypassed since it has already been performed at power-on and system performance depends on quick resume latency. The device, therefore, transmits ALIGNs at the speed determined at power-on.

The device initiates a wakeup from the partial or slumber states by entering the initialization sequence at the DP6: DR_COMWAKE state in the Device phy initializationstate machine (see Figure 40). Calibration and speed negotiation is bypassed since it has already been performed at power-on and system performance depends on quick resume latency. The device, therefore, transmits ALIGNs at the speed determined at power-on.

14.5.6.4 ON to Partial/Slumber

14.5.6.4.1 Host initiated

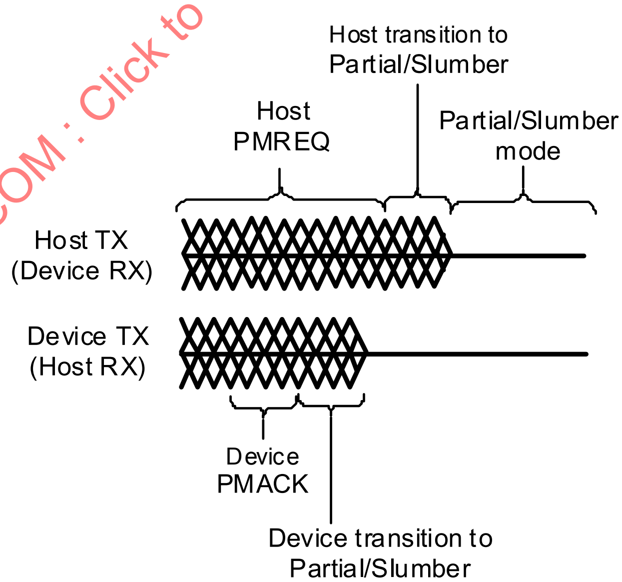


Figure 44 – ON to Partial/Slumber, host initiated

14.5.6.4.2 Detailed sequence

The following list describes the On to Partial/Slumber, host initiated diagram.

a) Host Application layer sends request to host Transport layer.
b) Host Transport layer transmits request to host Link layer.
c) Host Link layer encodes request as PMREQ primitive and transmits it four times to host Phy layer.
d) Host Phy layer serializes PMREQ primitives and transmits them to device Phy layer.
e) Device Phy de-serializes PMREQ primitives and transmits them to device Link layer.
f) Device Link layer decodes PMREQ primitives and transmits request to device Transport layer.
g) Device Transport layer transmits request to device Application layer.
h) Device Application layer processes and accepts request. Issues accept to device Transport layer.
i) Device Transport layer transmits acceptance to device Link layer.
j) Device Link layer encodes acceptance as PMACK primitive and transmits it four times to device Phy layer.
k) Device Phy layer transmits four PMACK primitives to host Phy layer.
l) Device Link layer places device Phy layer in Partial/Slumber state.
m) Host Phy layer de-serializes PMACK primitives and transmits them to host Link layer.
n) Host Link layer decodes PMACK primitives and transmits acceptance to host Transport layer.
o) Host Link layer places host Phy layer in Partial/Slumber State.
p) Host Transport layer transmits acceptance to host Application layer.

14.5.6.4.3 Device initiated

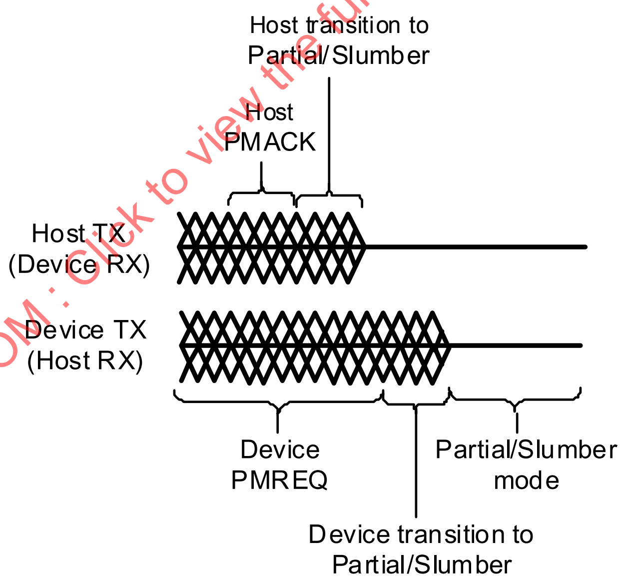


Figure 45 – ON to Partial/Slumber, device initiated

14.5.6.4.4 Detailed sequence

The following list describes the On to Partial/Slumber, device initiated diagram.

a) Device Application layer sends request to device Transport layer.
b) Device Transport layer transmits request to device Link layer.
c) Device Link layer encodes request as PMREQ primitive and transmits it to device Phy layer.
d) Device Phy layer serializes PMREQ primitives and transmits them to host Phy layer.
e) Host Phy de-serializes PMREQ primitives and transmits them to host Link layer.
f) Host Link layer decodes PMREQ primitives and transmits request to host Transport layer.

g) Host Transport layer transmits request to host Application layer.

NOTE In this context, the host Application layer does not necessarily imply BIOS or other host CPU programming. Rather, the Application layer is the intelligent control section of the chipset logic.

h) Host Application layer processes and accepts request. Issues accept to host Transport layer.
i) Host Transport layer transmits acceptance to host Link layer.
j) Host link layer encodes acceptance as PMACK primitive and transmits it four times to host Phy layer.
k) Host Phy layer transmits four PMACK primitives to device Phy layer.
l) Host Link layer asserts Partial/Slumber signal and places host Phy layer in Partial/Slumber state.
m) Host Phy layer negates Ready signal.
n) Device Phy layer de-serializes PMACK primitives and transmits them to device Link layer.
o) Device Link layer decodes PMACK primitives and transmits acceptance to device Transport layer.
p) Device Link layer asserts Partial/Slumber signal and places device Phy layer in Partial/Slumber State.
q) Device Phy layer negates Ready signal.
r) Device Transport layer transmits acceptance to device Application layer.

14.6 Elasticity buffer management

Elasticity buffer circuitry may be required to absorb the slight differences in frequencies between the host and device. The greatest frequency difference result from a SSC compliant device talking to a non-SSC device. The average frequency difference will be just over 0.25 % with excursions as much as 0.5 %.

Since an elasticity buffer has a finite length, there needs to be a mechanism at the physical layer protocol level that allows this receiver buffer to be reset without dropping or adding any bits to the data stream. This is especially important during reception of long continuous streams of data. This physical layer protocol supports oversampling architectures and accommodates unlimited frame sizes (the frame size is limited by the CRC polynomial).

The Link Layer shall keep track of a resettable counter that rolls over at most every 1 024 transmitted characters (256 DWORDs). Prior to, or at the pre-roll-over point (all 1's), the Link Layer shall trigger the issuance of dual, consecutive ALIGN primitives which shall be included in the DWORD count.

After communications have been established, the first and second words out of the Link Layer shall be the dual-ALIGN primitive sequence, followed by at most 254 non-ALIGN DWORDs. The cycle repeats starting with another dual-consecutive ALIGN primitive sequence. The Link may issue more than one dual ALIGN primitive sequence but shall not send an unpaired ALIGN primitive (i.e. ALIGN primitives are always sent in pairs) except as noted for retimed loopback.

14.7 BIST (Built in self test)

14.7.1 General

BIST provides loopback testing of portions of the physical layer. BIST is initiated by the BIST Activate FIS (see 16.2.8).

14.7.2 Loopback testing

14.7.2.1 General

Three types of Loopback test schemes are defined.

1) Far-End Retimed - Mandatory
2) Far-End Analog - Optional
3) Near-End Analog (Effectively Retimed) - Optional

Loopback - Far-end retimed

Figure 46 below illustrates the scope, at the architectural block diagram level, of the Far-End Retimed loopback. As this loopback scheme needs a specific action from the far-end connected interface, this mode shall be entered by way of the BIST Activate FIS described in 16.2.8.

The Far-End Interface shall remain in this Far-End Retimed Loopback, until receipt of the COMRESET/COMINIT OOB Signaling sequence.

As a minimum, Far-End Retimed Loopback shall involve far-end circuitry such that the datastream, at the Far-End interface, is extracted by the Serializer/Deserializer (SerDes) and data recovery circuit (DRC) before being sent back through the SerDes and Transmitter with appropriately inserted retiming ALIGN primitives. The data may be decoded and descrambled in order to provide testing coverage for those portions of the host/device, provided the data is re-scrambled using the same sequence of scrambler syndromes. The returned data shall be the same as the received data with the exception that the returned data may be encoded with different starting running disparity.

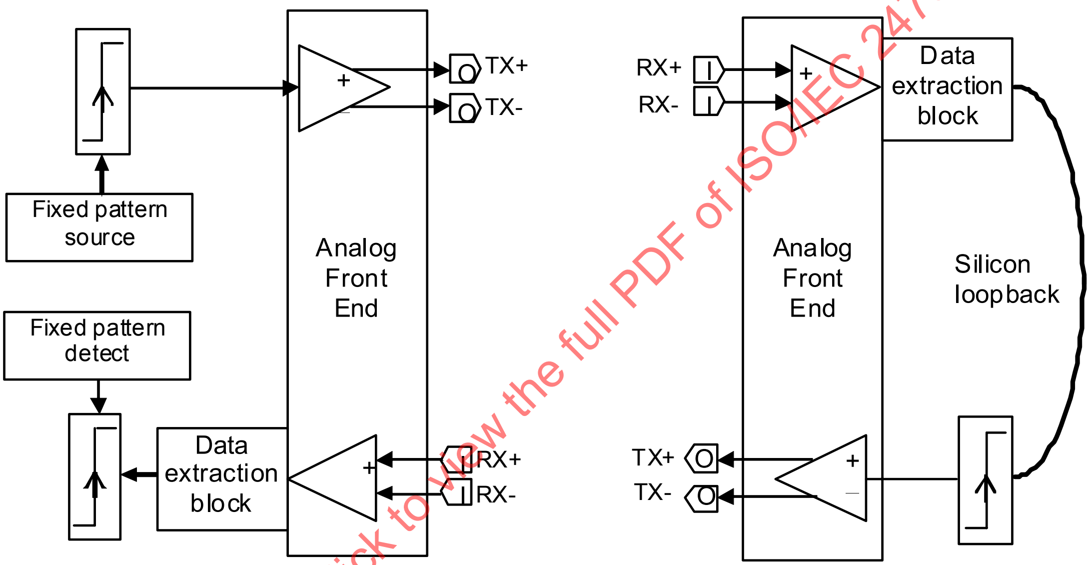


Figure 46 – Loopback, far-end retimed

The initiator of the retimed loopback mode shall account for the loopback host/device consuming up to two ALIGN primitives (one ALIGN sequence) every 256 DWORDs transmitted and, if it requires any ALIGN primitives to be present in the returned data stream, it shall insert additional ALIGNs in the transmitted stream. The initiator shall transmit additional ALIGN sequences in a single burst at the normal interval of every 256 DWORDs transmitted (as opposed to inserting ALIGN sequences at half N primitives from the received data. It may insert one or more ALIGN primitives if they are directly preceded or followed by the initiator inserted ALIGN primitives (resulting in ALIGN sequences consisting of at least two ALIGN primitives) or it may insert two or more ALIGN primitives if not preceded or followed by the initiator's ALIGN primitives. One side effect of the loopback retiming is that the returned data stream may have instances of an odd number of ALIGN primitives, however, returned ALIGNs are always in bursts of two and if the initiator transmitted dual ALIGN sequences (four consecutive ALIGNs), then the returned data stream shall include ALIGN bursts that are no shorter than two ALIGN primitives long (although the length of the ALIGN burst may be odd). The initiator of the retimed loopback mode shall not assume any relationship between the relative position of the ALIGNs returned by the loopback host/device and the relative position of the ALIGNs sent by the inithe interval).

The loopback host/device may remove zero, one or two ALIGtiator.

In retimed loopback mode, the initiator shall transmit only valid 8b/10b characters so the loopback host/device may 10b/8b decode it and re-encode it before retransmission. If the loopback host/device descrambles incoming data, it is responsible for rescrambling it with the same sequence of scrambling syndromes in order to ensure the returned data is unchanged from the received data. The loopback host/device's running disparity for its transmitter and receiver are not guaranteed to be the same and thus the loopback initiator shall 10b/8b decode the returned data rather than use the raw 10b returned stream for the purpose of data comparison. The loopback host/device shall return all received data unaltered and shall disregard protocol processing of primitives. Only the OOB signals and ALIGN processing is acted on by the loopback host/device, while all other data is retransmitted without interpretation.

14.7.2.2 Loopback, far-end analog (optional)

14.7.2.2.1 General

Figure 47 below, illustrates the scope, at the architectural block diagram level, of the Far-End Analog loopback. As this loopback scheme needs a specific action from the far-end connected interface, this mode, if implemented, shall be entered by way of the BIST Activate FIS described in 16.2.8.

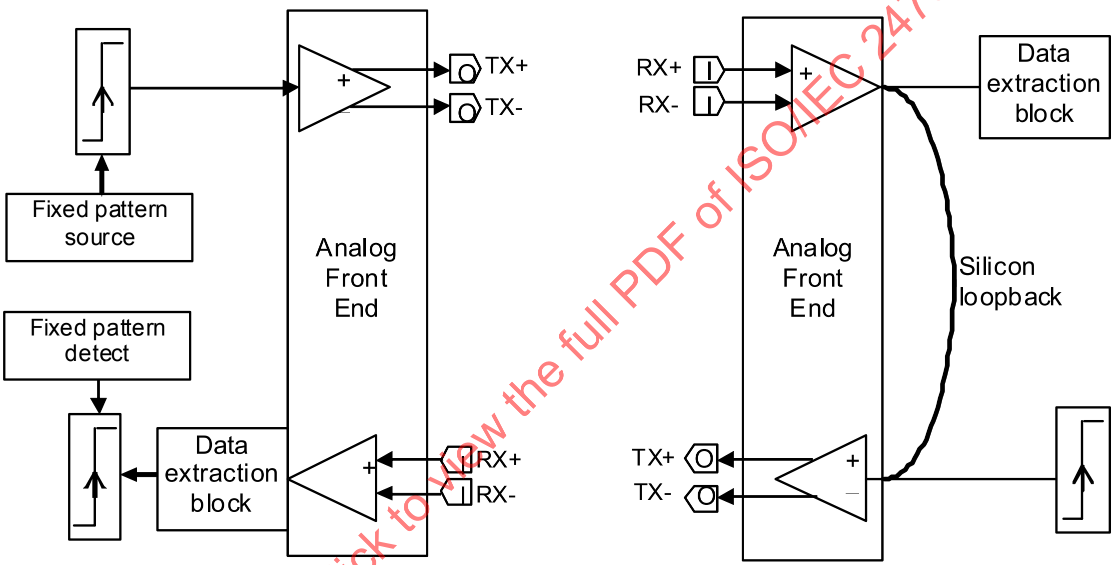


Figure 47 – Loopback, far-end analog

Once entered, the Far-End Interface shall remain in this Far-End Analog Loopback mode, until receipt of the COMRESET/COMINIT OOB Signalling sequence.

The implementation of Far End AFE Loopback is optional due to the round-trip characteristics of the test as well as the lack of retiming. This mode is intended to give a quick indication of connectivity, and test failure is not an indication of system failure.

14.7.2.2.2 Loopback, near-end analog (optional)

Figure 48 below, illustrates the scope, at the architectural block diagram level, of the Near-End Analog loopback. This loopback scheme, if implemented, needs the far-end connected interface to be in a non-transmitting mode, such as Slumber or Partial interface power management states.

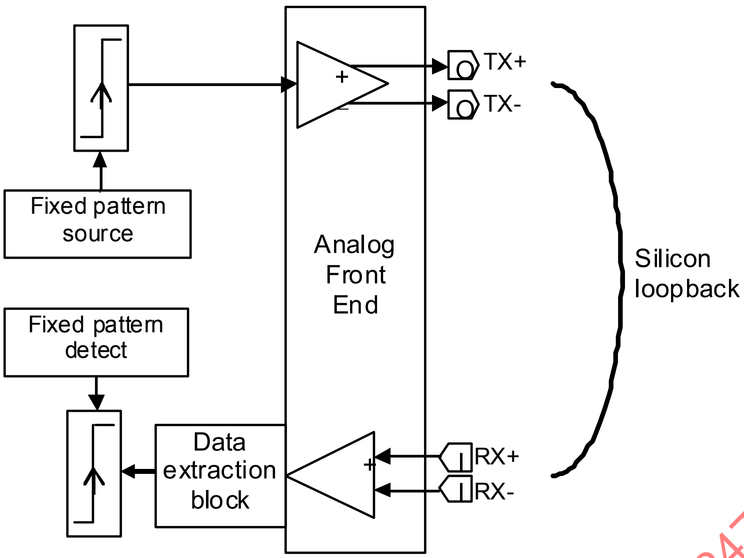


Figure 48 – Loopback, near-end analog

15 Serial interface Link layer

15.1 Overview

15.1.1 General

The Link layer transmits and receives frames, transmits primitives based on control signals from the Transport layer, and receives primitives from the Physical layer which are converted to control signals to the Transport layer. The Link layer need not be cognizant of the content of frames. Host and device Link layer state machines differ only in the fact that the host shall back-off in the event of a collision when attempting to transmit a frame (see Figure 59).

15.1.2 Frame transmission

When requested by the Transport layer to transmit a frame, the Link layer provides the following services.

- Negotiates with its peer Link layer to transmit a frame, resolves arbitration conflicts if both host and device request transmission.
- Inserts frame envelope around Transport layer data (i.e., SOF, CRC, EOF, etc.).
- Receives data in the form of DWORDs from the Transport layer.
- Calculates CRC on Transport layer data.
- Transmits frame.
- Provides frame flow control in response to requests from the FIFO or the peer Link layer.
- Receives frame receipt acknowledge from peer Link layer.
- Reports good transmission or Link/Physical layer errors to Transport layer.
- Performs 8b/10b encoding.
- Scrambles (transforms) control and data DWORDs in such a way as to distribute the potential EMI emissions over a broader range.

15.1.3 Frame receipt

When data is received from the Physical layer, the Link layer provides the following services.

- Acknowledges to the peer Link layer readiness to receive a frame.
- Receives data in the form of encoded characters from the Physical layer.
- Decodes the encoded 8b/10b character stream into aligned DWORDs of data.
- Removes the envelope around frames (i.e., SOF, CRC, EOF).
- Calculates CRC on the received DWORDs.
- Provides frame flow control in response to requests from the FIFO or the peer Link layer.
- Compares the calculated CRC to the received CRC.

- Reports good reception or Link/Physical layer errors to Transport layer and the peer Link layer.
- Descrambles (untransforms) the control and data DWORDs received from a peer Link layer.

15.2 Encoding method

15.2.1 General

Information to be transmitted over the serial interface shall be encoded a byte (eight bits) at a time along with a data or control character indicator into a 10-bit encoded character and then sent serially bit by bit. Information received over the serial interface shall be collected ten bits at a time, assembled into an encoded character and decoded into the correct data characters and control characters. The 8b/10b code allows for the encoding of all 256 combinations of eight-bit data. A smaller subset of the control character set is utilized by the serial implementation of ATA.

15.2.2 Notation and conventions

The coding scheme uses a letter notation for describing data bits and control variables. A description of the translation process between these notations follows. This subclause also describes a convention used to differentiate data characters from control characters. Finally, translation examples for both a data character and a control character are presented, see 3.3.10.

An unencoded byte of data is composed of eight bits A,B,C,D,E,F,G,H and the control variable Z. The encoding process results in a 10 bit character a,b,c,d,e,i,f,g,h,j. A bit is either a binary zero or binary one. The control variable, Z, has a value of D or K. When the control variable associated with a byte has the value D, the byte is referred to as a data character. When the control variable associated with a byte has the value K, the byte is referred to as a control character.

If a data byte is not accompanied with a specific control variable value the control variable Z is assumed to be Z = D and the data byte shall be encoded as a data character.

The following Figure 49 illustrates the association between the numbered unencoded bits in a byte, the control variable, and the letter-labeled bits in the encoding scheme:

Data byte notation	7	6	5	4	3	2	1	0	Control variable
Unencoded bit notation	H	G	F	E	D	C	B	A	Z

Figure 49 – Bit designations

Each character is given a name Zxx.y where Z is the value of the control variable (D for a data character, K for a control character), xx is the decimal value of the binary number composed of the bits E, D, C, B and A in that order, and y is the decimal value of the binary number composed of the bits H, G and F.

Figure 50, following, shows the relationship between the various representations.

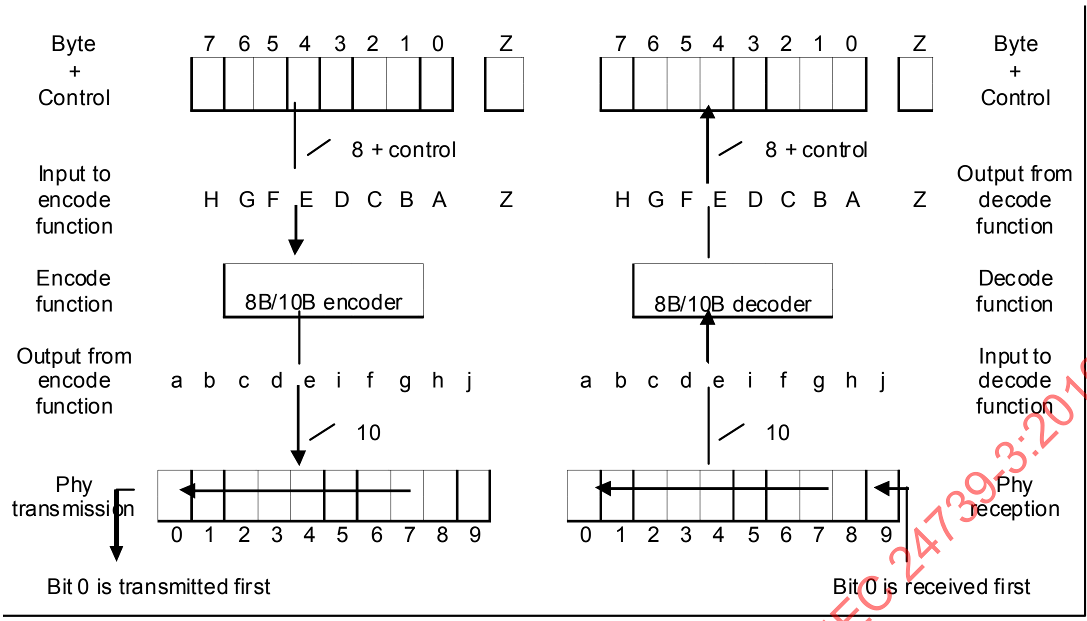


Figure 50 – Nomenclature reference

Figure 51 shows conversions from byte notation to character notation for a control and data byte.

Byte notation	BCh, control character		4Ah, data character	
Bit notation	76543210	Control variable	76543210	Control variable
	10111100	K	01001010	D
Unencoded bit notation	HGF EDCBA	Z	HGF EDCBA	Z
	101 11100	K	010 01010	D
Bit notation reordered to conform with Zxx.y convention	Z	EDCBA HGF	Z	EDCBA HGF
	K	11100 101	D	01010 010
Character name	K28.5		D10.2	

Figure 51 – Conversion examples

15.2.3 Character code

15.2.3.1 General

The coding scheme translates unencoded data and control bytes to characters. The encoded characters are then transmitted by the physical layer over the serial line where they are received from the physical layer and decoded into the corresponding byte and control value.

The coding scheme uses a subset of the 8b/10b coding method. The code used in the serial implementation of ATA uses all 256 data byte encodings while only two of the control codes are used. The reception of any unused code is a class of reception error referred to as a code violation, see 15.2.5.2.

15.2.3.2 Code construction

The 8b/10b coding process is defined in two stages. The first stage encodes the first five bits of the unencoded input byte into a six bit sub-block using a 5B/6B encoder. The input to this stage includes the current running disparity value. The second stage uses a 3B/4B encoder to encode the remaining three bits of the data byte and the running disparity as modified by the 5B/6B encoder into a four bit value.

In the derivations that follow, the control variable (Z) is assumed to have a value of D, and thus is an implicit input.

15.2.3.3 The concept of running disparity

Running Disparity is a binary parameter with either the value negative (-) or the value positive (+).

After transmitting any encoded character, the transmitter shall calculate a new value for its Running Disparity based on the value of the transmitted character.

After a COMRESET sequence, initial power-up, exiting any power management state, or exiting any diagnostic mode, the receiver shall assume either the positive or negative value for its initial Running Disparity. Upon reception of an encoded character the receiver shall determine whether the encoded character is valid according to the following rules and tables and shall calculate a new value for its Running Disparity based on the contents of the received character.

The following rules shall be used to calculate a new Running Disparity value for the transmitter after it sends an encoded character (transmitter's new Running Disparity) and for the receiver upon reception of an encoded character (receiver's new Running Disparity).

Running Disparity for an encoded character shall be calculated on two sub-blocks where the first six bits (abcdei) form one sub-block - the six-bit sub-block. The last four bits (fghj) form the second sub-block - the four-bit sub-block. Running Disparity at the beginning of the six-bit sub-block is the Running Disparity at the end of the last encoded character or the initial conditions described above for the first encoded character transmitted or received. Running Disparity at the beginning of the four-bit sub-block is the resulting Running Disparity from the six-bit sub-block. Running Disparity at the end of the encoded character - and the initial Running Disparity for the next encoded character - is the Running Disparity at the end of the four-bit sub-block.

Running Disparity for each of the sub-blocks shall be calculated as follows:

> Running Disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the six-bit sub-block if the value of the six-bit sub-block is 000111, and is positive at the end of the four-bit sub-block if the value of the four-bit sub-block is 0011.
> Running Disparity at the end of any sub-block is negative if the sub-block contains more zeroes than ones. It is also negative at the end of the six-bit sub-block if the value of the six-bit sub-block is 111000, and is negative at the end of the four-bit sub-block if the value of the four-bit sub-block is 1100.
> Otherwise, for any sub-block with an equal number of zeroes and ones, the Running Disparity at the end of the sub-block is the same as at the beginning of the sub-block. Sub-blocks with an equal number of zeroes and ones are said to have neutral disparity.

The 8b/10b code restricts the generation of the 000111, 111000, 0011 and 1100 sub-blocks in order to limit the run length of zeros and ones between sub-blocks. Sub-blocks containing 000111 or 0011 are generated only when the running disparity at the beginning of the sub-block is positive, resulting in positive Running Disparity at the end of the sub-block. Similarly, sub-blocks containing 111000 or 1100 are generated only when the running disparity at the beginning of the sub-block is negative and the resulting Running Disparity is negative.

The rules for Running Disparity result in generation of a character with disparity that is either the opposite of the previous character or neutral.

Sub-blocks with non-zero (non-neutral) disparity shall be of alternating disparity.

15.2.3.4 Data encoding

Table 17 and Table 18 describe the code and running disparity generation rules for each of the sub-blocks. The results can be used to generate the data in the data character tables.

In the tables which follow rd+ or rd- represent the current (incoming) running disparity and rd' represents the resulting Running Disparity. The resulting Running Disparity columns use -rd to indicate a change in Running Disparity polarity while rd indicates the resulting sub-block has neutral disparity.

Table 17 – 5b/6b coding

Inputs		abcdei outputs		rd'	Inputs		abcdei outputs		rd'
Dx	EDCBA	rd+	rd-		Dx	EDCBA	rd+	rd-	
D0	00000	011000	100111		D16	10000	100100	011011	-rd
D1	00001	100010	011101	-rd	D17	10001	100011		
D2	00010	010010	101101		D18	10010	010011		
D3	00011	110001		rd	D19	10011	110010		rd
D4	00100	001010	110101	-rd	D20	10100	001011		
D5	00101	101001			D21	10101	101010		
D6	00110	011001		rd	D22	10110	011010		
D7	00111	000111	111000		D23	10111	000101	111010	-rd
D8	01000	000110	111001	-rd	D24	11000	001100	110011	
D9	01001	100101			D25	11001	100110		rd
D10	01010	010101			D26	11010	010110		
D11	01011	110100		rd	D27	11011	001001	110110	-rd
D12	01100	001101			D28	11100	001110		rd
D13	01101	101100			D29	11101	010001	101110	
D14	01110	011100			D30	11110	100001	011110	-rd
D15	01111	101000	010111	-rd	D31	11111	010100	101011	

Table 18 – 3b/4b coding

Inputs		fghj outputs		rd'
Dx.y	HGF	rd+	rd-	
Dx.0	000	0100	1011	-rd
Dx.1	001	1001		
Dx.2	010	0101		rd
Dx.3	011	0011	1100	
Dx.4	100	0010	1101	-rd
Dx.5	101	1010		rd
Dx.6	110	0110		
Dx.P7	111	0001	1110	-rd
Dx.A7	111	1000	0111	
NOTE A7 replaces P7 if[(rd>0) and (e=i=0)] or [(rd<0) and (e=i=1)]				

15.2.3.5 Encoding examples

The coding examples in Figure 52 illustrate how the running disparity calculations are done.

The first conversion example completes the translation of data byte value 4Ah (which is the character name of D10.2) into an encoded character value of "abcdei fghj" = "010101 0101". This value has special significance because it is of neutral disparity and also contains an alternating zero/one pattern that represents the highest data frequency which can be generated.

In the second example the 8b/10b character named D11.7 is encoded. Assuming a positive value for the incoming Running Disparity, this example shows the Dx.P7/Dx.A7 substitution. With an initial rd+ value, D10 translates to an abcdei value of 110100, with a resulting Running Disparity of positive for the 6-bit sub-block. Encoding the 4-bit sub-block triggers the substitution clause of Dx.A7 for Dx.P7 since [(rd>0) AND (e=i=0)].

Initial rd	Character name	abcdei output	6-bit sub-block rd	fghj output	4-bit sub-block rd	Encoded character	Ending rd
-	D10.2	010101	-	0101	-	010101 0101	-
+	D11.7	110100	+	1000	-	110100 1000	-

Figure 52 – Coding examples

15.2.3.6 8b/10b valid encoded characters

15.2.3.6.1 General

The following tables define the valid data characters and valid control characters. These tables shall be used for generating encoded characters (encoding) for transmission. In the reception process, the table is used to look up and verify the validity of received characters (decoding).

In the tables, each data character and control character has two columns that represent two encoded characters. One column represents the output if the current Running Disparity is negative and the other is the output if the current Running Disparity is positive.

15.2.3.6.2 Data characters

Valid datea characters are listed in Table 19.

Table 19 – Valid data characters

Name	Byte	abcdei fghj output		Name	Byte	abcdei fghj output	
		Current rd-	Current rd+			Current rd-	Current rd+
D0.0	00h	100111 0100	011000 1011	D0.1	20h	100111 1001	011000 1001
D1.0	01h	011101 0100	100010 1011	D1.1	21h	011101 1001	100010 1001
D2.0	02h	101101 0100	010010 1011	D2.1	22h	101101 1001	010010 1001
D3.0	03h	110001 1011	110001 0100	D3.1	23h	110001 1001	110001 1001
D4.0	04h	110101 0100	001010 1011	D4.1	24h	110101 1001	001010 1001
D5.0	05h	101001 1011	101001 0100	D5.1	25h	101001 1001	101001 1001
D6.0	06h	011001 1011	011001 0100	D6.1	26h	011001 1001	011001 1001
D7.0	07h	111000 1011	000111 0100	D7.1	27h	111000 1001	000111 1001
D8.0	08h	111001 0100	000110 1011	D8.1	28h	111001 1001	000110 1001
D9.0	09h	100101 1011	100101 0100	D9.1	29h	100101 1001	100101 1001
D10.0	0Ah	010101 1011	010101 0100	D10.1	2Ah	010101 1001	010101 1001
D11.0	0Bh	110100 1011	110100 0100	D11.1	2Bh	110100 1001	110100 1001
D12.0	0Ch	001101 1011	001101 0100	D12.1	2Ch	001101 1001	001101 1001
D13.0	0Dh	101100 1011	101100 0100	D13.1	2Dh	101100 1001	101100 1001
D14.0	0Eh	011100 1011	011100 0100	D14.1	2Eh	011100 1001	011100 1001
D15.0	0Fh	010111 0100	101000 1011	D15.1	2Fh	010111 1001	101000 1001
D16.0	10h	011011 0100	100100 1011	D16.1	30h	011011 1001	100100 1001
D17.0	11h	100011 1011	100011 0100	D17.1	31h	100011 1001	100011 1001
D18.0	12h	010011 1011	010011 0100	D18.1	32h	010011 1001	010011 1001
D19.0	13h	110010 1011	110010 0100	D19.1	33h	110010 1001	110010 1001
D20.0	14h	001011 1011	001011 0100	D20.1	34h	001011 1001	001011 1001
D21.0	15h	101010 1011	101010 0100	D21.1	35h	101010 1001	101010 1001
D22.0	16h	011010 1011	011010 0100	D22.1	36h	011010 1001	011010 1001
D23.0	17h	111010 0100	000101 1011	D23.1	37h	111010 1001	000101 1001
D24.0	18h	110011 0100	001100 1011	D24.1	38h	110011 1001	001100 1001
D25.0	19h	100110 1011	100110 0100	D25.1	39h	100110 1001	100110 1001
D26.0	1Ah	010110 1011	010110 0100	D26.1	3Ah	010110 1001	010110 1001
D27.0	1Bh	110110 0100	001001 1011	D27.1	3Bh	110110 1001	001001 1001
D28.0	1Ch	001110 1011	001110 0100	D28.1	3Ch	001110 1001	001110 1001
D29.0	1Dh	101110 0100	010001 1011	D29.1	3Dh	101110 1001	010001 1001
D30.0	1Eh	011110 0100	100001 1011	D30.1	3Eh	011110 1001	100001 1001
D31.0	1Fh	101011 0100	010100 1011	D31.1	3Fh	101011 1001	010100 1001
D0.2	40h	100111 0101	011000 0101	D0.3	60h	100111 0011	011000 1100
D1.2	41h	011101 0101	100010 0101	D1.3	61h	011101 0011	100010 1100
D2.2	42h	101101 0101	010010 0101	D2.3	62h	101101 0011	010010 1100
D3.2	43h	110001 0101	110001 0101	D3.3	63h	110001 1100	110001 0011
D4.2	44h	110101 0101	001010 0101	D4.3	64h	110101 0011	001010 1100
D5.2	45h	101001 0101	101001 0101	D5.3	65h	101001 1100	101001 0011
D6.2	46h	011001 0101	011001 0101	D6.3	66h	011001 1100	011001 0011

Name	Byte	abcdei fghj output		Name	Byte	abcdei fghj output	
		Current rd-	Current rd+			Current rd-	Current rd+
D7.2	47h	111000 0101	000111 0101	D7.3	67h	111000 1100	000111 0011
D8.2	48h	111001 0101	000110 0101	D8.3	68h	111001 0011	000110 1100
D9.2	49h	100101 0101	100101 0101	D9.3	69h	100101 1100	100101 0011
D10.2	4Ah	010101 0101	010101 0101	D10.3	6Ah	010101 1100	010101 0011
D11.2	4Bh	110100 0101	110100 0101	D11.3	6Bh	110100 1100	110100 0011
D12.2	4Ch	001101 0101	001101 0101	D12.3	6Ch	001101 1100	001101 0011
D13.2	4Dh	101100 0101	101100 0101	D13.3	6Dh	101100 1100	101100 0011
D14.2	4Eh	011100 0101	011100 0101	D14.3	6Eh	011100 1100	011100 0011
D15.2	4Fh	010111 0101	101000 0101	D15.3	6Fh	010111 0011	101000 1100
D16.2	50h	011011 0101	100100 0101	D16.3	70h	011011 0011	100100 1100
D17.2	51h	100011 0101	100011 0101	D17.3	71h	100011 1100	100011 0011
D18.2	52h	010011 0101	010011 0101	D18.3	72h	010011 1100	010011 0011
D19.2	53h	110010 0101	110010 0101	D19.3	73h	110010 1100	110010 0011
D20.2	54h	001011 0101	001011 0101	D20.3	74h	001011 1100	001011 0011
D21.2	55h	101010 0101	101010 0101	D21.3	75h	101010 1100	101010 0011
D22.2	56h	011010 0101	011010 0101	D22.3	76h	011010 1100	011010 0011
D23.2	57h	111010 0101	000101 0101	D23.3	77h	111010 0011	000101 1100
D24.2	58h	110011 0101	001100 0101	D24.3	78h	110011 0011	001100 1100
D25.2	59h	100110 0101	100110 0101	D25.3	79h	100110 1100	100110 0011
D26.2	5Ah	010110 0101	010110 0101	D26.3	7Ah	010110 1100	010110 0011
D27.2	5Bh	110110 0101	001001 0101	D27.3	7Bh	110110 0011	001001 1100
D28.2	5Ch	001110 0101	001110 0101	D28.3	7Ch	001110 1100	001110 0011
D29.2	5Dh	101110 0101	010001 0101	D29.3	7Dh	101110 0011	010001 1100
D30.2	5Eh	011110 0101	100001 0101	D30.3	7Eh	011110 0011	100001 1100
D31.2	5Fh	101011 0101	010100 0101	D31.3	7Fh	101011 0011	010100 1100
D0.4	80h	100111 0010	011000 1101	D0.5	A0h	100111 1010	011000 1010
D1.4	81h	011101 0010	100010 1101	D1.5	A1h	011101 1010	100010 1010
D2.4	82h	101101 0010	010010 1101	D2.5	A2h	101101 1010	010010 1010
D3.4	83h	110001 1101	110001 0010	D3.5	A3h	110001 1010	110001 1010
D4.4	84h	110101 0010	001010 1101	D4.5	A4h	110101 1010	001010 1010
D5.4	85h	101001 1101	101001 0010	D5.5	A5h	101001 1010	101001 1010
D6.4	86h	011001 1101	011001 0010	D6.5	A6h	011001 1010	011001 1010
D7.4	87h	111000 1101	000111 0010	D7.5	A7h	111000 1010	000111 1010
D8.4	88h	111001 0010	000110 1101	D8.5	A8h	111001 1010	000110 1010
D9.4	89h	100101 1101	100101 0010	D9.5	A9h	100101 1010	100101 1010
D10.4	8Ah	010101 1101	010101 0010	D10.5	AAh	010101 1010	010101 1010
D11.4	8Bh	110100 1101	110100 0010	D11.5	ABh	110100 1010	110100 1010
D12.4	8Ch	001101 1101	001101 0010	D12.5	ACh	001101 1010	001101 1010
D13.4	8Dh	101100 1101	101100 0010	D13.5	ADh	101100 1010	101100 1010
D14.4	8Eh	011100 1101	011100 0010	D14.5	AEh	011100 1010	011100 1010
D15.4	8Fh	010111 0010	101000 1101	D15.5	AFh	010111 1010	101000 1010
D16.4	90h	011011 0010	100100 1101	D16.5	B0h	011011 1010	100100 1010
D17.4	91h	100011 1101	100011 0010	D17.5	B1h	100011 1010	100011 1010

Name	Byte	abcdei fghj output		Name	Byte	abcdei fghj output	
		Current rd-	Current rd+			Current rd-	Current rd+
D18.4	92h	010011 1101	010011 0010	D18.5	B2h	010011 1010	010011 1010
D19.4	93h	110010 1101	110010 0010	D19.5	B3h	110010 1010	110010 1010
D20.4	94h	001011 1101	001011 0010	D20.5	B4h	001011 1010	001011 1010
D21.4	95h	101010 1101	101010 0010	D21.5	B5h	101010 1010	101010 1010
D22.4	96h	011010 1101	011010 0010	D22.5	B6h	011010 1010	011010 1010
D23.4	97h	111010 0010	000101 1101	D23.5	B7h	111010 1010	000101 1010
D24.4	98h	110011 0010	001100 1101	D24.5	B8h	110011 1010	001100 1010
D25.4	99h	100110 1101	100110 0010	D25.5	B9h	100110 1010	100110 1010
D26.4	9Ah	010110 1101	010110 0010	D26.5	BAh	010110 1010	010110 1010
D27.4	9Bh	110110 0010	001001 1101	D27.5	BBh	110110 1010	001001 1010
D28.4	9Ch	001110 1101	001110 0010	D28.5	BCh	001110 1010	001110 1010
D29.4	9Dh	101110 0010	010001 1101	D29.5	BDh	101110 1010	010001 1010
D30.4	9Eh	011110 0010	100001 1101	D30.5	BEh	011110 1010	100001 1010
D31.4	9Fh	101011 0010	010100 1101	D31.5	BFh	101011 1010	010100 1010
D0.6	C0h	100111 0110	011000 0110	D0.7	E0h	100111 0001	011000 1110
D1.6	C1h	011101 0110	100010 0110	D1.7	E1h	011101 0001	100010 1110
D2.6	C2h	101101 0110	010010 0110	D2.7	E2h	101101 0001	010010 1110
D3.6	C3h	110001 0110	110001 0110	D3.7	E3h	110001 1110	110001 0001
D4.6	C4h	110101 0110	001010 0110	D4.7	E4h	110101 0001	001010 1110
D5.6	C5h	101001 0110	101001 0110	D5.7	E5h	101001 1110	101001 0001
D6.6	C6h	011001 0110	011001 0110	D6.7	E6h	011001 1110	011001 0001
D7.6	C7h	111000 0110	000111 0110	D7.7	E7h	111000 1110	000111 0001
D8.6	C8h	111001 0110	000110 0110	D8.7	E8h	111001 0001	000110 1110
D9.6	C9h	100101 0110	100101 0110	D9.7	E9h	100101 1110	100101 0001
D10.6	CAh	010101 0110	010101 0110	D10.7	EAh	010101 1110	010101 0001
D11.6	CBh	110100 0110	110100 0110	D11.7	EBh	110100 1110	110100 1000
D12.6	CCh	001101 0110	001101 0110	D12.7	ECh	001101 1110	001101 0001
D13.6	CDh	101100 0110	101100 0110	D13.7	EDh	101100 1110	101100 1000
D14.6	CEh	011100 0110	011100 0110	D14.7	EEh	011100 1110	011100 1000
D15.6	CFh	010111 0110	101000 0110	D15.7	EFh	010111 0001	101000 1110
D16.6	D0h	011011 0110	100100 0110	D16.7	F0h	011011 0001	100100 1110
D17.6	D1h	100011 0110	100011 0110	D17.7	F1h	100011 0111	100011 0001
D18.6	D2h	010011 0110	010011 0110	D18.7	F2h	010011 0111	010011 0001
D19.6	D3h	110010 0110	110010 0110	D19.7	F3h	110010 1110	110010 0001
D20.6	D4h	001011 0110	001011 0110	D20.7	F4h	001011 0111	001011 0001
D21.6	D5h	101010 0110	101010 0110	D21.7	F5h	101010 1110	101010 0001
D22.6	D6h	011010 0110	011010 0110	D22.7	F6h	011010 1110	011010 0001
D23.6	D7h	111010 0110	000101 0110	D23.7	F7h	111010 0001	000101 1110
D24.6	D8h	110011 0110	001100 0110	D24.7	F8h	110011 0001	001100 1110
D25.6	D9h	100110 0110	100110 0110	D25.7	F9h	100110 1110	100110 0001
D26.6	DAh	010110 0110	010110 0110	D26.7	FAh	010110 1110	010110 0001
D27.6	DBh	110110 0110	001001 0110	D27.7	FBh	110110 0001	001001 1110
D28.6	DCh	001110 0110	001110 0110	D28.7	FCh	001110 1110	001110 0001

Name	Byte	abcdei fghj output		Name	Byte	abcdei fghj output	
		Current rd-	Current rd+			Current rd-	Current rd+
D29.6	DDh	101110 0110	010001 0110	D29.7	FDh	101110 0001	010001 1110
D30.6	DEh	011110 0110	100001 0110	D30.7	FEh	011110 0001	100001 1110
D31.6	DFh	101011 0110	010100 0110	D31.7	FFh	101011 0001	010100 1110

15.2.3.6.3 Control characters

Valid control characters are listed in Table 20.

Table 20 – Valid control characters

Name	abcdei fghj output		Description
	Current rd-	Current rd+	
K28.3	001111 0011	110000 1100	Occurs only at byte 0 of all primitives except for the ALIGN primitive
K28.5	001111 1010	110000 0101	Occurs only at byte 0 of the ALIGN primitive

In the serial implementation of ATA, only the K28.3 and K28.5 control characters are valid and are always used as the first byte in a four-byte primitive. The K28.3 control character is used to prefix all primitives other than the ALIGN primitive, while the K28.5 control character is used to prefix the ALIGN primitive. The encoding of characters within primitives follow the same rules as that applied to non-primitives, when calculating the running disparity between characters and between subblocks of each character within the primitive. The control characters K28.3 and K28.5 invert the current running disparity.

The running disparity at the end of the ALIGN primitive is the same as the running disparity at the beginning of the ALIGN primitive.

15.2.4 Transmission summary

15.2.4.1 Transmission order

15.2.4.1.1 Bits within a byte

The bits within an encoded character are labeled a,b,c,d,e,i,f,g,h,j. Bit “a” shall be transmitted first, followed in order by “b”, “c”, “d”, “e”, “i”, “f”, “g”, “h” and “j”. Note that bit “i” is transmitted between bits “e” and “f”, and that bit “j” is transmitted last, and not in the order that would be indicated by the letters of the alphabet. Figure 53 illustrates the transmission order within a byte.

15.2.4.1.2 Bytes within a DWORD

For all transmissions and receptions, the serial implementation of ATA organizes all values as DWORDs. Even when representing a 32-bit value, the DWORD shall be considered a set of four bytes. The transmission order of the bytes within the DWORD shall be from the least-significant byte (byte 0) to the most-significant byte (byte 3). This right-to-left transmission order differs from Fibre Channel. Figure 53 illustrates how the bytes are arranged in a DWORD and the order in which bits are sent.

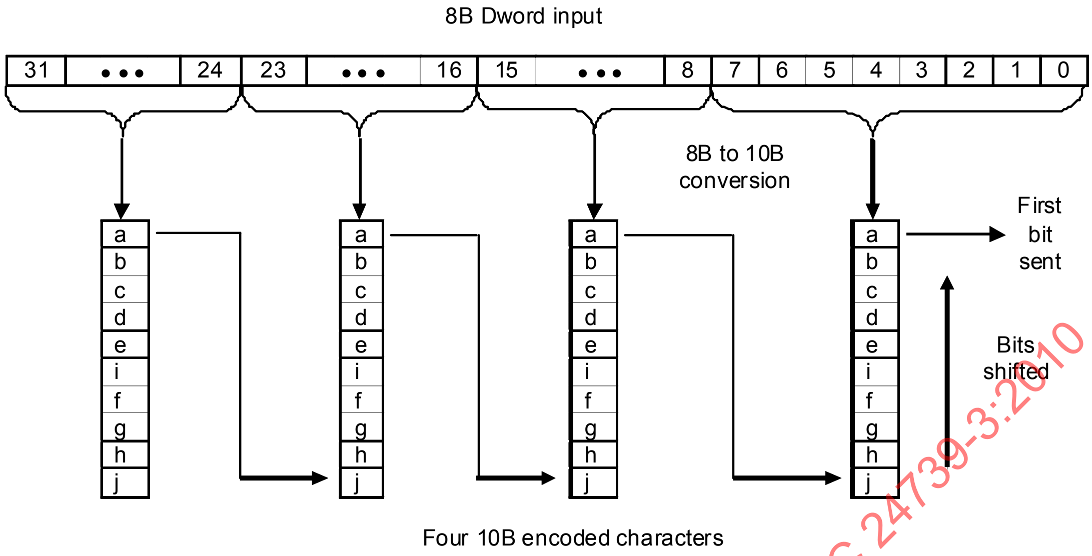


Figure 53 – Bit ordering and significance

15.2.4.1.3 DWORDs within a frame

A frame (as described in 16.2.1) shall be transmitted sequentially in ascending DWORD order starting with the SOF delimiter, followed by the DWORDs of the frame contents, followed by the CRC, and ending with the EOF delimiter.

NOTE Although the serial implementation of ATA employs a strict hierarchy of DWORD transmission as an ordered series of bytes, it is not the intent to restrict implementations from implementing a wider data path. It is possible, and even desirable, to perform transmission in word-sized fields. 8b/10b encoders with a 16(unencoded)/20(encoded) data path do exist. The only restriction is the transmission order of each byte and running disparity for each sub-block is preserved.

15.2.5 Reception

15.2.5.1 General

Upon reception of an encoded character the column corresponding to the receiver's current Running Disparity is searched for the encoded character value. If the encoded character value is found in the table the received encoded character shall be a legal character and decoded, and the decoded character value is made available to the Link layer.
If the received encoded character is not found in that column, then the encoded character shall be marked as code violation and reported to the Link layer.

15.2.5.2 Disparity and the detection of a code violation

Due to the propagation characteristics of the 8b/10b code, it is possible that a single bit error might not be detected until several characters after the introduction of the error. The following examples illustrate this effect. The first example, see Figure 54, shows a bit error being propagated two characters before being detected. The second example, see Figure 55, shows a single character of propagation.

It is important to note that the serial implementation of ATA sends data in DWORD increments, but the transmitter and receiver operate in units of a byte (character). The examples don't show DWORD boundaries, so it is possible that an error in either of these cases could be deferred by one full DWORD.

The frequency of disparity errors and code violations is an indicator of channel quality and corresponds directly to the bit error rate of the physical serial link between a host and a device. Implementations may elect to count such events and make them available to external firmware or software, although the method by which such counters are exposed is not defined in this standard.

Initial Running Disparity and the Running Disparity for each character is shown. In order to discover the errors note that Running Disparity is actually computed at the end of each sub-block and subsequently forwarded to the next sub-block. Footnotes indicate where the disparity error is detected. The error bit is underlined.

	rd	Character	rd	Character	rd	Character	rd
Transmitted character stream	-	D21.1	-	D10.2	-	D23.5	+
Transmitted bit stream	-	101010 1001	-	010101 0101	-	111010 1010	+
Received bit stream	-	101010 1011 (See note 1)	+	010101 0101	+	111010 1010 (See note 2)	+
Decoded character stream	-	D21.0	+	D10.2	+	Code violation (See note 2)	+(See note 3)

NOTE 1 Bit error introduced: 1001 → 1011

NOTE 2 Sub-blocks with non-neutral disparity shall alternate polarity (i.e., + → -). In this case, rd does not alternate (it stays positive for two sub-blocks in a row). The resulting encoded character does not exist in the rd+ column in the data or control code table, and so an invalid encoded character is recognized.

NOTE 3 Running disparity is computed on the received character regardless of the validity of the encoded character.

Figure 54 – Single bit error with two character delay

	rd	Character	rd	Character	rd	Character	rd
Transmitted character stream	-	D21.1	-	D23.4	-	D23.5	+
Transmitted bit stream	-	101010 1001	-	111010 0010	-	111010 1010	+
Received bit stream	-	101010 1011 (See note 1)	+	111010 0010 (See note 2)	-	111010 1010	+
Decoded character stream	-	D21.0	+	Code violation (See note 2)	-	D23.5	+ (See note 3)

NOTE 1 Bit error introduced: 1001 → 1011

NOTE 2 Sub-blocks with non-neutral disparity alternate polarity (i.e., + → -). In this case, rd does not alternate (it stays positive for two sub-blocks in a row). The resulting encoded character does not exist in the rd+ column in the data or control code table, and so an invalid encoded character is recognized.

NOTE 3 Running disparity is computed on the received character regardless of the validity of the encoded character.

Figure 55 – Single bit error with one character delay

15.3 Transmission method

The information on the serial line is a sequence of 8b/10b encoded characters. The smallest unit of communication is a DWORD. The contents of each DWORD are grouped to provide low-level control information or to transfer information between a host and an attached device.

The two types of structures are primitives and frames.

A primitive consists of a single DWORD and is the simplest unit of information that may be exchanged between a host and a device. When the bytes of a primitive are encoded the resulting pattern is difficult to misinterpret as any other primitive or random pattern. Primitives are used primarily to convey real-time state information, to control the transfer of information and coordinate host / device communication. All bytes in a primitive are constants and the first byte is always a control character. Since all of the bytes are constants, a primitive cannot be used to convey variable information. Later subclauses describe the contents of the primitives used in the serial implementation of ATA.

A frame consists of multiple DWORDs, and always starts with an SOF primitive, followed by a user payload called a Frame Information Structure (FIS), a CRC and ends with an EOF primitive. The CRC is defined to be the last non-primitive DWORD immediately preceding the EOF primitive. Some number of flow control primitives (HOLD or HOLDA, or a CONT stream to sustain a HOLD or HOLDA state) are allowed between the SOF and EOF primitives to throttle data flow for speed matching purposes.

The following Figure 56 shows an example of a sequence of transmissions.

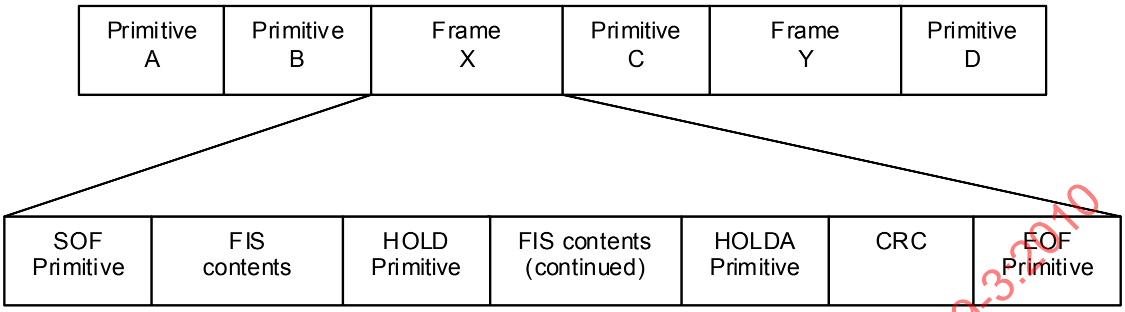


Figure 56 – Transmission structures

15.4 Primitives

15.4.1 Overview

15.4.1.1 General

Primitives are used to control and provide status of the serial line.

Primitives begin with a control character; all primitives use the K28.3 control character to signify the beginning of a primitive except for the ALIGN primitive which begins with the K28.5 control character. ALIGN thus represents the only primitive that contains the comma character (K28.5). Following the control character, three additional characters are encoded to complete the DWORD. Table 22 is a summary of the character combinations that make up each primitive.

15.4.1.2 Primitive disparity

Primitives begin with either positive or negative disparity and end in either positive or negative disparity Normal 8b/10b encoding disparity rules are applied when encoding primitives.

The ALIGN primitive is chosen to have neutral disparity so that it can be inserted into the stream without affecting the disparity of previously encoded characters. Disparity at the end of the ALIGN primitive is the same as the ending disparity of the last character transmitted before the ALIGN primitive.

Each primitive is described and the encoding defined in the following subclauses.

15.4.1.3 Primitive handshakes

Some primitives are transmitted in response to receipt of other primitives to acknowledge receipt. For example, the HOLDA primitive is transmitted in response to the receipt of HOLD primitives and R_OK or R_ERR is transmitted in response to WTRM primitives. Due to the different clock domains between to two ends of the cable, the number of response primitives may not match the number of primitives to which they are responding. For example, a host or device may send five HOLD primitives but receive six HOLDA primitives in response. Neither the sender nor recipient of these primitives need count the number of primitives or match the number sent and received. There are boundary cases where a zero number of response primitives such as HOLDA may be sent.

15.4.2 Primitive descriptions

The following Table 21 contains the primitive mnemonics and a brief description of each.

Table 21 – Description of primitives

Primitive	Name	Description	Usage
ALIGN	Physical layer control	Upon receipt of an ALIGN, the physical layer readjusts internal operations as necessary to perform its functions correctly. This primitive is always sent in pairs - there is no condition where an odd number of ALIGN primitives shall be sent (except as noted for retimed loopback).	Repeated
CONT	Continue repeating previous primitive	The CONT primitive allows long strings of repeated primitives to be eliminated. The CONT primitive is used to signal that the previously transmitted primitive pair is to be sustained until another primitive is received.	Single
DMAT	DMA terminate	This primitive is sent as a request to the transmitter to terminate a DMA data transmission early by computing a CRC on the data sent and ending with a EOF primitive. The transmitter context is assumed to remain stable after the EOF primitive has been sent.	Single
EOF	End of frame	EOF marks the end of a frame. The previous non-primitive DWORD is the CRC for the frame.	Single
HOLD	Hold data transmission	HOLD is transmitted in place of payload data within a frame when the transmitter does not have the next payload data ready for transmission. HOLD is also transmitted on the backchannel when a receiver is not ready to receive additional payload data.	Repeated NOTE
HOLDA	Hold acknowledge	This primitive is sent by a transmitter as long the HOLD primitive is received by its companion receiver.	Repeated NOTE
PMACK	Power management acknowledge	Sent in response to a PMREQ_S or PMREQ_P when a receiving node is prepared to enter a power mode state.	Repeated
PMNAK	Power management denial	Sent in response to a PMREQ_S or PMREQ_P when a receiving node is not prepared to enter a power mode state or when power management is not supported.	Repeated
PMREQ_P	Power management request to partial	This primitive is sent continuously until PMACK or PMNAK is received. When PMACK is received, current node (host or device) will stop PMREQ_P and enters the Partial power management state.	Repeated NOTE
PMREQ_S	Power management request to Slumber	This primitive is sent continuously until PMACK or PMNAK is received. When PMACK is received, current node (host or device) will stop PMREQ_S and enters the Slumber power management state.	Repeated NOTE
R_ERR	Reception error	Current node (host or device) detected error in received payload.	Repeated NOTE
R_IP	Reception in Progress	Current node (host or device) is receiving payload.	Repeated NOTE
R_OK	Reception with no error	Current node (host or device) detected no error in received payload.	Repeated NOTE
R_RDY	Receiver ready	Current node (host or device) is ready to receive payload.	Repeated NOTE
SOF	Start of frame	Start of a frame. Payload and CRC follow to EOF.	Single
SYNC	Synchronization	Synchronizing primitive - always idle.	Repeated NOTE
WTRM	Wait for frame termination	After transmission of any of the EOF, the transmitter will transmit WTRM while waiting for reception status from receiver.	Repeated NOTE
X_RDY	Transmission data ready	Current node (host or device) has payload ready for transmission	Repeated NOTE
NOTE This primitive may be continued by use of the CONT primitive.			

15.4.3 Primitive encoding

Table 22 defines the encoding for each primitive.

Table 22 – Primitive encoding

Primitive name	Byte 3 contents	Byte 2 contents	Byte 1 contents	Byte 0 contents
ALIGN	D27.3	D10.2	D10.2	K28.5
CONT	D25.4	D25.4	D10.5	K28.3
DMAT	D22.1	D22.1	D21.5	K28.3
EOF	D21.6	D21.6	D21.5	K28.3
HOLD	D21.6	D21.6	D10.5	K28.3
HOLDA	D21.4	D21.4	D10.5	K28.3
PMACK	D21.4	D21.4	D21.4	K28.3
PMNAK	D21.7	D21.7	D21.4	K28.3
PMREQ_P	D23.0	D23.0	D21.5	K28.3
PMREQ_S	D21.3	D21.3	D21.4	K28.3
R_ERR	D22.2	D22.2	D21.5	K28.3
R_IP	D21.2	D21.2	D21.5	K28.3
R_OK	D21.1	D21.1	D21.5	K28.3
R_RDY	D10.2	D10.2	D21.4	K28.3
SOF	D23.1	D23.1	D21.5	K28.3
SYNC	D21.5	D21.5	D21.4	K28.3
WTRM	D24.2	D24.2	D21.5	K28.3
X_RDY	D23.2	D23.2	D21.5	K28.3

15.4.4 ALIGN primitive

This primitive is always sent in pairs - there is no condition where an odd number of ALIGN primitives are sent (except as noted for retimed loopback).

The Link layer shall ignore reception of ALIGN primitives. The Physical layer may consume received ALIGN primitives. Implementations where the Phy does not consume received ALIGN primitives shall drop received ALIGN primitives at the input to the Link layer or shall include Link layer processing that yields behavior equivalent to the behavior produced if all received ALIGN primitives are consumed by the Phy and not presented to the Link.

After communications have been established, the first and second words out of the Link Layer shall be the dual-ALIGN primitive sequence, followed by at most 254 non-ALIGN DWORDs. The cycle repeats starting with another dual-consecutive ALIGN primitive sequence. The Link may issue more than one dual ALIGN primitive sequence but shall not send an unpaired ALIGN primitive (i.e. ALIGN primitives are always sent in pairs) except as noted for retimed loopback.

The ALIGN primitive encoding is defined in Table 22.

15.4.5 CONT primitive

15.4.5.1 General

The CONT primitive allows long strings of repeated primitives, as defined by Table 21 to be eliminated. The CONT primitive causes the previously received repeated primitive to be sustained until another primitive is received. The reception of any primitive other than CONT or ALIGN shall terminate the stream of repeated primitives. For an example of the usage of the CONT primitive, see Figure 57.

In order to accommodate EMI reductions, scrambling of data is incorporated in the serial implementation of ATA as described in 15.6. The scrambling of data is with a linear feedback shift register (LFSR) used in generating the scrambling pattern being reset at each SOF primitive, or rolling over every 2 048 DWORDs. However, the scrambling of primitives is not as effective or simple because of the small number of control characters available. In order to accommodate EMI reductions, repeated primitives are eliminated through the use of the CONT primitive.

Any repeated primitive (see Table 21) may be sustained through the use of the CONT primitive. The recipient of the CONT primitive shall ignore all data received after the CONT primitive until the reception of any primitive, excluding ALIGN. After transmitting the CONT character, the transmitter may send any sequence of data characters to the recipient provided that no primitives are included. The transmitter shall send a minimum of two identical repeated primitives (e.g SYNC SYNC, HOLD HOLD, etc) immediately preceding the CONT primitive, excluding ALIGNs. Valid CONT transmission sequences are shown in Table 23.

Table 23 – Valid CONT Transmission Sequences

PRIM	PRIM	CONT	XXXX	XXXX	XXXX
PRIM	ALIGN	ALIGN	PRIM	CONT	XXXX
PRIM	PRIM	ALIGN	ALIGN	CONT	XXXX
PRIM	PRIM	CONT	ALIGN	ALIGN	XXXX
PRIM = any primitive defined as repeated that may be continued in accordance with Table 21. XXXX = Output of LFSR or any primitive other than CONT.					

To improve overall protocol robustness and avoid potential timeout situations caused by a reception error in a primitive, all repeated primitives shall be transmitted a minimum of twice before a CONT primitive is transmitted. The first primitive correctly received is the initiator of any action within the receiver. This avoids scenarios, for example, where X_RDY is sent from the host, followed by a CONT, and the X_RDY is received improperly resulting in the device not returning an R_RDY and causing the system to deadlock until a timeout/reset condition occurs. Reception of a CONT primitive when one of the two preceding DWORDs is not a valid repeated primitive results in undefined behavior.

The transmission of a CONT primitive is optional, but the ability to receive and properly process the CONT primitive is required.

For a list of primitives that may be followed by a CONT, see Table 21.

The host PHY initialization state machine consumes the first few received primitives before communications between the host and device have been established (see state HP10:HR_SendAlign in 14.5.6.3.2). In order to ensure proper synchronization between the host and device after entry into the L1:L_IDLE state from the LS3:L_SendAlign state or the LPM8:L_WakeUp2 state (see 15.6.3.1 and 15.6.3.4), the use of the CONT primitive is not allowed after a transition from the LS3:L_SendAlign state or the LPM8:L_WakeUp2 state to the L1:L_IDLE state until either a minimum of 10 non-ALIGN primitives have been transmitted or until receipt of a primitive other than SYNC or ALIGN has been detected.

15.4.5.2 Scrambling of data following the CONT primitive

The data following the CONT shall be the output of an LFSR which implements the same polynomial as is used to scramble FIS contents. That polynomial is defined in 15.6.

The resulting LFSR value shall be encoded using the 8b/10b rules for encoding data characters before transmission by the Link layer.

The LFSR used to supply data after the CONT primitive shall be reset to the initial value upon detection of a COMINIT or COMRESET event.

Since the data following a CONT primitive is discarded by the Link layer, the value of the LFSR is undefined between CONT primitives. That is, the LFSR result used for CONT sequence N is not required to be continuous from the last LFSR result of CONT sequence N-1.

The sequence of LFSR values used to scramble the payload contents of an FIS shall not be affected by the scrambling of data used during repeated primitive suppression. That is, the data payload LFSR shall not be advanced during repeated primitive suppression and shall only be advanced for each payload data character that is scrambled using the data payload LFSR. See 15.6 for additional information on scrambling and repeated primitive suppression.

15.4.6 DMAT primitive

NOTE No consistent use of the DMAT (DMA Terminate) facility is defined, and its use may impact software compatibility. Implementations should tolerate reception of DMAT as defined in the standard but should avoid transmission of DMAT in order to minimize potential interaction problems. One valid response to reception of DMAT is to ignore it and complete the transfer.

The DMAT primitive is sent as a request to the transmitter to terminate a DMA data transmission early by computing a CRC on the data sent and ending with a EOF primitive. The DMA context is assumed to remain stable after the EOF primitive has been sent.

In a parallel implementation of ATA, devices can abort a DMA transfer and post an error, by negating DMA Request, updating the Error and Status registers, and possibly setting the INTRQ line. This can be performed for both reads from and writes to the device.

For example, in the serial implementation of ATA. a DMA read from device a device may terminate the transfer with an EOF, and send a Register Device to Host FIS to the host, with Error and Status registers updated appropriately. In the case of a DMA write to device, the device sends a DMA Activate FIS to the host, and then after receiving an SOF, accepts all data until receiving an EOF from the host. Since the device cannot terminate such a transfer once started, a DMAT primitive is used.

The DMA Terminate (DMAT) primitive may be sent on the back channel during transmission of a Data FIS to signal the transmitter to terminate the transfer in progress. It may be used for both host to device transfers and for device to host transfers. If processed, reception of the DMAT signal shall cause the recipient to close the current frame by inserting the CRC and EOF, and return to the idle state.

For host to device data transfers, upon receiving the DMAT signal the host may terminate the transfer in progress by deactivating its DMA engine and closing the frame with valid CRC and EOF. The host DMA engine shall preserve its state at the point it was deactivated so that the device may resume the transmission at a later time by transmitting another DMA Active FIS to re-activate the DMA engine. The device is responsible for either subsequently resuming the terminated transfer by transmitting another DMA Activate FIS or closing the affected command with appropriate status.

For device to host transfers, receipt of DMAT signal by the device results in permanent termination of the transfer and is not resumable. The device may terminate the transmission in progress and close the frame with a valid CRC and EOF, and shall thereafter clean up the affected command by indicating appropriate status for that command. No facility for resuming a device to host transfer terminated with the DMAT signal is provided.

Some implementations may have an implementation-dependent latency associated with closing the affected Data FIS in response to the DMAT signal. For example, a host adapter may have a small transmit FIFO, and in order for the DMA engine to accurately reflect a resumable state, the data already transferred by the DMA engine to the transmit FIFO may have to be transmitted prior to closing the affected Data FIS. Designs should minimize the DMAT response latency while being tolerant of other devices having a long latency.

15.4.7 EOF primitive

EOF marks the end of a frame. The previous non-primitive DWORD is the CRC for the frame.

15.4.8 HOLD/HOLDA primitives

15.4.8.1 General

HOLD is transmitted in place of payload data within a frame when the transmitter does not have the next payload data ready for transmission. HOLD is also transmitted on the backchannel when a receiver is not ready to receive additional payload data.

The HOLDA primitive is sent by a transmitter as long the HOLD primitive is received by its companion receiver.

15.4.8.2 Flow Control Signaling Latency

There is a finite pipeline latency in a round-trip handshake across the serial interface, In order to accommodate efficient system design with sufficient buffering headroom to avoid buffer overflow in flow control situations, the maximum tolerable latency from when a receiver issues a HOLD signal until it receives the HOLDA signal from the transmitter is bounded. This allows the limit to be set in the receive FIFO so as to avoid buffer overflow while avoiding excessive buffering/FIFO space.

In the case where the receiver wants to flow control the incoming data, it transmits HOLD characters on the back channel. Some number of received DWORDs later, valid data ceases, and HOLDA characters are received. The larger the latency between transmitting HOLD until receiving HOLDA, the larger the receive FIFO needs to be. The maximum allowed latency from the time the MSB of the HOLD primitive is on the wire, until the MSB of the HOLDA is on the wire shall be no more than 20 DWORD times. The LSB is transmitted first. A receiver shall be able to accommodate reception of 20 DWORDs of additional data after the time it transmits the HOLD flow control character to the transmitter, and the transmitter shall respond with a HOLDA in response to receiving a HOLD character within 20 DWORD times. See Figure 59 and Figure 60 for HOLD/HOLD transition cases that do respond with HOLDA within 20 DWORD latency times.

The specified maximum latency figure is based on the layers and states described throughout this document. It is recognized that the Link Layer may have two separate clock domains -- transmit clock domain, and the receive clock domain. It is also recognized that a Link state machine could run at the DWORD clock rate, implying synchronizers between three potential clock domains. In practice more efficient implementations would be pursued and the actual latencies may be less than indicated here. This accounting assumes a worst case synchronizer latency of 2.99 clocks in any clock domain and is rounded to three whole clocks. A one-meter cable contains less than one-half DWORD and is therefore rounded to 0. Two DWORDs of pipeline delay are assumed for the Phy, and the FIFO is assumed to run at the Link state machine rate. No synchronization is needed between the two.

Table 24 outlines the origin of the 20 DWORD latency standard. The example illustrates the components of a round trip delay when the receiver places a HOLD on the bus until reception of the HOLDA from the transmitter. This corresponds to the number of DWORDs that the receiver is able to accept after transmitting a HOLD character.

Table 25 is an example of an SRST write from host to device transmission breaking through a device to host Data FIS.

Table 24 – Latency example

Receiver sends HOLD:		
	1 DWORD	Convert to 40 bit data.
	1 DWORD	10b/8b conversion.
	1 DWORD	De scrambling.
	3 DWORDs	Synchronization between receive clock, and Link state machine clock.
	1 DWORD	Link state machine is notified that primitive has been received.
	1 DWORD	Link state machine takes action.
	1 DWORD	FIFO is notified of primitive reception.
	1 DWORD	FIFO stops sending data to Link layer.
	1 DWORD	Link is notified to insert HOLDA.
	1 DWORD	Link acts on notification and inserts HOLDA into data stream.
	1 DWORD	Scrambling
	1 DWORD	8b/10b conversion.
	1 DWORD	Synchronize to transmit clock (3 transmit clocks, which are four times the Link state machine rate).
	1 DWORD	Convert to 10 bit data.
	2 DWORDs	Phy, transmit side.
HOLDA on the cable.		

Table 25 – SRST write from host to device transmission breaking through a device to host Data FIS

Host driver	Device driver	Description
…	…	Previous activity abbreviated for clarity
R_IP	Data n	Device transmitting data
R_IP	Data n+1	
R_IP	HOLD	Device transmit FIFO empty, and flow control applied
R_IP	HOLD	Host receives and decodes HOLD flow control
HOLDA	HOLD	Host acknowledges flow control. Device internally deadlocked and no more data forthcoming (drive hung)
HOLDA	HOLD	
…	…	System in this state until host decides to reset drive
HOLDA	HOLD	Host detects SRST write to control register, needs to break deadlock
SYNC	HOLD	Host transmits SYNC primitive to abort current transmission
SYNC	HOLD	Device receives and decodes SYNC, abandons transmission in progress
SYNC	SYNC	Host sends SYNC / Device sends SYNC (both returned to idle state)
SYNC	SYNC	Host receives and decodes SYNC, may now initiate new FIS transmission
X_RDY	SYNC	Host ready to send Shadow Control Block for SRST write
X_RDY	SYNC	Device decodes X_RDY
X_RDY	R_RDY	Device indicates ready to receive
X_RDY	R_RDY	Host decodes R_RDY
SOF	R_RDY	Host starts a frame
etc.	etc.	etc.

15.4.9 PMREQ_P, PMREQ_S, PMACK, and PMNAK primitives

The PMREQ_P primitive is sent continuously to enter Partial power management state until PMACK or PMNAK is received. When PMACK is received, current node (host or device) will stop PMREQ_P and enters the Partial power management state.

The PMREQ_S primitive is sent continuously to enter Slumber power management state until PMACK or PMNAK is received. When PMACK is received, current node (host or device) will stop PMREQ_S and enters the Slumber power management state.

PMACK is sent in response to a PMREQ_S or PMREQ_P when a receiving node is prepared to enter a power mode state.

PMNAK is sent in response to a PMREQ_S or PMREQ_P when a receiving node is not prepared to enter a power mode state or when power management is not supported.

15.4.10 R_ERR primitive

Current node (host or device) detected an error in received payload.

15.4.11 R_IP primitive

Current node (host or device) is receiving payload.

15.4.12 R_OK primitive

Current node (host or device) detected no error in received payload.

15.4.13 R_RDY primitive

Current node (host or device) is ready to receive payload.

15.4.14 SOF primitive

Start of a frame. Payload and CRC follow to EOF.

15.4.15 SYNC primitive

Synchronizing primitive - Indicates physical interface is idle.

15.4.16 WTRM primitive

After transmission of the EOF primitive, the transmitter transmits WTRM while waiting for reception of a R_ERR or R_OK primitive from the receiver.

15.4.17 X_RDY primitive

Current node (host or device) has payload ready for transmission

15.4.18 Examples

The following examples illustrate basic primitive usage.

Figure 57 illustrates use of the CONT primitive in the transmission of an FIS.

Table 26 illustrates Shadow Command Block and Shadow Control Block transmission.

Table 27 illustrates data transmission from host to device.

Table 28 illustrates DMA data transmission from host to device with the device terminating transmission.

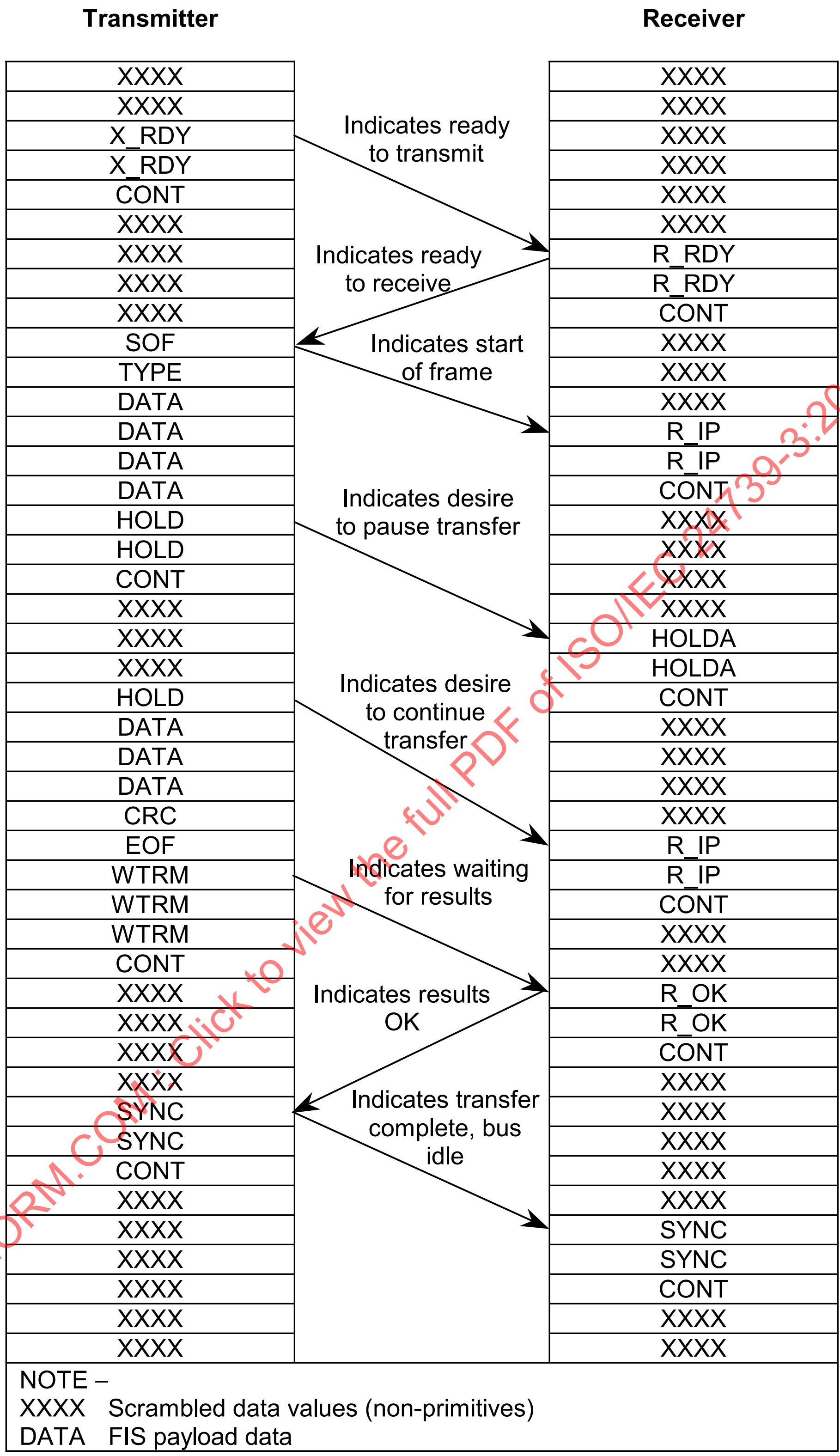


Figure 57 – CONT usage example

Table 26 – Shadow Command Block and Shadow Control Block transmission example

Host driver	Device driver	Description
SYNC	SYNC	Idle condition
SYNC	SYNC	Idle condition
X_RDY	SYNC	Host ready to send Shadow Command Block and Shadow Control Block
X_RDY	SYNC	Device decodes X_RDY
X_RDY	R_RDY	Device indicates ready to receive
X_RDY	R_RDY	Host decodes R_RDY
SOF	R_RDY	Host starts a frame
Hdr 0	R_RDY	Host sends Register FIS DWORD 0 / device decodes SOF
Hdr 1	R_IP	Host sends Register FIS DWORD 1 / device stores Hdr 0
…	…	…
Hdr n	R_IP	Host sends Register FIS DWORD n / device stores Hdr n-1
CRC	R_IP	Host sends CRC / device stores Hdr n
EOF	R_IP	Host sends EOF / device stores CRC
WTRM	R_IP	Device decodes EOF
WTRM	R_IP	Device computes good CRC and releases TF contents
WTRM	R_OK	Device sends good end
WTRM	R_OK	Host decodes R_OK as good results
SYNC	R_OK	Host releases interface
SYNC	R_OK	Device decodes release by host - is allowed to release
SYNC	SYNC	Idle condition

Table 27 – Data from host to device transmission example

Host driver	Device driver	Description
SYNC	SYNC	Idle condition
SYNC	SYNC	Idle condition
X_RDY	SYNC	Host ready to send Shadow Command Block and Shadow Control Block
X_RDY	SYNC	Device decodes X_RDY
X_RDY	R_RDY	Device indicates ready to receive
X_RDY	R_RDY	Host decodes R_RDY
SOF	R_RDY	Host starts a frame
Hdr 0	R_RDY	Host sends header DWORD 0 / device decodes SOF
Hdr 1	R_IP	Host sends header DWORD 1 / device stores header DWORD 0
…	..	…
Dat x	R_IP	Host sends data DWORD x / device stores data DWORD (x-1)
HOLD	R_IP	Host sends HOLD / device stores data DWORD (x) and decodes HOLD
HOLD	HOLDA	Device acknowledges HOLD
HOLD	HOLDA	Host decodes HOLDA - host may release HOLD at any time
Dat(n-3)	HOLDA	Host sends (n-2)th data DWORD / device decodes data DWORD
Dat(n-2)	R_IP	Host sends (n-1)th data DWORD / device Stores (n-2)th data DWORD
Dat(n-1)	R_IP	Host sends nth data DWORD / device ores (n-1)th data DWORD
CRC	R_IP	Host sends CRC / device stores nth data DWORD
EOF	R_IP	Host sends EOF / device stores CRC
WTRM	R_IP	Device decodes EOF
WTRM	R_IP	Device computes good CRC and releases data contents
WTRM	R_OK	Device sends good end
WTRM	R_OK	Host decodes R_OK as good results
SYNC	R_OK	Host releases interface
SYNC	R_OK	Device decodes release by host - is allowed to release
SYNC	SYNC	Idle condition

Table 28 – DMA data from host to device, device terminates transmission example

Host driver	Device driver	Description
SYNC	SYNC	Idle condition
SYNC	SYNC	Idle condition
X_RDY	SYNC	Host ready to send data
X_RDY	SYNC	Device decodes X_RDY
X_RDY	R_RDY	Device indicates ready to receive
X_RDY	R_RDY	Host decodes R_RDY
SOF	R_RDY	Host starts a frame
Hdr 0	R_RDY	Host sends header DWORD 0 / device decodes SOF
Dat 0	R_IP	Host sends data DWORD 0
…	..	Host sends data DWORD x / device stores data DWORD
Dat x	DMAT	Device decides to terminate, sends DMAT
Dat x	R_IP	Host decodes DMAT - host prepares to terminate
CRC	R_IP	Host sends current CRC value
EOF	R_IP	Host sends EOF / device stores CRC
WTRM	R_IP	Device decodes EOF
WTRM	R_IP	Device computes good CRC and releases data contents
WTRM	R_OK	Device sends good end
WTRM	R_OK	Host decodes R_OK as good results
SYNC	R_OK	Host releases interface
SYNC	R_OK	Device decodes release by host - is allowed to release
SYNC	SYNC	Idle condition

15.5 CRC calculation

The CRC (Cyclic Redundancy Check) of a frame is a DWORD (32-bit) field that shall follow the last DWORD of the contents of an FIS and precede the EOF primitive. The CRC calculation covers all of the FIS transport data between the SOF and EOF primitives, and excludes any intervening primitives and CONT stream contents. The CRC value shall be computed on the contents of the FIS before encoding for transmission (scrambling) and after decoding upon reception.

The CRC shall be calculated on DWORD quantities. If an FIS contains an odd number of words the last word of the FIS shall be padded with zeros to a full DWORD before the DWORD is used in the calculation of the CRC.

The CRC shall be aligned on a DWORD boundary.

The CRC shall be calculated using the following 32-bit generator polynomial:

$$G(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

The CRC value shall be initialized with a value of 52325032h before the calculation begins.

The maximum number of DWORDs between the SOF primitive to the EOF primitive shall not exceed 2064 DWORDs including the FIS type and CRC.

15.6 Scrambling

15.6.1 Frame content scrambling

15.6.1.1 General

The contents of a frame, that is, all data words between the SOF and EOF including the CRC, shall be scrambled before transmission by the physical layer.

Scrambling shall be performed on DWORD quantities by XORing the data to be transmitted with the output of a linear feedback shift register (LFSR). The shift register shall implement the following polynomial:

$$G(X) = X^{16} + X^{15} + X^{13} + X\,X^{4} + 1$$

The serial shift register shall be initialized with a value of FFFFh before the first shifted output. The shift register shall be initialized to the seed value before the SOF primitive.

15.6.1.2 Relationship between scrambling and CRC

The order of application of scrambling shall be as follows. For a DWORD of data following the SOF primitive the DWORD shall be used in the calculation of the CRC. The same DWORD value shall be XORed with the scrambler output, and the resulting DWORD submitted to the 8b/10b encoder for transmission. Similarly, on reception, the DWORD shall be decoded using a 10b/8b decoder, the scrambler output shall be XORed with the resulting DWORD, and the resulting DWORD presented to the Link layer and subsequently used in calculating the CRC. The CRC DWORD shall be scrambled according to the same rules.

15.6.2 Repeated primitive suppression

15.6.2.1 General

A second linear feedback shift register shall be used to provide scrambled data for the suppression of repeated primitives with the CONT primitive (see 15.4.5). This LFSR shall implement the same polynomial as the frame scrambling LFSR but need not be reinitialized.

15.6.2.2 Relationship between scrambling of FIS data and repeated primitives

There are two separate scramblers used in the serial implementation of ATA. One scrambler is used for the data payload encoding and a separate scrambler is used for repeated primitive suppression. The scrambler used for data payload encoding shall maintain consistent and contiguous context over the scrambled payload data characters of a frame (between SOF and EOF), and shall not have its context affected by the scrambling of data used for repeated primitive suppression.

Scrambling is applied to all data (non-primitive) DWORDs. Primitives, including ALIGN, do not get scrambled and shall not advance the data payload LFSR register. Similarly, the data payload LFSR shall not be advanced during transmission of DWORDs during repeated primitive suppression (i.e. after a CONT primitive). Since it is possible for a repeated primitive stream to occur in the middle of a data frame - multiple HOLD/HOLDA primitives are likely - care should be taken to ensure that the data payload LFSR is only advanced for each data payload data character that it scrambles and that it is not advanced for primitives or for data characters transmitted as part of repeated primitive suppression.

15.6.3 Link layer state diagrams

15.6.3.1 Link idle state diagram

Figure 58 shows the link idle state diagram.

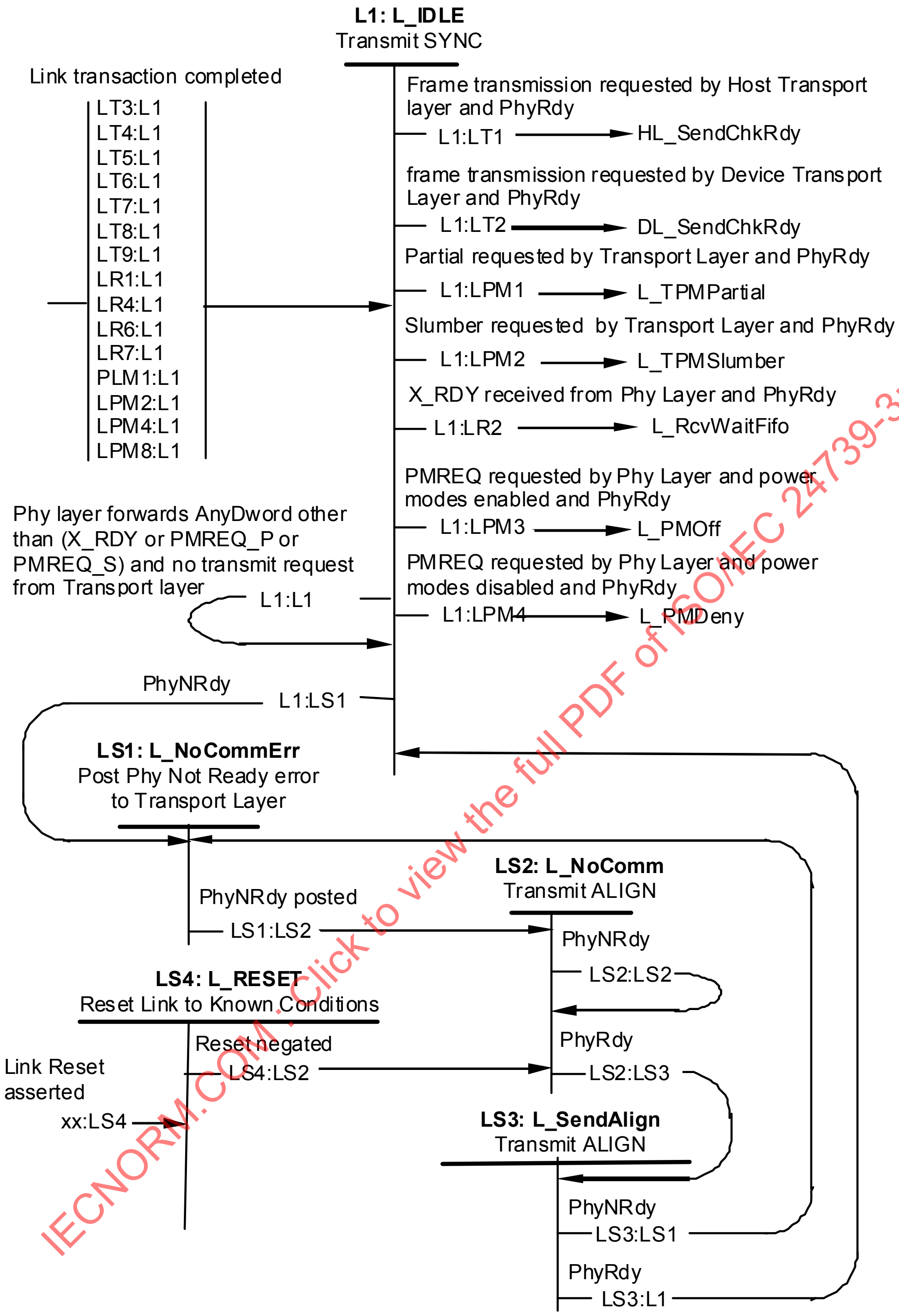


Figure 58 – Link idle state diagram (States L1, LS1-LS3)

L1: L_IDLE state: This state is entered when a frame transmission has been completed by the Link layer or from Power Management or as a result of errors in the Link Transmit or Link Receive states. When entered from the LS3:L1 SendAlign state or the LPM8:L1 WakeUp2 state, use of the CONT primitive is subject to restrictions as outlined in 15.4.5.

When in this state, the Link layer transmits the SYNC primitive and waits for an X_RDY primitive from the Physical layer or a frame transmission request from the Transport layer.

Transition L1:LT1: When the host Link layer receives a request to transmit a frame from the Transport layer and the Physical layer is ready, the Link layer shall make a transition to the LT1: HL_SendChkRdy state. This transition is taken even if errors such as 10b decoding errors are detected.

Transition L1:LT2: When the device Link layer receives a request to transmit a frame from the Transport layer and the Physical layer is ready, the Link layer shall make a transition to the LT2: DL_SendChkRdy state. This transition is taken even if errors such as 10b decoding errors are detected.

Transition L1:LPM1: When the Link layer receives a request to enter the Partial power mode from the Transport layer and the Physical layer is ready, the Link layer shall make a transition to the LPM1: L_TPMPartial state. This transition is taken even if errors such as 10b decoding errors are detected.

Transition L1:LPM2: When the Link layer receives a request to enter the Slumber power mode from the Transport layer and the Physical layer is ready, the Link layer shall make a transition to the LPM2: L_TPMSlumber state. This transition is taken even if errors such as 10b decoding errors are detected.

Transition L1:LR2: When the Link layer receives an X_RDY from the Physical layer, the Link layer shall make a transition to the LR2: L_RcvWaitFifo state.

Transition L1:LPM3: Phy layer forwards (PMREQ_P or PMREQ_S) and power modes are enabled: When the Link layer receives a PMREQ_P or PMREQ_S from the Physical layer and is enabled to perform power management modes, the Link layer shall make a transition to the LPM3: L_PMOff state.

Transition L1:LPM4: Phy layer forwards (PMREQ_P or PMREQ_S) and power modes are disabled: When the Link layer receives a PMREQ_P or a PMREQ_S from the Physical layer and is not enabled to perform power management modes, the Link layer shall make a transition to the LPM4: L_PMDeny state.

Transition L1:L1: Phy layer forwards AnyDword other than (X_RDY or PMREQ_P or PMREQ_S) and no transmit request from Transport layer: When the Link layer does not receive a request to transmit a frame from the Transport layer, does not receive a request to go to a power mode from the Transport layer, does not receive an X_RDY from the Physical layer or does not receive a PMREQ_P or PMREQ_S from the Physical layer the Link layer shall make a transition to the L1: L_IDLE state. This transition is taken even if errors such as 10b decoding errors are detected. This state ignores any unrecognized sequences or primitives not defined by this standard.

Transition L1:LS1: If the Physical layer becomes not ready even if the Transport layer is requesting an operation, the Link layer transitions to the L_NoCommErr state.

LS1: L_NoCommErr state: This state is entered upon detection of a not ready condition of the Physical layer while attempting to process another state. The entry into this state indicates an error condition in the Link layer.

Transition LS1:LS2: The transition is made to LS2:L_NoComm when the error has been passed to the transport layer.

LS2: L_NoComm state: This state is entered directly from the LS1:L_NoCommErr state or the LS4:L_RESET State. The Link Layer remains in this state until the Phy signals that it has established communications and is ready.

While in this state, enable Physical Layer and transmit ALIGN primitives.

Transition LS2:LS2: If the Physical layer signals it is not ready, the transition is made to LS2: L_NoComm.

Transition LS2:LS3: When the Physical layer signals it is ready, a transition is made to LS3: L_SendAlign.

LS3: L_SendAlign state: This state is entered when PhyRdy is detected.

When in this state ALIGN is transmitted.

Transition LS3:LS1: If the Physical layer becomes not ready, then a transition is made to LS1: L_NoCommErr.

Transition LS3:L1: If the Phy indicates that it is ready, a transition is made to the L1: L_IDLE state.

LS4: L_RESET state: This state is entered whenever the Link Reset control is active. All Link layer hardware is initialized to and held at initial conditions. While in this state all requests or triggers from other layers are ignored. While in this state, the Phy reset signal is also asserted.

Transition xx:LS4: While the RESET control is active a transition is made back to the LS4: L_RESET state.

Transition LS4:LS2: When the RESET control goes inactive a transition is made to the LS2: L_NoComm state.

15.6.3.2 Link transmit state diagram

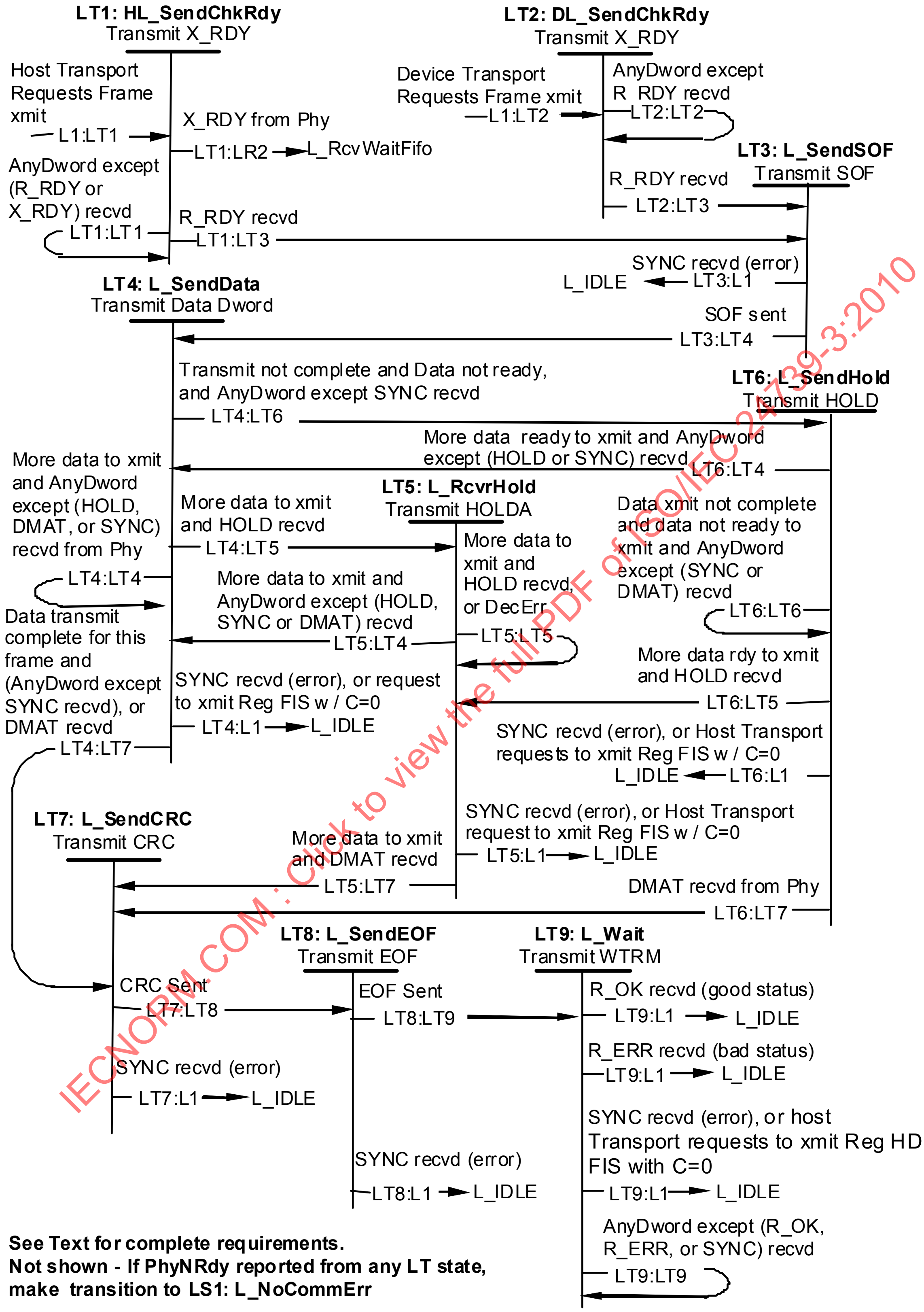


Figure 59 – Link transmit state diagram (States LT1-LT9)

In any state of the Link transmit state diagram, if the Link layer detects the Phy layer is not ready, the Link layer will notify the Transport layer, fail the attempted transfer, and make a transition to LS1: L_NoCommErr. These transitions are not shown in the state diagram to improve clarity.

LT1: HL_SendChkRdy state: This state is entered when a frame transmission has been requested by the host Transport layer.

When in this state, the Link layer transmits an X_RDY primitive and waits for an X_RDY primitive or R_RDY primitive from the Physical layer.

NOTE It is possible that both the host and the device simultaneously request frame transmission by transmitting X_RDY. If the host receives X_RDY while transmitting X_RDY, the host shall back-off and enter the LR1: L_RcvChkRdy state, postponing its desired frame transmission until the device has completed its frame transmission and the bus is idle.

Transition LT1:LT3: When the host Link layer receives an R_RDY primitive from the Physical layer, the Link layer shall make a transition to the LT3: L_SendSOF state.

Transition LT1:LR2: When the host Link layer receives an X_RDY primitive from the Physical layer, the Link layer shall make a transition to the LR2: L_RcvWaitFifo state.

Transition LT1:LT1: AnyDword other than (R_RDY or X_RDY) received from Phy layer: When the host Link layer receives any DWORD other than an R_RDY or an X_RDY primitive from the Physical layer, the Link layer shall make a transition to the LT1: HL_SendChkRdy state. Any received errors such as 10b decoding errors and invalid primitives are ignored.

Transition LT1:LS1: When the host Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition, fail the attempted transfer, and make a transition to the LS1: L_NoCommErr state.

LT2: DL_SendChkRdy state: This state is entered when a frame transmission has been requested by the device Transport layer.

When in this state, the Link layer transmits an X_RDY primitive and waits for an R_RDY primitive from the Physical layer.

Transition LT2:LT3: When the device Link layer receives an R_RDY primitive from the Physical layer, the Link layer shall make a transition to the LT3: L_SendSOF state.

Transition LT2:LT2: AnyDword other than R_RDY received from Phy: When the device Link layer does not receive an R_RDY primitive from the Physical layer, the Link layer shall make a transition to the LT2: DL_SendChkRdy state.

Transition LT2:LS1: When the device Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LT3: L_SendSOF state: This state is entered an R_RDY primitive has been received from the Physical layer.

When in this state, the Link layer transmits an SOF primitive.

Transition LT3:LT4: When the device Link layer has transmitted an SOF primitive, the Link layer shall make a transition to the LT4: L_SendDATA state. Any received errors such as 10b decoding errors and invalid primitives are ignored.

Transition LT3:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition, fail the attempted transfer, and make a transition to the LS1: L_NoCommErr state.

Transition LT3:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1:L_IDLE state.

LT4: L_SendData state: This state is entered when an SOF primitive has been transmitted.

When in this state, the Link layer takes a data DWORD from the Transport layer, encodes the DWORD, and transmits it. The DWORD is also entered into the CRC calculation before encoding.

Transition LT4:LT4: More data to transmit and AnyDword other than (HOLD or DMAT or SYNC) received from Phy: When the Link layer receives any DWORD other than a HOLD, DMAT, or SYNC primitive from the Physical layer and the Transport layer indicates a DWORD is available for transfer, the Link layer may make a transition to the LT4: L_SendData state. (Use of DMAT is not recommended, see 15.4.6.) The DMAT signal is advisory and data transmission should be halted at the earliest opportunity but is not required to cease immediately. It is therefore allowable to stay in the LT4: L_SendData state when there is more data to transmit and DMAT is received. Any received errors such as 10b decoding errors and invalid primitives are ignored.

Transition LT4:LT5: When the Link layer receives a HOLD primitive from the Physical layer, the Link layer shall make a transition to the LT5: L_RcvrHold state.

Transition LT4:LT6: Data transmit not complete and data not ready to transmit and AnyDword other than SYNC received from Phy: When the Transport layer indicates that the next DWORD is not available to transfer and any DWORD other than a SYNC primitive has been received from the Physical layer, the Link layer shall make a transition to the LT6: L_SendHold state.

Transition LT4:LT7: DMAT received from Phy or data transmit complete and AnyDword other than SYNC received from Phy: When the Transport layer indicates that all data for the frame has been transferred and any DWORD other than a SYNC primitive has been received from the Physical layer, the Link layer shall make a transition to the LT7: L_SendCRC state. (Use of DMAT is not recommended, see 15.4.6) When the Link layer receives a DMAT primitive from the Physical layer, it may notify the Transport layer and terminate the transmission in progress as described in 15.4.6 and shall transition to the LT7: L_SendCRC state. The DMAT signal is advisory and data transmission should be halted at the earliest opportunity but is not required to cease immediately. It is therefore allowable to stay in the LT4: L_SendData state when there is more data to transmit and DMAT is received. Any received errors such as 10b decoding errors and invalid primitives are ignored.

Transition LT4:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall notify the Transport layer of the illegal transition error condition, fail the attempted transfer, and shall make a transition to the L1:L_IDLE state.

When the host Link layer receives notification from the host Transport layer that a Register FIS with C=0 is pending for transmission (typically due to SRST being toggled), the current transfer shall be aborted and a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the receiving Link layer to also transition to the L1: L_IDLE state. The Register FIS with C=0 may then be transmitted as a Register FIS Host to Device. When this condition is true, the associated transition has priority over all other transitions exiting this state.

Transition LT4:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition, fail the attempted transfer, and make a transition to the LS1: L_NoCommErr state.

LT5: L_RcvrHold state: This state is entered when a HOLD primitive has been received from the Physical layer.

When in this state, the Link layer shall transmit the HOLDA primitive.

Transition LT5:LT4: More data to transmit and AnyDword other than (HOLD or SYNC or DMAT) received from Phy: When the Link layer receives any DWORD other than a HOLD, SYNC, or a DMAT primitive from the Physical layer and the Transport layer indicates that a

DWORD is available for transfer, the Link layer shall make a transition to the LT4: L_SendData state.

Transition LT5:LT5: When the Link layer receives a HOLD primitive from the Physical layer or a decoding error was detected, the Link layer shall make a transition to the LT5: L_RcvrHold state.

Transition LT5:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall notify the Transport layer of the illegal transition error, fail the attempted transfer, and shall make a transition to the L1: L_IDLE state.

When the host Link layer receives notification from the host Transport layer that a Register FIS with C=0 is pending for transmission (typically due to SRST being toggled), the current transfer shall be aborted and a transition to the L1:L_IDLE state shall be made. The return to the L1:L_IDLE state results in the transmission of SYNC primitives, which causes the receiving Link layer to also transition to the L1:L_IDLE state. The Register FIS with C=0 may then be transmitted as a Register FIS Host to Device. When this condition is true, the associated transition has priority over all other transitions exiting this state.

Transition LT5:LT7: When the Link layer receives a DMAT primitive from the Physical layer, it may notify the Transport layer and terminate the transmission in progress as described in 15.4.6 and may transition to the LT7: L_SendCRC state. (Use of DMAT is not recommended, see 15.4.6.) The DMAT signal is advisory and data transmission should be halted at the earliest opportunity but is not required to cease immediately. It is therefore allowable to stay in the LT5: L_RcvrHold state when there is more data to transmit and DMAT is received. Any received errors such as 10b decoding errors and invalid primitives are ignored.

Transition LT5:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition, fail the attempted transfer, and make a transition to the LS1: L_NoCommErr state.

LT6: L_SendHold state: This state is entered when the Transport layer indicates a DWORD is not available for transfer and a HOLD primitive has not been received from the Physical layer.

When in this state, the Link layer shall transmit the HOLD primitive.

Transition LT6:LT4: More data ready to transmit and AnyDword other than (HOLD or SYNC) received from Phy: When the Link layer receives any DWORD other than a HOLD or SYNC primitive from the Physical layer and the Transport layer indicates that a DWORD is available for transfer, the Link layer shall make a transition to the LT4: L_SendData state.

Transition LT6:LT5: When the Link layer receives a HOLD primitive from the Physical layer and the Transport layer indicates a DWORD is available for transfer, the Link layer shall make a transition to the LT5: L_RcvrHold state.

Transition LT6:LT6: Data transmit not complete and data not ready to transmit and AnyDword other than (SYNC or DMAT) received from Phy: When the Transport layer indicates that a DWORD is not available for transfer and any DWORD other than a SYNC or DMAT primitive is received from the Physical layer, the Link layer shall make a transition to the LT6: L_SendHold state.

Transition LT6:LT7: When the Link layer receives a DMAT primitive from the Physical layer, it may notify the Transport layer and terminate the transmission in progress as described in 15.4.6 and transition to the LT7:L_SendCRC state. (Use of DMAT is not recommended, see 15.4.6.) The DMAT signal is advisory and data transmission should be halted at the earliest opportunity but is not required to cease immediately. It is therefore allowable to stay in the LT6: L_SendHold state when there is more data to transmit and DMAT is received. Any received errors such as 10b decoding errors and invalid primitives are ignored.

Transition LT6:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall make a transition to the L1: L_IDLE state. The Transport layer shall be notified of the illegal transition error condition and fail the attempted transfer.

When the host Link layer receives notification from the host Transport layer that a Register FIS with C=0 is pending for transmission (typically due to SRST being toggled), the current transfer shall be aborted and a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the receiving Link layer to also transition to the L_IDLE state. The Register FIS with C=0 may then be transmitted as a Register FIS Host to Device. When this condition is true, the associated transition has priority over all other transitions exiting this state.

Transition LT6:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition, fail the attempted transfer, and make a transition to the LS1: L_NoCommErr state.

LT7: L_SendCRC state: This state is entered when the Transport layer indicates that all data DWORDs have been transferred for this frame.

When in this state, the Link layer shall transmit the calculated CRC for the frame.

Transition LT7:LT8: When the CRC has been transmitted, the Link layer shall make a transition to the LT8: L_SendEOF state.

Transition LT7:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition, fail the attempted transfer, and make a transition to the LS1: L_NoCommErr state.

Transition LT7:L1: If the Phy is Ready and the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall notify the Transport layer of the illegal transition error condition, fail the attempted transfer, and shall make a transition to the L1:L_IDLE state.

LT8: L_SendEOF state: This state is entered when the CRC for the frame has been transmitted.

When in this state, the Link layer shall transmit the EOF primitive.

Transition LT8:LT9: When the EOF primitive has been transmitted, the Link layer shall make a transition to the LT9: L_Wait state.

Transition LT8:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition, fail the attempted transfer, and make a transition to the LS1: L_NoCommErr state.

Transition LT8:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall notify the Transport layer of the illegal transition error condition, fail the attempted transfer, and shall make a transition to the L1:L_IDLE state.

LT9: L_Wait state: This state is entered when the EOF primitive has been transmitted.

When in this state, the Link layer shall transmit the WTRM primitive.

Transition LT9:L1: When the Link layer receives a R_OK primitive from the Physical layer, the Link layer shall notify the Transport layer and make a transition to the L1: L_IDLE state.

Transition LT9:L1: When the Link layer receives a R_ERR primitive from the Physical layer, the Link layer shall notify the Transport layer, fail the attempted transfer, and make a transition to the L1: L_IDLE state.

Transition LT9:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall notify the Transport layer, fail the attempted transfer, and make a transition to the L1: L_IDLE state. If the Host transport requests to transmit a Register HD FIS with C=0, the transport shall make a transition to the L1: L_IDLE state.

Transition LT9:LT9: AnyDword other than (R_OK or R_ERR , or SYNC) received from Phy: When the Link layer receives any DWORD other than an R_OK, R_ERR, or SYNC primitive from the Physical layer, the Link layer shall make a transition to the LT9: L_Wait state.

Transition LT9:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition, fail the attempted transfer, and make a transition to the LS1: L_NoCommErr state.

15.6.3.3 Link receive state diagram

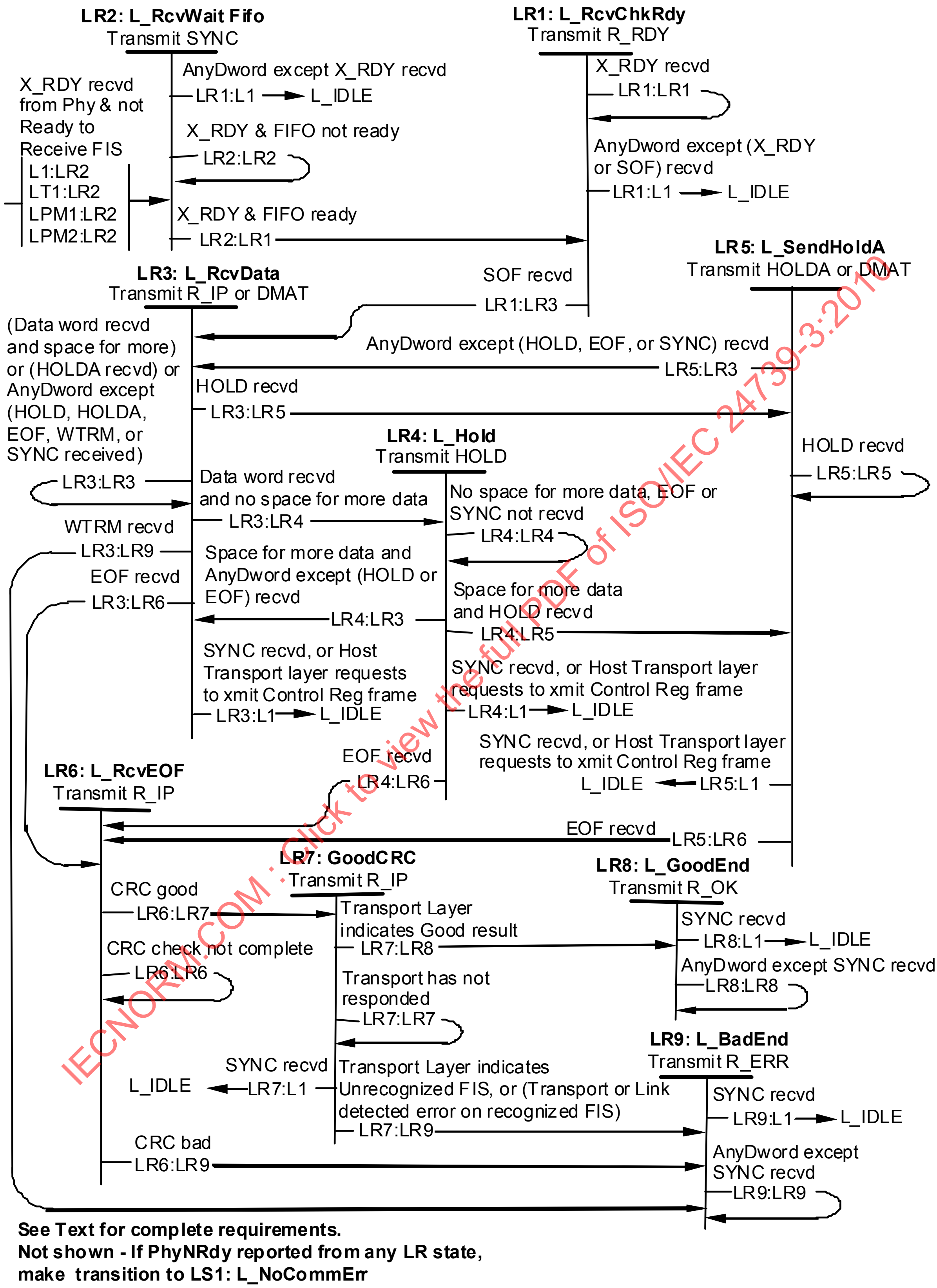


Figure 60 – Link receive state diagram (States LR1-LR9)

In any state of the Link receive state diagram, if the Link layer detects the Phy layer is not ready, the Link layer will notify the Transport layer, fail the attempted transfer and make a transition to LS1: L_NoCommErr. These transitions are not shown in the state diagram to improve clarity.

LR1: L_RcvChkRdy state: This state is entered when an X_RDY primitive has been received from the Physical layer.

When in this state, the Link layer shall transmit an R_RDY primitive and wait for an SOF primitive from the Physical layer.

Transition LR1:LR1: When the Link layer receives an X_RDY primitive from the Physical layer, the Link layer shall make a transition to the LR1: L_RcvChkRdy state.

Transition LR1:LR2: When the Link layer receives an SOF primitive from the Physical layer, the Link layer shall make a transition to the LR2: L_RcvData state.

LR1:L1: Any DWORD other than (X_RDY or SOF) received from Phy: When the Link layer receives any DWORD other than an X_RDY or SOF primitive from the Physical layer, the Link layer shall notify the Transport layer of the condition and make a transition to the L1: L_IDLE state.

Transition LR1:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall Notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR2: L_RcvWaitFifo state: This state is entered when an X_RDY has been received, and the FIFO is not ready to receive an FIS.

When in this state, the Link layer shall transmit the SYNC primitive.

Transition LR2:LR1: When the Link layer receives a X_RDY primitive from the Physical layer and the FIFO is ready to accept data, the Link layer shall make a transition to the LR1: L_RcvChkRdy state.

Transition LR2:LR2: When the Link layer receives a X_RDY primitive from the Physical layer and the FIFO is not ready to accept data, the Link layer shall make a transition to the LR2: L_RcvWaitFifo state.

Transition LR2:L1: When the Link layer receives any DWORD other than an X_RDY primitive from the Physical layer, the Link layer shall notify the Transport layer of the condition and make a transition to the L1: L_IDLE state.

Transition LR2:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR3: L_RcvData state: This state is entered when an SOF primitive has been received from the Physical layer.

When in this state, the Link layer receives an encoded character sequence from the Physical layer, decodes it into a DWORD, and passes the DWORD to the Transport layer. The DWORD is also entered into the CRC calculation. When in this state the Link layer either transmits a R_IP primitive to signal transmission to continue or transmits a DMAT primitive (Use of DMAT is not recommended, see 15.4.6) to signal the transmitter to terminate the transmission.

Transition LR3:LR3: (Data DWORD received from Phy and FIFO space) or HOLDA received from Phy: When the Transport layer indicates that space is available in its FIFO, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR3:LR3: AnyDword other than (HOLD or EOF or HOLDA or SYNC or WTRM) received from Phy: When the Link layer receives any DWORD other than a HOLD, HOLDA, EOF or SYNC or WTRMprimitive from the Physical layer, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR3:LR4: When the Transport layer indicates that sufficient space is not available in its FIFO, the Link layer shall make a transition to the LR4: L_Hold state.

Transition LR3:LR5: When the Link layer receives a HOLD primitive from the Physical layer, the Link layer shall make a transition to the LR5: L_SendHoldA state.

Transition LR3:LR6: When the Link layer receives an EOF primitive from the Physical layer, the Link layer shall make a transition to the LR6: L_RcvEOF state.

Transition LR3:LR9: When the Link layer receives a WTRM primitive from the Physical layer, the Link layer shall make a transition to the LR9: L_BadEnd state.

Transition LR3:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall notify the Transport layer that reception was aborted and shall make a transition to the L1: L_IDLE state.

When the host Link layer receives notification from the host Transport layer that a Control Register Frame is pending for transmission (typically due to SRST being toggled), a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the transmitting Link layer to also transition to the L1:L_IDLE state. The Register FIS with C=0 may then be transmitted as a Register FIS from host to device.

Transition LR3:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR4: L_Hold state: This state is entered when the Transport layer indicates that sufficient space is not available in its receive FIFO.

When in this state, the Link layer shall transmit the HOLD primitive and may receive an encoded character from the Physical layer.

Transition LR4:LR3: FIFO space available and AnyDword other than HOLD or EOF received from Phy: When the Link layer receives any DWORD other than a HOLD primitive or EOF primitive from the Physical layer and the Transport layer indicates that sufficient space is now available in its receive FIFO, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR4:LR5: When the Link layer receives a HOLD primitive from the Physical layer and the Transport layer indicates that space is now available in its FIFO, the Link layer shall make a transition to the LR5: L_SendHoldA state.

Transition LR4:LR6: When the Link layer receives a EOF primitive from the Physical layer, the Link layer shall make a transition to the LR6: L_RcvEOF state. Note that due to pipeline latency, an EOF may be received when in the L_Hold state in which case the receiving Link shall use its FIFO headroom to receive the EOF and close the frame reception.

Transition LR4:LR4: No FIFO space available and EOF not received from Phy and SYNC not received from Phy and PhyRdy: When the Transport layer indicates that there is not sufficient space available in its FIFO and the Physical layer is ready, the Link layer shall make a transition to the LR4: L_Hold state.

Transition LR4:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR4:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1: L_IDLE state.

When the host Link layer receives notification from the host Transport layer that a Control Register Frame is pending for transmission (typically due to SRST being toggled), a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the transmitting Link layer to also transition to the L1:L_IDLE state. The Register FIS with C=0 may then be transmitted as a Register FIS from host to device.

LR5: L_SendHoldA state: This state is entered when a HOLD primitive has been received from the Physical layer.

When in this state, the Link layer shall either transmit the HOLDA primitive to signal transmission to proceed when the transmitter becomes ready or transmit a DMAT primitive (Use of DMAT is not recommended, see 15.4.6) to signal the transmitter to terminate the transmission.

Transition LR5:LR3: AnyDword other than (HOLD or EOF or SYNC) received from Phy: When the Link layer receives any DWORD other than a HOLD or EOF or SYNC primitive from the Physical layer, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR5:LR5: When the Link layer receives a HOLD primitive from the Physical layer, the Link layer shall make a transition to the LR5: L_SendHoldA state.

Transition LR5:LR6: When the Link layer receives a EOF primitive from the Physical layer, the Link layer shall make a transition to the LR6: L_RcvEOF state.

Transition LR5:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall make a transition to the L1: L_IDLE state. The Transport layer shall be notified of the illegal transition error condition.

When the host Link layer receives notification from the host Transport layer that a Control Register Frame is pending for transmission (typically due to SRST being toggled), a transition to the L_IDLE state shall be made. The return to the L_IDLE state results in the transmission of SYNC primitives, which causes the transmitting Link layer to also transition to the L1:L_IDLE state. The Register FIS with C=0 may then be transmitted as a Register FIS from host to device.

Transition LR5:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR6: L_RcvEOF state: This state is entered when the Link layer has received an EOF primitive from the Physical layer.

When in this state, the Link layer shall check the calculated CRC for the frame and transmit one or more R_IP primitives.

Transition LR6:LR6: If the CRC calculation and check is not yet completed, the Link layer shall make a transition to the LR6: L_RcvEOF state.

Transition LR6:LR7: When the CRC indicates no error, the Link layer shall notify the Transport layer and make a transition to the LR7: L_GoodCRC state.

Transition LR6:LR9: When the CRC indicates an error has occurred, the Link layer shall notify the Transport layer and make a transition to the LR9: L_BadEnd state.

Transition LR6:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR7: L_GoodCRC state: This state is entered when the CRC for the frame has been checked and determined to be good.

Upon entering this state for the first time, the Link layer shall notify the Transport layer that the CRC for this frame is valid. When in this state, the Link layer shall wait for the Transport Layer to check the frame and transmit one or more R_IP primitives.

Transition LR7:LR8: When the Transport Layer indicates a good result, the Link Layer shall transition to the LR8: L_GoodEnd state.

Transition LR7:LR9: When the Transport Layer indicates an unrecognized FIS, the Link Layer shall transition to the LR9: L_BadEnd state.

When the Transport layer or Link layer indicates an error was encountered during the reception of the recognized FIS, the Link layer shall transition to the LR9: L_BadEnd state.

Transition LR7:LR7: If the Transport Layer has not supplied status, then the Link Layer shall transition to the LR7: L_GoodCRC state.

Transition LR7:LS1: When the Link layer detects that the Physical layer is not ready, the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR7:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall notify the Transport layer of the illegal transition error condition and shall make a transition to the L1: L_IDLE state.

LR8: L_GoodEnd state: This state is entered when the CRC for the frame has been checked and determined to be good.

When in this state, the Link layer shall transmit the R_OK primitive.

Transition LR8:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall make a transition to the L1: L_IDLE state.

Transition LR8:LR8: When the Link layer receives any DWORD other than a SYNC primitive from the Physical layer, the Link layer shall make a transition to the LR8: L_GoodEnd state.

Transition LR8:LS1: When the Link layer detects that the Physical layer is not ready, the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR9: L_BadEnd state: This state is entered when the CRC for the frame has been checked and determined to be bad or when the Transport layer has notified the Link layer that the received FIS is invalid.

When in this state, the Link layer shall transmit the R_ERR primitive.

Transition LR9:L1: When the Link layer receives a SYNC primitive from the Physical layer, the Link layer shall make a transition to the L1: L_IDLE state.

Transition LR9:LR9: When the Link layer receives any DWORD other than a SYNC primitive from the Physical layer, the Link layer shall make a transition to the LR8:BadEnd state.

Transition LR9:LS1: When the Link layer detects that the Physical layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

15.6.3.4 Link power mode state diagram

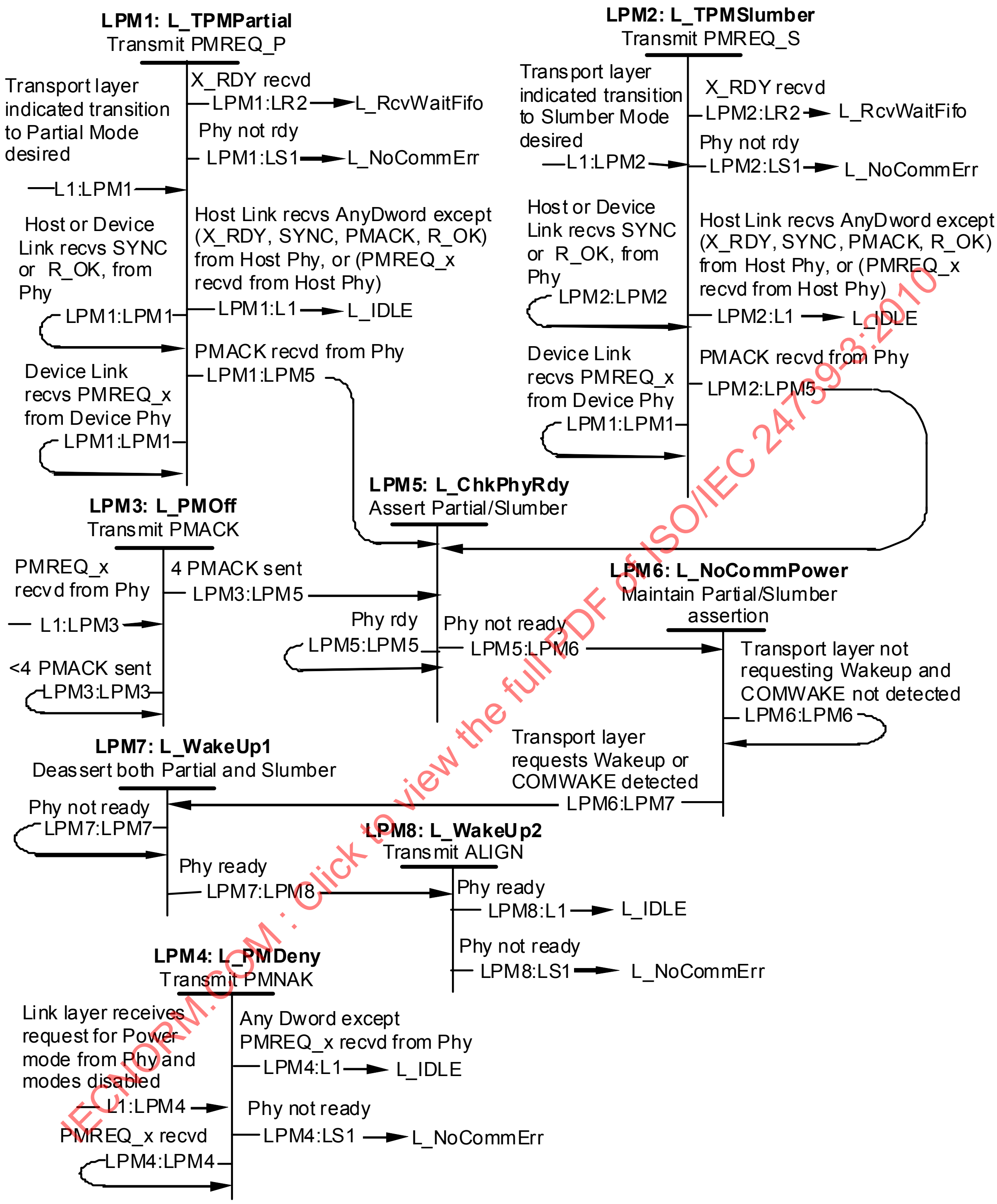


Figure 61 – Link power mode state diagram (States LPM1-LPM8)

LPM1: L_TPMPartial state: This state is entered when the Transport layer has indicated that a transition to the Partial power state is desired. When in this state a PMREQ_P primitive shall be transmitted

Transition LPM1:LPM5: When the Link layer receives a PMACK primitive a transition to the LPM5: L_ChkPhyRdy state shall be made.

Transition LPM1:LR2: If the Link layer receives an X_RDY primitive a transition shall be made to the LR2: L_RcvWaitFifo state. This aborts the request from the Transport layer to enter a power mode. A status indication to the Transport layer of this event is required.

Transition LPM1:LPM1: If the Link layer receives a SYNC or R_OK primitive, then it is assumed that the opposite side has not yet processed the PMREQ_P primitive yet and time is needed. A transition to the LPM1: L_TPMPartial state shall be made.

Transition LPM1:LPM1: The host Link layer shall not make this transition as it applies only to the device Link layer. If the device Link layer receives a PMREQ_P or PMREQ_S primitive from the host, it shall remain in this state by transitioning back to LPM1: L_TPMPartial.

Transition LPM1:L1: AnyDword other than (PMACK or X_RDY or SYNC or R_OK or PMREQ_P or PMREQ_S) received from Phy layer: If the host Link layer receives any DWORD from the Physical layer other than a PMACK, X_RDY, SYNC, or R_OK primitive, then the request to enter the partial state is aborted and a transition to L1: L_IDLE shall be made. This transition includes the case where a PMNAK is received. The device link layer shall not make this transition if it receives a PMREQ_P or PMREQ_S primitive while in this state.

Transition LPM1:LS1: If the Link layer detects that the Physical layer has become not ready, this is interpreted as an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1: L_NoCommErr state.

LPM2: L_TPMSlumber state: This state is entered when the Transport layer has indicated that a transition to the Slumber power state is desired. When in this state a PMREQ_S primitive shall be transmitted.

Transition LPM2:LPM5: When the Link layer receives a PMACK primitive, a transition to the LPM5: L_ChkPhyRdy state shall be made.

Transition LPM2:LR2: If the Link layer receives an X_RDY primitive, a transition to the LR2: L_RcvWaitFifo state shall be made. This aborts the request from the Transport layer to enter a power mode. A status indication to the Transport layer of this event is required.

Transition LPM2:LPM2: If the Link layer receives a SYNC or R_OK primitive, then it is assumed that the opposite side has not yet processed the PMREQ_S primitive yet and time is needed. The transition to the LPM2: L_TPMSlumber state shall be made.

Transition LPM2:LPM2: The host Link layer shall not make this transition as it applies only to the device Link layer. If the device Link layer receives a PMREQ_P or PMREQ_S primitive from the host, it shall remain in this state by transitioning back to LPM2: L_TPMSlumber.

Transition LPM2:L1: AnyDword other than (PMACK or X_RDY or SYNC or R_OK or PMREQ_P or PMREQ_S) received from Phy layer: If the host Link layer receives any DWORD from the Physical layer other than a PMACK, X_RDY, SYNC or R_OK primitive, then the request to enter the slumber state is aborted and a transition to L1: L_IDLE shall be made. This transition includes the case where a PMNAK is received. The device link layer shall not make this transition if it receives a PMREQ_P or PMREQ_S primitive while in this state.

Transition LPM2:LS1: If the Link layer detects that the Physical layer has become not ready, this is interpreted as an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1:L_NoCommErr state.

LPM3: L_PMOff state: This state is entered when either a PMREQ_S or PMREQ_P primitive was received by the Physical layer. A flag is set according to whether PMREQ_P or PMREQ_S was received from the Physical layer. The Link layer transmits a PMACK primitive for each execution of this state.

Transition LPM3:LPM5: If four PMACK primitives have been transmitted, a transition shall be made to the LPM5:L_ChkPhyRdy state.

Transition LPM3:LPM3: If less than four PMACK primitives have been transmitted, a transition shall be made to LPM3:L_PMOff state.

LPM4: L_PMDeny state: This state is entered when any primitive is received by the Link layer to enter a power mode and power modes are currently disabled. The Link layer shall transmit a PMNAK primitive to inform the opposite end that a power mode is not allowed.

Transition LPM4:LPM4: If the Link layer continues to receive a request to enter any power mode, then a transition back to the same LPM4: L_PMDeny state shall be made.

Transition LPM4:L1: AnyDword other than (PMREQ_P or PMREQ_S) received from Phy layer: If the Link layer receives any DWORD other than a power mode request primitive, then the Link layer assumes that the power mode request has been removed and shall make a transition to the L1: L_IDLE state.

Transition LPM4:LS1: If the Link layer detects that the Physical layer has become not ready, this is interpreted as an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1: L_NoCommErr state.

LPM5: L_ChkPhyRdy state: This state is entered whenever it is desired for the Physical layer to enter a low power condition. For each execution in this state a request is made to the Physical layer to enter the state and deactivate the PhyRdy signal. Partial or Slumber is asserted to the Phy as appropriate.

Transition LPM6:LPM5: If the Physical layer has not yet processed the request to enter the power saving state and not deactivated the PhyRdy signal, then the Link layer shall remain in the LPM5: L_ChkPhyRdy state and continue to request the Physical layer to enter the power mode state.

Transition LPM6:LPM6: When the Physical layer has processed the power mode request and has deactivated the PhyRdy signal, then a transition shall be made to the LPM6: L_NoCommPower state.

LPM6: L_NoCommPower state: This state is entered when the Phy has negated its PhyRdy signal indicating that it is in either Partial or Slumber state. In this state, the Link layer waits for the out of band detector to signal reception of the COMWAKE sequence, or for the Transport layer to request a wakeup.

Transition LPM6:LPM7: If the Transport layer requests a wakeup or the out of band signal detector indicates reception of the COMWAKE signal, then a transition shall be made to LPM7: L_WakeUp1

Transition LPM6:LPM6: If the Transport layer does not request a wakeup and the out of band detector does not indicate reception of the COMWAKE signal, then a transition shall be made to LPM6: L_NoCommPower.

LPM7: L_WakeUp1 state: This state is entered when the Transport layer has initiated a wakeup. In this state, the Link layer shall deassert both Partial and Slumber to the Phy, and wait for the PhyRdy signal from the Phy to be asserted. While in this state the Phy is performing the wakeup sequence.

Transition LPM7:LPM8: When the Phy asserts its PhyRdy signal, a transition shall be made to LPM8: L_WakeUp2.

Transition LPM7:LPM7: When the Phy remains not ready, a transition shall be made to LPM7: L_WakeUp1.

LPM8: L_WakeUp2 state: This state is entered when the Phy has acknowledged an initiated wakeup request by asserting its PhyRdy signal. In this state, the Link layer shall transmit the ALIGN sequence and transition to the L1: L_IDLE state.

Transition LPM8:L1: If the Phy keeps PhyRdy asserted, a transition shall be made to the L1: L_IDLE state.

Transition LPM8:LS1: If the Phy deasserts PhyRdy, this is an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1: L_NoCommErr state.

16 Serial interface Transport layer

16.1 Transport layer overview

16.1.1 General

The Transport layer need not be cognizant of how frames are transmitted and received. The Transport layer constructs Frame Information Structures (FISs) for transmission and decomposes received Frame Information Structures. Host and device Transport layer state differ in that the source of the FIS content differs. The Transport layer maintains no context in terms of ATA commands or previous FIS content.

16.1.2 FIS construction

When requested to construct an FIS by a higher layer, the Transport layer provides the following services.

- Gathers FIS content based on the type of FIS requested.
- Places FIS content in the proper order.
- Notifies the Link layer of required frame transmission and passes FIS content to Link.
- Manages Buffer/FIFO flow, notifies Link of required flow control.
- Receives frame receipt acknowledge from Link layer.
- Reports good transmission or errors to requesting higher layer.

16.1.3 FIS decomposition

When an FIS is received from the Link layer, the Transport layer provides the following services.

- Receives the FIS from the Link layer.
- Determines FIS type.
- Distributes the FIS content to the locations indicated by the FIS type.
- For the host Transport layer, receipt of an FIS may also cause the construction of an FIS to be returned to the device.
- Reports good reception or errors to higher layer.

16.2 Frame Information Structure (FIS)

16.2.1 Overview

A frame is a group of DWORDs that convey information between host and device as described previously. Primitives are used to define the boundaries of the frame and may be inserted to control the rate of the information flow. This subclause describes the information content of the frame - heretofore referred to as payload - and assumes the reader is aware of the Primitives that are needed to support the information content.

16.2.2 Payload content

The type and layout of the payload is indicated by the Frame Information Type field located in byte 0 of the first DWORD of the payload. For Shadow Command Block and Shadow Control Block registers, see Clause 5.

16.2.3 FIS types

16.2.3.1 General

The following sections define the structure of each individual FIS.

16.2.3.2 Register, Host to Device

	31–24	23–16	15	14	13	12–8	7–0
0	Features	Command	C	R	R	Reserved (0)	FIS Type (27h)
1	Device	LBA High	LBA Mid				LBA Low
2	Features (exp)	LBA High (exp)	LBA Mid (exp)				LBA Low (exp)
3	Control	Reserved (0)	Sector Count (exp)				Sector Count
4	Reserved (0)	Reserved (0)	Reserved (0)				Reserved (0)

Figure 62 – Register, Host to Device FIS layout

The following Field definitions apply.

FIS Type - Set to a value of 27h. Defines the rest of the FIS fields. Defines the length of the FIS as five DWORDs.

C - This bit is set to one when the register transfer is due to an update of the Command register. The bit is cleared to zero when the register transfer is due to an update of the Device Control register.

Setting C =1 and SRST=1 in the Device Control Field is invalid and will result in inderminate behavior.

Command - Contains the contents of the Command register of the Shadow Command Block.

LBA Low - Contains the contents of the LBA Low register of the Shadow Command Block.

LBA Low (exp) - Contains the contents of the expanded address field of the Shadow Command Block.

Control - Contains the contents of the Device Control register of the Shadow Command Block.

LBA Mid - Contains the contents of the LBA Mid register of the Shadow Command Block.

LBA Mid (exp) - Contains the contents of the expanded address field of the Shadow Command Block

LBA High - Contains the contents of the LBA High register of the Shadow Command Block.

LBA High (exp) - Contains the contents of the expanded address field of the Shadow Command Block.

Device - Contains the contents of the Device register of the Shadow Command Block.

Features - Contains the contents of the Features register of the Shadow Command Block.

Features (exp) - Contains the contents of the expanded address field of the Shadow Command Block.

R - Reserved.

Sector Count - Contains the contents of the Sector Count register of the Shadow Command Block.

Sector Count (exp) - Contains the contents of the expanded address field of the Shadow Command Block.

16.2.3.3 Description

The Register - Host to Device FIS is used to transfer the contents of the Shadow Command Block and Shadow Control Block from the host to the device, see Figure 62. This is the mechanism for issuing the parallel implementation of ATA commands to the device.

16.2.3.4 Transmission

Transmission of a Register - Host to Device FIS is initiated by a write operation to either the Command register, or a write to the Device Control register with a value different than is currently in the Device Control register. There are BIOS and drivers that write the Device Control register to enable the interrupt just prior to issuing a command. To avoid unnecessary overhead, this FIS should be transmitted to the device only upon a change of state from the previous value in the host adapter's Shadow Control Block. Upon initiating transmission, the current contents of the Shadow Command Block and Shadow Control Block are transmitted and the C bit in the FIS is set according to whether the transmission was a result of the Command register being written or the Device Control register being written. The host adapter shall set the BSY bit in the Shadow Status register to one within 400 ns after the write operation to the Command register that initiated the transmission. The host adapter shall set the BSY bit in the Shadow Status register to one within 400 ns after a write operation to the Device Control register if the write to the Device Control register changes the state of the SRST bit from 0 to 1 but shall not set the BSY bit in the Shadow Status register for writes to the Device Control register that do not change the state of the SRST bit from 0 to 1.

It is important to note that serial implementations of ATA host adapters enforce the same access control to the Shadow Command Block and Shadow Control Block as the parallel implementation of ATA devices enforce to the Command Block Registers. When either BSY or DRQ is set in the Status Register the host should not write the Command Block registers. Any write to the Command Block Registers when BSY or DRQ is set will result in indeterminate behavior unless the write is to issue a DEVICE RESET command.

16.2.3.5 Reception

Upon reception of a valid Register - Host to Device FIS the device updates its local copy of the Command and Control Block Register contents and, if the C bit is set to 1, initiates execution of the command indicated in the command register.

16.2.4 Register, Device to Host

16.2.4.1 General

	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12 11 10 9 8	7 6 5 4 3 2 1 0
0	Error	Status	R	I	R	Reserved (0)	FIS Type (34h)
1	Device	LBA High	LBA Mid				LBA Low
2	Reserved (0)	LBA High (exp)	LBA Mid (exp)				LBA Low (exp) (0)
3	Reserved (0)	Reserved (0)	Sector Count (exp)				Sector Count
4	Reserved (0)	Reserved (0)	Reserved (0)				Reserved (0)

Figure 63 – Register, Device to Host FIS layout

The following Field definitions apply.

FIS Type - Set to a value of 34h. Defines the rest of the FIS fields. Defines the length of the FIS as five DWORDs.

LBA Low - Contains the contents to be placed in the LBA Low register of the Shadow Command Block.

LBA Low (exp) - Contains the contents of the expanded address field of the Shadow Command Block.

LBA Mid - Contains the contents to be placed in the LBA Mid register of the Shadow Command Block.

LBA Mid (exp) - Contains the contents of the expanded address field of the Shadow Command Block.

LBA High - Contains the contents to be placed in the LBA High register of the Shadow Command Block.

LBA High (exp) - Contains the contents of the expanded address field of the Shadow Command Block.

Device - Contains the contents to be placed in the Device register of the Shadow Command Block.

Error - Contains the contents to be placed in the Error register of the Shadow Command Block.

I - Interrupt bit. This bit reflects the Interrupt Pending state of the device. Devices shall not modify the behavior of this bit based on the state of the nIEN bit received in Register Host to Device FISes.

NOTE Some implementations prior to this standard modify the behavior of this bit based on the state of nIEN in received Register Host to Device FISes, see 18.2.

R - Reserved

Sector Count - Contains the contents to be placed in the Sector Count register of the Shadow Command Block.

Sector Count (exp) - Contains the contents of the expanded address field of the Shadow Command Block.

Status - Contains the contents to be placed in the Status (and Alternate status) Register of the Shadow Command Block.

16.2.4.2 Description, Register Device to Host FIS

The Register - Device to Host FIS is used to by the device to update the contents of the host adapter's Shadow Command Block and Shadow Control Block, see Figure 63. This is the mechanism by which devices indicate command completion status or otherwise change the contents of the host adapter's Shadow Command Block and Shadow Control Block.

16.2.4.3 Transmission

Transmission of a Register - Device to Host FIS is initiated by the device in order to update the contents of the host adapter's Shadow Command Block and Shadow Control Block. Transmission of the Register - Device to Host FIS is typically as a result of command completion by the device.

The Register - Device to Host FIS shall only be used to set the SERV bit in the Status Register to request service for a bus released command if the BSY bit or the DRQ bit is currently set in the Status Register; the Set Device Bits - Device to Host FIS shall be used to set the SERV bit when the BSY bit and DRQ bit are both cleared to 0 in the Status Register. The SERV bit transmitted with the Register - Device to Host FIS will be written to the shadow Status Register and so the bit should accurately reflect the state of pending service requests when the FIS is transmitted as a result of a command completion by the device.

16.2.4.4 Reception

Upon reception of a valid Register - Device to Host FIS the received register contents are transferred to the host adapter's Shadow Command Block and Shadow Control Block.

If the BSY bit and the DRQ bit in the Shadow Status Register are both cleared when a Register - Device to Host FIS is received by the host adapter, then the host adapter shall discard the contents of the received FIS and not update the contents of any Shadow Command Block or Shadow Control Block.

16.2.5 Set Device Bits - Device to Host

16.2.5.1 General

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Error								R	Status Hi			R	Status Lo			R	I	R	Reserved (0)					FIS Type (A1h)							
1	Reserved (0)																															

Figure 64 – Set Device Bit, Device to Host FIS layout

The following Field definitions apply.

FIS Type- Set to a value of A1h. Defines the rest of the FIS fields. Defines the length of the FIS as two DWORDs.

I- Interrupt Bit. This bit signals the host adapter to enter an Interrupt Pending state if both the BSY bit and the DRQ bit in the Shadow Status register are zero when the frame is received. Devices shall not modify the behavior of this bit based on the state of the nIEN bit received in Register Host to Device FISes.

NOTE Some implementations prior to this standard modify the behavior of this bit based on the state of nIEN in received Register Host to Device FISes, see 18.2.

Error- Contains the contents to be placed in the Error register of the Shadow Command Block.

Status-Hi- Contains the contents to be placed in bits 6, 5 and 4 of the Status register of the Shadow Command Block.

Status-Lo- Contains the contents to be placed in bits 2,1 and 0 of the Status register of the Shadow Command Block.

R- Reserved.

16.2.5.2 Description Set Device Bits Device to Host FIS

The Set Device Bits - Device to Host FIS is used by the device to load Shadow Command Block bits for which the device has exclusive write access, see Figure 64. These bits are the eight bits of the Error register and six of the eight bits of the Status register. This FIS does not alter bit 7, BSY or bit 3, DRQ, of the Shadow Status register.

The FIS includes a bit to signal the host adapter to generate an interrupt if the BSY bit and the DRQ bit in the Shadow Status Register are both cleared to zero when this FIS is received.

Some of the serial-to-parallel bridge solutions may not support this FIS. Upon reception, such bridge solutions may process this FIS as if it were an invalid FIS type and return the R_ERR end of frame handshake. Read and Write DMA Queued commands may not be possible if this FIS is not implemented.

Transmission

The device transmits a Set Device Bits - Device to Host to alter the Error register and bits in the Status register other than BSY and DRQ in the Shadow Command Block. This FIS should be used by the device to set the SERV bit in the Status register to request service for a bus released command. When used for this purpose the device shall set the Interrupt bit to one.

The Register - Device to Host FIS shall only be used to set the SERV bit in the Status Register to request service for a bus released command if the BSY bit or the DRQ bit is currently set in the Status Register; the Set Device Bits - Device to Host FIS shall be used to set the SERV bit when the BSY bit and DRQ bit are both cleared to 0 in the Status Register. The SERV bit transmitted with the Register - Device to Host FIS will be written to the shadow Status Register and so the bit should accurately reflect the state of pending service requests when the FIS is transmitted as a result of a command completion by the device.

16.2.5.3 Reception

Upon receiving a Set Device Bits - Device to Host, the host adapter shall load the data from the Error field into the Shadow Error register, the data from the Status-Hi field into bits 6, 5 and 4, of the Shadow Status register, and the data from the Status-Lo field into bits 2, 1 and 0 of the Shadow Status register. Bit 7, BSY and bit 3, DRQ, of the Shadow Status register shall not be changed. If the I bit in the FIS is set to a one, and if both the BSY bit and the DRQ bit in the Shadow Status register are cleared to zero when this FIS is received, then the host adapter shall enter an Interrupt Pending state.

16.2.6 DMA Activate, Device to Host

16.2.6.1 General

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved (0)								Reserved (0)								R	R	R	Reserved (0)					FIS Type (39h)							

Figure 65 – DMA Activate, Device to Host FIS layout

The following Field definitions apply.

FIS Type - Set to a value of 39h. Defines the rest of the FIS fields. Defines the length of the FIS as one DWORD.

R - Reserved

16.2.6.2 Description

The DMA Activate - Device to Host FIS is used by the device to signal the host to proceed with a DMA data transfer of data from the host to the device, see Figure 65. This is similar to the DMARQ mechanism by which a parallel implementation of ATA device signals its readiness to receive DMA data from the host.

A situation may arise where the host needs to send multiple Data FISes in order to complete the overall data transfer request. The host shall wait for a successful reception of a DMA Activate FIS before sending each of the Data FIS's that are needed.

16.2.6.3 Transmission

The device transmits a DMA Activate - Device to Host to the host in order to initiate the flow of DMA data from the host to the device as part of the data transfer portion of a corresponding DMA write command. Before transmitting this FIS, the device shall be prepared to receive a Data - Host to Device FIS from the host with the DMA data for the corresponding command.

16.2.6.4 Reception

Upon receiving a DMA Activate - Device to Host, if the host adapter's DMA controller has been programmed and armed, the host adapter shall initiate the transmission of a Data FIS and shall transmit in this FIS the data corresponding to the host memory regions indicated by the DMA controller's context. If the host adapter's DMA controller has not yet been programmed and armed, the host adapter shall set an internal state indicating that the DMA controller has been activated by the device, and as soon as the DMA controller has been programmed and armed, a Data FIS shall be transmitted to the device with the data corresponding to the host memory regions indicated by the DMA controller context.

16.2.7 First Party DMA Setup, Device to Host or Host to Device (bidirectional)

16.2.7.1 General

<table>
<tr><th></th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="8">Reserved (0)</td><td colspan="8">Reserved (0)</td><td>R</td><td>I</td><td>D</td><td colspan="5">Reserved (0)</td><td colspan="8">FIS Type (41h)</td></tr>
<tr><td>1</td><td colspan="32">DMA Buffer Identifier Low</td></tr>
<tr><td>2</td><td colspan="32">DMA Buffer Identifier High</td></tr>
<tr><td>3</td><td colspan="32">Reserved (0)</td></tr>
<tr><td>4</td><td colspan="32">DMA Buffer Offset</td></tr>
<tr><td>5</td><td colspan="32">DMA Transfer Count</td></tr>
<tr><td>6</td><td colspan="32">Reserved (0)</td></tr>
</table>

Figure 66 – First Party DMA Setup – Device to Host FIS layout

The following Field definitions apply.

FIS Type - Set to a value of 41h. Defines the rest of the FIS fields. Defines the total length of the FIS as seven DWORDs.

D - Indicates whether subsequent data transferred after this FIS is from sender to recipient or from recipient to sender 1 = sender to recipient, 0 = recipient to sender .

DMA Buffer Identifier Low/High - This field is used to identify a DMA buffer region in host and device memory. The contents are not described in this standard and are command protocol dependent. The buffer identifier is supplied by the host by a command protocol specific means to the device and the device echoes it back to the host, or the buffer identifier is supplied by the device to the host and the host echoes it back to the device. This allows the implementation to pass a physical address or, in more complex implementations, the buffer identifier could be a scatter gather list or other information that can identify a DMA “channel”.

DMA Buffer Offset - This is the byte offset into the buffer. Bits <1:0> shall be zero as the addressing granularity in on a DWORD boundary.

DMA Transfer Count -This is the number of bytes that will be transferred. Bit zero shall be zero.

I Interrupt - If the Interrupt bit is set to 1 an Interrupt Pending shall be generated when the DMA transfer count is exhausted.

R - Reserved

16.2.7.2 Description

The First Party DMA Setup - Device to Host or Host to Device FIS is the mechanism by which First Party DMA access to memory is initiated, see Figure 66. This FIS is used to request the host or device to program its DMA controller before transferring data. The FIS allows the memory regions to be abstracted (depending on implementation) by having memory regions referenced via a base memory descriptor representing a memory region to which the host or device has been granted access. The specific implementation for the memory descriptor abstraction is not defined.

The device or host is informed of the 64-bit DMA buffer identifier/descriptor at some previous time by a command protocol specific mechanism. Random access within a buffer is accomplished by using the buffer offset.

First Party DMA is a superset capability not supported by parallel implementations of ATA or it's device drivers.

16.2.7.3 Transmission

A device or host transmits a First Party DMA Setup - Device to Host or Host to Device FIS as the first step in performing a First Party DMA access. The purpose of the First Party DMA Setup - Device to Host or Host to Device is to establish DMA hardware context for one or more data transfers.

A First Party DMA Setup - Device to Host or Host to Device is required only when the DMA context is to be changed. Multiple Data - Host to Device or Device to Host FIS's can follow in either direction, for example, if the transfer count exceeds the maximum Data - Host to Device or Device to Host transfer length or when a data transfer is interrupted. When multiple Data - Host to Device or Device to Host FISes follow a First Party DMA Setup - Device to Host or Host to Device FIS, the device or host shall place the data contained in the FIS in sequential addresses; that is, if the last DWORD of an FIS is placed in (or obtained from) address N, the first DWORD of a subsequent Data - Host to Device or Device to Host shall be placed in (or obtained from) address N+4 unless an intervening First Party DMA Setup - Device to Host or Host to Device FIS is used to alter the DMA context. This mechanism allows for the efficient streaming of data into a buffer.

16.2.7.4 Reception

Upon receiving a First Party DMA Setup - Device to Host or Host to Device FIS, the recipient of the FIS shall validate the received First Party DMA Setup request, and provided that the buffer identifier and the specified offset/count are valid, program and arm the adapter's DMA controller using the information in the FIS. The specific implementation of the buffer identifier and buffer/address validation is not specified. After a valid First Party DMA Setup - Device to Host or Host to Device FIS with the D bit set to 0, the recipient of the First Party DMA Setup - Device to Host or Host to Device FIS responds with one or more Data - Host to Device or Device to Host FIS's until the DMA count is exhausted. After a valid First Party DMA Setup - Device to Host or Host to Device FIS with the D bit set to 1, the recipient of the FIS shall be prepared to accept one or more Data - Host to Device or Device to Host FIS's until the DMA count is exhausted.

An Interrupt Pending condition shall be generated upon the completion of the DMA transfer if the I bit is set to 1. The definition of DMA Transfer Completion is command protocol dependent but typically includes the exhaustion of the transfer count or the detection of an error by the DMA controller..

16.2.8 BIST Activate, bidirectional

16.2.8.1 General

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	Reserved (0)								Pattern definition								R	R	R	Reserved (0)					FIS Type (58h)							
									T	A	S	L	F	P	R	V																
1	Data [31:24]								Data [23:16]								Data [15:8]								Data [7:0]							
2	Data [31:24]								Data [23:16]								Data [15:8]								Data [7:0]							

Figure 67 – BIST Activate, Bidirectional

The following Field definitions apply.

FIS Type - Set to a value of 58h. Defines the rest of the FIS fields.
F - Far End Analog (AFE) Loopback (Optional)
L - Far End Retimed Loopback. Transmitter shall insert additional ALIGNS)
R - Reserved
T - Far end transmit only mode
A - ALIGN Bypass (Do not Transmit Align Primitives) (valid only in combination with T Bit) (optional behavior)

S - Bypass Scrambling (valid only in combination with T Bit) (optional behavior)
P - Primitive bit. (valid only in combination with the T- Bit) (optional behavior)
V - Vendor Specific Test Mode. Causes all other bits to be ignored

16.2.8.2 Description

The BIST Activate FIS shall be used to place the receiver in one of several loopback modes, see Figure 67.

The BIST Activate FIS is a bi-directional request in that it may be sent by either the host or the device. The sender and recipient have distinct responsibilities in order to insure proper cooperation between the two parties. The state machines for transmission and reception of the FIS are symmetrical.

The state machines for the transmission of the FIS do not specify the actions the sender takes once successful transmission of the request has been performed. After the sender's Application layer is notified of the successful transmission of the FIS the sender's Application layer will prepare its own Application, Transport and Physical layers into the appropriate states that support the transmission of a stream of data. The FIS shall not be considered successfully transmitted until the receiver has acknowledged reception of the FIS. The sender of the BIST Activate FIS transmits continuous SYNC primitives after reception of the R_OK primitive until such a time that it is ready to interact with the receiver in the BIST exchange.

The state machines for the reception of the FIS do not specify the actions of the receiver's application layer. Once the FIS has been received, the recipient's application layer places its own Application, Transport and Physical layers into states that perform the appropriate retransmission of the sender's data. The recipient shall not enter the BIST state until after it has properly received a good BIST Activate FIS (good CRC), indicated a successful transfer of the FIS to the transmitting side via the R_OK primitive and has received at least one good SYNC primitive. Once in the self-test mode, a receiver shall continue to allow processing of the COMINIT or COMRESET signals in order to exit from the self-test mode.

F: The Far End Analog (Analog Front End - AFE) Loopback, optional, mode where the raw data is received, and retransmitted, without any retiming or re-synchronization, etc. (see 14.7.2.2).

L: The Far End Retimed Loopback, shall be defined as a mode where the recipient retimes the data, and retransmits the retimed data (see 14.7.2.1). This mode is mandatory.

T: The Far-End Transmit Mode mode is used to invoke the Far-End Interface to send data patterns, upon receipt of the FIS BIST Activate, as defined by Pattern DWords #1, and #2. Note that Pattern DWords #1, and #2 shall be applicable only when the T bit is active, indicating "Far-End Transmit Mode".

This data is modified by the P, A, S and V bits.

NOTE This mode is intended for Inspection/Observation Testing, as well as support for conventional laboratory equipment, rather than for in-system automated testing.

P: The transmit primitives bit. When this bit is set in far end transmit mode, the lowest order byte of the two following DWORDs will be treated as K Characters. It is the responsibility of the sender of the BIST FIS to ensure that the data in byte 0 of the DWORDs are valid D character versions of the K character (i.e. BCh for K28.5). Applicable only when the T bit is set.

A: The ALIGN primitive sequence bypass mode. When set to 1, no ALIGN primitives are sent. When the A-bit is not asserted, ALIGN Primitives are sent normally as defined in this document. Applicable only when the T bit is set.

S: The Bypass Scrambling mode is used to send data or patterns, during BIST activation, that are not scrambled, however are encoded and decoded to normal and legal 8b/10b values. Applicable only when the T bit is set.

V: This mode is vendor specific. All other bits are vendor specific in this mode.

16.2.8.3 Transmission

The initiator transmits a BIST Activate to the recipient in order to initiate the BIST mode of operation.

16.2.8.4 Reception

Upon receiving a BIST Activate, the recipient shall begin operations as per the BIST Activate FIS, described above.

16.2.9 PIO Setup, Device to Host

16.2.9.1 General

	31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13	12 11 10 9 8	7 6 5 4 3 2 1 0
0	Error	Status	R	I	D	Reserved (0)	FIS Type (5Fh)
1	Device	LBA High	LBA Mid				LBA Low
2	Reserved (0)	LBA High (exp)	LBA Mid (exp)				LBA Low (exp) (0)
3	E_Status	Reserved (0)	Sector Count (exp)				Sector Count
4	Reserved (0)		Transfer Count				

Figure 68 – PIO Setup, Device to Host FIS layout

The following Field definitions apply.

FIS Type - Set to a value of 5Fh. Defines the rest of the FIS fields. Defines the length of the FIS as five DWORDs.
LBA Low - Holds the contents of the LBA Low register of the Command Block.
LBA Low (exp) - Contains the contents of the expanded address field of the Shadow Command Block.
LBA Mid - Holds the contents of the LBA Mid register of the Shadow Command Block.
LBA Mid (exp) - Contains the contents of the expanded address field of the Shadow Command Block.
LBA High - Holds the contents of the LBA High register of the Command Block.
LBA High (exp) - Contains the contents of the expanded address field of the Shadow Command Block.
D - Indicates whether host memory is being written or read by the device. 1 = write (device to host), 0 = read (host to device).
Device - Holds the contents of the Device register of the Command Block.
Status - Contains the value of the Status register of the Command Block for initiation of host data transfer.
Error - Contains the value of the Error register of the Command Block at the conclusion of all subsequent Data to Device frames.
I - Interrupt bit. This bit reflects the Interrupt Pending status of the device.
R - Reserved
Sector Count - Holds the contents of the sector count register of the Command Block.
Sector Count (exp) - Contains the contents of the expanded address field of the Shadow Command Block.
E_Status - Contains the value to be placed in the Shadow Status register of the Command Block at the expiration of the Transfer Count for this PIO Setup.
Transfer Count - Holds the number of bytes to be transferred in the subsequent Data FIS. The Transfer Count value shall be non-zero and the low order bit shall be zero (even number of bytes transferred).

16.2.9.2 Description

The PIO Setup - Device to Host FIS is used by the device to provide the host adapter with sufficient information regarding a PIO data phase to allow the host adapter to efficiently handle PIO data transfers, see Figure 68. For PIO data transfers, the device shall send to the host a PIO Setup - Device to Host FIS just before every data transfer FIS that is required to complete the data transfer. Data transfers from Host to Device as well as data transfers from Device to Host shall follow this algorithm. Because of the stringent timing constraints in the parallel implementation of ATA, the PIO Setup FIS includes both the starting status from the Status field and ending status from the E_Status field values. These are used by the host adapter to first signal to host software readiness for PIO write data (BSY cleared to zero and DRQ set to one), and to properly signal host software by negating DRQ and possibly asserting BSY after reception of the Data FIS.

16.2.9.3 Transmission of PIO Setup by Device Prior to a Data Transfer from Host to Device

The device transmits a PIO Setup - Device to Host FIS to the host in preparation for a Data FIS just before every Data FIS required to complete the total data transfer for a command. The device includes in the PIO Setup FIS the values to be placed in the Shadow Status register at the initiation of the Data FIS and the E_Status value to be placed in the Shadow Status register at the end of the Data FIS. The device shall be prepared to receive a Data FIS in response to transmitting a PIO Setup FIS.

16.2.9.4 Reception of PIO Setup by Host Prior to a Data Transfer from Host to Device

Upon receiving a PIO Setup - Device to Host FIS, the host shall transfer the Status and Error values into the Shadow Status and Error registers and shall hold the E_Status value in a temporary register. The Transfer Count value shall be loaded into a countdown register. Upon detecting the change in the Shadow Status register, host software proceeds to perform a series of write operations to the Shadow Data register, which the host adapter collects to produce a Data FIS to the device. Each write of the Shadow Data register results in another word of data being concatenated into the Data FIS, and the countdown register being decremented accordingly. The E_Status value shall be transferred to the Shadow Status register within 400 ns after the countdown register reaching terminal count. In the case that the transfer length represents an odd number of words, the last word shall be placed in the low order (word 0) of the final DWORD and the high order word (word 1) of the final DWORD shall be padded with zeros before transmission. This process is repeated for every Data FIS needed to complete the overall data transfer of a command.

16.2.9.5 Transmission of PIO Setup by Device Prior to a Data Transfer from Device to Host

The device transmits a PIO Setup - Device to Host FIS to the host in preparation for a PIO Data FIS just before every PIO Data FIS required to complete the total data transfer for a command. The device includes in the FIS the values to be placed in the Shadow Status register at the beginning of the PIO Data FIS and the E_Status value to be placed in the Shadow Status register at the end of the Data FIS. The device shall be prepared to transmit a Data FIS following the transmittal of a PIO Setup FIS.

16.2.9.6 Reception of PIO Setup by Host Prior to a Data Transfer from Device to Host

Upon receiving a PIO Setup - Device to Host FIS for a device to host transfer, the host shall hold the Status, Error and E_Status values in temporary registers. The Transfer Length value shall be loaded into a countdown register. Upon reception of a Data FIS from the device, the Status and Error values are loaded into the Shadow Status and Shadow Error registers and host proceeds to perform a series of read operations from the Shadow Data register. Each read of the Shadow Data register results in a countdown register being decremented accordingly. The E_Status value shall be transferred to the Shadow Status register within 400 ns after the countdown register reaching terminal count. This process is repeated for every Data FIS needed to complete the overall data transfer of a command.

16.2.10 Data, Host to Device or Device to Host (bidirectional)

16.2.10.1 General

	31–24	23–16	15	14	13	12–8	7–0
0	Reserved (0)	Reserved (0)	R	R	R	Reserved (0)	FIS Type (46h)
…	N DWORDs of data (minimum of one DWORD - maximum of 2048 DWORDs)						
…							
n							

Figure 69 – Data, Host to Device or Device to Host FIS layout

The following Field definitions apply.

FIS Type - Set to a value of 46h. Defines the rest of the FIS fields. Defines the length of the FIS as n+1 DWORDs.

DWORDs of data - Contain the actual data to transfer. Only 32 bit fields are transferred. The last DWORD is padded with zeros when only a partial DWORD is to be transmitted.

NOTE The maximum amount of user data that can be sent in a single Data - Host to Device or Data - Device to Host FIS is limited. See description 16.2.10.2.

R – Reserved.

16.2.10.2 Description

The Data - Host to Device and the Data - Device to Host FISes are used for transporting payload data, such as the data read from or written to a number of sectors on a hard drive, see Figure 69. The FIS may either be generated by the device to transmit data to the host or may be generated by the host to transmit data to the device. This FIS is only one element of a sequence of transactions leading up to a data transmission and the transactions leading up to and following the Data FIS establish the proper context for both the host and device.

The byte count of the payload is not an explicit parameter, rather it is inferred by counting the number of DWORDs between the SOF and EOF primitives, and discounting the FIS type and CRC DWORDs. The payload size shall be no more than 2048 DWORDs (8192 bytes). Non-packet devices shall report a SET MULTIPLE limit of 16 sectors or less in word 47 of their IDENTIFY DEVICE information.

In the case that the transfer length represents an odd number of words, the last word shall be placed in the low order (word 0) of the final DWORD and the high order word (word 1) of the final DWORD shall be padded with zeros before transmission.

16.2.10.3 Transmission

The device transmits a Data - Device to Host FIS to the host during the data transfer phase of PIO reads, DMA reads and First Party DMA writes to host memory. The device shall precede a Data FIS with any necessary context-setting transactions as appropriate for the particular command sequence. For example, a First Party DMA host memory write is preceded by a First Party DMA Setup - Device to Host FIS to establish proper context for the Data FIS that follows.

The host transmits a Data - Host to Device FIS to the device during the data transfer phase of PIO writes, DMA writes and First Party DMA reads of host memory. The FIS shall be preceded with any necessary context-setting transactions as appropriate for the particular command sequence. For example, a DMA write to the device is preceded by a DMA Activate - Device to Host FIS with the DMA context having been pre-established by the host.

When used for transferring data for DMA operations multiple Data - Host to Device or Device to Host FIS's can follow in either direction. Segmentation can occur when the transfer count exceeds the maximum Data - Host to Device or Device to Host transfer length or if a data transfer is interrupted.

When used for transferring data in response to a PIO Setup all of the data specified in the transfer count in the PIO Setup shall be transmitted in a single Data FIS.

In the event that a transfer is broken into multiple FISes, all intermediate FISes shall contain an integral number of full DWORDs. If the total data transfer is for an odd number of words, then the high order word (word 1) of the last DWORD of the last FIS shall be padded with zeros before transmission and discarded on reception.

16.2.10.4 Reception

Neither the host nor device is expected to buffer an entire Data FIS in order to check the CRC of the FIS before processing the data. Incorrect data reception for a Data FIS is reflected in the overall command completion status.

16.3 Host transport states

16.3.1 Host transport idle state diagram

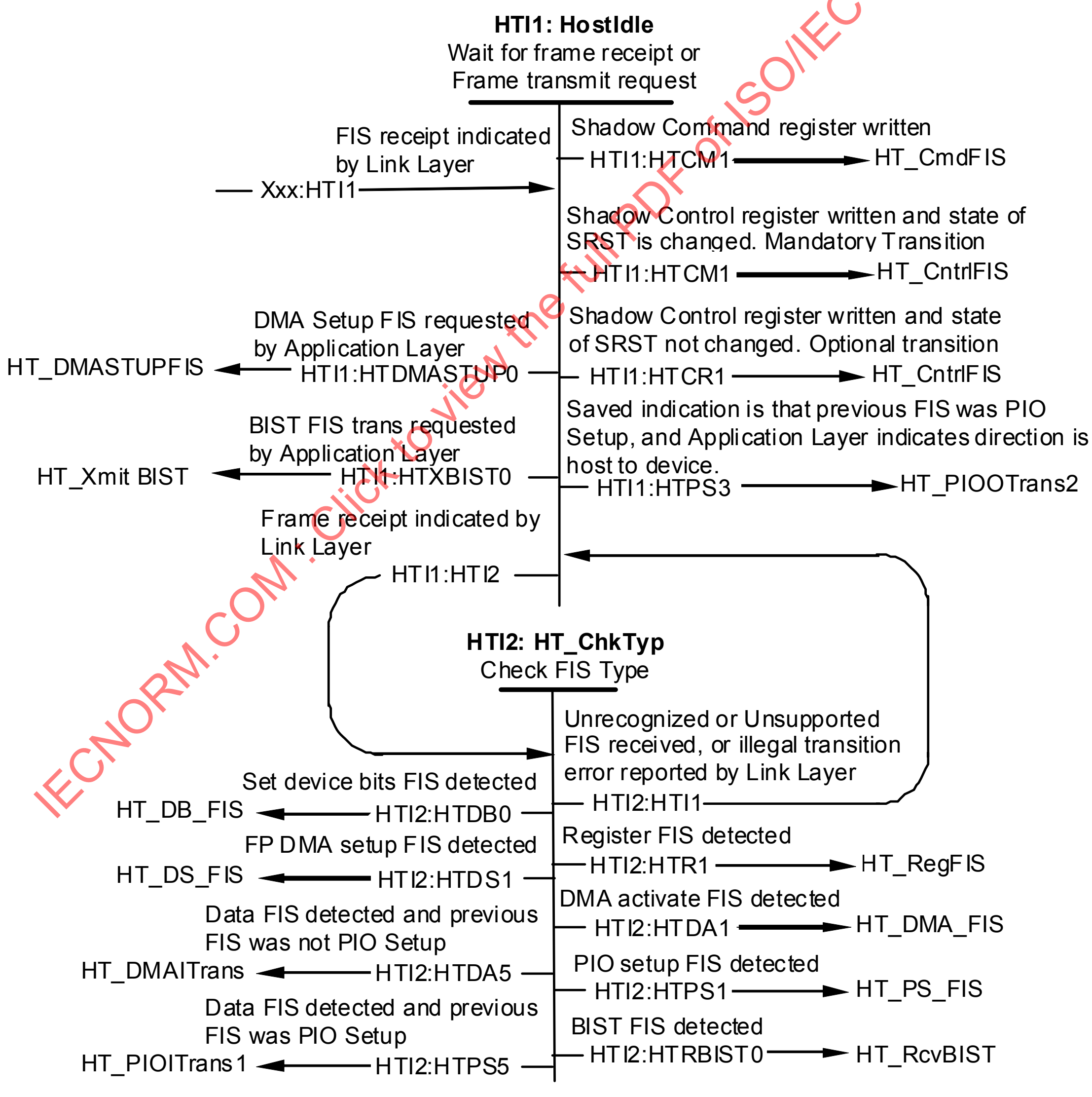


Figure 70 – Host transport idle state diagram (States HTI1-HTI2)

HTI1: HT_HostIdle state: This state is entered when a Frame Information Structure (FIS) transaction has been completed by the Transport layer, see Figure 70.

When in this state, the Transport layer waits for the Shadow Command register to be written, the Shadow Device Control register to be written, or the Link layer to indicate that an FIS is being received.

TransitionHTI1:HTCM1: When the Shadow Command register is written, the Transport layer shall make a transition to the HTCM1: HT_CmdFIS state.

Transition HTI1:HTCR1: When the Shadow Control register is written, and the state of the SRST bit is changed from its previous state, the Transport layer shall make a transition to the HTCR1: HT_CntrlFIS state. This transition is optional upon a write operation to the Shadow Control register if the state of the SRST bit is not changed.

Transition HTI1:HTI2: When the Link layer indicates that an FIS is being received, the Transport layer shall make a transition to the HTI2: HT_ChkTyp state.

Transition HTI1:HTDMASTUP0: When the Application layer indicates that a First Party DMA Setup FIS is to be sent, the Transport layer shall make a transition to the HTDMASTUP0:HT_DMASTUPFIS state.

Transition HTI1:HTXBIST1: When the host's application layer requests the transmission of a BIST request to the device the Transport layer shall make a transition to the HTXBIST1:HT state.

Transition HTI1:HTPS3: When the host's application layer requests the transmission of data to the device and the saved indication shows that the previous FIS was a PIO Setup type, the Transport layer shall make a transition to the HTPS3:HT_PIOOTrans2 state. The Host Transport layer shall save an indication that a PIO Setup FIS was the last FIS received. The host's application layer writes data to the device by performing writes to the Data register in the Shadow Command Block.

HTI2: HT_ChkTyp state: This state is entered when the Link layer indicates that an FIS is being received, see Figure 70.

When in this state, the Transport layer checks the FIS type of the incoming FIS.

Transition HTI2:HTR1: When the incoming FIS is a register type, the Transport layer shall notify the Link Layer that it has received a valid FIS type, and make a transition to the HTR1: HT_RegFIS state.

Transition HTI2:HTDB0: When the incoming FIS is a Set Device Bits type, the transport layer shall notify the Link Layer that it has received a valid FIS type and make a transition to the HTDB0:HT_DB_FIS state.

Transition HTI2:HTDA1: When the incoming FIS is a DMA Activate type, the Transport layer shall notify the Link Layer that it has received a valid FIS type and make a transition to the HTDA1: HT_DMA_FIS state.

Transition HTI2:HTPS1: When the incoming FIS is a PIO Setup type, the Transport layer shall notify the Link Layer that it has received a valid FIS type and make a transition to the HTPS1: HT_PS_FIS state. The Host Transport layer shall save an indication that a PIO Setup FIS is the last FIS received.

Transition HTI2:HTDS1: When the incoming FIS is a First Party DMA Setup type, the Transport layer shall notify the Link Layer that it has received a valid FIS type and make a transition to the HTDS1: HT_DS_FIS state.

Transition HTI2:HTRBIST1: When the incoming FIS is a BIST Activate type, the Transport layer shall notify the Link Layer that it has received a valid FIS type and make a transition to the HTRBIST1:HT_RcvBIST state.

Transition HTI2:HTDA5: When the incoming FIS is a Data type, and the previous FIS was not a PIO Setup type, the Transport layer shall notify the Link Layer that it has received a valid FIS type, and make a transition to the HTDA5:HT_DMAITrans state. The Host Transport Layer saved an indication that a PIO Setup FIS was the last FIS received, so that this state can determine whether to transition to DMA data transfer or PIO data transfer.

Transition HTI2:HTPS5: When the incoming FIS is a Data type, and the previous FIS was a PIO Setup type, the Transport layer shall notify the Link Layer that it has received a valid FIS type and make a transition to the HTPS5:HT_PIOITrans state. The Host Transport Layer saved an indication that a PIO Setup FIS was the last FIS received, so that this state can determine whether to transition to DMA data transfer or PIO data transfer.

Transition HTI2:HTI1: When the received FIS is of an unrecognized or unsupported type, the Transport layer shall notify the Link Layer that it has received an unrecognized FIS and make a transition to the HTI1: HT_HostIdle state.

When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

16.3.2 Host Transport transmit command FIS diagram

The protocol shown in Figure 71 builds an FIS that contains the host adapter Shadow Command Block and Shadow Control Block and sends it to the device when the software driver or BIOS writes the host adapter Shadow Command register.

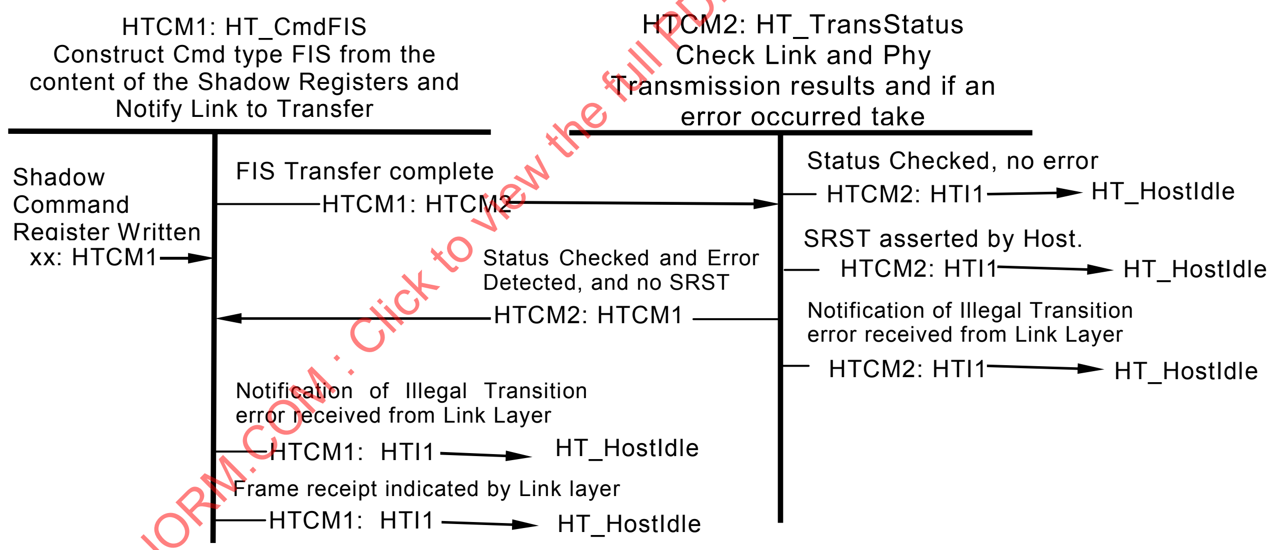


Figure 71 – Host transport transmit command FIS diagram (States HTCM1-HTCM2)

HTCM1: HT_CmdFIS state: This state is entered when the Shadow Command register is written.

When in this state, the Transport layer shall construct a register FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer.

Transition HTCM1:HTCM2: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmit is complete and make a transition to the HTCM2: HT_TransStatus state.

Transition HTCM1:HTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

Transition HTCM1:HTI1: When the Link layer indicates that an FIS is being received, the Transport Layer shall mak e a transition to the HTI1:HT_HostIdle state.

HTCM2: HT_TransStatus state: This state is when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition HTCM2:HTI1: When the FIS status has been handled and no errors detected, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTCM2:HTCM1: When the FIS status has been handled and an error has been detected, and the host has not asserted the SRST by writing to the Device Control register, the Transport layer shall transition to the HTCM1: HT_CmdFIS state.

Transition HTCM2:HTI1: When the FIS status has been handled, an error has been detected, and the host has asserted the SRST by writing to the Device Control register, or a DEVICE RESET command has been written to an ATAPI device, the Transport layer shall inform the Link layer to send a SYNC primitive, and transition to the HTI1: HT_HostIdle state.

Transition HTCM2:HTCM1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

16.3.3 Host Transport transmit control FIS diagram

The protocol shown in Figure 72 builds an FIS that contains the host adapter Shadow Command Block and Shadow Control Block content and sends it to the device when the software driver or BIOS writes the host adapter Shadow Device Control register.

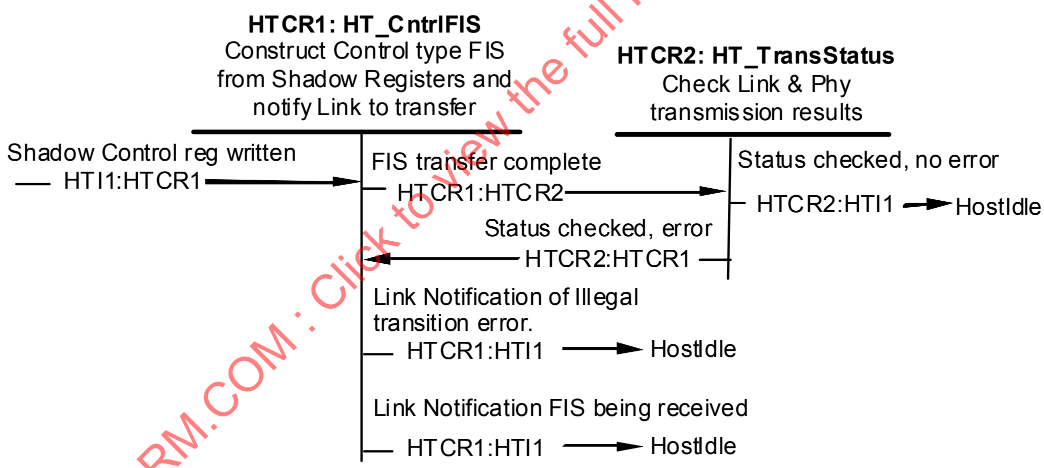


Figure 72 – Host transport transmit control FIS diagram (States HTCR1-HTCR2)

HTCR1: HT_Cntrl_FIS state: This state is entered when the Shadow Device Control register is written.

When in this state, the Transport layer shall construct a register FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer.

Transition HTCR1:HTCR2: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmit is complete and make a transition to the HTCR2: HT_TransStatus state.

Transition HTCR1:HTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

When the Link layer indicates that an FIS is being received, the Transport Layer shall make a transition to the HTI1:HT_HostIdle state.

HTCR2: HT_TransStatus state: This state is when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition HTCR2:HTI1: When the FIS status has been handled and no errors have been detected, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTCR2:HTCR1: When the FIS status has been handled and at least one error has been detected, the Transport layer shall transition to the HTCR1: HT_Cntrl_FIS.

16.3.4 Host Transport transmit First Party DMA Setup, Device to Host or Host to Device FIS state diagram

The protocol shown in Figure 73 transmits a First Party DMA Setup - Device to Host or Host to Device FIS to a receiver. This FIS is a request by a sender for the recipient to program its DMA controller for a First Party DMA transfer and is followed by one or more Data FISes that transfer data. The First Party DMA Setup - Device to Host or Host to Device FIS request includes the transfer direction indicator, the host buffer identifier, the host buffer offset, the byte count and the interrupt flag.

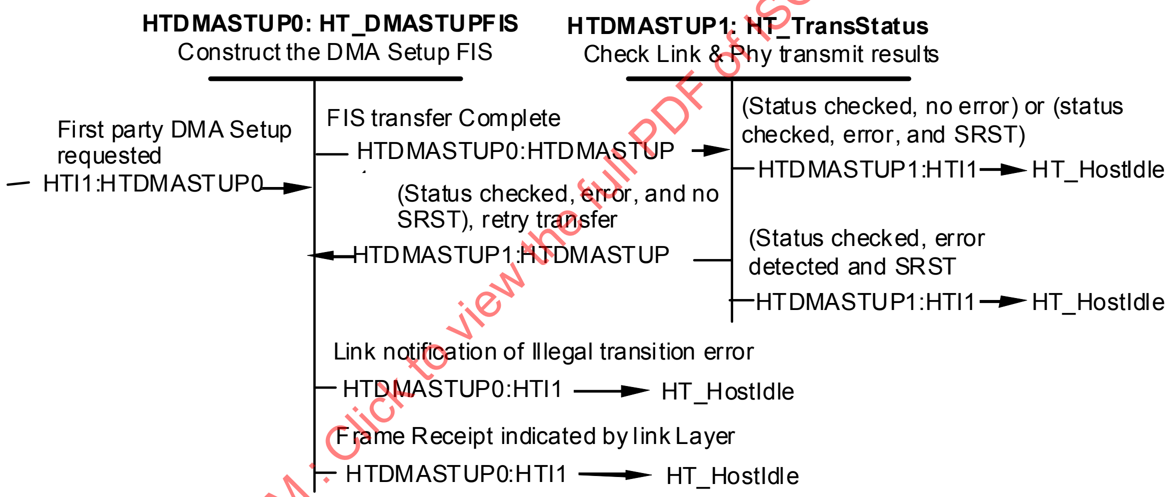


Figure 73 – Host transport transmit First Party DMA setup – Device to host or host to device FIS (States HTDMASTUP0-HTDMASTUP1)

HTDMASTUP0: HT_DMASTUPFIS state: This state is entered when the Application requests the transmission of a First Party DMA Setup - Host to Device or Device to Host FIS.

When in this state, the Transport layer shall construct a First Party DMA Setup - Host to Device or Device to Host FIS, notify the Link layer that the FIS is to be transmitted and pass the FIS to the Link layer.

Transition HTDMASTUP0:HTDMASTUP1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the HTDMASTUP1: HT_TransStatus state.

Transition HTDMASTUP0:HTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

Transition HTDMASTUP0:HTI1: When the Link layer indicates that an FIS is being received, the Transport Layer shall make a transition to the HTI1:HT_HostIdle state.

HTPDMASTUP1: HT_TransStatus state: This state is entered when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition HTDMASTUP1:HTI1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTDMASTUP1:HTDMASTUP0: When the FIS status has been handled, an error detected and the host has not asserted the SRST by writing to the Device Control register, the Transport layer shall report status to the Link layer, and retry this transfer by transitioning to the HTDMASTUP0: HT_DMASTUPFIS state.

Transition HTDMASTUP1:HTI1: When the host has asserted the SRST bit by writing to the Device Control register or the DEVICE RESET command is issued, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1: HT_HostIdle state.

16.3.5 Host Transport transmit BIST Activate FIS

The protocol shown in Figure 74 builds a BIST Activate FIS that tells the device to prepare to enter the appropriate Built-in Self-test mode. After successful transmission, the host Transport layer enters the idle state. The application layer, upon detecting successful transmission to the device shall then cause the host's Transport layer, Link layer and Physical layer to enter the appropriate mode for the transmission of the Built-in Test data defined by the FIS. The means by which the Transport, Link and Physical layers are placed into self-test mode are not defined by this standard.

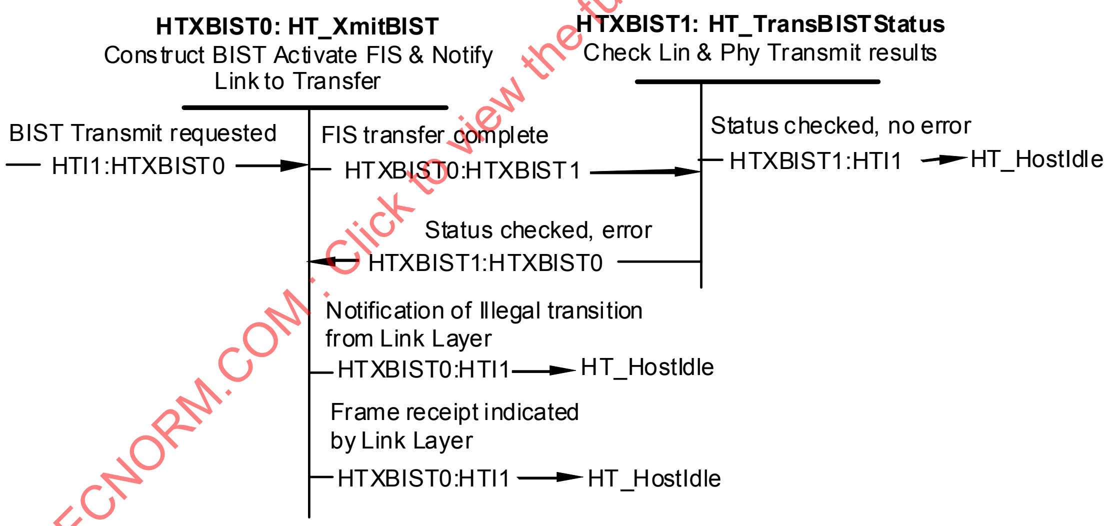


Figure 74 – Host transport transmit BIST activate FIS (States HTXBIST0-HTXBIST1)

HTXBIST0: HT_XmitBIST state: This state is entered to send a BIST FIS to the device.

Transition HTXBIST0:HTXBIST1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the HTXBIST1:HT_TransBISTStatus state.

Transition HTXBIST0:HTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

Transition HTXBIST0:HTI1: When the Link layer indicates that an FIS is being received, the Transport Layer shall make a transition to the HTI1:HT_HostIdle state.

HTXBIST1: HT_TransBISTStatus state: This state is entered when the entire FIS has been passed to the Link layer.

Transition HTXBIST1:HTI1: When the FIS transmission is completed and no errors have been detected the Transport layer shall transition to the HTI1:HT_HostIdle state.

Transition HTXBIST1:HTXBIST0: When the FIS transmission is completed and at least one error is detected the Transport layer shall transition to the HTXBIST0: HT_XmitBIST state.

16.3.6 Host Transport decompose Register FIS diagram

The protocol shown in Figure 75 receives an FIS from the device containing new Shadow Command Block and Shadow Control Block content and places that content into the Shadow Command and Shadow Control registers.

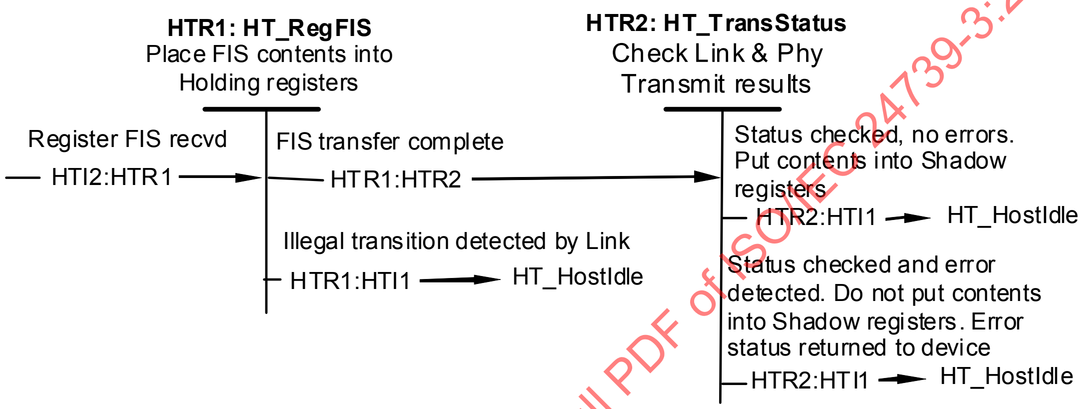


Figure 75 – Host transport decompose register FIS diagram (States HTR1- HTR2)

HTR1: HT_RegFIS state: This state is entered when the Link layer has indicated that an FIS is being received and that the FIS is of the register type.

When in this state, the Transport layer shall decompose the register FIS and place the contents into the appropriate holding registers.

Transition HTR1:HTR2: When the entire FIS has been placed into the holding registers, the Transport layer shall make a transition to the HTR2: HT_TRANS_STATUS state.

Transition HTR1:HTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

HTR2: HT_TransStatus state: This state is entered when the entire FIS has been placed into the holding registers.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition HTR2:HTI1: When the FIS status has been handled and no errors detected, the contents of the holding registers shall be placed in the Shadow Command Block and Shadow Control Block and if the interrupt bit is set, the Transport layer shall set the Interrupt Pending flag. The Transport layer shall transition to the HTI1: HT_HostIdle state. When the FIS status has been handled and at least one error detected, the contents of the holding registers shall not be transferred to the Shadow Command Block and Shadow Control Block, error status shall be returned to the device, and the Transport layer shall transition to the HTI1: HT_HostIdle state.

16.3.7 Host Transport decompose a Set Device Bits FIS state diagram

The protocol shown in Figure 76 receives an FIS from the device containing new Shadow Error and Shadow Status register content and places that content into the Shadow Error and Shadow Status registers.

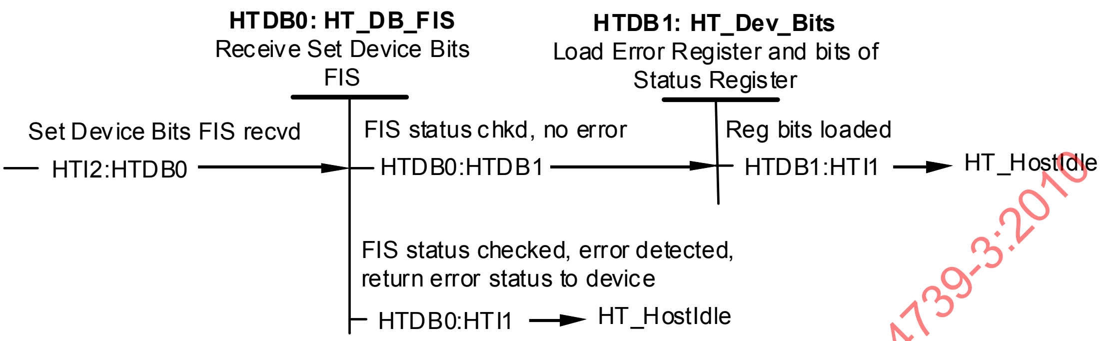


Figure 76 – Host transport decompose Set Device Bits FIS state diagram (States HTDB0-HTDB1)

HTDB0:HT_DB_FIS state: This state is entered when the Link layer has indicated that an FIS being received and that the FIS is a Set Device Bits type.

When in this state, the Transport layer shall wait for the FIS reception to complete and for Link and Phy ending status to be posted.

Transition HTDB0:HTDB1: When the FIS reception is complete with no errors detected, the Transport layer shall transition to the HTDB1:HT_Dev_Bits state.

Transition HTDB0:HTI1: When the FIS reception is complete with errors detected, the Transport layer shall return error status to the device and transition to the HTI1:HT_HostIdle state.

HTDB1:HT_Dev_Bits state: This state is entered when a Set Device Bits FIS has been received with no errors.

When in this state, the data in the Error field of the received FIS shall be loaded into the host adapter's Shadow Error register. The data in the Status-Hi field of the received FIS shall be loaded into bits 6, 5 and 4 of the Shadow Status register. The data in the Status-Lo field of the received FIS shall be loaded into bits 2, 1 and 0 of the Shadow Status register. Bit 7, BSY and bit 3, DRQ, in the Shadow Status register shall not be changed. If the I bit in the FIS is set to one and if both the BSY bit and the DRQ bit in the Shadow Status register are cleared to zero, then the host adapter shall enter an Interrupt Pending state.

Transition HTDB1:HTI1: The Transport layer shall transition to the HTI1:HT_HostIdle state.

16.3.8 Host Transport decompose a DMA Activate FIS diagram and DMA Data Transfer

The protocol shown in Figure 77 receives an FIS that requests a legacy DMA data transfer. If the data transfer is from the host to the device the DMA Activate FIS causes the host adapter to transmit the data in a subsequent Data FIS If the data transfer is from the device to the host, the host receives the subsequent data FIS.

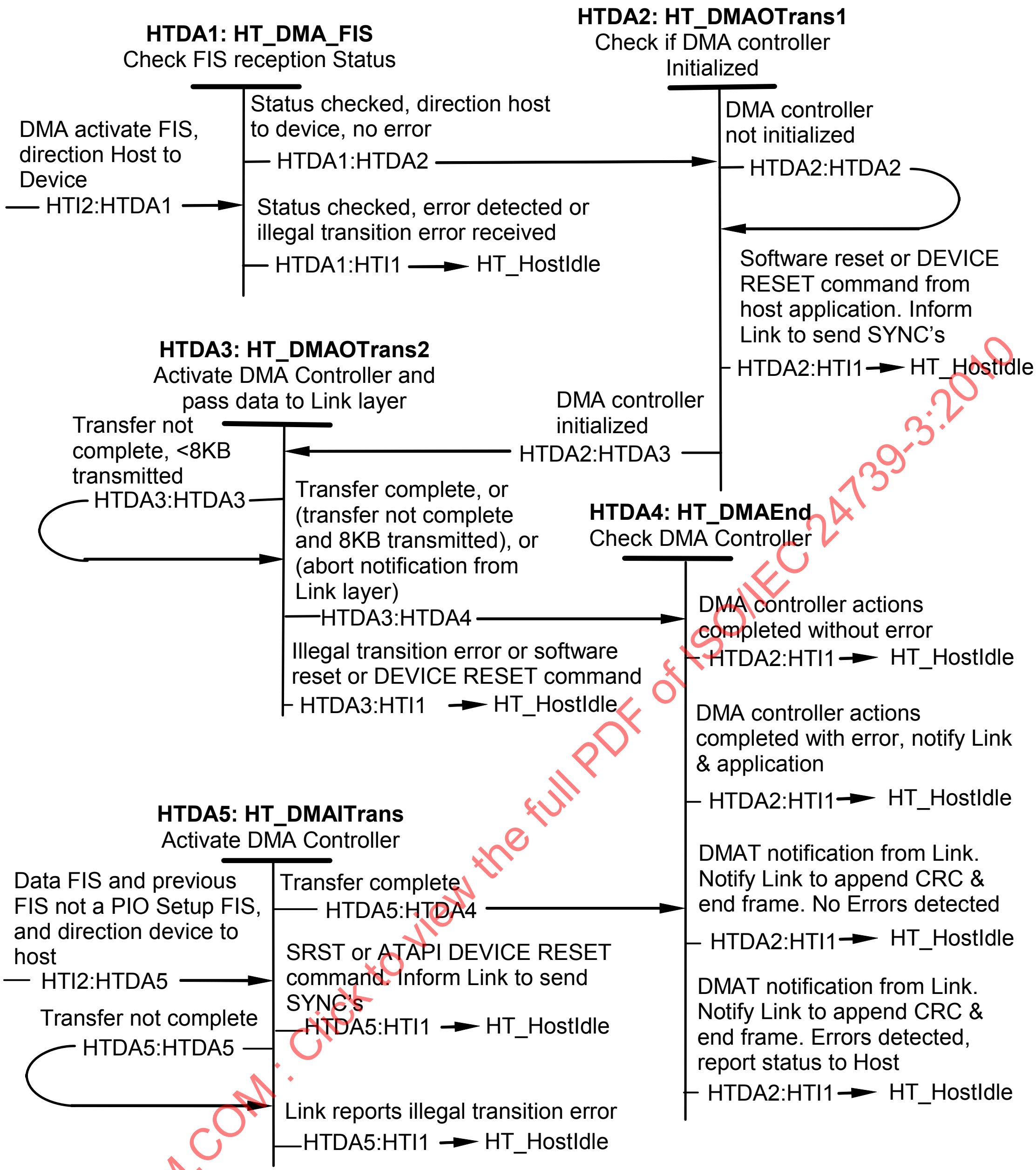


Figure 77 – Host transport decompose DMA activate FIS diagram (States HTDA1-HTDA5)

HTDA1: HT_DMA_FIS state: This state is entered when the Link layer has indicated that an FIS is being received and the Transport layer has determined that a DMA Activate FIS is being received.

Transition HTDA1:HTDA2: The Transport layer shall make a transition to the HTDA2: HT_DMAOTrans1 state. This transition occurs if no error is detected.

Transition HTDA1:HTI1: When an error is detected, status is conveyed to the Link layer and to the application layer. The Transport layer shall make a transition to the HTI1:HT_HostIdle state.

When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

HTDA2: HT_DMAOTrans1 state: This state is entered when the FIS Reception has been checked and no errors are detected.

When in this state, the Transport layer shall determine if the DMA controller has been initialized.

Transition HTDA2:HTDA2: When the DMA controller has not yet been initialized, the Transport layer shall transition to the HTDA2: HT_DMAOTrans1 state.

Transition HTDA2:HTDA3: When the DMA controller has been initialized, the Transport layer shall transition to the HTDA3: HT_DMAOTrans2 state.

Transition HTDA2:HTI1: When the host has asserted the SRST bit by writing to the Device Control register, or the DEVICE RESET command is written to the Command register, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1:HT_HostIdle state.

HTDA3: HT_DMAOTrans2 state: This state is entered when the DMA controller has been initialized.

When in this state, the Transport layer shall activate the DMA controller and pass data to the Link layer.

Transition HTDA3:HTDA3: When the transfer is not complete and less than 8KB of payload data has been transmitted, the Transport layer shall transition to the HTDA3: HT_DMAOTrans2 state.

Transition HTDA3:HTDA4: When the transfer is not complete but 8KB of payload data has been transmitted, the Link layer shall be notified to close the current frame and the Transport layer shall deactivate the DMA engine and transition to the HTDA4: HT_DMAEnd state.

When notified by the Link layer that the DMAT primitive was received, the Transport layer may transition to the HTDA4: HT_DMAEnd state. Use of DMAT is not recommended, see 15.4.6.

When the requested DMA transfer is complete, the Transport layer shall transition to the HTDA4: HT_DMAEnd state.

Transition HTDA3:HTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

When the host has asserted the SRST bit by writing to the Device Control register, or the DEVICE RESET command is written to the Command register, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1:HT_HostIdle state.

HTDA4: HT_DMAEnd state: This state is entered when the DMA data transfer is complete.

When in this state, the Transport layer shall ensure that the activities of the DMA controller have completed.

Transition HTDA4:HTI1: When the DMA controller has completed its activities, whether it has exhausted its transfer count or has been deactivated as a result of reaching the 8KB data payload limit, the Transport layer shall transition to the HTI1: HT_HostIdle state. This transition occurs if no error is detected.

When an error is detected, status shall be reported to the Link and application layers. The Transport layer shall transition to the HTI1:HT_HostIdle state.

When notified by the Link layer that a DMAT primitive was received, the transfer may be truncated, and the Link layer notified to append CRC and end the frame. (Use of DMAT is not recommended, see 15.4.6.)

When it is determined that the transfer is completed with an error, the Transport layer shall report status to the host and make a transition to the HTI1:HT_HostIdle state.

When it is determined that the transfer is completed with no error, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTDA5: HT_DMAITrans state: This state is entered when the Transport layer has determined that the DMA transfer being activated is from device to host.

When in this state, the Transport layer shall activate the DMA controller if the DMA controller is initialized. A data frame will be received from the device and a received data DWORD shall be placed in the data FIFO.

When in this state, the Transport layer shall wait until the Link layer has begun to receive the DMA data frame and data is available to be read by the host.

Transition HTDA5:HTDA5: If the transfer is not complete, the Transport layer shall transition to the HTDA5: HT_DMAITrans state. This includes the condition where the host DMA engine has not yet been programmed and the transfer is therefore held up until the DMA engine is prepared to transfer the received data to the destination memory locations.

Transition HTDA5:HTI1: When the SRST bit is asserted by the host writing the Device Control register, or a DEVICE RESET command has been written to an ATAPI device, the Link layer shall be informed to send a SYNC primitive, and the Transport layer shall transition to the HTI1:HT_Idle state.

When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1: HT_HostIdle state.

Transition HTDA5:HTDA4: When the requested DMA transfer is complete, the Transport layer shall transition to the HTDA4: HT_DMAEnd state.

16.3.9 Host Transport decompose a PIO Setup FIS state diagram

The protocol shown in Figure 78 receives a PIO Setup FIS that requests a PIO data transfer. If the direction is from host to device, the Transport layer transmits a Data FIS to the device containing the PIO data. If the direction of transfer is from device to host, the Transport layer receives a Data FIS from the device. There shall be one data FIS for each corresponding PIO Setup FIS.

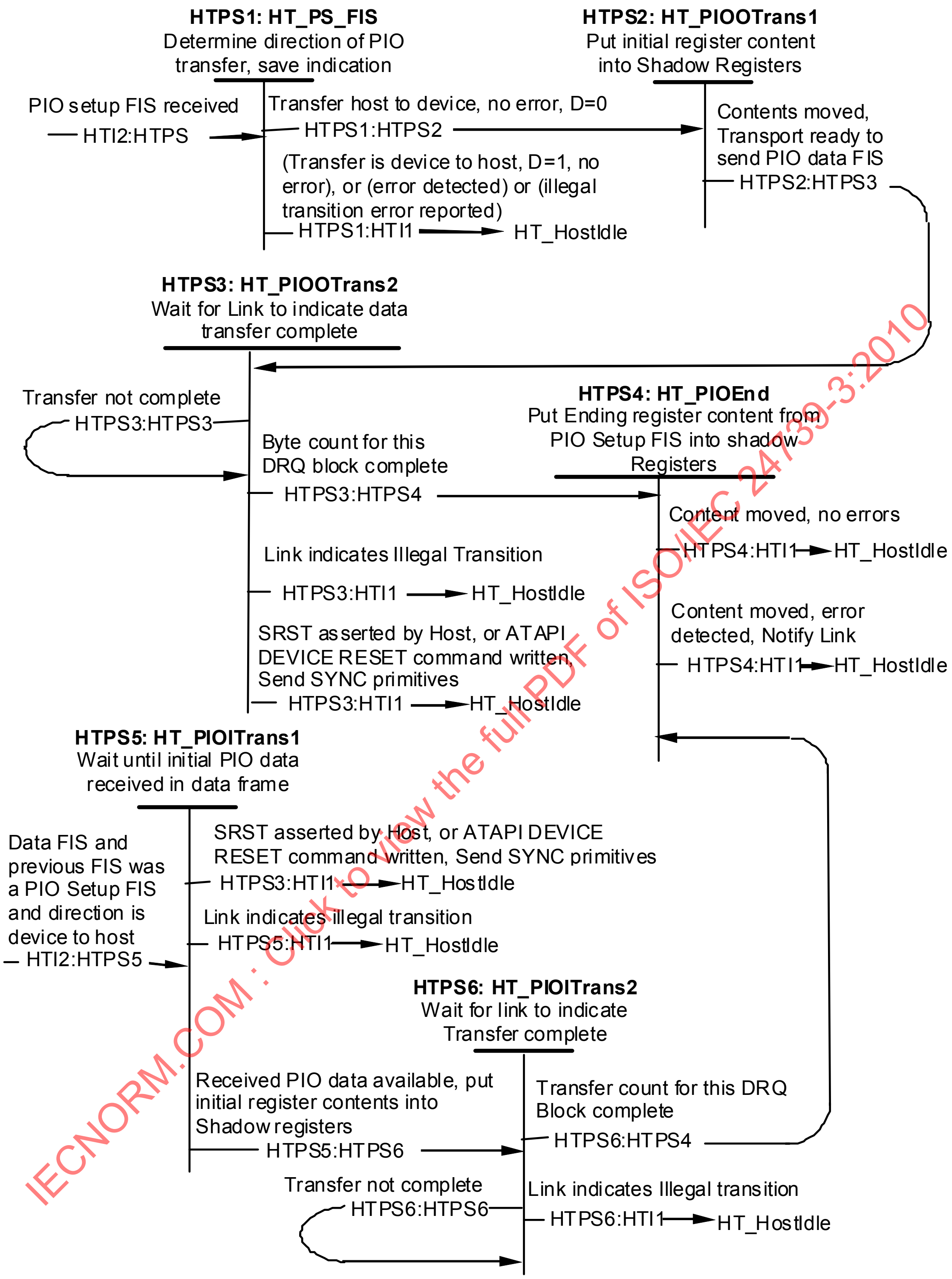


Figure 78 – Host transport decompose PIO setup FIS state diagram (States HTPS1-HTPS6)

HTPS1: HT_PS_FIS state: This state is entered when the Link layer has indicated that an FIS is being received and that the Transport layer has determined a PIO Setup FIS is being received.

When in this state, the Transport layer shall determine the direction of the requested PIO transfer and indicate that the last FIS sent was a PIO Setup.

Transition HTPS1:HTPS2: When the direction of transfer requested is from host to device (D=0), the Transport layer shall make a transition to the HTPS2: HT_PIOOTrans1 state. This transition occurs if no error is detected.

Transition HTPS1:HTI1: When the direction of transfer requested is from device to host (D=1), the Transport layer shall make a transition to the HTI1:HT_HostIdle state. This transition occurs if no error is detected.

When an error is detected, status shall be reported to the Link layer. The Transport layer shall make a transition to the HTI1:HT_HostIdle state.

When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTPS2: HT_PIOOTrans1 state: This state is entered when the direction of the requested PIO data transfer is from host to device.

When in this state, the Transport layer shall place the FIS initial register content into the Shadow Command Block and Shadow Control Block, the FIS byte count, and set the Interrupt Pending flag if the FIS indicates to do so.

Transition HTPS2:HTPS3: When the FIS initial register content has been placed into the Shadow Command Block and Shadow Control Block, Interrupt Pending set if requested, and the Transport layer is ready to begin transmitting the requested PIO Data FIS, the Transport layer shall make a transition to the HTPS3: HT_PIOOTrans2 state.

HTPS3: HT_PIOOTrans2 state: This state is entered when PIO data is available in the PIO FIFO to be passed the Link layer.

When in this state, the Transport layer shall wait for the Link layer to indicate that all data has been transferred.

NOTE Since the software driver or BIOS sees DRQ set and BSY cleared, it continues writing the Data register filling the PIO FIFO.

Transition HTPS3:HTPS3: If the transfer is not complete, the Transport layer shall transition to the HTPS3: HT_PIOOTrans2 state.

Transition HTPS3:HTPS4: When the byte count for this DRQ data block is reached, the Transport layer shall transition to the HTPS4: HT_PIOEnd state.

Transition HTPS3:HTI1: If the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

If the host has asserted the SRST bit by writing to the Device Control register or the DEVICE RESET command is written to the Command register, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1:HT_HostIdle state.

HTPS4: HT_PIOEnd state: This state is entered when the byte count for the DRQ data block is reached (see 16.2.9.4).

When in this state, the Transport layer shall place the E_Status field from the previously received PIO Setup FIS into the Shadow Status Register.

Transition HTPS4:HTI1: When the E_Status field for the previously received PIO Setup FIS has been placed into the Shadow Command Block and Shadow Control Block and there were no errors detected with the transfer, the Transport layer shall transition to the HTI1: HT_HostIdle state.

When the E_Status field from the previously received PIO Setup FIS has been placed into the Shadow Command Block and Shadow Control Block, the Transport layer shall transition to the HTI1:HT_HostIdle state. For data in transfers, the Transport layer shall notify the Link layer of any error encountered during the transfer, and the error shall be reflected in the end of frame handshake. If the transfer was not the final transfer for the PIO data in command, the device shall reflect the error status by transmitting an appropriate Register FIS to the host. If the transfer was the final transfer for the associated PIO data in command, the error condition is not detectable. For data out transfers, errors detected by the device shall be reflected in the end of frame handshake. The device shall reflect the error status by transmitting an appropriate Register FIS to the host.

HTPS5: HT_PIOITrans1 state: This state is entered when the direction of the PIO data transfer is device to host and the previous FIS was a PIO Setup FIS.

When in this state, the Transport layer shall wait until the Link layer has begun to receive the PIO data frame and data is available to be read by the host.

Transition HTPS5:HTPS6: When data is available for the host to read in the Shadow Data register, the Transport layer shall place the initial register content received in the PIO Setup frame into the Shadow Command Block and Shadow Control Block and transition to the HTPS6: HT_PIOITrans2 state.

Transition HTPS5:HTI1: When the host has asserted the SRST bit by writing to the Device Control register, or the DEVICE RESET command is issued, the Transport layer shall inform the Link layer to send SYNC primitives, and the Transport layer shall transition to the HTI1: HT_HostIdle state.

When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTPS6: HT_PIOITrans2 state: This state is entered when PIO data is available in the PIO FIFO to be read by the host and the initial Shadow Command Block and Shadow Control Block content has been set.

When in this state, the Transport layer shall wait for the Link layer to indicate that the data transfer is complete

Transition HTPS6:HTPS6: When the transfer is not complete, the Transport layer shall transition to the HTPS6: HT_PIOITrans2 state.

Transition HTPS6:HTPS4: When the byte count for this DRQ data block is reached, the Transport layer shall transition to the HTPS4: HT_PIOEnd state.

Transition HTPS6:HTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

16.3.10 Host Transport decompose a First Party DMA Setup FIS state diagram

The protocol shown in Figure 79 receives an FIS that sets up the host adapter DMA controller to allow the transfer of a First Party DMA Data FIS to the host. The DMA Activate FIS will be transmitted as a separate DMA Activate FIS protocol following the completion of this protocol.

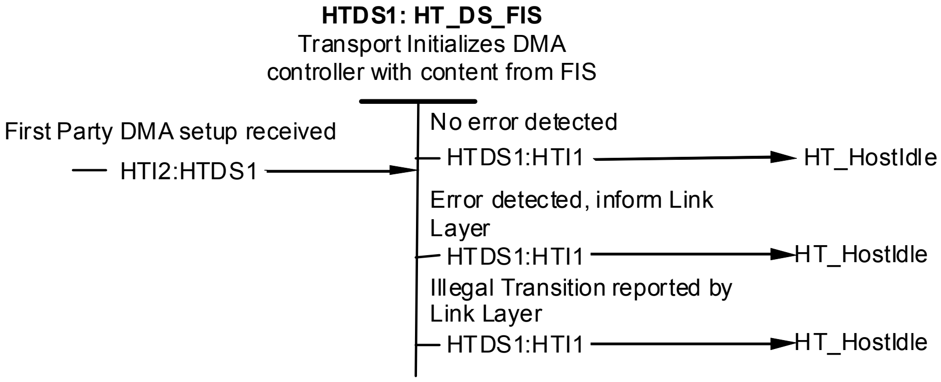


Figure 79 – Host transport decompose First Party DMA Setup FIS state diagram (State HTDS1)

HTDS1: HT_DR_FIS state: This state is entered when the Link layer has indicated that an FIS is being received and that the Transport layer has determined the FIS is of the First Party DMA in request type.

When in this state, the Transport layer shall initialize the DMA controller with content from the FIS.

Transition HTDS1:HTI1: When the DMA controller has been initialized, the Transport layer shall transition to the HTI1: HT_HostIdle state. This transition is made if no error is detected.

Transition HTDS1:HTI1: If an error is detected, status shall be reported to the Link layer. The Transport layer shall transition to the HTI1:HI_HostIdle state.

Transition HTDS1:HTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

16.3.11 Host transport decompose a BIST Activate FIS state diagram

The protocol shown in Figure 80 receives an FIS that instructs the host to enter one of several Built-in Self-test modes that cause the host to retransmit the data it receives. If the mode is supported the Host's application layer will place both the transmit and receive portions of the Transport, Link and/or Physical layers into appropriate state to perform the loopback operation.

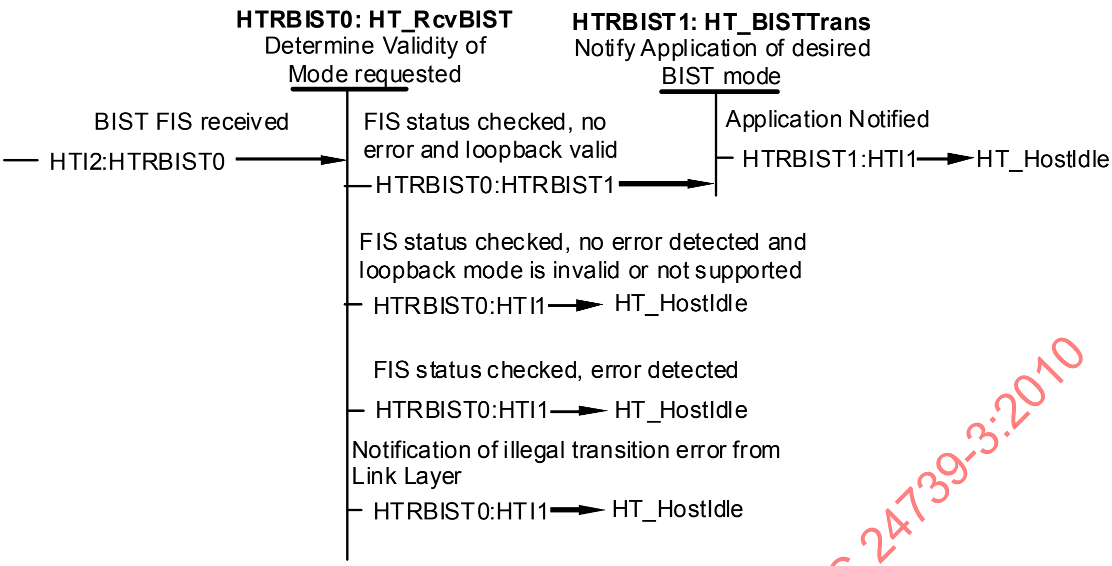


Figure 80 – Host transport decompose BIST activate FIS state diagram (State HTRBIST0-HTRBIST1)

HTRBIST0: HT_RcvBIST state: This state is entered when the link layer has indicated that an FIS is being received and the Transport layer has determined that a BIST Activate FIS is being received.

When in this state, the Transport layer shall determine the validity of the loopback request.

Transition HTRBIST0:HTRBIST1: If no reception error is detected and the FIS contents indicate a form of loopback request that is supported by the host the Transport layer shall make a transition to the HTRBIST2: HT_BISTTrans1 state.

Transition HTRBIST0:HTI1: If no reception error is detected and the FIS contents indicate a form of loopback request that is not supported by the host the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

Transition HTRBIST0:HTI1: If a reception error is indicated the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

Transition HTRBIST0:HTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTRBIST1: HT_BISTTrans1 state: This state is entered when the Transport layer has determined that a valid BIST Activate FIS has been received.

Having received a valid FIS, the Transport layer informs the host's application layer that it should place the Transport, Link and Physical layers into the appropriate modes to loop the received data back to the transmitter. The method by which this is performed is vendor specific.

Transition HTRBIST1:HTI1: When the host's application layer has been notified the Transport layer shall transition to the HTI1:HostIdle state.

16.4 Device transport states

16.4.1 Device transport idle state diagram

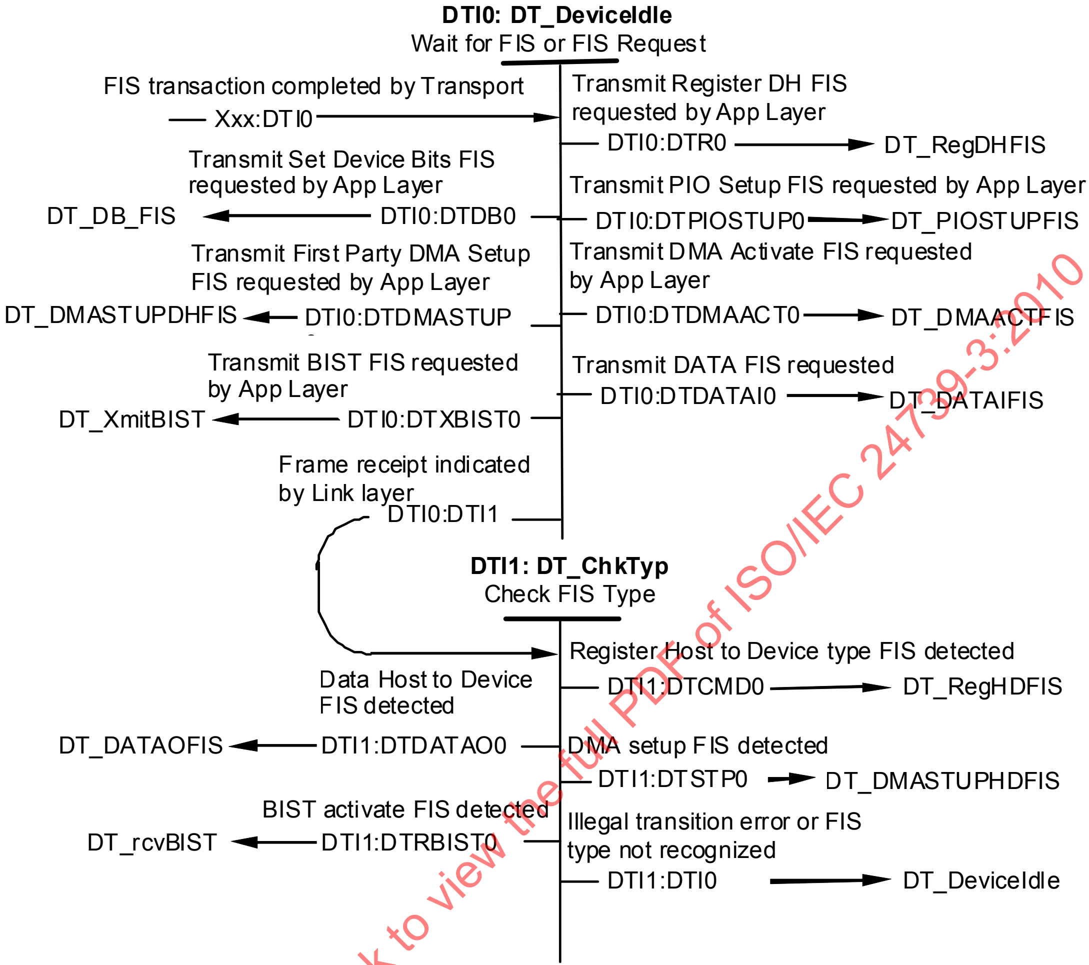


Figure 81 – Device transport idle state diagram (States DTI0-DTI1)

DTI0: DT_DeviceIdle state: This state is entered when a Frame Information Structure (FIS) transaction has been completed by the Transport layer, see Figure 81.

When in this state, the Transport layer waits for the device's application layer to indicate that an FIS is to be transmitted or the Link layer to indicate that an FIS is being received.

Transition DTI0:DTR0: When the device's Application layer indicates that a register FIS is to be transmitted, the Transport layer shall make a transition to the DTR0: DT_RegDHFIS state.

Transition DTI0:DTDB0: When the device's Application layer indicates that a Set Device Bits FIS is to be transmitted, the Transport layer shall make a transition to the DTDB0:DT_DB_FIS state.

Transition DTI0:DTPIOSTUP0: When the device's Application layer indicates that a PIO Setup FIS is to be transmitted, the Transport layer shall make a transition to the DTPIOSTUP0: DT_PIOSTUPFIS state.

Transition DTI0:DTDMAACT0: When the device's Application layer indicates that a DMA Activate FIS is to be transmitted, the Transport layer shall make a transition to the DTDMAACT0: DT_DMAACTFIS state.

Transition DTI0:DTDMASTUP0: When the device's Application layer indicates that a First Party DMA Setup FIS is to be transmitted, the Transport layer shall make a transition to the DTDMASTUP0: DT_DMASTUPDHFIS state.

Transition DTI0:DTDATAI0: When the device's Application layer indicates that a Data FIS is to be transmitted, the Transport layer shall make a transition to the DTDATAI0: DT_DATAIFIS state.

Transition DTI0:DTXBIST1: When the device's Application layer indicates that a BIST Activate FIS is to be transmitted, the Transport layer shall make a transition to the DTXBIST1:DT XmitBIST state.

Transition DTI0:DTI1: When the Link layer indicates that an FIS is being received, the Transport layer shall make a transition to the DTI1: DT_ChkTyp state.

DTI1: DT_ChkTyp state: This state is entered when the Transport layer is idle and Link layer indicates that an FIS is being received, see Figure 81.

When in this state, the Transport layer checks the FIS type of the incoming FIS.

Transition DTI1:DTCMD0: When the incoming FIS is a Register - Host to Device FIS type, the Transport layer shall make a transition to the DTCMD0: DT_RegHDFIS state.

Transition DTI1:DTDATAO0: When the incoming FIS is a Data - Host to Device FIS type, the Transport layer shall make a transition to the DTDATAO0: DT_DATAOFIS state.

Transition DTI1:DTSTP0: When the incoming FIS is a First Party DMA Setup FIS type, the Transport layer shall make a transition to the DTSTP0: DT_ DMASTUPHDFIS state.

Transition DTI1:DTRBIST1: When the incoming FIS is a BIST Activate FIS type, the Transport layer shall make a transition to the DTRBIST1:DT Rcv BIST state.

Transition DTI1:DTI0: When the Transport layer receives notification from the Link layer of an illegal state transition or the FIS type is not recognized, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

16.4.2 Device Transport send Register, Device to Host state diagram

The protocol shown in Figure 82 builds a Register - Device to Host FIS that contains the register content and sends it to the host when the device's Application layer requests the transmission.

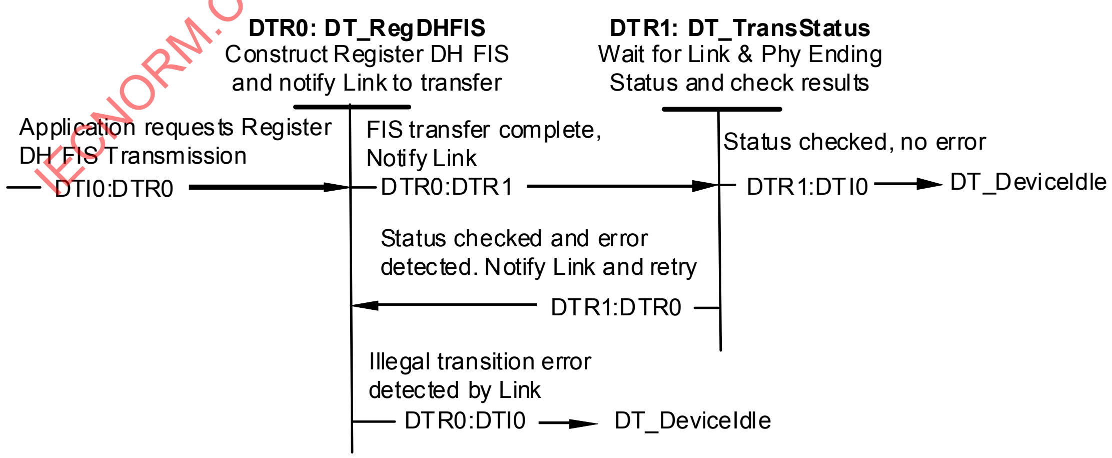


Figure 82 – Device transport send register – Device to host state diagram (DTR0-DTR1)

DTR0: DT_RegDHFIS state: This state is entered the Application requests the transmission of a Register - Device to Host FIS.

When in this state, the Transport layer shall construct a Register - Device to Host FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer.

Transition DTR0:DTR1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTR1: DT_TransStatus state.

Transition DTR0:DTI0: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTR1: DT_TransStatus state: This state is entered when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTR1:DTI0: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTR1:DTR0: When the FIS status has been handled, and an error detected, the Transport layer shall report status to the Link layer, and retry this transfer by transitioning to the DTR0: DT_RegDHFIS state.

16.4.3 Device Transport send Set Device Bits FIS state diagram

The protocol shown in Figure 83 sends a Set Device Bits FIS to the host adapter when the device's Application layer requests the transmission.

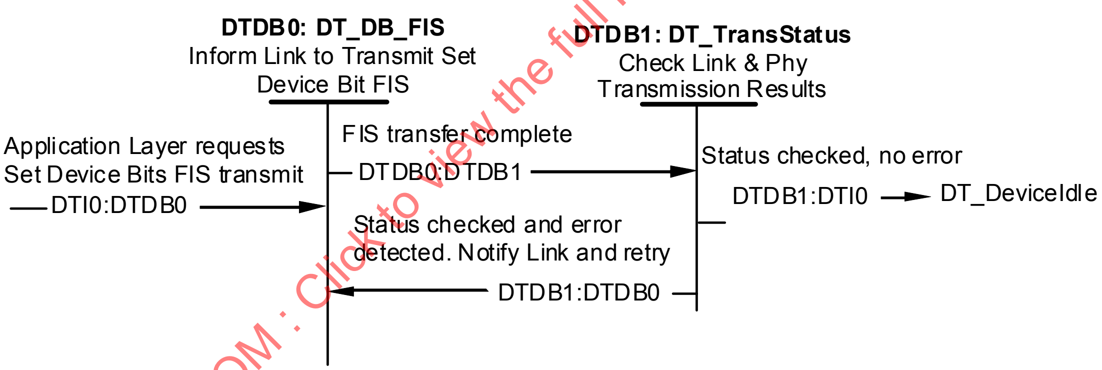


Figure 83 – Device transport send set device bits FIS state diagram (DTDB0-DTDB1)

DTDB0:DT_DB_FIS state: This state is entered when the device's Application layer requests the transmission of a Set Device Bits FIS.

When in this state, the Transport layer shall construct a Set Device Bits FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer.

Transition DTDB0:DTDB1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTDB1:DT_TransStatus state.

DTDB1:DT_TransStatus state: This state is entered when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTDB1:DTI0: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0:DT_DeviceIdle state.

Transition DTDB1:DTDB0: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DTDB0:DT_DB_FIS state.

16.4.4 Device Transport transmit PIO Setup, Device to Host FIS state diagram

The protocol shown in Figure 84 transmits a PIO Setup - Device to Host FIS to the host. Following this PIO Setup frame, a single data frame containing PIO data shall be transmitted or received depending on the state of the D bit in the PIO Setup frame.

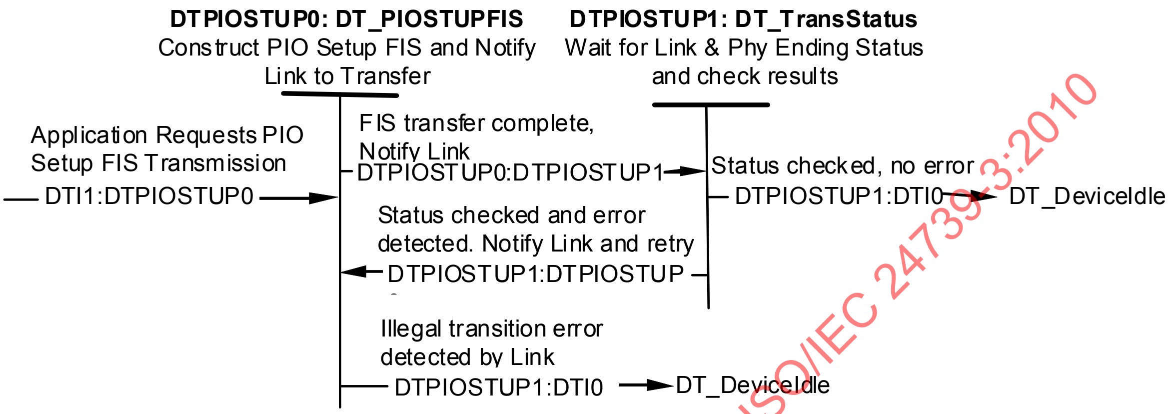


Figure 84 – Device transport transmit PIO setup – Device to Host FIS state diagram (States DTPIOSTUP0-DTPIOSTUP1)

DTPIOSTUP0: DT_PIOSTUPFIS state: This state is entered the device's Application layer requests the transmission of a PIO Setup - Device to Host FIS.

When in this state, the Transport layer shall construct a PIO Setup - Device to Host FIS, notify the Link layer that the FIS is to be transmitted and pass the FIS to the Link layer.

Transition DTPIOSTUP0:DTPIOSTUP1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTPIOSTUP1: DT_TransStatus state.

Transition DTPIOSTUP0:DTI0: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTPIOSTUP1: DT_TransStatus state: This state is entered when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTPIOSTUP1:DTI0: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTPIOSTUP1:2: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DTPIOSTUP0: DT_PIOSTUPFIS state.

16.4.5 Device Transport transmit DMA Activate FIS state diagram

The protocol shown in Figure 85 transmits a DMA Activate FIS to the host adapter. Following this DMA Activate frame, a data frame of DMA data shall be sent from the host to the device.

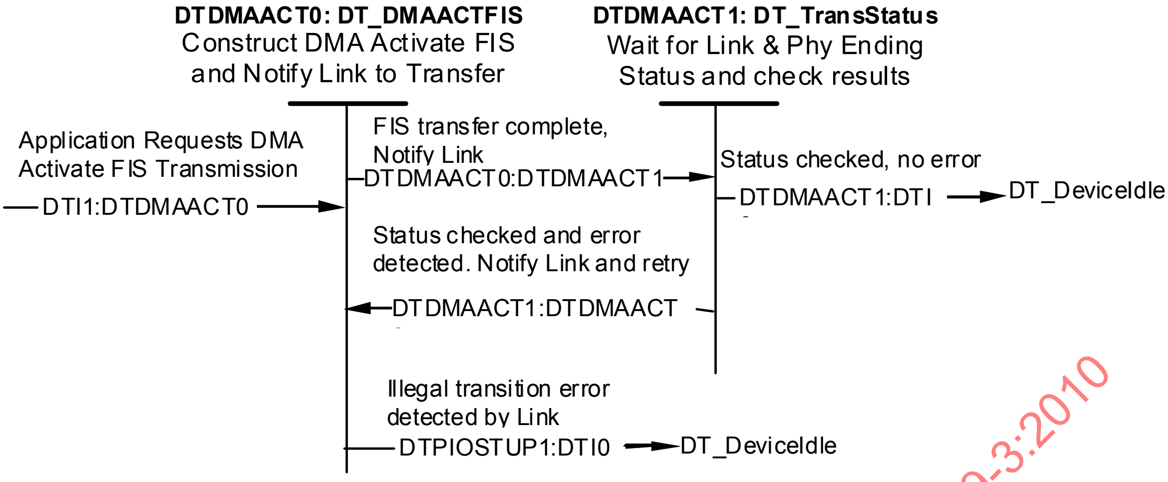


Figure 85 – Device transport transmit DMA activate FIS state diagram (States DTDMAACT0-DTDMAACT1)

DTDMAACT0: DT_DMAACTFIS state: This state is entered the Application requests the transmission of a DMA Activate FIS.

When in this state, the Transport layer shall construct a DMA Activate FIS, notify the Link layer that the FIS is to be transmitted and pass the FIS to the Link layer.

Transition DTDMAACT0:DTDMAACT11: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTDMAACT1: DT_TransStatus state.

Transition DTDMAACT0:DTI0: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTDMAACT1: DT_TransStatus state: This state is entered is when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTDMAACT1:DTI0: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDMAACT1:DTDMAACT0: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DTDMAACT0: DT_DMAACTFIS state.

16.4.6 Device Transport transmit First Party DMA Setup, Device to Host FIS state diagram

The protocol shown in Figure 86 transmits a First Party DMA Setup - Device to Host FIS to the host adapter. This FIS is a request by the device for the host adapter to program the DMA controller for a First Party DMA transfer and is followed by one or more Data FISes that transfer the data to or from the host adapter depending on the direction of the transfer. The First Party DMA Setup - Device to Host request includes the transfer direction indicator, the host buffer identifier, the host buffer offset, the byte count and the interrupt flag.

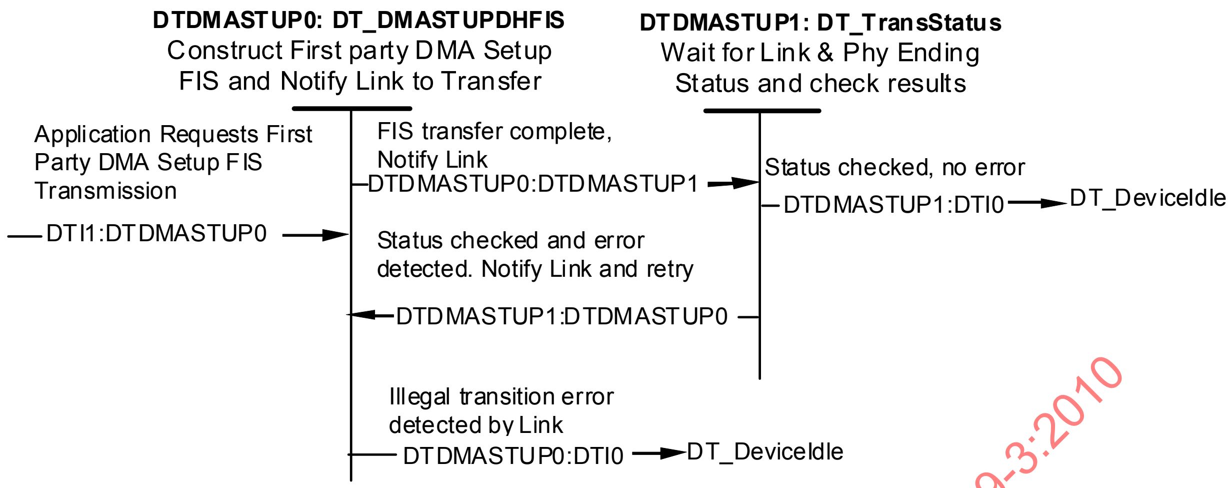


Figure 86 – Device transport transmit First Party DMA setup – Device to Host state diagram (States DTDMASTUP0-DTDMASTUP1)

DTDMASTUP0: DT_DMASTUPDHFIS state: This state is entered when the Application requests the transmission of a First Party DMA Setup - Device to Host FIS.

When in this state, the Transport layer shall construct a First Party DMA Setup - Device to Host FIS, notify the Link layer that the FIS is to be transmitted and pass the FIS to the Link layer.

Transition DTDMASTUP0:DTDMASTUP1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTDMASTUP1: DT_TransStatus state.

Transition DTIDMASTUP0:DTI0: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTPDMASTUP1: DT_TransStatus state: This state is entered when the entire FIS has been passed to the Link layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTDMASTUP1:DTI0: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDMASTUP1:DTDMASTUP0: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DTDMASTUP0: DT_DMASTUPFIS state.

16.4.7 Device Transport transmit Data, Device to Host FIS diagram

The protocol shown in Figure 87 builds a Data - Device to Host FIS.

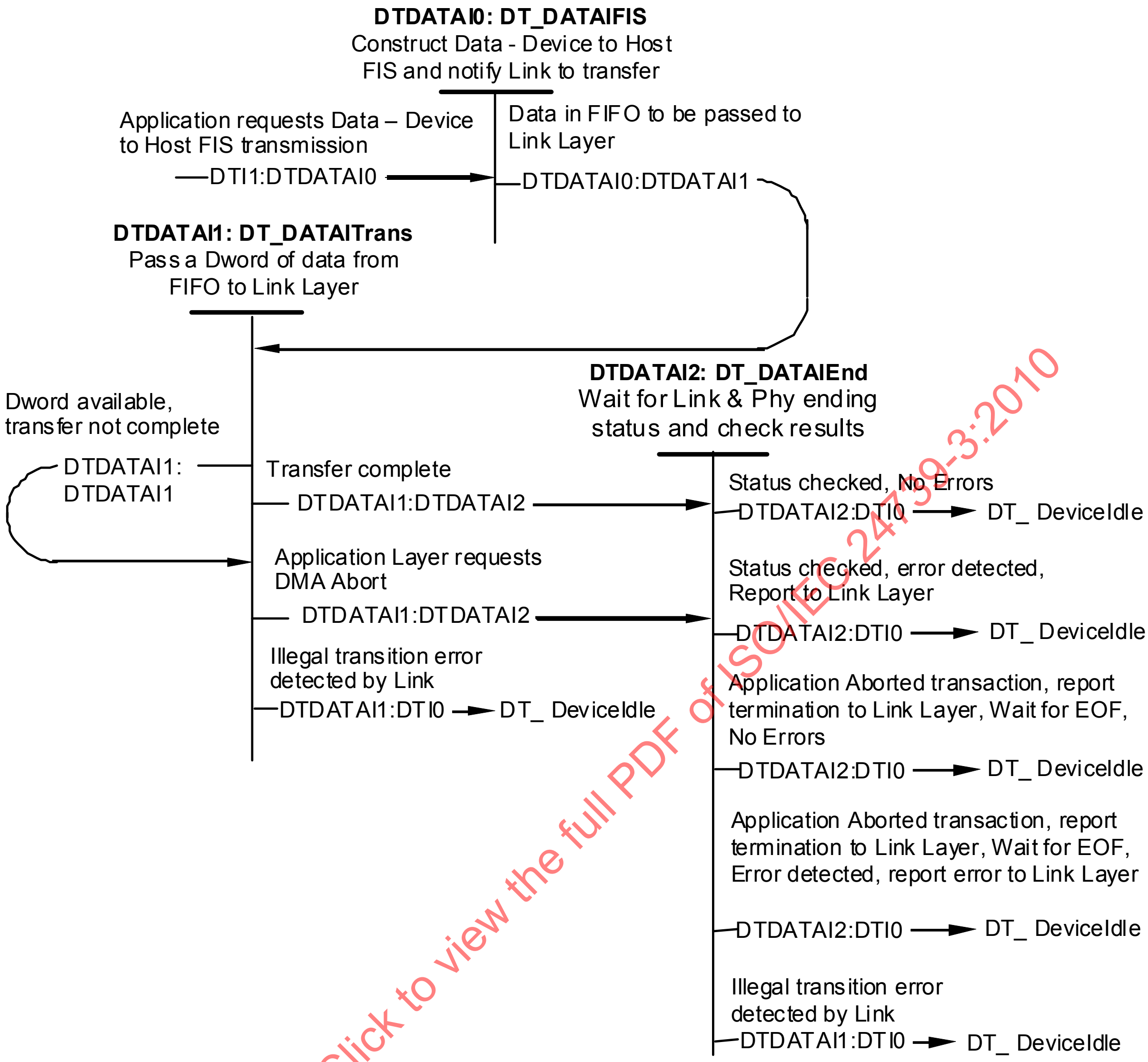


Figure 87 – Device transport transmit data – Device to Host FIS diagram (State DTDATAI0-DTDATAI2)

DTDATAI0: DT_DATAFIS state: This state is entered when the device's Application layer has requested the transmission of a Data - Device to Host FIS.

When in this state the Transport layer shall pass DMA data to the Link layer.

Transition DTDATAI0:DTDATAI1: When ready and there is data to be passed to the Link layer, the Transport layer shall transition to the DTDATAI1: DT_DATAITrans state.

DTDATAI1: DT_DATAITrans state: This state is entered when data is available to be passed the Link layer.

When in this state, the Transport layer shall pass a DWORD of data to the Link layer.

Transition DTDATAI1:DTDATAI1: If the transfer is not complete, the Transport layer shall transition to the DTDATAI1: DT_DATAITrans state.

Transition DTDATAI1:DTDATAI2: When the transfer is complete, the Transport layer shall transition to the DTDATAI2: DT_DATAIEnd state.

When the device's Application layer requests that a DMA operation is to be aborted, the Transport Layer shall transition to the DTDATAI2:DT_DATAIEnd state.

Transition DTDATAI1:DTI0: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTDATAI2: DT_DATAIEnd state: This state is entered when the data transfer is complete or an abort has been requested by the application layer.

When in this state, the Transport layer shall wait for the Link and Phy ending status for the FIS and take appropriate error handling action if required.

Transition DTDATAI2:DTI0: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI1: DT_DeviceIdle state.

When the FIS status has been handled and an error detected, status shall be reported to the Link layer. The Transport layer shall transition to the DTI0: DT_DeviceIdle state.

When the device's application layer requests the termination of a DMA data in transaction, it will report the abort condition to the Link, terminate the FIS normally with good CRC, and if no error is detected, make a transition to the DTI0:DT_DeviceIdle state. This transition is optional in response to a DMAT reception. Use of DMAT is not recommended, see 15.4.6.

When the device's application layer requests the termination of a DMA data in transaction, it will report the abort condition to the Link, terminate the FIS normally with good CRC and if an error is detected, report the error to the Link layer, and make a transition to the DTI0:DT_DeviceIdle state. This transition is optional in response to a DMAT reception. Use of DMAT is not recommended, see 15.4.6.

When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

16.4.8 Device Transport transmit BIST Activate FIS diagram

The protocol shown in Figure 88 builds a BIST Activate FIS that tells the host to prepare to enter the appropriate Built-in Self Test mode. After successful transmission, the device Transport layer enters the idle state. The application layer, upon detecting successful transmission to the host shall then cause the device's Transport layer, Link layer and Physical layer to enter the appropriate mode for the transmission of the built-in test data defined by the FIS. The means by which the Transport, Link and Physical layers are placed into self-test mode are not defined by this standard.

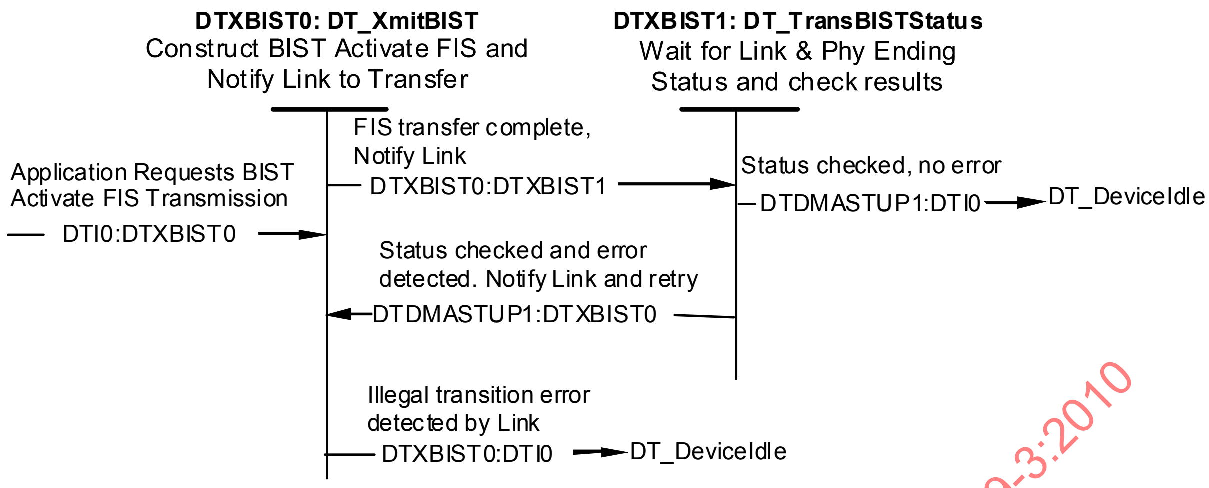


Figure 88 – Device transport transmit BIST activate FIS diagram (States DTXBIST0-DTXBIST1)

DTXBIST0: DT_XmitBIST state: This state is entered to send a BIST FIS to the host.

Transition DTXBIST0:DTXBIST1: When the entire FIS has been passed to the Link layer, the Transport layer shall indicate to the Link layer that the FIS transmission is complete and make a transition to the DTXBIST1:DT_TransBISTSTatus state.

Transition DTXBIST1:DTI0: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTXBIST1: DT_TransBISTStatus state: This state is entered when the entire FIS has been passed to the Link layer.

Transition DTXBIST1:DTI0: When the FIS transmission is completed and no errors have been detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTXBIST1:DTXBIST0: When the FIS transmission is completed and at least one error is detected, the Transport layer shall transition to the DTXBIST0: DT_XmitBIST state.

16.4.9 Device Transport decompose Register, Host to Device state diagram

The protocol shown in Figure 89 receives a Register - Host to Device FIS, places received register content into the device registers, and notifies the device's Application layer of the FIS receipt.

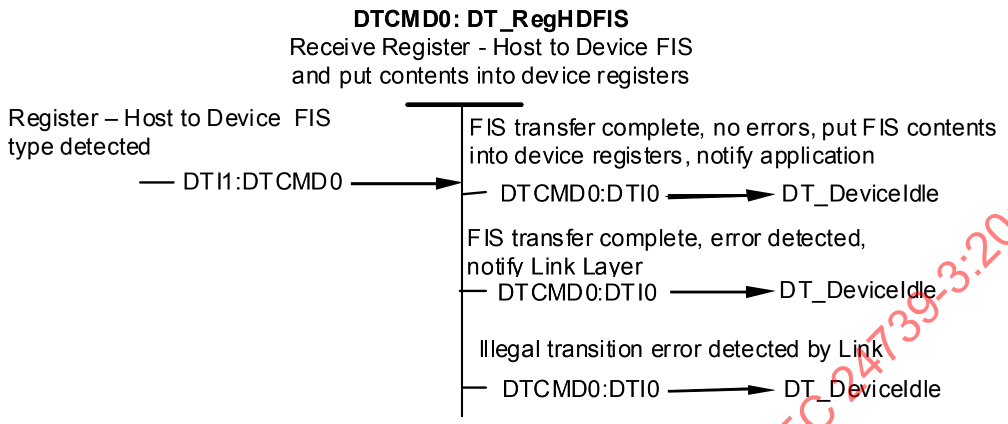


Figure 89 – Device transport decompose register – Host to Device state diagram (State DTCMD0)

DTCMD0: DT_RegHDFIS state: This state is entered when the receipt of a DT_RegHDFISFIS is recognized.

When in this state, the Transport layer shall receive the FIS and place the contents of the FIS into the device registers when it is determined that the FIS was received without error.

Transition DTCMD0:DTI0: When the entire FIS has been received from the Link layer without error, the Transport layer shall indicate to the device's Application layer that a command FIS was received and make a transition to the DTI0: DT_DeviceIdle state.

Transition DTCMD0:DTI0: When the entire FIS has been received from the Link layer and an error has been detected, status shall be sent to the Link layer. The Transport layer shall make a transition to the DTI0:DT_DeviceIdle state.

Transition DTCMD0:DTI0: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

16.4.10 Device Transport decompose Data (Host to Device) FIS state diagram

The protocol shown in Figure 90 receives a Data - Host to Device FIS.

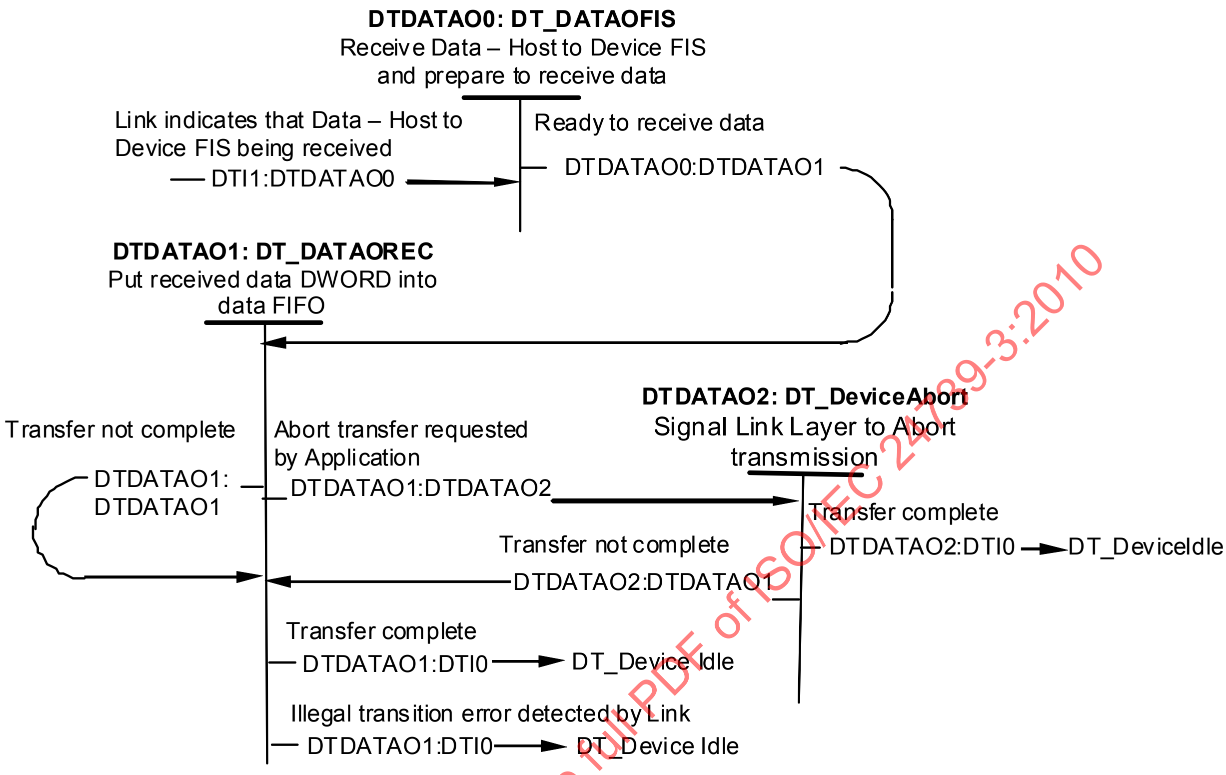


Figure 90 – Device transport decompose data (host to device) FIS state diagram (States DTDATAO0-DTDATAO2)

DTDATAO0: DT_DATAOFIS state: This state is entered when the Link layer has indicated that an FIS is being received and that the Transport layer has determined the FIS is of Data - Host to Device type.

When in this state, the Transport layer shall prepare to receive the data.

Transition DTDATAO0:DTDATAO1: When ready to receive the data, the Transport layer shall make a transition to the DTDATAO1: DT_DATAOREC state.

DTDATAO1: DT_DATAOREC state: This state is entered when the Transport layer is ready to receive the data.

When in this state, the Transport layer shall wait for the Link layer to indicate the transfer is complete.

Transition DTDATAO1:DTDATAO1: When Link layer has not indicated that the end of the FIS has been reached, the Transport layer shall transition to the DTDATAO1: DT_DATAOREC state.

Transition DTDATAO1:DTI0: When the Link layer indicates that the end of the FIS has been reached, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

Transition DTDATAO1:DTDATAO2: When the device's Application layer indicates that the FIS is to be aborted, the Transport layer shall transition to the DTDATAO2: DT_DeviceAbort state.

DTDATAO2: DT_DeviceAbort state: This state is entered when the device's application layer indicates that the current transfer is to be aborted. (Use of DMAT is not recommended, see 15.4.6.)

When in this state, the Transport layer shall signal the Link layer to transmit DMAT primitive and return to either DT_DATAOREC or if the abort occurs coincident with an end of transfer indication from the Link, then transition to DTI0: DT_DeviceIdle. After signalling the Link Layer to transmit DMAT, the Transport returns to normal data transfer and awaits the end of transfer indication from the Link.

Transition DTDATAO2:DTDATAO1: Inform Link layer to issue an abort. When Transfer is not complete, the Transport layer shall transition to the DTDATAO1: DT_DATAOREC state.

Transition DTDATAO2:DTI0: Inform Link layer to issue an abort. When the Link layer indicates that the end of the FIS has been reached, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

16.4.11 Device Transport decompose First Party DMA Setup FIS, Host to Device or Device to Host state diagram

The protocol shown in Figure 91 receives a First Party DMA Setup - Host to Device or Device to Host FIS, passes received First Party DMA Setup content, and notification of FIS receipt to the Application layer.

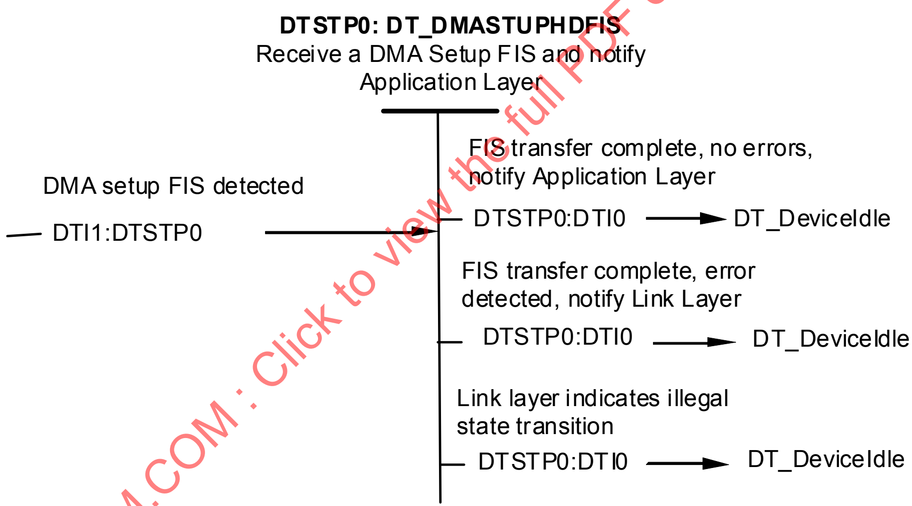


Figure 91 – Device transport decompose First Party DMA Setup FIS – Host to Device or device to host state diagram (State DTDMASTUP0)

DTSTP0: DT_DMASTUPHDFIS state: This state is entered when the receipt of a DT_DMASTUP FIS is recognized.

Transition DTSTP0:DTI0: When the entire FIS has been received from the Link layer without error, the Transport layer shall indicate to the device's Application layer that a First Party DMA Setup FIS was received and make a transition to the DTI0: DT_DeviceIdle state.

When the entire FIS has been received from the Link layer and an error has been detected, status shall be sent to the link layer. The Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

16.4.12 Device Transport decompose a BIST Activate FIS state diagram

The protocol shown in Figure 92 receives an FIS that instructs the device to enter one of several Built-in Self-test modes that cause the device to retransmit the data it receives. If the mode is supported the Device's application layer will place both the transmit and receive portions of the Transport, Link and/or Physical layers into appropriate state to perform the loopback operation.

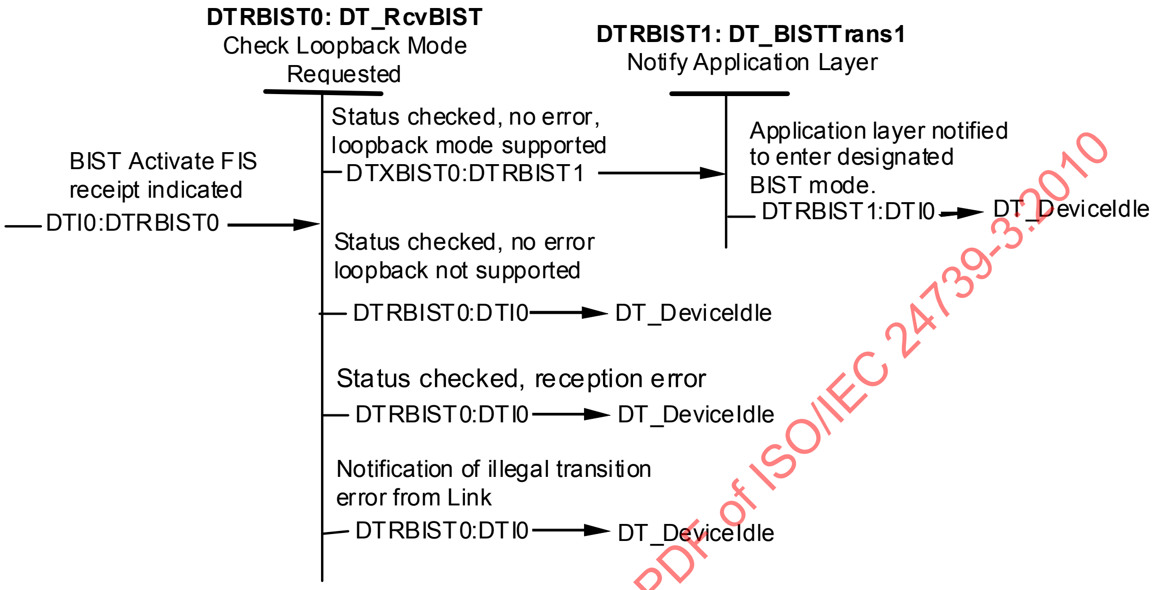


Figure 92 – Device transport decompose BIST activate FIS (States DTRBIST0-DTRBIST1)

DTRBIST0: DT_RcvBIST state: This state is entered when the Link layer has indicated that an FIS is being received and the Transport layer has determined that a BIST Activate FIS is being received.

When in this state, the Transport layer shall determine the validity of the loopback request.

Transition DTRBIST0:DTRBIST1: If no reception error is detected and the FIS contents indicate a form of loopback request that is supported by the host the Transport layer shall make a transition to the DTRBIST1: DT_BISTTrans1 state.

Transition DTRBIST0:DTI12: If no reception error is detected and the FIS contents indicate a form of loopback request that is not supported by the host the Transport layer shall make a transition to the DTI1:DT_DeviceIdle state.

Transition DTRBIST0:DTI0: If a reception error is indicated the Transport layer shall make a transition to the DTI1:DT_DeviceIdle state.

Transition DTRBIST0:DTI1: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the DTI0: DT_DeviceIdle state.

DTRBIST1: DT_BISTTrans1 state: This state is entered when the Transport layer has determined that a valid BIST Activate FIS has been received.

Having received a valid FIS, the Transport layer informs the device's application layer that it should place the Transport, Link and Physical layers into the appropriate modes to loop the received data back to the transmitter. The method by which this is performed is vendor specific.

Transition DTRBIST1:DTI0: When the device's Application layer has been notified the Transport layer shall transition to the DTI0:DT_DeviceIdle state.

17 Serial interface Device Command Layer Protocol

17.1 COMRESET or SRST sent by Host

In the following Device Command Layer Protocols, if the host sends COMRESET before the device has completed executing a command layer protocol, then the device shall immediately start executing the COMRESET protocol from the beginning. If the host asserts SRST in the Device Control register before the device has completed executing a command layer protocol, then the device shall immediately start executing its software reset protocol from the beginning.

17.2 Power-on and COMRESET protocol diagram

If the host asserts Hard Reset by transmiting a COMRESET Sequence the device shall execute the hardware reset protocol regardless of the power management mode or the current device command layer state.

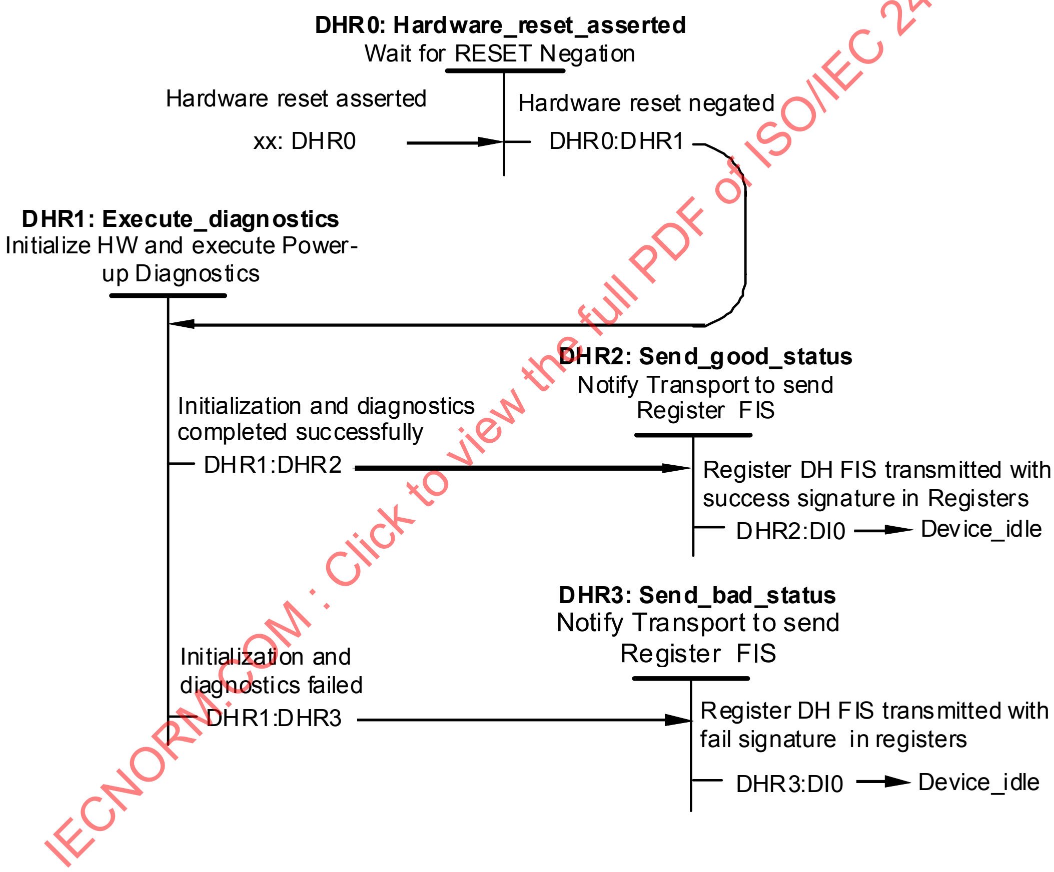


Figure 93 – Power on and COMRESET protocol (States DHR0-DHR3)

DHR0: Hardware_reset_asserted: This state is entered when the Transport layer indicates that the COMRESET signal is asserted, see Figure 93.

When in this state, the device awaits the negation of the COMRESET signal.

Transition DHR0:DHR1: When the Transport layer indicates that the COMRESET signal has been negated, the device shall transition to the DHR1: Execute_diagnostics state.

DHR1: Execute_diagnostics: This state is entered when the Transport layer indicates that the COMRESET signal has been negated, see Figure 93.

When in this state, the device initializes the device hardware and executes its power-up diagnostics.

Transition DHR1:DHR2: When the device hardware has been initialized and the power-up diagnostics successfully completed, the device shall transition to the DHR2: Send_good_status state.

Transition DHR1:DHR3: When the device hardware has been initialized and the power-up diagnostics failed, the device shall transition to the DHR3: Send_bad_status state.

DHR2: Send_good_status: This state is entered when the device hardware has been initialized and the power-up diagnostics successfully completed, see Figure 93.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. The device signature shall be set in the registers as defined in Clause 5 of ISO/IEC 24739-1:2009.

Transition DHR2:DI0: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

DHR3: Send_bad_status: This state is entered when the device hardware has been initialized and the power-up diagnostics failed, see Figure 93.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. The device signature shall be set in the registers as defined in Clause 5 of ISO/IEC 24739-1:2009 .

Transition DHR3:DI0: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

17.3 Device Idle protocol

The state diagram below, see Figure 94, describes the idle protocol for a device. States and transitions preceded by an * are utilized only when queuing is implemented (see Clause 4).

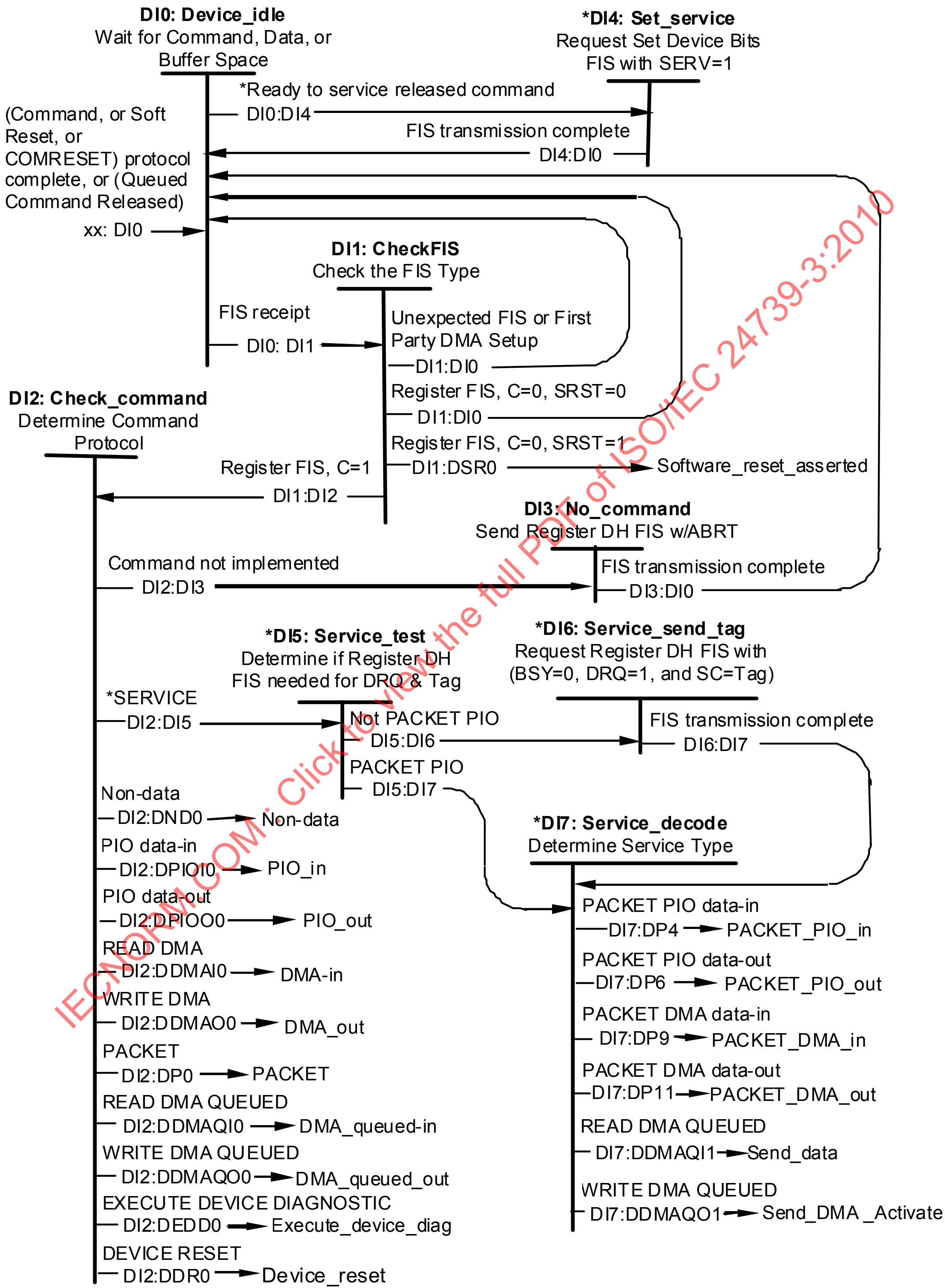


Figure 94 – Device idle protocol (States DI0-DI7)

DI0: Device_Idle: This state is entered when the device has completed the execution of a command protocol, a COMRESET protocol, a software reset protocol, or a queued command has been released.

When in this state, the device is awaiting a command. If queuing is supported, the device may be waiting to acquire data or establish buffer space to complete a queued command.

Transition DI0:DI1: When the device receives an FIS from the Transport layer, the device shall transition to the DI1: Check_FIS state.

*** Transition DI0:DI4:** When the device is ready to complete the data transfer for a queued command, the device shall transition to the DI4: Set_service state.

DI1: Check_FIS state: This state is entered when the device receives an FIS from the Transport layer.

When in this state, the device shall check the FIS type.

Transition DI1:DSR0: If the FIS type is a Register FIS, the C bit in the FIS is cleared, and the SRST bit in the FIS is set, the device shall transition to the DSR0: Software_reset_asserted state.

Transition DI1:DI0: If the FIS type is a Register FIS, the C bit in the FIS is cleared, and the SRST bit in the FIS is cleared, the device shall transition to the DI0: Device_idle state.

Transition DI1:DI2: If the FIS type is a Register FIS and the C bit in the FIS is set, the device shall transition to the DI2: Check_command state.

Transition DI1:DI0: If the FIS type is a First Party DMA Setup FIS, the device shall inform the Transport layer of the reception of the First Party DMA Setup FIS and transition to the DI0: Device_idle state.

Transition DI1:DI0: For any other FIS, the device shall transition to the DI0: Device_idle state.

DI2: Check_command state: This state is entered when the device recognizes that the received Register FIS contains a new command.

NOTE This state shows transitions for all commands. If a device does not implement any particular command, then transition DI2:11 to state DI3:No_command shall be made.

When in this state, the device shall check the command protocol required by the received command.

Transition DI2:DND0: When the received command is a non-data transfer command, the device shall transition to the DND0: Non-data state.

Transition DI2:DPIOI0: When the received command is a PIO data-in command, the device shall transition to the DPIOI0: PIO_in state.

Transition DI2:DPIOO0: When the received command is a PIO data-out command, the device shall transition to the DPIOO0: PIO_out state.

Transition DI2:DDMAI0: When the received command is a READ DMA command, the device shall transition to the DDMAI0: DMA_in state.

Transition DI2:DDMAO0: When the received command is a WRITE DMA command, the device shall transition to the DDMAO0: DMA_out state.

Transition DI2:DP0: When the received command is a PACKET command, the device shall transition to the DP0: PACKET state.

Transition DI2:DDMAQI0: When the received command is a READ DMA QUEUED command, the device shall transition to the DDMAQI0: DMA_queued_in state.

Transition DI2:DDMAQO0: When the received command is a WRITE DMA QUEUED command, the device shall transition to the DDMAQO0: DMA_queued_out state.

Transition DI2:DEDD0: When the received command is an EXECUTE DEVICE DIAGNOSTICS command, the device shall transition to the DEDD0: Execute_device_diag state.

Transition DI2:DDR0: When the received command is an RESET DEVICE command, the device shall transition to the DDR0: Device_reset state.

Transition DI2:DI3: When the received command is not implemented by the device, the device shall transition to the DI3: No_command state.

Transition DI2:DI5: When the received command is a SERVICE command, the device shall transition to the DI5: Service state.

DI3: No_command state: This state is entered when the device recognizes that the received command is not implemented by the device. The device shall abort any outstanding Queued commands.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the appropriate command description and the I bit set to one.

Transition DI2:DI0: When the Transport layer has transmitted the Register FIS, the device shall transition to the DI0: Device_idle state.

*** DI4: Set_service state:** This state is entered when the ready to complete the data transfer for a queued command.

When in this state, the device shall request that the Transport layer transmit a Set Device Bits FIS with the SERV bit set in the Status register and with all other bits in the Error and Status fields the same as the current contents of the respective registers, and the I bit set to one.

Transition DI4:DI0: When the Transport layer has transmitted the Set Device Bits FIS, the device shall transition to the DI0: Device_idle state.

*** DI5: Service_test state:** This state is entered when the SERVICE command has been received.

When in this state, the device shall determine the type of command that the device has requested service to complete. The PACKET_PIO command includes its own register update to set DRQ and send the command tag, but other queued commands require a Register FIS.

Transition DI5:DI7: When the command to be serviced is a PIO data-in or PIO data-out command, the device shall transition to the DI7: Service_decode state.

Transition DI5:DI6: When the command to be serviced is neither a PIO data-in nor a PIO data-out command, the device shall transition to the DI6: Service_send_tag state.

*** DI6: Service_send_tag state:** This state is entered when the SERVICE command has been received and sending a Register Device to Host FIS is necessary for the command being serviced.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register contents, including the desired command tag for the command being serviced.

Transition DI6:DI7: When the Transport layer has transmitted the Register FIS, the device shall transition to theDI7: Service_decode state.

*** DI7: Service_decode state:** This state is entered when a register FIS has been transmitted, if necessary to send the register contents, including the desired command tag, in response to a SERVICE command.

When in this state, the device shall determine the type of command that the device has requested service to complete, and branch to that command's data transfer and completion.

Transition DI7:DP4: When the command to be serviced is a PIO data-in command, the device shall transition to the DP4: PACKET_PIO_in state.

Transition DI7:DP6: When the command to be serviced is a PIO data-out command, the device shall transition to the DP6: PACKET_PIO_out state.

Transition DI7:DP9: When the command to be serviced is a DMA data-in command, the device shall transition to the DP9: PACKET_DMA_in state.

Transition DI7:DP11: When the command to be serviced is a DMA data-out command, the device shall transition to the DP11: PACKET_DMA_out state.

Transition DI7:DDMAQI1: When the command to be serviced is a READ DMA QUEUED command, the device shall transition to the DDMAQI1: Send_data state.

Transition DI7:DDMAQO1: When the command to be serviced is a WRITE DMA QUEUED command, the device shall transition to the DDMAQO1: Send_DMA_activate state.

17.4 Software reset protocol

When the host sends a Register FIS with a one in the SRST bit position of the Device Control register byte, the device shall execute the software reset protocol regardless of the power management mode.

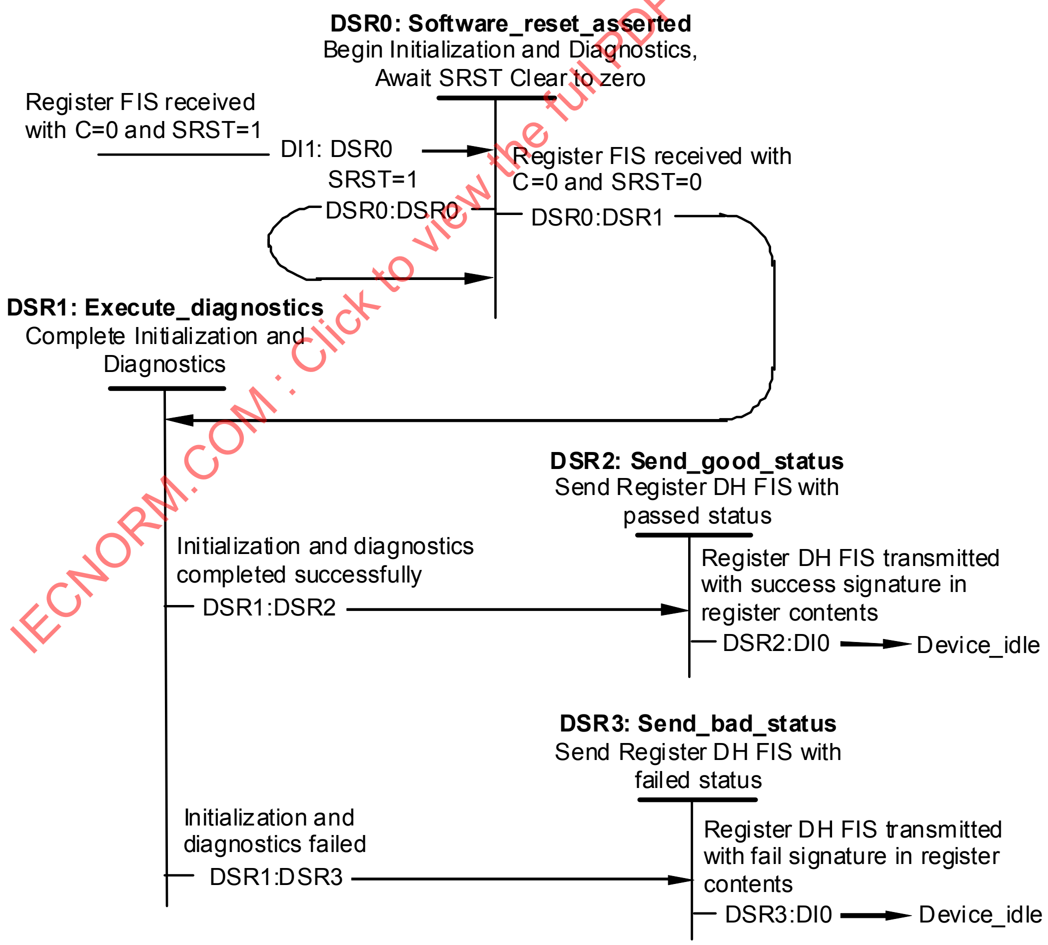


Figure 95 – Software reset protocol (States DSR0-DSR3)

DSR0: Software_reset_asserted: This state is entered when a Register FIS is received with the C bit in the FIS cleared to zero and the SRST bit set to one in the Device Control register.

When in this state, the device begins its initialization and diagnostics execution and awaits the clearing of the SRST bit.

Transition DSR0:DSR1: When a Register FIS is received with the C bit in the FIS cleared to zero and the SRST bit cleared to zero in the Device Control register, the device shall transition to the DSR1: Execute_diagnostics state.

Transition DSR0:DSR0: If a Register FIS is received with the C bit in the FIS set to one, or the SRST bit set to one in the Device Control register, the device shall transition to the DSR0: Software_reset_asserted state.

DSR1: Execute_diagnostics: This state is entered when a Register FIS is received with the C bit in the FIS cleared to zero and the SRST bit cleared to zero in the Device Control register.

When in this state, the device completes initialization and execution of its diagnostics.

Transition DSR1:DSR2: When the device has been initialized and the diagnostics successfully completed, the device shall transition to the DSR2: Send_good_status state.

Transition DSR1:DSR3: When the device has been initialized and the diagnostics failed, the device shall transition to the DSR3: Send_bad_status state.

DSR2: Send_good_status: This state is entered when the device has been initialized and the diagnostics successfully completed.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. The device signature shall be set in the registers as defined in Clause 5 of ISO/IEC 24739-1:2009.

Transition DSR2:DI0: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

DSR3: Send_bad_status: This state is entered when the device has been initialized and the diagnostics failed.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. The device signature shall be set in the registers as defined in Clause 5 of ISO/IEC 24739-1:2009.

Transition DSR3:DI0: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

17.5 EXECUTE DEVICE DIAGNOSTIC command protocol

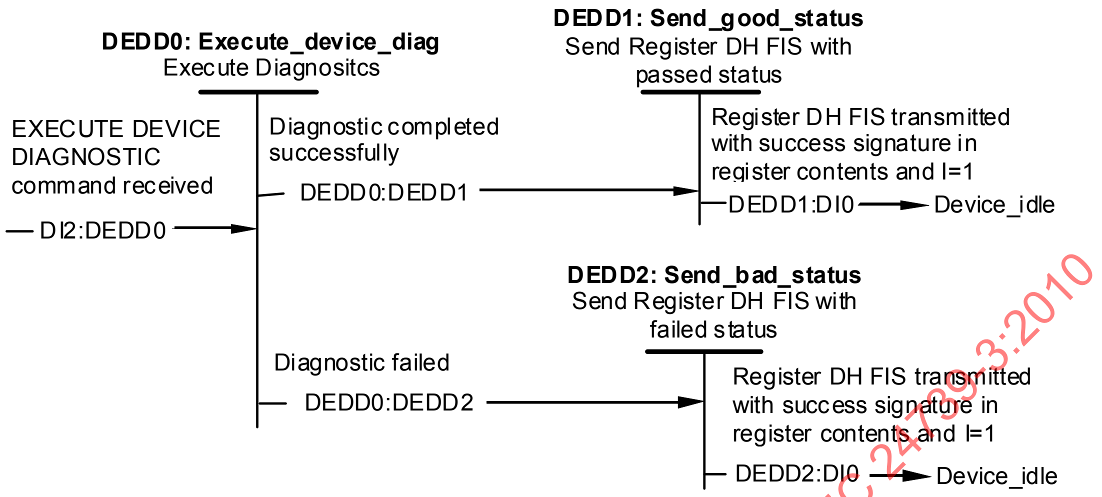


Figure 96 – EXECUTE DEVICE DIAGNOSTIC command protocol (States DEDD0-DEDD2)

DEDD0: Execute_device_diag: This state is entered when an EXECUTE DEVICE DIAGNOSTIC command is received, see Figure 96.

When in this state, the device executes its diagnostics.

Transition DEDD0:DEDD1: When the device successfully completed the diagnostics, the device shall transition to the DEDD1: Send_good_status state.

Transition DEDD1:DEDD2: When the device has failed the diagnostics, the device shall transition to the DEDD2: Send_bad_status state.

DEDD1: Send_good_status: This state is entered when the device has successfully completed the diagnostics. When in this state, the device shall request that the Transport layer transmit a Register FIS to the host, with the I bit set to one. The device signature shall be set in the registers as defined in in Clause 5 of ISO/IEC 24739-1:2009, see Figure 96.

Transition DEDD1:DI0: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

DEDD2: Send_bad_status: This state is entered when the device has been initialized and the diagnostics failed.

When in this state, the device shall request that the Transport layer transmit a Register FIS to the host, with the I bit set to one. The device signature shall be set in the registers as defined in in Clause 5 ISO/IEC 24739-1:2009, see Figure 96.

Transition DEDD2:DI0: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

17.6 DEVICE RESET command protocol

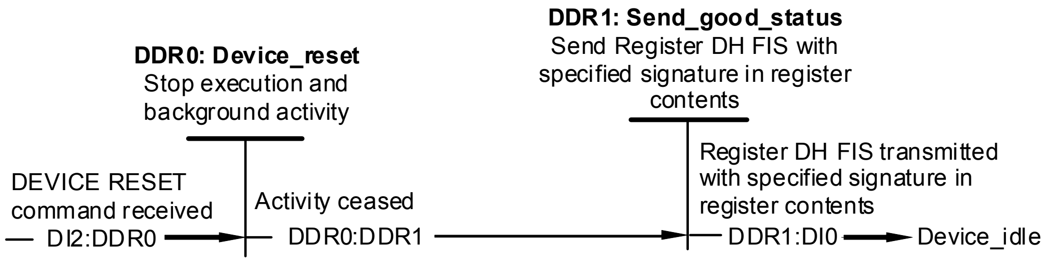


Figure 97 – DEVICE RESET command protocol (States DDR0-DDR1)

DDR0: Device_reset: This state is entered when an DEVICE RESET command is received, see Figure 97.

When in this state, the device stops any execution or activity in progress.

Transition DDR0:DDR1: When the device has ceased any execution or activity and has completed its internal diagnostics, the device shall transition to the DDR1: Send_good_status state.

DDR1: Send_good_status: This state is entered when the device has been initialized and the diagnostics successfully completed, see Figure 97.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. The device signature shall be set in the registers as defined in in Clause 5 of ISO/IEC 24739-1:2009.

Transition DDR1:DI0: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

17.7 Non-data command protocol

Non-Data Commands are listed in Annex B of ISO/IEC 24739-1:2009.

Execution of these commands involves no data transfer. See the NOP command description and the SLEEP command description for additional protocol requirements.

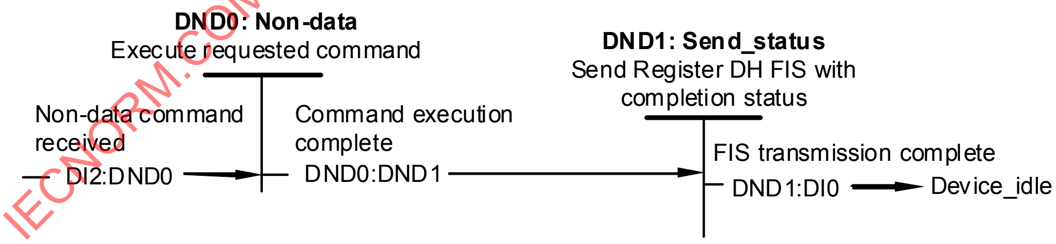


Figure 98 – Non-data command protocol (States DND0-DND1)

DND0: Non-data State: This state is entered when a received command is a non-data command, see Figure 98.

When in this state, the device shall execute the requested command if supported.

Transition DND0:DND1: When command execution completes, the device shall transition to the DND1: Send_status state.

DND1: Send_status State: This state is entered when the execution of the non-data command has been completed, see Figure 98.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the appropriate command description and the I bit set to one.

Transition DND1:DI0: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

17.8 PIO data-in command protocol

PIO Data In commands are listed in Annex B of ISO/IEC 24739-1:2009.

Execution of this class of command includes the PIO transfer of one or more blocks of data from the device to the host.

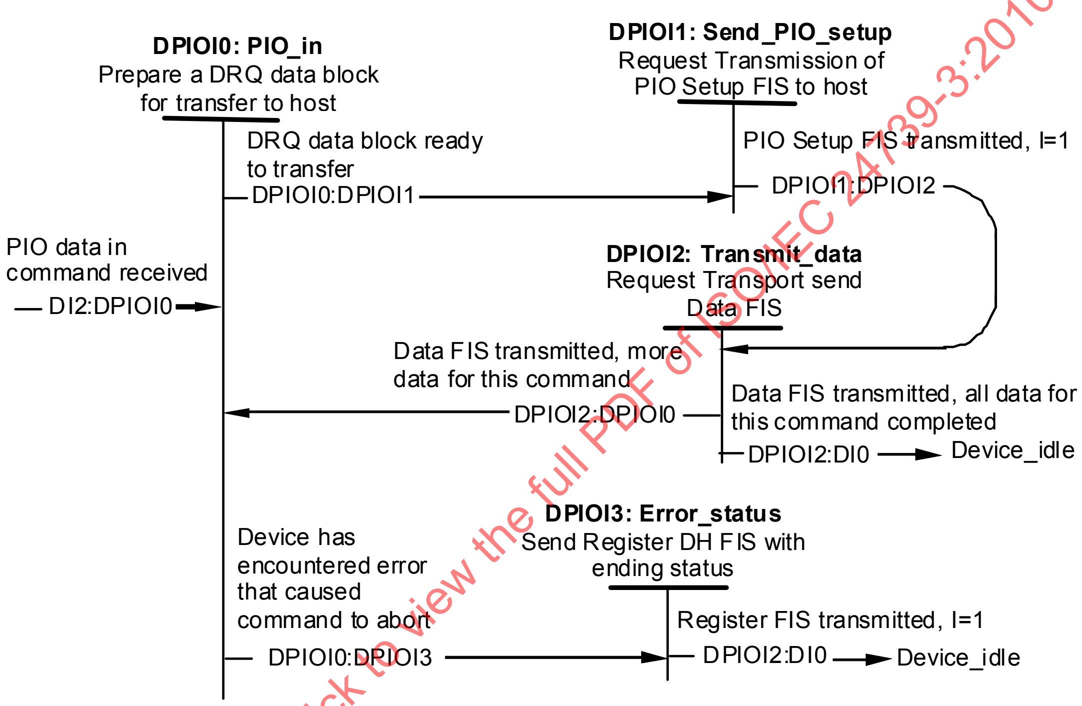


Figure 99 – PIO data-in command protocol (States DPIOI0-DPIOI3)

DPIOI0: PIO_in State: This state is entered when the device receives a PIO data-in command or the transmission of one or more additional DRQ data blocks is required to complete the command, see Figure 99.

When in this state, device shall prepare a DRQ data block for transfer to the host.

Transition DPIOI0:DPIOI1: When the device has a DRQ data block ready to transfer, the device shall transition to the DPIOI1: Send_PIO_setup state.

Transition DPIOI0:DPIOI3: When the device has encountered an error that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DPIOI3: Error_status state.

DPIOI1: Send_PIO_setup: This state is entered when the device is ready to transmit a DRQ data block from the host, see Figure 99.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS. The initial status shall have BSY cleared to zero and DRQ set to one and with register content as described in the appropriate command description. The I bit shall be set. If this is the last DRQ data block requested by the command, the E_Status field shall have BSY cleared to zero and DRQ cleared to zero. If this is not the last data block requested by the command, the E_Status field shall have BSY set to one and DRQ cleared to zero.

Transition DPIOI1:DPIOI2: When the PIO Setup FIS has been transferred, the device shall transition to the DPIOI2: Transmit_data state.

DPIOI2: Transmit_data: This state is entered when the device has transmitted a PIO Setup FIS to the host, see Figure 99.

When in this state, the device shall request that the Transport layer transmit a Data FIS containing the DRQ data block.

Transition DPIOI2:DI0: When the Data FIS has been transferred and all data requested by this command have been transferred, the device shall transition to the DI0: Device_idle.

Transition DPIOI2:DPIOI0: When the Data FIS has been transferred but all data requested by this command has not been transferred, or the 8KB transfer limit has been reached, then the device shall transition to the DPIOI0: PIO_in state.

DPIOI3: Error_status: This state is entered when the device has encountered an error that causes the command to abort before completing the transfer of the requested data, see Figure 99.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the appropriate command description and the I bit set to one.

Transition DPIOI3:DI0: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

17.9 PIO data-out command protocol

PIO Data-out commands are listed in Annex B of ISO/IEC 24739-1:2009. Execution of this class of command includes the PIO transfer of one or more blocks of data from the host to the device.

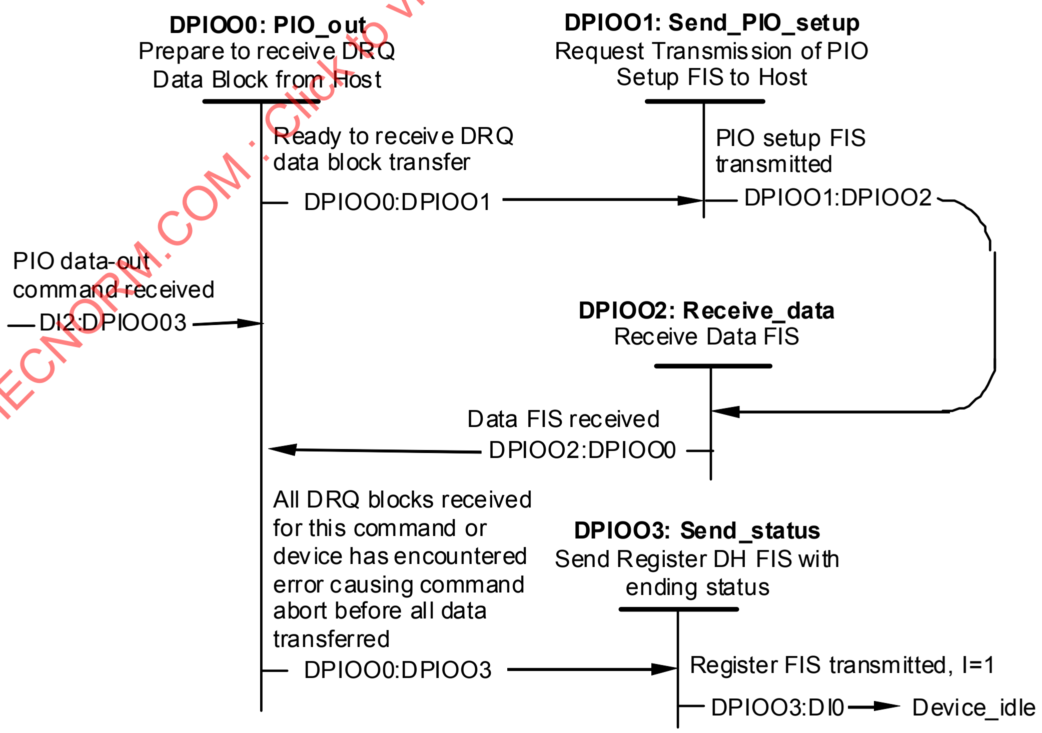


Figure 100 – PIO data-out command protocol (States DPIOO0-DPIOO3)

DPIOI0: PIO_out State: This state is entered when the device receives a PIO data-out command or the receipt of one or more DRQ data blocks is required to complete this command, see Figure 100.

When in this state, device shall prepare to receive a DRQ data block transfer from the host.

Transition DPIOO0:DPIOO1: When the device is ready to receive a DRQ data block, the device shall transition to the DPIOO1: Send_PIO_setup state.

Transition DPIOO0:DPIOO3: When the device has received all DRQ data blocks requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DPIOO3: Send_status state.

DPIOO1: Send_PIO_setup: This state is entered when the device is ready to receive a DRQ data block from the host, see Figure 100.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS. The initial status shall have BSY cleared to zero and DRQ set to one. If this is the first DRQ data block for this command, the I bit shall be cleared to zero. If this is not the first DRQ data block for this command, the I bit shall be set to one. The E_Status field shall have BSY set to one and DRQ cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DPIOO1:DPIOO2: When the PIO Setup FIS has been transferred, the device shall transition to the DPIOO2: Receive_data state.

DPIOO2:Receive_data: This state is entered when the device has transmitted a PIO Setup FIS to the host, see Figure 100.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DPIOO2:DPIOO0: When the Data FIS has been received, the device shall transition to the DPIOO0: PIO_out state.

DPIOO3: Send_status: This state is entered when the device has received all DRQ data blocks requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, see Figure 100.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the appropriate command description and the I bit set to one.

Transition DPIOO3:DI0: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

17.10 DMA data-in command protocol

DMA data-in commands are listed in Annex B of ISO/IEC 24739-1:2009.

Execution of this class of command includes the transfer of one or more blocks of data from the device to the host using DMA transfer.

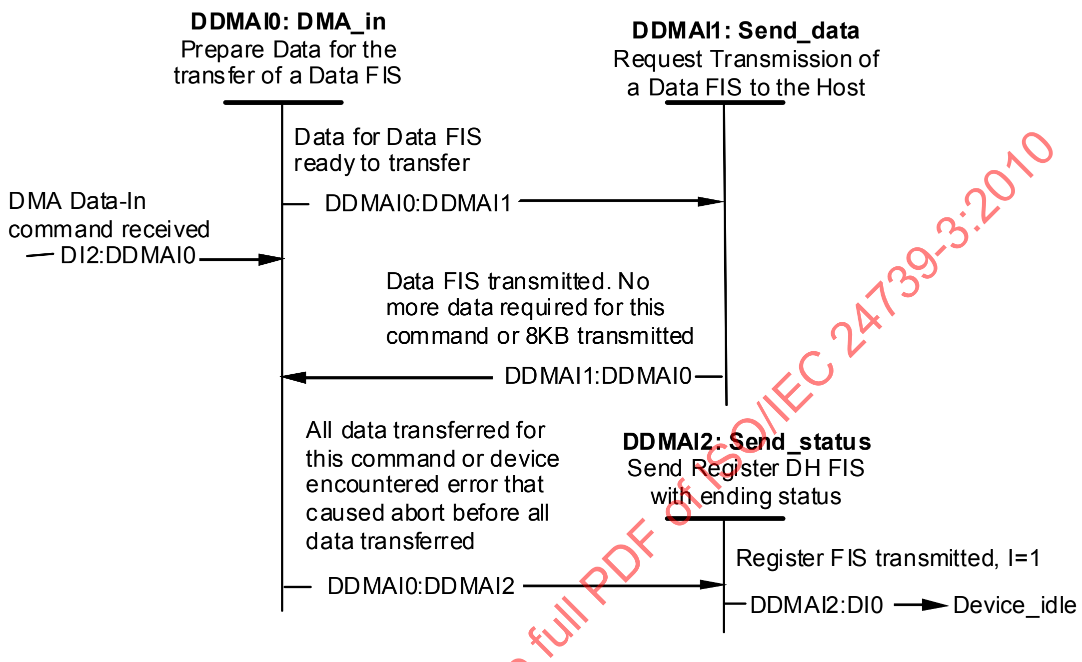


Figure 101 – DMA data-in command protocol (States DDMAI0-DDMAI1)

DDMAI0: DMA_in State: This state is entered when the device receives a DMA data-in command or the transmission of one or more Data FIS is required to complete the command, see Figure 101.

When in this state, device shall prepare the data for transfer of a Data FIS to the host.

Transition DDMAI0:DDMAI1: When the device has the data ready to transfer a Data FIS, the device shall transition to the DDMAI1: Send_data state.

Transition DDMAI0:DDMAI2: When the device has transferred all of the data requested by this command or has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DDMAI2: Send_status state.

DDMAI1: Send_data: This state is entered when the device has the data ready to transfer a Data FIS to the host, see Figure 101.

When in this state, the device shall request that the Transport layer transmit a Data FIS containing the data. The device command layer shall request a Data FIS size of no more than 8KB.

Transition DDMAI1:DDMAI0: When the Data FIS has been transferred, the device shall transition to the DDMAOI0: DMA_in state.

DDMAI2: Send_status: This state is entered when the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data, see Figure 101.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the appropriate command description and the I bit set to one.

Transition DDMAI2:DI0: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

17.11 DMA data out command protocol

DMA data out commands are listed in Annex B of ISO/IEC 24739-1:2009.

Execution of this class of command includes the transfer of one or more blocks of data from the host to the device using DMA transfer. A single interrupt is issued at the completion of the command.

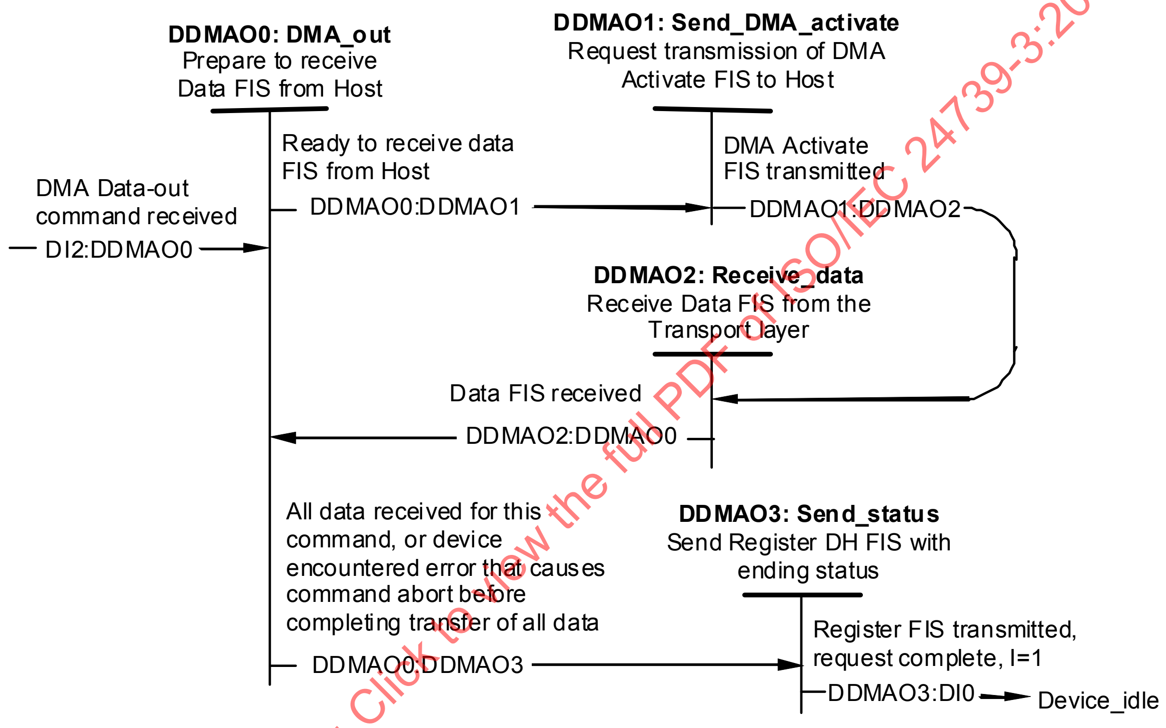


Figure 102 – DMA data-out command protocol (States DDMAO0-DDMAO3)

DDMAO0: DMA_out State: This state is entered when the device receives a DMA data-out command or the receipt of one or more Data FIS is required to complete this command, see Figure 102.

When in this state, device shall prepare to receive a Data FIS from the host.

Transition DDMAO0:DDMAO1: When the device is ready to receive a Data FIS, the device shall transition to the DDMAO1: Send_DMA_activate state.

Transition DDMAO0:DDMAO3: When the device has received all the data requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DDMAO3: Send_status state.

DDMAO1: Send_DMA_activate: This state is entered when the device is ready to receive a Data FIS from the host, see Figure 102.

When in this state, the device shall request that the Transport layer transmit a DMA Activate FIS.

Transition DDMAO1:DDMAO2: When the DMA Activate FIS has been transferred, the device shall transition to the DDMAO2: Receive_data state.

DDMAO2: Receive_data: This state is entered when the device transmitted a DMA Activate FIS to the host, see Figure 102.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DDMAO2:DDMAO0: When the Data FIS has been received, the device shall transition to the DDMAO0: DMA_out state.

DDMAO3: Send_status: This state is entered when the device has received all the data requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, see Figure 102.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the appropriate command description and the I bit set to one.

Transition DDMAO3:DI0: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

17.12 PACKET protocol

The PACKET protocol is illustrated in Figure 103. States marked with an * in Figure 103 are only utilized when queuing is implemented (see Clause 4 of ISO/IEC 24739-1:2009).

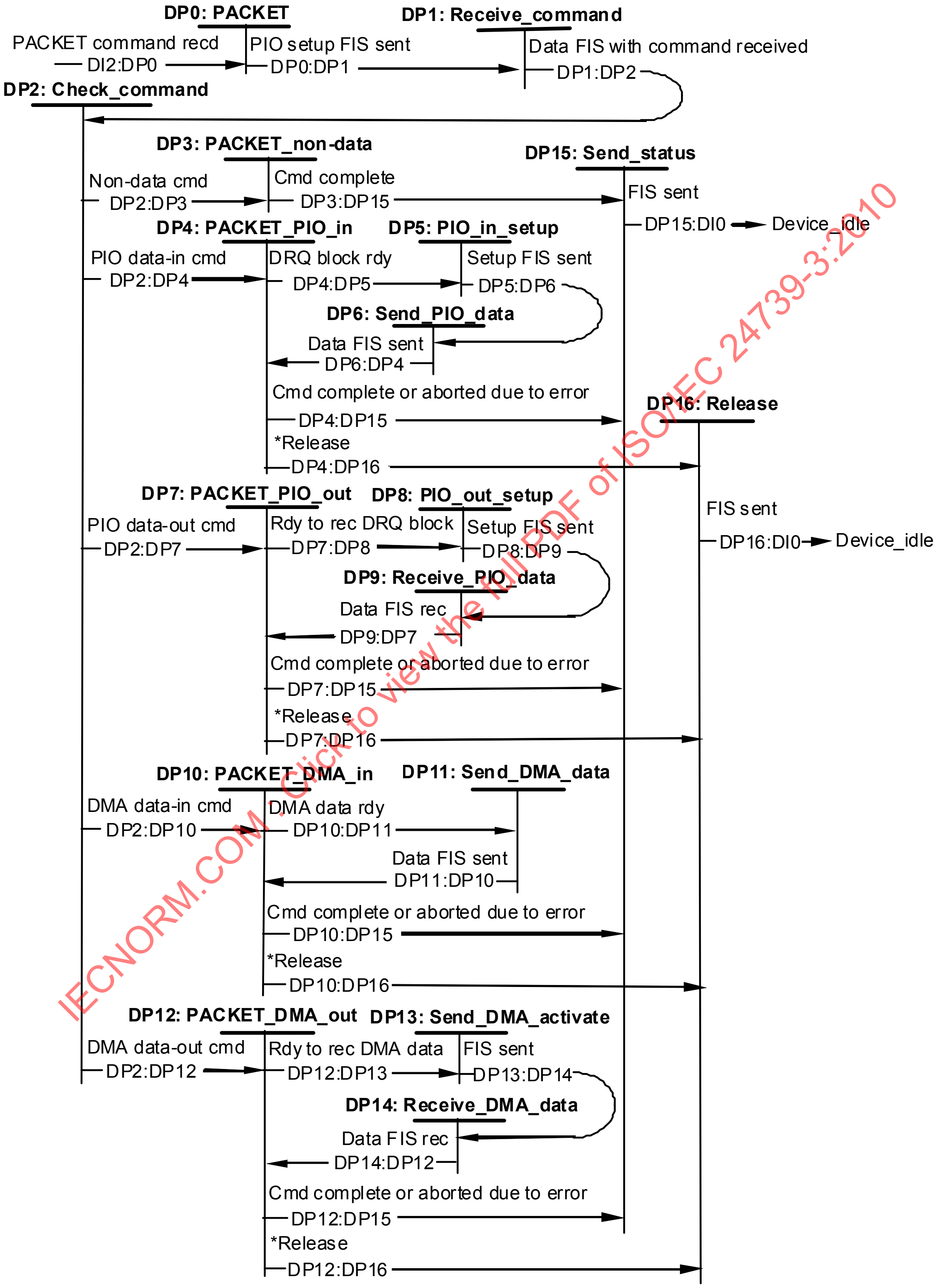


Figure 103 – PACKET command protocol (States DP0-DP16)

DP0: PACKET: This state is entered when the device receives a PACKET command.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS to acquire the command packet associated with this command. The initial status shall have BSY cleared to zero and DRQ set to one. The I bit shall be cleared to zero. The E_Status field shall have BSY set to one and DRQ cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DP0:DP1: When the PIO Setup FIS has been transferred, the device shall transition to the DP1: Receive_command state.

DP1: Receive_command: This state is entered when the device transmitted a PIO Setup FIS to the host to get the command packet.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DP1:DP2: When the Data FIS has been received, the device shall transition to the DP2: Check_command state.

DP2: Check_command: This state is entered when the Data FIS containing the command packet has been received.

When in this state, the device shall determine the protocol for the command contained in the command packet.

Transition DP2:DP3: When the command is a non-data transfer command, the device shall transition to the DP3: PACKET_non-data state.

Transition DP2:DP4: When the command is a PIO data-in transfer command, the device shall transition to the DP4: PACKET_PIO_in state.

Transition DP2:DP7: When the command is a PIO data-out transfer command, the device shall transition to the DP7: PACKET_PIO_out state.

Transition DP2:DP10: When the command is a DMA data-in transfer command, the device shall transition to the DP10: PACKET_DMA_in state.

Transition DP2:DP12: When the command is a DMA data-out transfer command, the device shall transition to the DP12: PACKET_DMA_out state.

DP3: PACKET_non-data State: This state is entered when a received command is a non-data command.

When in this state, the device shall execute the requested command.

Transition DP3:DP15: When command execution completes, the device shall transition to the DP15: Send_status state.

DP4: PACKET_PIO_in State: This state is entered when the device receives a PIO data-in command or the transmission of one or more DRQ data blocks is required to complete the command.

When in this state, device shall prepare a DRQ data block for transfer to the host.

Transition DP4:DP5: When the device has a DRQ data block ready to transfer, the device shall transition to the DP5: PIO_in_setup.

Transition DP4:15: When all of the data requested by this command has been transferred or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DP15: Send_status state.

* **Transition DP4:DP16:** When the device supports overlap and queuing and does not have a DRQ data block ready to transfer immediately, the device shall transition to the DP16: Release state.

DP5: PIO_in_setup: This state is entered when the device is ready to transfer a DRQ block to the host.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS. The initial status shall have BSY cleared to zero and DRQ set to one. The I bit shall be set to one. The E_Status field shall have BSY set to one and DRQ cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DP5:DP6: When the PIO Setup FIS has been transferred, the device shall transition to the DP6:SendPIO_data state.

DP6:Send_PIO_data: This state is entered when the device is ready to transfer a DRQ data block to the host.

When in this state, the device shall request that the Transport layer transmit a Data FIS containing the DRQ data block.

Transition DP6:DP5: When the Data FIS has been transferred, the device shall transition to the DP5: PACKET_PIO_in state.

DP7: PACKET_PIO_out State: This state is entered when the device receives a PIO data-out command or the receipt of one or more DRQ data blocks is required to complete the command.

When in this state, device shall prepare to receive a DRQ data block transfer from the host.

Transition DP7:DP8: When the device is ready to receive a DRQ data block transfer, the device shall transition to the DP8: PIO_out_setup state.

Transition DP7:DP15: When the device has received all DRQ data blocks requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DP15: Send_status state.

* **Transition DP7:DP16:** When the device supports overlap and queuing and can not accept a DRQ data block immediately, the device shall transition to the DP16: Release state.

DP8: PIO_out_setup: This state is entered when the device is ready to receive a DRQ data block from the host.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS. The initial status shall have BSY cleared to zero and DRQ set to one. The I bit shall be set to one. The E_Status field shall have BSY set to one and DRQ cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DP8:DP9: When the PIO Setup FIS has been transferred, the device shall transition to the DP9: Receive_PIO_data state.

DP9: Receive_PIO_data: This state is entered when the device transmitted a PIO Setup FIS to the host.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DP9:DP7: When the Data FIS has been received, the device shall transition to the DP7: PACKET_PIO_out state.

DP10: PACKET_DMA_in State: This state is entered when the device receives a DMA data-in command or the transmission of one or more Data FIS is required to complete the command.

When in this state, device shall prepare the data for transfer of a Data FIS to the host.

Transition DP10:DP11: When the device has the data ready to transfer a Data FIS, the device shall transition to the DP11: Send_DMA_data state.

Transition DP10:DP15: When the device has transferred all of the data requested by this command or has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DP15: Send_status state.

*** Transition DP10:DP16:** When the device supports overlap and queuing and does not have data ready to transfer immediately, the device shall transition to the DP16: Release state.

DP11: Send_DMA_data: This state is entered when the device has the data ready to transfer a Data FIS to the host.

When in this state, the device shall request that the Transport layer transmit a Data FIS containing the data.

Transition DP11:DP10: When the Data FIS has been transferred, the device shall transition to the DP10: PACKET_DMA_in state. The device command layer shall request a Data FIS size of no more than 8KB.

DP12: PACKET_DMA_out State: This state is entered when the device receives a DMA data-out command or the receipt of one or more Data FIS is required to complete the command.

When in this state, device shall prepare to receive a Data FIS from the host.

Transition DP12:DP13: When the device is ready to receive a Data FIS, the device shall transition to the DP13: Send_DMA_activate state.

Transition DP12:DP15: When the device has received all the data requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DP15: Send_status state.

*** Transition DP12:DP16:** When the device supports overlap and queuing and can not accept a Data FIS immediately, the device shall transition to the DP16: Release state.

DP13: Send_DMA_activate: This state is entered when the device is ready to receive a Data FIS from the host.

When in this state, the device shall request that the Transport layer transmit a DMA Activate FIS.

Transition DP13:DP14: When the DMA Activate FIS has been transferred, the device shall transition to the DP14: Receive_DMA_data state.

DP14: Receive_DMA_data: This state is entered when the device transmitted a DMA Activate FIS to the host.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DP14:DP12: When the Data FIS has been received, the device shall transition to the DP12: PACKET_DMA_out state.

DP15: Send_status: This state is entered when the device has received all the data requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the appropriate command description, and the I bit set to one.

Transition DP15:DI0: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

*** DP16: Release:** This state is entered when the device is not able to do a data transfer immediately.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the appropriate command description with the REL bit set to one, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the I bit set to one.

Transition DP16:DI0: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

17.13 READ DMA QUEUED command protocol

The Read DMA Queued command protocol is illustrated in Figure 104. Read DMA Queued commands are listed in Annex B of ISO/IEC 24739-1:2009.

Execution of this class of command includes the transfer of one or more blocks of data from the device to the host using DMA transfer. All data for the command may be transferred without a bus release between the command receipt and the data transfer. This command may bus release before transferring data. When data transfer is begun, all data for the request shall be transferred without a bus release.

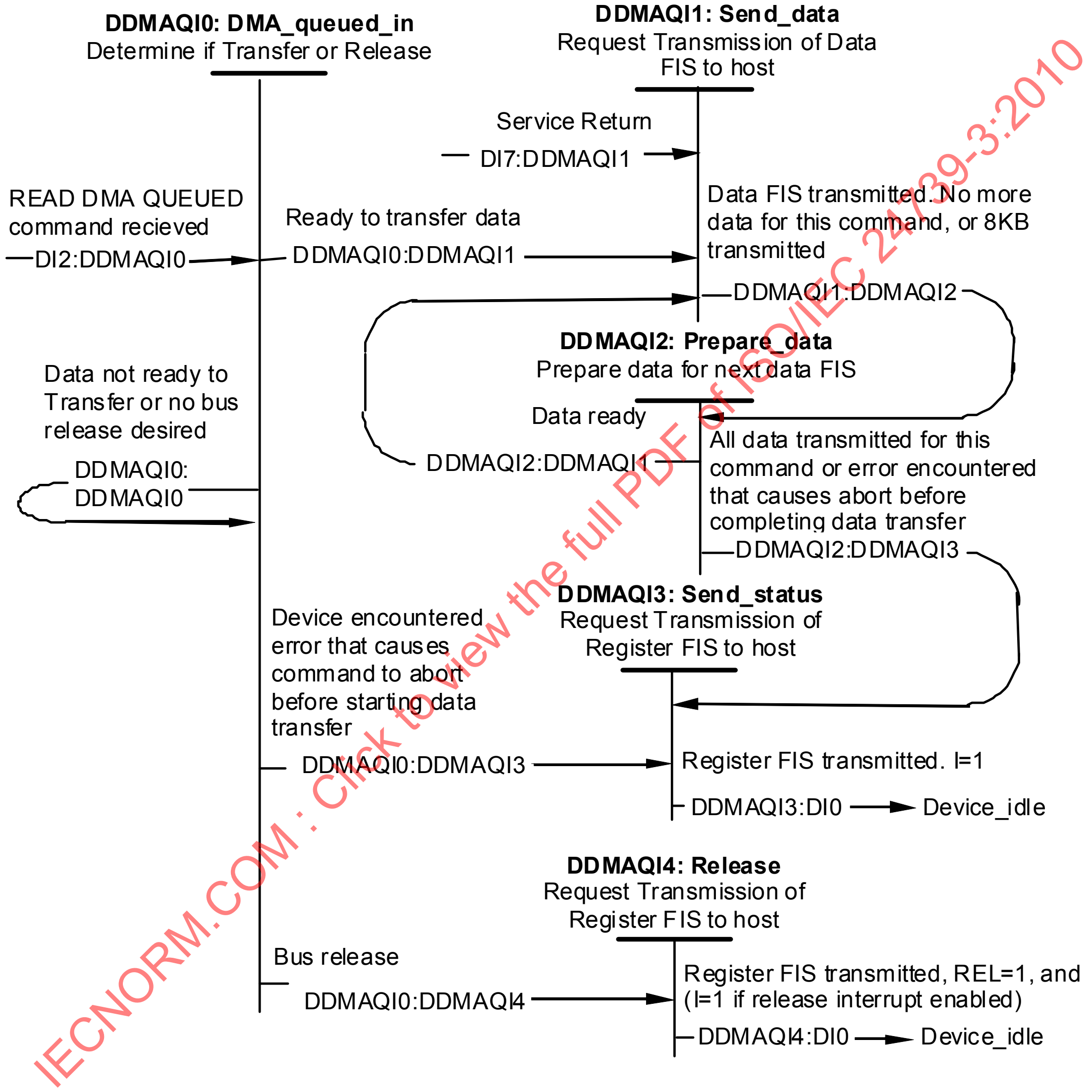


Figure 104 – READ DMA QUEUED command protocol (States DDMAQI0-DDMAQI4)

DDMAQI0: DMA_queued_in: This state is entered when the device receives a READ DMA QUEUED command.

When in this state, device shall determine if the requested data is ready to transfer to the host.

Transition DDMAQI0:DDMAQI1: When the device has the requested data ready to transfer a Data FIS immediately, the device shall transition to the DDMAQI1: Send_data state.

Transition DDMAQI0:DDMAQI3: When the device has encountered an error that causes the command to abort before starting the transfer of the requested data, the device shall transition to the DDMAQI3: Send_status state.

Transition DDMAQI0:DDMAQI4: When the device does not have the requested data ready to transfer a Data FIS, and a release is desired, the device shall transition to the DDMAQI4: Release state.

Transition DDMAQI0:DDMAQI0: When the device does not have the requested data ready to transfer, or no bus release is desired, the device shall transition to the DDMAQI0: DMA_queued_in state.

DDMAQI1: Send_data: This state is entered when the device has the data ready to transfer a Data FIS to the host.

When in this state, the device shall request that the Transport layer transmit a Data FIS containing the data.

Transition DDMAQI1:DDMAQI2: When the Data FIS has been transferred, the device shall transition to the DDMAQI2: Prepare_data state. The device command layer shall request a Data FIS size of no more than 8KB.

DDMAQI2: Prepare_data: This state is entered when the device has completed the transfer a Data FIS to the host.

When in this state, the device shall prepare the data for the next Data FIS.

Transition DDMAQI2:DDMAQI1: When data is ready for the Data FIS, the device shall transition to the DDMAQI1: Send_data state.

Transition DDMAQI2:DDMAQI3: When all data requested for the command has been transmitted or an error has been encountered that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQI3: Send_status state.

DDMAQI3: Send_status: This state is entered when the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the appropriate command description and the **I** bit set to one.

Transition DDMAQI3:DI0: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

DDMAQI4: Release: This state is entered when the device does not have the requested data available for immediate transfer.

When in this state, the device shall request that the Transport layer transmit a Register FIS with the REL bit set to one, with register content as described in the appropriate command description, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the I bit set to one.

Transition DDMAQI4:DI0: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

17.14 WRITE DMA QUEUED command protocol

The Write DMA Queued command protocol is illustrated in Figure 105. Write DMA Queued commands are listed in Annex B of ISO/IEC 24739-1:2009.

Execution of this class of command includes the transfer of one or more blocks of data from the device to the host using DMA transfer. All data for the command may be transferred without a bus release between the command receipt and the data transfer. This command may bus release before transferring data. The host shall initialize the DMA controller prior to transferring data. When data transfer is begun, all data for the request shall be transferred without a bus release.

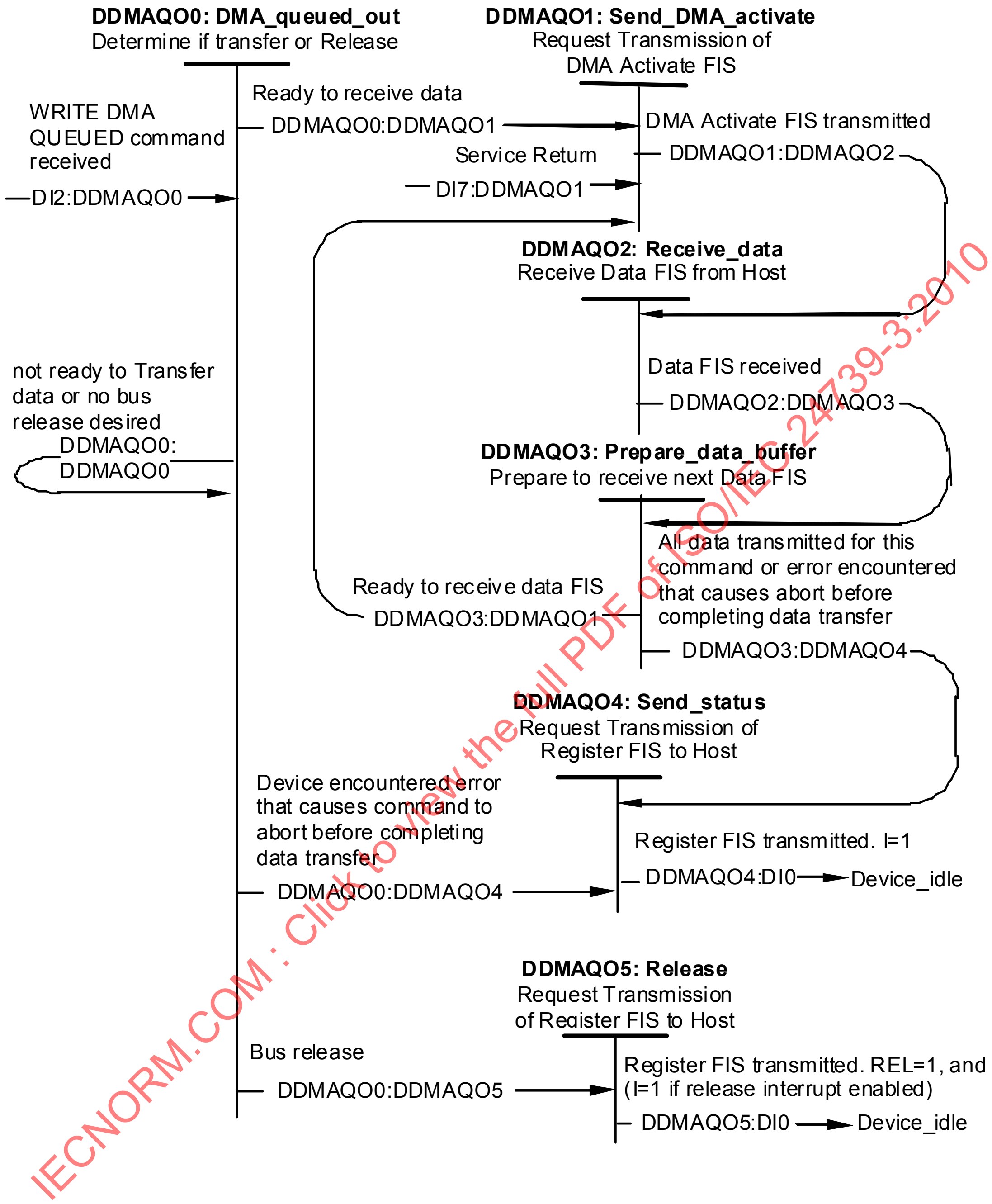


Figure 105 – WRITE DMA QUEUED command protocol (DDMAOQ0-DDMAQO5)

DDMAQO0: DMA_queued_out: This state is entered when the device receives a WRITE DMA QUEUED command.

When in this state, device shall determine if it is ready to accept the requested data from the host.

Transition DDMAQO0:DDMAQO1: When the device is ready to receive a Data FIS immediately, the device shall transition to the DDMAQO1: Send_DMA_activate state.

Transition DDMAQO0:DDMAQO4: When the device has encountered an error that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQO4: Send_status state.