

**Working Draft
American National
Standard**

**T13
Project 1697D**

Revision 0c
16 Jun 2005

**Information Technology -
AT Attachment - 8
ATA Serial Transport
(ATA8-AST)**

This is a draft proposed American National Standard of Accredited Standards Committee INCITS. As such this is not a completed standard. The T13 Technical Committee may modify this document as a result of comments received during public review and its approval as a standard. Use of the information contained here in is at your own risk.

Permission is granted to members of INCITS, its technical committees, and their associated task groups to reproduce this document for the purposes of INCITS standardization activities without further permission, provided this notice is included. All other rights are reserved. Any commercial or for-profit replication or republication is prohibited.

T13 Technical Editor:

John Masiewicz
Western Digital
20511 Lake Forest Drive
Lake Forest, CA 92630
USA

Tel: 949-672-7686
Fax: 949-672-5455

Reference number
ISO/IEC *****:200x
ANSI INCITS.*** - xxxx
Printed June, 17, 2005 5:26PM

Points of Contact:

T13 Chair

Dan Colgrove
Hitachi Global Storage Technologies
2903 Carmelo Dr
Henderson, NV 89402
Tel: 702-614-6199
Fax: 702-614-7955

T13 Vicechair

Jim Hatfield
Seagate Technology
389 Disc Drive
Longmont CO 80503
Tel: 720-684-2120
Fax: 720-684-2711

INCITS Secretariat

NCITS Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005
Email: NCITS@ITIC.ORG

Tel: 202-737-8888
Fax: 202-638-4922

T13 Reflector

See the T13 Web Site at <http://www.t13.org> for reflector information.

T13 Web Site

<http://www.t13.org>

T13 Anonymous FTP Site

<ftp.t13.org>

Document Distribution

INCITS Online Store <http://www.techstreet.com/incits.html>
managed by Techstreet Telephone: 1-734-302-7801
1327 Jones Drive or 1-800-699-9277
Ann Arbor, MI 48105 Facsimile: 1-734-302-7811

or

Global Engineering <http://global.ihs.com/>
15 Inverness Way East Telephone: 1-303-792-2181
Englewood, CO 80112-5704 or 1-800-854-7179
Facsimile: 1-303-792-2192

DOCUMENT STATUS

Revision 0 – September 21, 2004

Document created from ATA/ATAPI-7 Volume 3 (1532D Volume 3, revision 4b)
Added editorial changes requested for ATA/ATAPI-7 at the August 24-26, 2004 plenary meeting.
Formatted as discussed at the August 24-26, 2004 plenary including transition of document to a Serial Transport based format, with Parallel Emulation moved to an Annex, and including changes described in e0140r0 and e0141r1.

Revision 0.a - Updated glossary, corrected references to other volumes to refer to projects, fixed terminology to conform to serial transport format. Revised to change major subclauses into chapters. Revised indenting for better outline format.

Revision 0b - 2/22/05 T13 Plenary action. Accepted all changes with agreed modifications from plenary review. Changed references from FIS registers to be FIS fields. Changed Update Bits FIS back to Set Device Bits FIS. Incorporated ANSI editor's ATA-7 changes as approved by ATA7 plenary. Changed Response FIS types to be 34h (corrected edit error). Revised Annex D to remove parallel task file descriptions and make generic for Serial Transport.

Revision 0c: 6/16/05 Deleted Clause 5.5.8 per plenary vote. Restored Clauses referencing "Freeze" and added new language referencing "timeout" and upper layer handling. Changed Sector Count Fields to be Count Field. Accepted all changes reviewed at plenary.

American National Standard
for Information Technology —

AT Attachment - 8
Serial Transport
(ATA8-AST)

Secretariat
Information Technology Industry Council

Approved mm dd yy

American National Standards Institute, Inc.

Abstract

This standard specifies the serial transport for the AT Attachment Interface between host systems and storage devices. It includes the serial transport Protocols and Physical Interconnect for AT Attachment devices commonly known as Serial ATA.

This standard maintains a high degree of compatibility with the Serial implementation of the AT Attachment Interface with Packet Interface - 7 (ATA/ATAPI-7), INCITS 397-2005, and may provide additional functions requiring new hardware or software for implementation. It is not intended to require changes to presently installed devices or existing software.

**ANSI
INCITS. -200x
American
National
Standard**

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard. As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Published by
**American National Standards Institute
11 West 42nd Street, New York, New York 10036**

Copyright © 2004 by Information Technology Industry Council (ITI)
All rights reserved.

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without prior written permission of ITI, 1250 Eye Street NW, Suite 200, Washington, DC 20005.

Printed in the United States of America

Contents**Page**

| | |
|--|-------------|
| Points of Contact: | ii |
| CONTENTS | I |
| TABLES | VIII |
| FIGURES | VIII |
| FOREWORD | XI |
| INTRODUCTION | XII |
| 1 Scope | 1 |
| 2 Normative references | 2 |
| 2.1 Approved references | 2 |
| 2.1.1 ANSI References | 2 |
| 2.1.2 ISO References | 3 |
| 2.2 References under development..... | 3 |
| 2.3 Other references | 3 |
| 3 Definitions, abbreviations, and conventions | 5 |
| 3.1 Definitions..... | 5 |
| 3.2 Symbols and abbreviations | 10 |
| 3.3 Conventions | 12 |
| 3.3.1 Precedence | 12 |
| 3.3.1.1 Lists..... | 12 |
| 3.3.2 Keywords | 12 |
| 3.3.3 Editorial conventions | 13 |
| 3.3.4 Signal conventions..... | 13 |
| 3.3.5 State diagram conventions..... | 13 |
| 3.3.6 Bit and byte ordering | 14 |
| 3.3.7 Byte, word and dword Relationships | 15 |
| 4 Serial transport general overview | 16 |
| 4.1 Overview | 16 |
| 4.2 Sub-module operation..... | 16 |
| 5 Serial transport physical layer | 18 |
| 5.1 Overview | 18 |
| 5.1.1 List of services | 18 |
| 5.2 Connectors specifications | 18 |
| 5.2.1 Overview | 18 |
| 5.2.1.1 General descriptions | 18 |
| 5.2.2 Connector drawings | 20 |
| 5.2.2.1 Device plug connector | 20 |
| 5.2.2.2 Signal cable receptacle connector | 23 |
| 5.2.2.3 Signal host plug connector | 25 |
| 5.2.2.4 Host receptacle connector..... | 25 |
| 5.2.2.5 Power cable receptacle connector | 28 |
| 5.2.2.6 Optional latching power cable receptacle | 29 |
| 5.2.3 Connector pinouts | 30 |

| | | |
|------------|--|----|
| 5.2.4 | Backplane connector configuration and blind-mating tolerance | 31 |
| 5.2.5 | Connector locations | 32 |
| 5.2.6 | Connector conformance requirements | 35 |
| 5.2.6.1 | Signal | 35 |
| 5.2.6.2 | Housing and contact electrical requirements | 41 |
| 5.2.6.3 | Mechanical and environmental requirements | 42 |
| 5.2.6.4 | Sample selection | 43 |
| 5.2.6.5 | Test sequence | 44 |
| 5.3 | Cable assemblies | 44 |
| 5.4 | Phy (physical layer electronics) | 46 |
| 5.4.1 | Block diagram (Informative) | 46 |
| 5.4.2 | Electrical specifications | 49 |
| 5.4.3 | Frame error-rate measurements | 53 |
| 5.4.4 | Receiver Differential voltage | 53 |
| 5.4.5 | Receiver Common-mode voltage | 53 |
| 5.4.6 | Transmitter Differential voltage | 53 |
| 5.4.7 | Transmitter Common-mode voltage | 53 |
| 5.4.8 | Rise/fall times | 53 |
| 5.5 | Electrical features | 55 |
| 5.5.1 | Definitions | 55 |
| 5.5.2 | Test Methodology and Test Patterns | 55 |
| 5.5.2.1 | Test bit patterns and sequence characteristics | 55 |
| 5.5.2.2 | Low transition density bit pattern sequences | 56 |
| 5.5.2.3 | High transition density bit pattern sequences | 58 |
| 5.5.2.4 | Low frequency spectral content bit pattern sequences | 58 |
| 5.5.2.5 | Simultaneous switching outputs bit pattern sequences | 59 |
| 5.5.2.6 | Composite bit pattern sequences | 59 |
| 5.5.3 | Bit error rate testing - Informative | 60 |
| 5.5.3.1 | Error-burst-rate-thresholding measurement - Informative | 60 |
| 5.5.3.2 | Bit-error-rate measurements - Informative | 60 |
| 5.5.4 | Frame error rate testing | 61 |
| 5.5.4.1 | Frame error-rate patterns | 61 |
| 5.5.4.1.1 | Loopback test | 61 |
| 5.5.5 | Test requirements - non-compliant patterns | 61 |
| 5.5.6 | Test requirements - compliant frame patterns | 61 |
| 5.5.7 | Test requirements - loopback | 62 |
| 5.5.7.1 | Test requirements - loopback - far-end retimed | 62 |
| 5.5.7.2 | Test requirements - loopback - far-end analog (vendor specific) | 62 |
| 5.5.7.3 | Test requirements - loopback - near-end analog (vendor specific) | 62 |
| 5.5.8 | Differential voltage/timing (EYE) diagram | 63 |
| 5.5.8.1 | Jitter output/tolerance mask | 63 |
| 5.5.8.2 | Sampling differential noise budget | 65 |
| 5.5.8.3 | Jitter output | 65 |
| 5.5.8.3.1 | Jitter measurements | 65 |
| 5.5.9 | Spread spectrum clocking (SSC) | 66 |
| 5.5.10 | Common-mode biasing | 68 |
| 5.5.11 | Matching | 68 |
| 5.5.12 | Out of band signaling | 69 |
| 5.5.12.1 | Idle bus status | 70 |
| 5.5.12.2 | Phy Layer OOB Sequences (Informative) | 71 |
| 5.5.12.3 | Host Phy initialization state machine | 72 |
| 5.5.12.3.1 | Device power-up and Phy Layer state machine | 78 |
| 5.5.12.3.2 | Power-up and COMRESET | 84 |
| 5.5.12.3.3 | COMINIT Description | 86 |
| 5.5.12.3.4 | COMWAKE Description | 88 |
| 5.5.12.3.5 | Interface power modes | 88 |
| 5.5.12.3.6 | Power-on sequence timing diagram | 90 |
| 5.5.12.3.7 | Active to Partial or Slumber mode | 91 |

| | | |
|------------|---|-----------|
| 5.5.12.3.8 | Partial mode or Slumber mode to Active | 91 |
| 5.5.12.4 | On to Partial mode or Slumber mode | 91 |
| 5.5.12.4.1 | Host initiated | 91 |
| 5.5.12.4.2 | Detailed sequence | 92 |
| 5.5.12.4.3 | Device initiated | 93 |
| 5.5.12.4.4 | Detailed sequence | 93 |
| 5.6 | Elasticity buffer management | 94 |
| 5.7 | BIST (Built in self test) | 95 |
| 5.7.1 | Loopback testing | 95 |
| 5.7.1.1 | Loopback -- Far end retimed | 95 |
| 5.7.1.2 | Loopback -- far-end analog (Optional) | 96 |
| 5.7.1.2.1 | Loopback -- near-end analog (Optional) | 97 |
| 6 | Serial transport link layer | 98 |
| 6.1 | Overview | 98 |
| 6.1.1 | Frame transmission | 98 |
| 6.1.2 | Frame receipt | 98 |
| 6.2 | 8b10 coding method | 98 |
| 6.2.1 | Notation and conventions | 99 |
| 6.2.2 | Character code | 100 |
| 6.2.2.1 | Code construction | 100 |
| 6.2.2.2 | The concept of running disparity | 100 |
| 6.2.2.3 | Data encoding | 102 |
| 6.2.2.4 | Encoding examples | 102 |
| 6.2.2.5 | 8b/10b valid encoded characters | 104 |
| 6.2.2.5.1 | Data characters | 104 |
| 6.2.2.5.2 | Control characters | 108 |
| 6.2.3 | Transmission summary | 109 |
| 6.2.3.1 | Transmission order | 109 |
| 6.2.3.1.1 | Bits within a byte | 109 |
| 6.2.3.1.2 | Bytes within a dword | 109 |
| 6.2.3.1.3 | dwords within a frame | 109 |
| 6.2.4 | Reception | 110 |
| 6.2.4.1 | Disparity and the detection of a code violation | 110 |
| 6.3 | Transmission Method | 112 |
| 6.4 | Primitives | 113 |
| 6.4.1 | Overview | 113 |
| 6.4.1.1 | Primitive disparity | 113 |
| 6.4.1.2 | Primitive handshakes | 113 |
| 6.4.2 | Primitive descriptions | 114 |
| 6.4.3 | Primitive encoding | 116 |
| 6.4.4 | ALIGN primitive | 117 |
| 6.4.5 | CONT primitive | 117 |
| 6.4.5.1 | Scrambling of data following the CONT primitive | 118 |
| 6.4.6 | DMAT primitive | 118 |
| 6.4.7 | EOF primitive | 119 |
| 6.4.8 | HOLD/HOLDA primitives | 119 |
| 6.4.8.1 | Flow Control Signaling Latency | 120 |
| 6.4.9 | PMREQ_P, PMREQ_S, PMACK, and PMNAK primitives | 121 |
| 6.4.10 | R_ERR primitive | 121 |
| 6.4.11 | R_IP primitive | 121 |
| 6.4.12 | R_OK primitive | 122 |
| 6.4.13 | R_RDY primitive | 122 |
| 6.4.14 | SOF primitive | 122 |
| 6.4.15 | SYNC primitive | 122 |
| 6.4.16 | WTRM primitive | 122 |
| 6.4.17 | X_RDY primitive | 122 |
| 6.4.18 | Examples | 122 |

| | | |
|----------|--|------------|
| 6.5 | CRC calculation..... | 127 |
| 6.6 | Scrambling..... | 128 |
| 6.6.1 | Frame content scrambling..... | 128 |
| 6.6.1.1 | Relationship between scrambling and CRC..... | 128 |
| 6.6.2 | Repeated primitive suppression..... | 128 |
| 6.6.2.1 | Relationship between scrambling of FIS data and repeated primitives..... | 128 |
| 6.7 | link layer state diagrams..... | 128 |
| 6.7.1.1 | Link idle state diagram..... | 129 |
| 6.7.1.2 | Link transmit state diagram..... | 132 |
| 6.7.1.3 | Link receive state diagram..... | 138 |
| 6.7.1.4 | Link power mode state diagram..... | 143 |
| 7 | Transport layer..... | 147 |
| 7.1 | transport layer overview..... | 147 |
| 7.1.1 | FIS construction..... | 147 |
| 7.1.2 | FIS decomposition..... | 147 |
| 7.2 | Frame Information Structure (FIS)..... | 147 |
| 7.3 | Overview..... | 147 |
| 7.4 | Payload content..... | 147 |
| 7.5 | FIS types..... | 147 |
| 7.5.1 | Command FIS..... | 147 |
| 7.5.1.1 | Command FIS - 48-bit command mapping..... | 148 |
| 7.5.1.2 | Command FIS - 28-bit command and non-data command mapping..... | 150 |
| 7.5.2 | Control FIS..... | 152 |
| 7.5.2.1 | Control FIS – control mapping..... | 152 |
| 7.5.3 | Response FIS..... | 154 |
| 7.5.3.1 | Response FIS - 48-bit command response mapping..... | 154 |
| 7.5.3.2 | Response FIS - 28-bit command and non-data command response mapping..... | 156 |
| 7.5.4 | Set Device Bits FIS..... | 158 |
| 7.5.4.1 | Set Device Bits FIS..... | 158 |
| 7.5.5 | DMA Activate - device to host..... | 159 |
| 7.5.5.1 | Description..... | 159 |
| 7.5.5.2 | Transmission..... | 159 |
| 7.5.5.3 | Reception..... | 159 |
| 7.5.6 | First Party DMA Setup (bidirectional)..... | 160 |
| 7.5.6.1 | Description..... | 160 |
| 7.5.6.2 | Transmission..... | 161 |
| 7.5.6.3 | Reception..... | 161 |
| 7.5.7 | BIST Activate - Bidirectional..... | 162 |
| 7.5.7.1 | Description..... | 162 |
| 7.5.7.2 | Transmission..... | 163 |
| 7.5.7.3 | Reception..... | 163 |
| 7.5.8 | PIO Setup FIS..... | 164 |
| 7.5.8.1 | PIO Setup FIS - 48-bit command PIO setup mapping..... | 164 |
| 7.5.8.2 | PIO Setup FIS - 28-bit command PIO setup mapping..... | 166 |
| 7.5.9 | Data (bidirectional)..... | 168 |
| 7.5.9.1 | Description..... | 168 |
| 7.5.9.2 | Transmission..... | 168 |
| 7.5.9.3 | Reception..... | 169 |
| 7.6 | Host transport states..... | 170 |
| 7.6.1 | Host transport idle state diagram..... | 170 |
| 7.6.2 | Host Transport transmit Command FIS diagram..... | 173 |
| 7.6.3 | Host Transport transmit control FIS diagram..... | 175 |
| 7.6.4 | Host Transport transmit First Party DMA Setup - device to host or host to device FIS state diagram ... | 176 |
| 7.6.5 | Host Transport transmit BIST Activate FIS..... | 178 |
| 7.6.6 | Host Transport decompose Response FIS diagram..... | 179 |
| 7.6.7 | Host Transport decompose a Set Device Bits FIS state diagram..... | 180 |
| 7.6.8 | Host Transport decompose a DMA Activate FIS diagram and DMA Data Transfer..... | 181 |

| | | |
|-----------|--|------------|
| 7.6.9 | Host Transport decompose a PIO Setup FIS state diagram..... | 184 |
| 7.6.10 | Host Transport decompose a First Party DMA Setup FIS state diagram..... | 187 |
| 7.6.11 | Host transport decompose a BIST Activate FIS state diagram | 188 |
| 7.7 | Device transport states | 190 |
| 7.7.1 | Device transport idle state diagram | 190 |
| 7.7.2 | Device Transport send Response FIS to host state diagram..... | 192 |
| 7.7.3 | Device Transport send Set Device Bits FIS state diagram | 193 |
| 7.7.4 | Device Transport transmit PIO Setup - device to host FIS state diagram..... | 194 |
| 7.7.5 | Device Transport transmit DMA Activate FIS state diagram | 195 |
| 7.7.6 | Device Transport transmit First Party DMA Setup - device to host FIS state diagram..... | 196 |
| 7.7.7 | Device Transport transmit Data - device to host FIS diagram | 197 |
| 7.7.8 | Device Transport transmit BIST Activate FIS diagram | 199 |
| 7.7.9 | Device Transport decompose Command or Control FIS state diagram..... | 200 |
| 7.7.10 | Device Transport decompose Data (host to device) FIS state diagram..... | 201 |
| 7.7.11 | Device Transport decompose First Party DMA Setup FIS - host to device or device to host state diagram 203 | |
| 7.7.12 | Device Transport decompose a BIST Activate FIS state diagram | 204 |
| 8 | Serial Transport Device Command Layer Protocol | 206 |
| 8.1 | COMRESET or SRST sent by Host..... | 206 |
| 8.2 | Power-on and COMRESET protocol diagram..... | 206 |
| 8.3 | Device Idle protocol..... | 208 |
| 8.4 | Software reset protocol | 212 |
| 8.5 | EXECUTE DEVICE DIAGNOSTIC command protocol | 214 |
| 8.6 | DEVICE RESET command protocol..... | 215 |
| 8.7 | Non-data command protocol | 216 |
| 8.8 | PIO data-in command protocol..... | 217 |
| 8.9 | PIO data-out command protocol | 219 |
| 8.10 | DMA data-in command protocol..... | 221 |
| 8.11 | DMA data out command protocol..... | 223 |
| 8.12 | PACKET protocol..... | 225 |
| 8.13 | READ DMA QUEUED command protocol | 230 |
| 8.14 | WRITE DMA QUEUED command protocol | 232 |
| 9 | Serial transport host adapter register interface | 235 |
| 9.1 | Overview | 235 |
| 9.2 | SStatus, SError and SControl registers..... | 235 |
| 9.2.1 | SStatus register | 236 |
| 9.2.2 | SErrror register..... | 237 |
| 9.2.3 | SControl register..... | 239 |
| 10 | Serial Transport error handling | 240 |
| 10.1 | Architecture..... | 240 |
| 10.2 | Phy error handling overview | 242 |
| 10.2.1 | Error detection | 242 |
| 10.2.2 | Error control actions | 242 |
| 10.2.2.1 | No device present..... | 242 |
| 10.2.2.2 | OOB signaling sequence failure..... | 243 |
| 10.2.2.3 | Phy internal error | 243 |
| 10.2.3 | Error reporting | 243 |
| 10.3 | Link error handling overview | 244 |
| 10.3.1 | Error detection | 244 |
| 10.3.2 | Error control actions | 244 |
| 10.3.2.1 | Invalid state transitions | 244 |
| 10.3.2.2 | Data integrity errors:..... | 244 |
| 10.3.3 | Error reporting | 245 |
| 10.4 | Transport error handling..... | 246 |
| 10.4.1 | Overview..... | 246 |

| | | |
|--|---|------------|
| 10.4.2 | Error detection | 246 |
| 10.4.3 | Error control actions | 246 |
| 10.4.3.1 | Internal errors..... | 246 |
| 10.4.3.2 | Frame errors | 247 |
| 10.4.3.3 | Protocol and state transition errors..... | 247 |
| 10.4.4 | Error reporting | 248 |
| 10.5 | Software error handling overview | 249 |
| 10.5.1 | Error detection | 249 |
| 10.5.2 | Error control actions | 249 |
| ANNEX A. SAMPLE CODE FOR SERIAL CRC SCRAMBLING (INFORMATIVE)..... | | 251 |
| A.1 | CRC calculation | 251 |
| A.1.1 | Overview..... | 251 |
| A.1.2 | aximum frame size..... | 251 |
| A.1.3 | Example code for CRC algorithm | 251 |
| A.1.4 | Example CRC implementation output | 254 |
| A.2 | Scrambling calculation..... | 255 |
| A.2.1 | Overview..... | 255 |
| A.2.2 | Example code for scrambling algorithm..... | 255 |
| A.2.3 | Example scrambler implementation..... | 258 |
| A.3 | Example frame | 259 |
| ANNEX B. FIS TYPE FIELD VALUE SELECTION (INFORMATIVE)..... | | 260 |
| B.1 | Overview | 260 |
| B.2 | Type field values..... | 260 |
| ANNEX C. PHYSICAL LAYER IMPLEMENTATION EXAMPLES (INFORMATIVE) | | 261 |
| C.1 | Cable construction example..... | 261 |
| C.2 | Contact material and plating..... | 262 |
| C.3 | Relationship of frequency to the jitter specification..... | 262 |
| C.4 | Sampling BER and jitter formulas | 264 |
| C.5 | DC and AC coupled transmitter examples | 265 |
| C.6 | OOB signaling and squelch detector examples | 266 |
| ANNEX D. COMMAND PROCESSING EXAMPLE (INFORMATIVE)..... | | 269 |
| D.1 | Non-data commands | 269 |
| D.1.1 | DMA read by host from device | 269 |
| D.1.2 | DMA write by host to device | 270 |
| D.1.3 | PIO data read from the device | 271 |
| D.1.4 | PIO data write to the device | 272 |
| D.1.5 | READ DMA QUEUED example..... | 273 |
| D.1.6 | WRITE DMA QUEUED example..... | 274 |
| D.1.7 | ATAPI PACKET commands with PIO data-in | 275 |
| D.1.8 | ATAPI PACKET commands with PIO data out..... | 276 |

| | | |
|--|--|------------|
| D.1.9 | ATAPI PACKET commands with DMA data-in | 277 |
| D.1.10 | ATAPI PACKET commands with DMA data-out | 278 |
| D.1.11 | First party DMA read of host memory by device | 279 |
| D.1.12 | First party DMA write of host memory by device..... | 279 |
| D.1.13 | Odd word count considerations | 279 |
| D.1.13.1 | DMA read from device for odd word count..... | 279 |
| D.1.13.2 | DMA write by host to device for odd word count | 280 |
| D.1.13.3 | PIO data read from the device | 280 |
| D.1.13.4 | PIO data write to the device | 281 |
| D.1.13.5 | First party DMA read of host memory by device | 281 |
| D.1.13.6 | First party DMA write of host memory by device..... | 281 |
| ANNEX E. PARALLEL ATA EMULATION (INFORMATIVE) | | 283 |
| E.1 | Parallel ATA Emulation (Optional) | 283 |
| E.1.1 | Software reset..... | 283 |
| E.1.2 | Device 0-only emulation..... | 284 |
| E.1.3 | Device 0/Device 1 emulation (optional) | 284 |
| E.1.3.1 | Software reset | 285 |
| E.1.3.2 | EXECUTE DEVICE DIAGNOSTICS | 285 |
| E.1.3.3 | Restrictions and limitations | 285 |
| E.2 | Host Command Layer state diagram for Parallel ATA Emulation..... | 285 |
| E.3 | Overview | 285 |
| E.4 | Device Emulation of nIEN with Interrupt Pending (Informative) | 289 |
| E.5 | Frame Information Structure (FIS) Mapping for Parallel ATA Emulation | 291 |
| E.5.1 | Description | 291 |
| E.5.2 | Register - host to device | 291 |
| E.5.2.1 | Description | 292 |
| E.5.2.2 | Transmission | 292 |
| E.5.2.3 | Reception | 292 |
| E.5.3 | Register - device to host | 293 |
| E.5.3.1 | Description - Register device to host FIS | 293 |
| E.5.3.2 | Transmission | 294 |
| E.5.3.3 | Reception | 294 |
| E.5.4 | Set Device Bits - device to host..... | 295 |
| E.5.4.1 | Description Set Device Bits device to host FIS | 295 |
| E.5.4.2 | Transmission | 295 |
| E.5.4.3 | Reception | 296 |
| E.5.5 | PIO Setup - device to host | 297 |
| E.5.5.1 | Description..... | 297 |
| E.5.5.2 | Transmission of PIO Setup by Device Prior to a Data Transfer from host to device | 298 |
| E.5.5.3 | Reception of PIO Setup by Host Prior to a Data Transfer from host to device | 298 |
| E.5.5.4 | Transmission of PIO Setup by Device Prior to a Data Transfer from device to host | 298 |
| E.5.5.5 | Reception of PIO Setup by Host Prior to a Data Transfer from device to host | 298 |
| E.5.6 | Data - host to device or device to host (Bidirectional) | 299 |
| E.5.6.1 | Description | 299 |
| E.5.6.2 | Transmission | 299 |
| E.5.6.3 | Reception | 300 |

Tables

| | Page |
|---|-------------|
| Table 1 - Standards bodies | 2 |
| Table 2 - ISO and American numbering conventions | 13 |
| Table 3 - Data dword containing a value..... | 15 |
| Table 4 - Data dword containing four one-byte fields | 15 |
| Table 5 - Device plug connector pin definition | 30 |
| Table 6 - Signal integrity requirements and test procedures | 36 |
| Table 7 - Housing and contact electrical parameters, test procedures, and requirements | 41 |
| Table 8 - Mechanical test procedures and requirements..... | 42 |
| Table 9 - Environmental parameters, test procedures, and requirements..... | 43 |
| Table 10 - Additional requirement | 43 |
| Table 11 - Connector test sequences | 44 |
| Table 12 - Physical Layer Electrical Requirements..... | 49 |
| Table 13 - Voltage / Timing Margin Definition..... | 63 |
| Table 14 - Sampling differential noise budget..... | 65 |
| Table 15 - Desired peak amplitude reduction by SSC | 68 |
| Table 16 - Out of band signal times | 70 |
| Table 17 - Interface power modes | 88 |
| Table 18 - 5b/6b coding..... | 102 |
| Table 19 - 3b/4b coding..... | 102 |
| Table 20 - Valid data characters | 104 |
| Table 21 - Valid control characters | 108 |
| Table 22 - Description of primitives | 114 |
| Table 23 - Primitive encoding..... | 116 |
| Table 24 - Valid CONT Transmission Sequences | 117 |
| Table 25 - Latency example | 120 |
| Table 26 - SRST write from host to device transmission breaking through a device to host Data FIS | 121 |
| Table 27 - device to host FIS transmission example | 124 |
| Table 28 - Data from host to device transmission example | 125 |
| Table 29 - DMA data from host to device, device terminates transmission example | 126 |
| Table 30 - SCR definition | 235 |
| Table 31 - SCR Definition..... | 235 |
| Table 32 - CRC and scrambler calculation example - PIO Write Command | 259 |
| Table 33 - Type field values | 260 |

Figures

| | Page |
|---|-------------|
| Figure 1 - ATA document relationships | 1 |
| Figure 2 - State diagram convention | 14 |
| Figure 3 - Byte, word and data dword relationships..... | 15 |
| Figure 4 - The serial transport connectivity..... | 16 |
| Figure 6 - Serial implementation connector examples..... | 19 |
| Figure 7 - Device plug connector part 1 of 2..... | 21 |
| Figure 8 - Device Plug Connector part 2 of 2..... | 22 |
| Figure 9 - Non-Latching Signal Cable receptacle connector interface dimensions | 23 |
| Figure 10 - Optional Latching Signal Cable Receptacle connector interface dimensions | 24 |
| Figure 11 - Host plug connector interface dimension | 25 |
| Figure 12 - Host receptacle connector interface dimensions..... | 26 |
| Figure 13 - Non-latching power cable receptacle connector interface dimensions | 28 |
| Figure 14 - Optional latching power cable receptacle..... | 29 |
| Figure 15 - Connector pair blind-mate misalignment tolerance | 31 |
| Figure 16 - Device-backplane mating configuration..... | 31 |
| Figure 17 - Device plug connector location on 3.5" device..... | 32 |
| Figure 18 - Device plug connector location on 2.5" device..... | 33 |
| Figure 19 - Recommended host plug spacing for Non-Latching Connectors | 34 |
| Figure 20 - Recommended host plug connector clearance & Orientation for Optional Latching Connectors..... | 35 |

| | |
|--|-----|
| Figure 21 - Signals and grounds assigned in direct connect and cabled | 45 |
| Figure 22 - Physical plant simplified block diagram (Informative)..... | 46 |
| Figure 23 - Signal rise and fall times..... | 53 |
| Figure 24 - Transmit test fixture | 54 |
| Figure 25 - Receive test fixture | 54 |
| Figure 26 - Low transition density pattern..... | 57 |
| Figure 27 - Half-rate / quarter-rate high transition density pattern..... | 58 |
| Figure 28 - Low frequency spectral content pattern | 58 |
| Figure 29 - Simultaneous switching outputs patterns | 59 |
| Figure 30- Composite patterns | 59 |
| Figure 31- Compliant test patterns..... | 62 |
| Figure 32 - Voltage / timing margin base diagram..... | 63 |
| Figure 33 - Jitter output/tolerance mask | 64 |
| Figure 34 - Jitter measurement example | 66 |
| Figure 35 - Triangular frequency modulation profile | 67 |
| Figure 36 - Spectral fundamental frequency comparison | 67 |
| Figure 37 - Out of band signals..... | 69 |
| Figure 39 – Host Phy initialization state machine (States HP1-HP13) | 73 |
| Figure 40 – Device initialization and Phy layer state machine (States DP1-DP12)..... | 79 |
| Figure 41 - COMRESET Description | 84 |
| Figure 42 – Device Initiated ComlNit Sequence | 86 |
| Figure 43 - Power-on Sequence | 90 |
| Figure 44 - On to Partial or Slumber mode - host initiated | 91 |
| Figure 45 - ON to Partial or Slumber - device initiated | 93 |
| Figure 46 - Loopback far-end retimed..... | 95 |
| Figure 47 - Loopback far-end analog..... | 96 |
| Figure 48 - Loopback - near-end analog..... | 97 |
| Figure 49 - Bit designations | 99 |
| Figure 50 - Nomenclature reference | 99 |
| Figure 51 - Conversion examples | 100 |
| Figure 52 - Coding examples..... | 103 |
| Figure 53 - Bit ordering and significance | 109 |
| Figure 54 - Single bit error with two character delay | 111 |
| Figure 55 - Single bit error with one character delay | 111 |
| Figure 56 - Transmission structures | 112 |
| Figure 57 - CONT usage example | 123 |
| Figure 58 – Link idle state diagram (States L1, LS1-LS3) | 129 |
| Figure 59 – Link transmit state diagram (States LT1-LT9) | 132 |
| Figure 60 – Link receive state diagram (States LR1-LR9)..... | 138 |
| Figure 61 – Link power mode state diagram (States LPM1-LPM8)..... | 143 |
| Figure 62 - Command FIS - 48-bit command transport FIS layout..... | 148 |
| Figure 63 - Command FIS - 28-bit command transport and con-data command transport FIS layout | 150 |
| Figure 64 - Control FIS – control transport FIS layout | 152 |
| Figure 65 - Response FIS - 48-bit command response transport FIS layout | 154 |
| Figure 66 - Response FIS - 28-bit command response and non-data command response transport FIS layout | 156 |
| Figure 67 – Set Device Bits - device to host transport FIS layout | 158 |
| Figure 68 - DMA Activate - device to host FIS layout..... | 159 |
| Figure 69 - First Party DMA Setup - device to host FIS layout | 160 |
| Figure 70 - BIST Activate - Bidirectional | 162 |
| Figure 71 - PIO Setup – 48-bit command device to host transport FIS layout | 164 |
| Figure 72 - PIO Setup – 28-bit command device to host transport FIS layout | 166 |
| Figure 73 - Data FIS layout..... | 168 |
| Figure 74 – Host transport idle state diagram (States HTI1-HTI2) | 170 |
| Figure 75 – Host transport transmit Command FIS diagram (States HTCM1-HTCM2) | 173 |
| Figure 76 – Host transport transmit control FIS diagram (States HTCR1-HTCR2)..... | 175 |
| Figure 77 – Host transport transmit First Party DMA setup - device to host or host to device FIS (States HTDMASTER0-HTDMASTER1)..... | 176 |

| | |
|--|-----|
| Figure 78 – Host transport transmit BIST activate FIS (States HTXBIST0-HTXBIST1) | 178 |
| Figure 79 - Host transport decompose Response FIS diagram (States HTR1- HTR2)..... | 179 |
| Figure 80 – Host transport decompose Set Device Bits FIS state diagram (States HTDB0-HTDB1) | 180 |
| Figure 81 – Host transport decompose DMA activate FIS diagram (States HTDA1-HTDA5)..... | 181 |
| Figure 82 – Host transport decompose PIO setup FIS state diagram (States HTPS1-HTPS6) | 184 |
| Figure 83 – Host transport decompose First Party DMA Setup FIS state diagram (State HTDS1) | 187 |
| Figure 84 – Host transport decompose BIST activate FIS state diagram (State HTRBIST0-HTRBIST1)..... | 188 |
| Figure 85 – Device transport idle state diagram (States DTIO-DTI1)..... | 190 |
| Figure 86 – Device transport send Response state diagram (DTR0-DTR1) | 192 |
| Figure 87 – Device transport send device bits FIS state diagram (DTDB0-DTDB1) | 193 |
| Figure 88 – Device transport transmit PIO setup - device to host FIS state diagram (States DTPIOSTUP0- DTPIOSTUP1) | 194 |
| Figure 89 – Device transport transmit DMA activate FIS state diagram (States DTDMACT0-DTDMACT1) | 195 |
| Figure 90 – Device transport transmit First Party DMA Setup - device to host state diagram (States DTDMASTERUP0-DTDMASTERUP1)..... | 196 |
| Figure 91 – Device transport transmit data - device to host FIS diagram (State DTDATAI0-DTDATAI2)..... | 197 |
| Figure 92 – Device transport transmit BIST activate FIS diagram (States DTXBIST0-DTXBIST1) | 199 |
| Figure 93 – Device transport decompose Command or Control FIS state diagram (State DTCMD0) | 200 |
| Figure 94 – Device transport decompose data (host to device) FIS state diagram (States DTDATAO0- DTDATAO2) | 201 |
| Figure 95 – Device transport decompose First Party DMA Setup FIS - host to device or device to host state diagram (State DTDMASTERUP0)..... | 203 |
| Figure 96 - Device transport decompose BIST activate FIS (States (DTRBIST0-DTRBIST1)..... | 204 |
| Figure 97 - Power on and COMRESET protocol (States DHR0-DHR3)..... | 206 |
| Figure 98 - Device idle protocol (States DI0-DI7) | 208 |
| Figure 99 - Software reset protocol (States DSR0-DSR3)..... | 212 |
| Figure 100 - EXECUTE DEVICE DIAGNOSTIC command protocol (States DEDD0-DEDD2)..... | 214 |
| Figure 101 - DEVICE RESET command protocol (States DDR0-DDR1) | 215 |
| Figure 102 - Non-data command protocol (States DND0-DND1)..... | 216 |
| Figure 103 - PIO data-in command protocol (States DPIOI0-DPIOI3) | 217 |
| Figure 104 - PIO data-out command protocol (States DPIOO0-DPIOO3) | 219 |
| Figure 105 - DMA data-in command protocol (States DDMAI0-DDMAI1)..... | 221 |
| Figure 106 - DMA data-out command protocol (States DDMAO0-DDMAO3) | 223 |
| Figure 107 - PACKET command protocol (States DP0-DP16)..... | 225 |
| Figure 108 - READ DMA QUEUED command protocol (States DDMAQI0-DDMAQI4)..... | 230 |
| Figure 109 - WRITE DMA QUEUED command protocol (DDMAOQ0-DDMAOQ5)..... | 232 |
| Figure 110 - Error handling architecture | 240 |
| Figure 111 - Cable construction example | 261 |
| Figure 112 - Jitter as a function of frequency..... | 263 |
| Figure 113 - Sampling bit error rate formulas | 264 |
| Figure 114 - Transmitter examples | 265 |
| Figure 115 - OOB signaling detector..... | 267 |
| Figure 116 - Squelch detector | 268 |
| Figure 117 – Host adapter state diagram (States HA0-HA2)..... | 286 |
| Figure 118 - Register - host to device FIS layout | 291 |
| Figure 119 - Register - device to host FIS layout..... | 293 |
| Figure 120 - Set Device Bit - device to host FIS layout | 295 |
| Figure 121 - PIO Setup - device to host FIS layout | 297 |
| Figure 122 - Data - host to device or device to host FIS layout | 299 |

Foreword

(This foreword is not part of this standard.)

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, ITI, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by InterNational Committee for Information Technology Standards (INCITS). Committee approval of this standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, INCITS had the following members:

Karen Higginbottom, Chair
David Michael, Vice-chair
Monica Vago, Secretary

Technical Committee T13 on ATA Interfaces, that reviewed this standard, had the following members and additional participants:

Dan Colgrove, Chairman

Jim Hatfield, Vice-Chairman

Mark Overby, Secretary

John Masiewicz, Technical Editor

Introduction

This standard encompasses the following:

(editors note: this introduction is of little value, and we should write a better introduction.)

Formatted: Font: Bold, Font color: Red

Subclause 1 describes the scope.

Subclause 2 provides normative references for the entire standard.

Subclause 3 provides definitions, abbreviations, and conventions used within the entire standard.

Subclause 4 describes the general operating requirements of the command layer.

Subclause 5 describes the I/O registers.

Subclause 6 contains descriptions of the commands.

American National Standard
for Information Systems —

**Information Technology —
AT Attachment 8
ATA Serial Transport
(ATA8-AST)**

1 Scope

This standard specifies the serial transport for 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.

ATA8-ACS defines the commands used by devices implementing the standard. This document 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 commands, data, and status over the interface for the serial transport. Figure 1 shows the relationship of these documents.

(Editor's Note: Need new figure)

Formatted: Font: Bold, Font color: Red

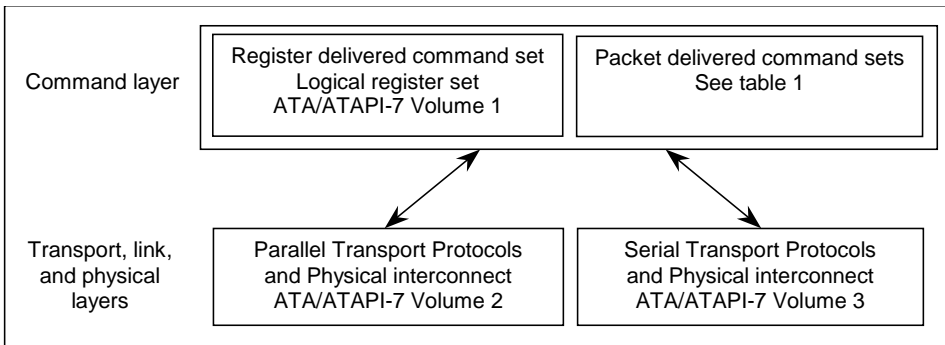


Figure 1 – ATA document relationships

This standard maintains a high degree of compatibility with the AT Attachment with Packet Interface - 7 standard (ATA/ATAPI-7), INCITS 397-2005, and while providing additional functions, is not intended to require changes to presently installed devices or existing software.

2 Normative references

The following standards contain provisions that, by reference in the text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

Copies of the following documents may be obtained from ANSI:

- a) approved ANSI standards;
- b) approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITU-T); and
- c) approved and draft foreign standards (including BSI, JIS, and DIN).

For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at <http://www.ansi.org>.

Additional availability contact information is provided below as needed.

Table 1 shows standards bodies and their web sites.

Table 1 - Standards bodies

| Abbreviation | Standards body | Web site |
|--------------|---|---|
| ANSI | American National Standards Institute | http://www.ansi.org |
| BSI | British Standards Institution | http://www.bsi-global.com |
| CEN | European Committee for Standardization | http://www.cenorm.be |
| CENELEC | European Committee for Electrotechnical Standardization | http://www.cenelec.org |
| DIN | German Institute for Standardization | http://www.din.de |
| IEC | International Engineering Consortium | http://www.iec.ch |
| IEEE | Institute of Electrical and Electronics Engineers | http://www.ieee.org |
| INCITS | International Committee for Information Technology Standards | http://www.incits.org |
| ISO | International Standards Organization | http://www.iso.ch |
| ITI | Information Technology Industry Council | http://www.itic.org |
| ITU-T | International Telecommunications Union Telecommunications Standardization Sector | http://www.itu.int |
| JIS | Japanese Industrial Standards Committee | http://www.jisc.org |
| T10 | INCITS T10 SCSI storage interfaces | http://www.t10.org |
| T11 | INCITS T11 Fibre Channel interfaces | http://www.t11.org |
| T13 | INCITS T13 ATA storage interface | http://www.t13.org |

2.1 Approved references

The following approved ANSI standards, approved international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), may be obtained from the international and regional organizations who control them.

2.1.1 ANSI References

| | |
|---------------------------------|--|
| SCSI-3 Block Commands (SBC) | [ANSI INCITS 306-1998] (PACKET command feature set commands) |
| SCSI-3 Primary Commands (SPC) | [ANSI X3.301-1997] (PACKET command feature set device types) |
| SCSI-3 Streaming Commands (SSC) | [ANSI INCITS 335-2000] (PACKET command feature set commands) |
| Multimedia Commands (MMC) | [ANSI X3.304-1997] (PACKET command feature set sense codes) |
| Multimedia Commands - 2 (MMC-2) | [ANSI INCITS 333-2000] (PACKET command feature set commands) |

| | |
|--|--|
| Multimedia Commands - 3 (MMC-3) | [ANSI INCITS 360-2002] (PACKET command feature set commands) |
| Protected Area Run Time Interface Extensions (PARTIES) | [ANSI INCITS 346-2001] |
| SCSI Primary Commands - 2 (SPC-2) | [ANSI INCITS 351-2001] (PACKET command feature set commands) |
| ATA Attachment with Packet Interface Extension (ATA/ATAPI-4) | [ANSI INCITS.317-1998] |

2.1.2 ISO References

| | |
|---|---|
| Control and Status Register (CSR) Architecture for microprocessor buses | [ISO/IEC 13213:1994] (World wide names) |
|---|---|

To obtain copies of these documents, contact Global Engineering or INCITS. Additional information may be available at <http://www.t10.org> and <http://www.t13.org>.

2.2 References under development

At the time of publication, the following referenced standards were still under development. For information on the current status of the document, or regarding availability, contact the relevant standards body or other organization as indicated.

| | |
|-----------------------------------|--|
| SCSI Block Commands - 2 (SBC-2) | [T10/1417-D] |
| SCSI Primary Commands - 3 (SPC-3) | [T10/1416-D] (PACKET command feature set commands) |
| ATAPI for Rewritable Media | [SFF8070i] (PACKET command feature set commands) |
| Multimedia Commands - 4 (MMC-4) | [T10/1545D] (PACKET command feature set commands) |

For more information on the current status of the T10 documents, contact INCITS. To obtain copies of T10 or SFF documents, contact Global Engineering.

2.3 Other references

The following standard and specifications are also referenced.

PC Card Standard, February 1995, PCMCIA (68-pin Connector)

For the PC Card Standard published by the Personal Computer Memory Card International Association, contact PCMCIA at 408-433-2273 or <http://www.pc-card.org>.

CompactFlash™ Association Specification, Revision 1.4

For the CompactFlash™ Association Specification published by the CompactFlash™ Association, contact the CompactFlash™ Association at <http://www.compactflash.org>.

ATA Packet Interface (ATAPI) for Streaming Tape QIC-157 revision D

For QIC specifications published by Quarter-Inch Cartridge Drive Standards, Inc., contact them at 805 963-3853 or <http://www.qic.org>.

EIA-364-09 TP-09C - Durability test procedure for electrical connectors and contacts

EIA-364-13 Mating and unmating forces test procedures for electrical connectors
EIA-364-17 TP-17B - Temperature life with or without electrical load test procedure for electrical connectors and sockets
EIA-364-18 Visual and dimensional inspection for electrical connectors
EIA-364-20 TP-20B - Withstanding voltage test procedure for electrical connectors, sockets, and coaxial contacts
EIA-364-21 Insulation resistance test procedure for electrical connectors, sockets, and coaxial contacts
EIA-364-23 Low level contact resistance test procedure for electrical connectors and sockets
EIA-364-27 Mechanical pulse (specified pulse) for electrical connectors
EIA-364-28 TP-28D Vibration test procedure for electrical connectors and sockets
EIA-364-31 Humidity test procedure for electrical connectors and sockets
EIA-364-32 Thermal shock (temperature cycling) test procedure for electrical connectors and sockets
EIA-364-38 TP-38B - Cable pull-out test procedure for electrical connectors
EIA-364-41 TP-41C - Cable flexing test procedure for electrical connectors
EIA-364-65 TP-65A Mixed flowing gas

For EIA specifications, contact them at <http://www.eia.org>.

3 Definitions, abbreviations, and conventions

3.1 Definitions

For the purposes of this standard, the following definitions apply:

8b10b coding: A coding scheme that represents an 8-bit byte (i.e., a control byte or data byte) as a 10-bit character (i.e., a control character or data character). See 6.2

ATA command set (ACS): A command formatted in accordance with the ATA8 Command Set (ACS).

application client: An object that is the source of ATA commands (see ATA8-ACS)

ASCII character: Designates 8-bit value that is encoded using the ASCII character set.

AT Attachment (ATA): A standard for the internal attachment of storage devices to hosts.

asserted: A signal driven to the true state.

ATA device: A storage peripheral

ATA domain: An I/O system consisting of a set of ATA hosts and ATA devices that communicate with one another by means of a service delivery subsystem

ATA host: A host that originates requests to be processed by an ATA device. See "host".

ATAPI (AT Attachment Packet Interface) device: A device implementing the Packet Command feature set. See "device".

backchannel: When transmitting a FIS, the backchannel is the receive channel.

BER (bit error rate): The statistical probability of a transmitted encoded bit being erroneously received in a communication system. See C.4

big-endian: A format for storage or transmission of binary data in which the most significant byte appears first. In a multi-byte value, the byte containing the most significant bit is stored in the lowest memory address and transmitted first and the byte containing the least significant bit is stored in the highest memory address and transmitted last (e.g., for the value 0080h, the byte containing 00h is stored in the lowest memory address and the byte containing 80h is stored in the highest memory address).

burst time: The part of an OOB signaling where ALIGN primitives are being transmitted. See 5.5.12

bus release: Clearing of both BSY and DRQ to zero before the action requested by a command operation is complete. See 8.13 and 8.14.

byte: A sequence of eight contiguous bits considered as a unit. A byte is encoded as a character using 8b10b coding (see xx).

CFA (CompactFlash™ Association): The CompactFlash™ Association which created the specification for compact flash memory that uses the ATA interface.

character: A sequence of ten contiguous bits considered as a unit. A byte is encoded as a character using 8b10b coding (see xx).

code violation: An error that occurs in the decoding of an encoded character. See 6.2.2

COMINIT: A device initiated hardware reset sequence from the issuance of a ComInit Sequence until the device PhyRdy signal is asserted. See 5.5.12.2 and 5.5.12.3.3.

ComInit Sequence: A device to host OOB reset sequence that consists of transmission bursts and quiescent gaps including negation times as defined in subclause 5.5.12.

ComInit signal: The host physical layer detector output that indicates a valid ComReset Sequence is being received. See Figure 22.

COMRESET: A host initiated hardware reset sequence from the issuance of a ComReset Sequence until the host PhyRdy signal is asserted. See 5.5.12.2 and 5.5.12.3.2.1.

ComReset Sequence: A host to device OOB reset sequence that consists of transmission bursts and quiescent gaps including negation times as defined in subclause 5.5.12.

ComReset signal: The host physical layer detector output that indicates a valid ComReset Sequence is being received. See Figure 22.

COMWAKE: A host or device initiated OOB awake sequence from Partial or Slumber power modes, from the issuance of a ComWake Sequence until the PhyRdy signal is asserted. See 5.5.12.2 and 5.5.12.3.4.

ComWake Sequence: A host to device or device to host OOB operation that consists of transmission bursts and quiescent gaps including negation times as defined in subclause 5.5.12 and 5.5.12.2.

ComWake signal: The host or device physical layer detector output that indicates a valid ComWake Sequence is being received. See Figure 22.

command aborted: Command completion with **ERR set to one in the Status field and** ABRT set to one in the Error field

Deleted: and ERR set to one in the Status register

command acceptance: A positive acknowledgment of a Command FIS with an R_OK primitive sequence (see 6.4.2).

command completion: The positive acknowledgment with an R_OK primitive sequence by the host to the device of the Response FIS of the action requested by a command operation or the termination of the command with an error, the placing of the appropriate error and status bits in the HBA response registers, with both BSY and DRQ cleared to zero and the Interrupt Pending bit set to one.

Command FIS: A host to device FIS type 27h, with the C bit set to one. See 7.5.1

command operation: A higher layer request that a command as defined by the ATA Command Set (ATA8-ACS) be transmitted to a device. Synonymous to “command register written” in prior revisions of this standard.

command released: A queued command is released when a valid response FIS is received by the host with BSY and DRQ cleared before command completion, and the REL bit is set to one.

command response: A Response FIS transmitted by the device that contains intermediate response information, ending status, or both.

control byte: A byte containing control information defined in Table 21.

control character (Kxx.y): A character containing control information defined in Table 21.

Control FIS: A host to device FIS type 27h, with the C bit cleared to zero. See 7.5.2

control operation: A higher layer request that a control operation as defined by the ATA Command Set (ATA8-ACS) be transmitted to a device. Typically used to transmit soft reset (SRST) to the device. Synonymous to “control register written” in prior revisions of this standard.

cyclic redundancy check (CRC): An error checking mechanism that checks data integrity by computing a polynomial algorithm based checksum. See xx

data byte: A byte containing data information defined in Table 20.

data character (Dxx.y): A character containing data information defined in Table 20.

data dword: A dword containing four data bytes, or four data characters with correct disparity.

data-in: The protocol that moves data from the device to the host. Such transfers are initiated by READ commands.

data-out: The protocol that moves data from the host to the device. Such transfers are initiated by WRITE commands.

deterministic jitter: Jitter with a non-Gaussian probability density function. Deterministic jitter is always bounded in amplitude and has specific causes. Four kinds of deterministic jitter are identified: duty cycle distortion, data dependent, sinusoidal, and uncorrelated (to the data) bounded. Deterministic jitter is characterized by its bounded, peak-to-peak value.

device: A storage peripheral. 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.

disparity: The difference between the number of ones and the number of zeros in an encoded character. See 6.2.2.2

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

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

DRQ data block: A unit of data words transferred during a single assertion of DRQ bit of the Status field when using PIO data transfer. Editors Note: This is being reviewed in all volumes.

Formatted: Font: Bold

dword: A sequence of four contiguous bytes or four contiguous characters considered as a unit. The meaning depends on the context (e.g., when discussing the bits being transmitted over a physical link, dword represents four characters (i.e., 40 bits). When discussing the contents of a frame after 10b8b decoding (see 3.1.3), a data dword represents four bytes (i.e., 32 bits).

elasticity buffer: A portion of the receiver where character slipping and/or character alignment is performed.

encoded character: In a serial transport implementation, the output of the 8b/10b encoder. See 6.2.2.3.

field: A group of one or more contiguous bits, or bits interpreted as contiguous even if separated by other fields.

first party DMA (FPDMA): A method by which a device accesses host memory. First party DMA differs from DMA in that the device sends a First Party DMA Setup FIS with information to configure host tags, offsets, and transfer counts; whereas for DMA the host configures the DMA controller.

Forced Unit Access (FUA): Forced Unit Access requires that user data be transferred to or from the device media before command completion even if caching is enabled.

frame: A unit of information exchanged between the host adapter and a device. A frame consists of an SOF primitive, a Frame Information Structure (FIS), a CRC calculated over the contents of the FIS, and an EOF primitive.

Frame Information Structure (FIS): The payload of a frame and does not include the SOF primitive, CRC, and EOF primitive. (Plural FISes, possessive FIS's) See 6.3.

Gen1 dword time: The time it takes to transmit a 40 bit encoded value at 1.5 Gb/Sec.

hard reset: A SATA device action in response to a reset event (COMRESET or power on reset) in which the device performs the operations described in subclause 5.5.12.3.2.

HBA Control register: Implementation specific hardware where HBA requests a control operation from device. Synonymous with Shadow Control Register in ATA/ATAPI-7.

HBA Data register: Implementation specific hardware that receives the data from device data FISes. Synonymous with Shadow Data Register in ATA/ATAPI-7.

HBA response register: Implementation specific hardware that receives the response from device response FISes. Synonymous with Shadow Command Block and Shadow Control Block in ATA/ATAPI-7.

host adapter: The implementation specific hardware of the host transport, link, and physical layers.

host bus adapter (HBA): Synonymous with host adapter.

host: The 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.

idle time: The part of an OOB sequence (see 5.5.12) where the transmitter output is quiescent (see 5.5.12.1).

indication: A message passed between states of the same or separate state machines.

Deleted: from a lower layer state machine to a higher layer

Interrupt Pending: An internal state of the host adapter. This state is entered by reception of a FIS with the I field set to one.

Deleted: , usually relaying a request (see xx) from a peer higher layer state machine.

intersymbol interference (ISI): Reduction in the distinction of a pulse caused by overlapping energy from neighboring pulses. ISI may result in data dependent deterministic jitter.

invalid character: A character that is not a control character (see Table 21) or a data character (see Table 20).

jitter: Abrupt and unwanted variations in the interval between successive pulses.

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

least significant bit (LSB): In a binary code, the bit or bit position with the smallest numerical weighting in a group of bits that, when taken as a whole, represent a numerical value (e.g., in the number 0001b, the bit that is set to one).

Linear Feedback Shift Register (LFSR): (See 6.4.5.1)

link reset: Performing the link reset sequence (see 6.7.1.1).

link: The layer that manages the Phy layer to achieve the delivery and reception of frames. (see 6.7)

little-endian: A format for storage or transmission of binary data in which the least significant byte appears first. In a multi-byte value, the byte containing the least significant bit is stored in the lowest memory address and transmitted first and the byte containing the most significant bit is stored in the highest memory address and transmitted last (e.g., for the value 0080h, the byte containing 80h is stored in the lowest memory address and the byte containing 00h is stored in the highest memory address).

most significant bit (MSB): In a binary code, the bit or bit position with the largest numerical weighting in a group of bits that, when taken as a whole, represent a numerical value (e.g., in the number 1000b, the bit that is set to one).

negated: A signal driven to the false state (equivalent to not asserted).

OOB sequence: A sequence where Phys transmit and detect OOB signaling. See 5.5.12 and 5.5.12.2.

OOB signaling: The transmission and detection of OOB sequences.

packet delivered command: A 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.

Phy: physical layer See 5.4.

physical layer: physical layer electronics. See 5.4.

physical link: Two differential signal pairs, one pair in each direction, that connect two physical phys. See Figure 21 and subclause 5.4.

PIO (programmed input/output) data transfer: Data transfers using PIO commands and protocol. See subclauses 7.5.8, 7.7.4, and 8.8.

power on: Power being applied.

primitive sequence: A set of primitives treated as a single entity.

primitive: A dword containing a K28.3 or K28.5 control character in byte zero with correct disparity followed by three data characters with correct disparity. See 6.4

queued: Concurrent commands issued to the same device which have been accepted but not completed. Only commands included in the Overlapped feature (see ATA8-ACS) set may be queued.

random jitter: Jitter that is assumed to have a Gaussian distribution. See 5.5.8.2

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

receiver: The recipient of a signal.

register delivered command: A command that is delivered to the device by placing the command and all of the parameters for the command into a Command FIS. Also see packet delivered command.

register: A register may be a physical hardware register, or a vendor specific implementation that is interpreted as a logical field.

Response FIS: A device to host FIS type 34h. See 7.5.3

response structure: A response as defined by the ATA Command Set (ATA8-ACS).

running disparity: A binary value indicating the cumulative encoded signal imbalance between the one and zero signal state of all characters since dword synchronization has been achieved (see xx).

SATA device port: A storage device object in an ATA domain that interfaces to the service delivery subsystem with SATA

SATA device: A storage device that contains a SATA device port in an ATA domain

SATA host port: A host device object in an ATA domain that interfaces to the service delivery subsystem with SATA

SATA host: A host device containing a SATA host port in an ATA domain

SATA Phy: A Phy in a SATA device that interfaces to a service delivery subsystem

scrambling: Modifying data by XORing each bit with a pattern generated by a linear feedback shift register to minimize repetitive character patterns. See 6.4.5.1.

sector: A uniquely addressable set of data.

Serial ATA (SATA): The serial implementation of ATA. Synonymous with serial transport.

serial transport: The physical implementation of the serial implementation of ATA defined by this standard. Also see transport.

signature: A unique set of values placed in the HBA response registers by the device sending a Response FIS, (e.g. to allow the host to distinguish devices implementing the PACKET Command feature set from those devices not implementing the PACKET Command feature set).

speed negotiation: A sequence in which two phys negotiate the operational physical link rate. See 5.5.12.2

spread spectrum clocking (SSC): The technique of modulating the operating frequency of a transmitted signal to reduce the measured peak amplitude of radiated emissions. See 5.5.9.

state machine variable: A variable that exists within the context of a state machine that may log status determined in one state that is used in another state of the same state machine affecting subsequent state transitions or state machine outputs. (editors note: use this to replace “saved indication” in state machines and text)

total jitter: Measured jitter including deterministic jitter and random jitter.

transceiver: A physical entity that contains both a transmitter and a receiver.

transmitter: The source or generator of a signal.

transport: transport layer. The layer that manages the lower layers (link and Phy) as well as constructing and parsing FISes. (Also see serial transport).

unit interval (UI): The time required to transmit one bit on a physical link (i.e., 666.6 ps at 1.5 Gbps and 333.3 ps at 3.0 Gbps).

vendor specific (VS): Bits, bytes, fields, and code values that are reserved for vendor specific purposes. These bits, bytes, fields, and code values are not described in this standard, and may vary among vendors. This term is also applied to levels of functionality whose definition is left to the vendor. NOTE – 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.

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

3.2 Symbols and abbreviations

See xx for abbreviations of standards bodies (e.g., ISO). Additional symbols and abbreviations used in this standard include:

| Abbreviation | Meaning |
|--------------|---|
| A.C. | alternating current |
| ACK | acknowledge primitive |
| ACS | ATA Command Set |
| ATA | AT attachment |
| ATAPI | AT attachment packet interface |
| ATA/ATAPI-7 | AT Attachment with Packet Interface - 7 standard |
| ATA8-ACS | AT Attachment 8 - ATA Command Set |
| AWG | American wire gauge |
| BER | bit error rate |
| BIST | built in self test |
| CDB | command descriptor block |
| CFA | The CompactFlash™ Association |
| CJTPAT | compliant jitter test pattern |
| CRC | cyclic redundancy check |
| dB | decibel |
| D.C. | direct current |
| DMA | direct memory access |
| Dxx.y | data character |
| e | 2.71828..., the base of the natural (i.e., hyperbolic) system of logarithms |
| EMI | electromagnetic interference |
| EOF | end of frame primitive |
| FIS | frame information structure |
| FUA | force unit access |
| Gen1 | generation 1 physical link rate (1.5 Gbps) |
| Gen2 | generation 2 physical link rate (3.0 Gbps) (defined in a future version of this standard) |
| Gen3 | generation 3 physical link rate (defined in a future version of this standard) |
| Gbps | gigabits per second (10^9 bits per second) |
| GHz | gigahertz (10^9 transitions per second) |
| ISI | intersymbol interference |
| JTPAT | jitter test pattern |
| kHz | kilohertz (10^3 bits per second) |
| Kxx.y | control character |
| LBA | logical block address |
| LFSR | linear feedback shift register |
| LSB | least significant bit |
| μ A | microampere (10^{-6} amperes) |
| μ s | microsecond (10^{-6} seconds) |
| Mbaud | megabaud (10^6 transitions per second) |
| MBps | megabytes per second (10^6 bytes per second) |
| MHz | megahertz (10^6 bits per second) |
| MSB | most significant bit |
| ms | millisecond (10^{-3} seconds) |
| mV | millivolt (10^{-3} volts) |
| N/A | not applicable |
| NAK | negative acknowledge primitive |
| nF | nanofarad (10^{-9} Farads) |
| ns | nanosecond (10^{-9} seconds) |
| OOB | out-of-band |
| PIO | Programmed input output protocol |
| PLL | phase lock loop |
| P-P | peak-to-peak |
| ppm | parts per million (10^{-6}) |

| | |
|--------|---|
| ps | picosecond (10^{-12} seconds) |
| ρ | reflection coefficient (rho) |
| RCDT | rate change delay time |
| RRDY | receiver ready primitive |
| Rx | receive |
| SATA | Serial ATA |
| SCSI | Small Computer System Interface family of standards |
| SOF | start of frame primitive |
| SSC | spread spectrum clocking |
| s | second (unit of time) |
| Tx | transmit |
| UI | unit interval |
| V | volt |
| XOR | exclusive logical OR |
| x | multiplication |
| / | division |

3.3 Conventions

3.3.1 Precedence

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

3.3.1.1 Lists

Lists sequenced by letters (e.g., a) red, b) blue, c) green) show no ordering relationship between the listed items. Listed sequenced by numbers (e.g., 1) first, 2) second, 3) third) show an ordering between the listed items.

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

- 1) first
- 2) second
- 3) third

Unordered list are of the form:

- a) red
- b) blue
- c) green

3.3.2 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: A keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may not”).

obsolete: A keyword indicating 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 command 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: A keyword that 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.

reserved: A keyword indicating 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: A keyword indicating 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: A keyword indicating a mandatory requirement (equivalent to “is required to”). Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

should: A keyword indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase “it is strongly recommended”.

3.3.3 Editorial conventions

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 subclause 3.1 or in the text where they first appear.

Uppercase is used for defined terms (e.g. COMRESET) and may have different meaning than normal or mixed-case (e.g. ComReset).

Mixed-Case is used for defined terms (e.g. PhyRdy).

Normal case is used for words having the normal English meaning.

A binary number is represented in this standard by any sequence of digits comprised of only the Western-Arabic numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included between characters in binary number representations to increase readability or delineate field boundaries (e.g., 0 0101 1010b or 0_0101_1010b).

A hexadecimal number is represented in this standard by any sequence of digits comprised of only the Western-Arabic numerals 0 through 9 and/or the upper-case English letters A through F immediately followed by a lower-case h (e.g., FA23h). Underscores or spaces may be included between characters in hexadecimal number representations to increase readability or delineate field boundaries (e.g., B FD8C FA23h or B_FD8C_FA23h).

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

A decimal number is represented in this standard by any sequence of digits comprised of only the Western-Arabic numerals ‘0’ through ‘9’ not immediately followed by a lower-case ‘b’ or lower-case ‘h’ (e.g., 25).

This standard uses the American convention for representing decimal numbers (e.g., the thousands and higher multiples are separated by a comma, and a period is used as the decimal point). Table 2 shows some examples of decimal numbers using the ISO and American numbering conventions.

Table 2 - ISO and American numbering conventions

| ISO | American |
|--------------|--------------|
| 0,6 | 0.6 |
| 3,141 592 65 | 3.14159265 |
| 1 000 | 1,000 |
| 1 323 462,95 | 1,323,462.95 |

3.3.4 Signal conventions

Signal names are shown in all uppercase letters.

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. 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.

3.3.5 State diagram conventions

State diagrams are as shown in Figure 2.

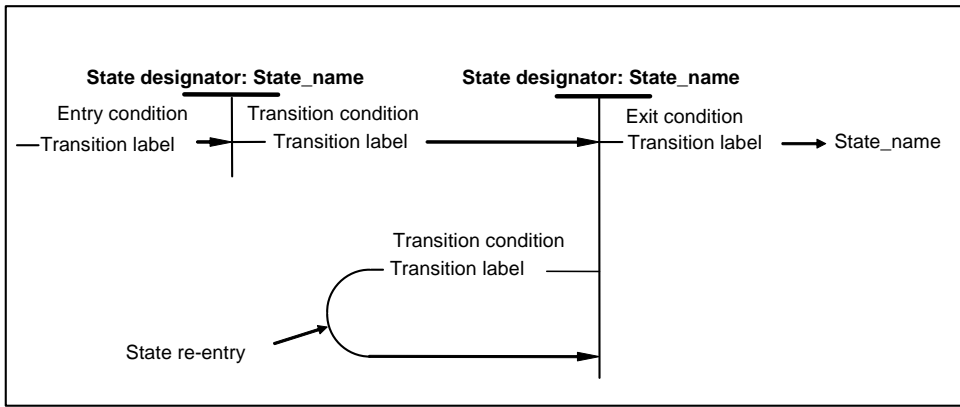


Figure 2 – State diagram convention

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 is typically followed by a sequence or identifying number. The state name is typically 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 typically 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.

The state diagrams do not constitute complete requirements for actions within a state or for state transitions, and the accompanying text description for the state is included as part of the requirements.

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 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. For re-entry transitions, transition labels may be omitted to simplify diagrams.

Transitions from state to state shall be instantaneous.

3.3.6 Bit and byte ordering

In a field in a table consisting of more than one bit that contains a single value (e.g., a number), the least significant bit (LSB) is shown on the right and the most significant bit (MSB) is shown on the left (e.g., in a byte, bit 7 is the MSB and is shown on the left; bit 0 is the LSB and is shown on the right). The MSB and LSB are not labeled if the field consists of 8 or fewer bits.

In a field in a table consisting of more than one byte that contains multiple fields each with their own values (e.g., a descriptor), there is no MSB and LSB of the field itself and thus there are no MSB and LSB labels. Each individual field has an MSB and LSB, but they are not labeled.

In a field containing a text string (e.g., ASCII or UTF-8), the MSB label is the MSB of the first character and the LSB label is the LSB of the last character.

Multiple byte fields are represented with only two rows, with the non-sequentially increasing byte number indicating the presence of additional bytes.

A data dword consists of 32 bits. Table 4 shows a data dword containing a single value, where the MSB is on the left in bit 31 and the LSB is on the right in bit 0.

Table 3 - Data dword containing a value

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|-------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | LSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 5 shows a data dword containing four one-byte fields, where byte 0 (the first byte) is on the right and byte 3 (the fourth byte) is on the left. Each byte has an MSB on the left and an LSB on the right.

Table 4 - Data dword containing four one-byte fields

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|--------|---|---|---|---|---|---|---|---------------|--|--|--|--|--|--|--|-----|--|--|--|--|--|--|--|--------|--|--|--|--|--|--|--|-----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|---------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| MSB | | | | | | | | byte 3 | | | | | | | | LSB | | | | | | | | byte 2 | | | | | | | | byte 1 | | | | | | | | MSB | | | | | | | | byte 0 | | | | | | | | LSB | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (Fourth byte) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (Third byte) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (Second byte) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | (First byte) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

3.3.7 Byte, word and dword Relationships

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

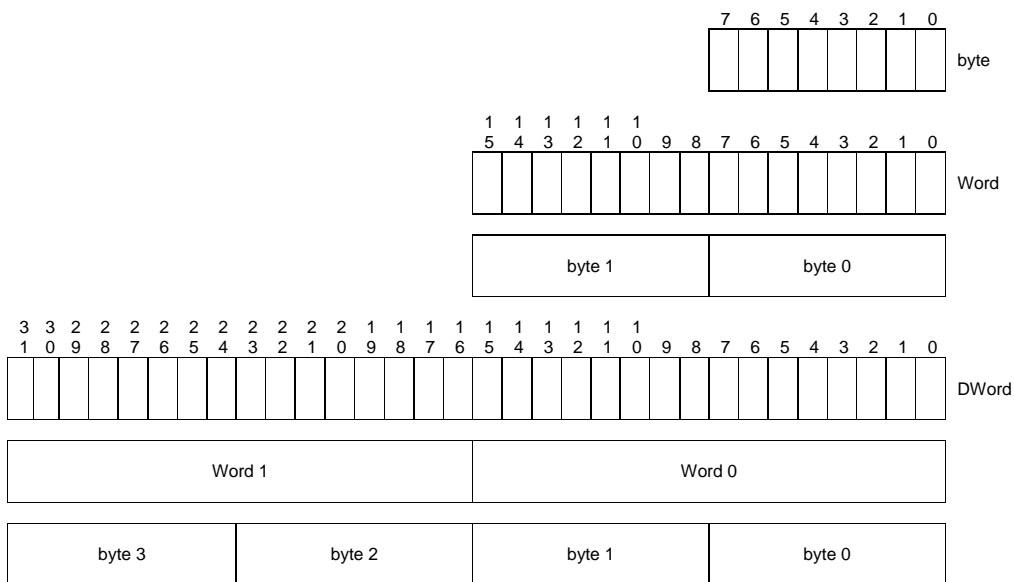


Figure 3 - Byte, word and data dword relationships

4 Serial transport general overview

4.1 Overview

The ATA serial transport is a high-speed serial implementation of the ATA attachment. The serial transport employed is a high-speed differential layer that utilizes Gigabit technology and 8b/10b encoding.

Figure 4 illustrates how two devices are connected using a host adapter with a serial transport. The serial transport structure connects each of the two drives with their own respective cables in a point-to-point fashion.

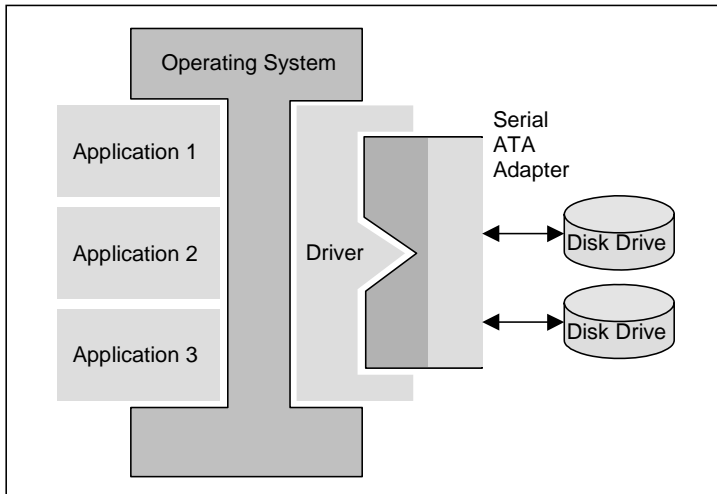


Figure 4 - The serial transport connectivity

4.2 Sub-module operation

Figure 5 shows how the machines communicate their various needed parameters in a hierarchal model. Each layer communicates with its counterpart directly or indirectly.

The host interacts with the transport control through an interface that implements the ATA8-ACS command sets and the transport control layer breaks down these operations into a sequence of actions that are exchanged with the link layer.

The transport layer and the link layer are the two core sub-modules that control overall operation. The link controls the operation(s) related to the serial line and the transport controls the operation(s) relating to the host platform. The two layers coordinate their actions and utilize resources to transfer data between a host computer and attached device. The host link layer communicates via the serial line to a corresponding link layer located in the device. The host transport layer also likewise communicates with a corresponding device transport layer. The two link layers ensure that control sequences between the two transport layers are properly exchanged. Each layer is defined by state machines that define layer operations.

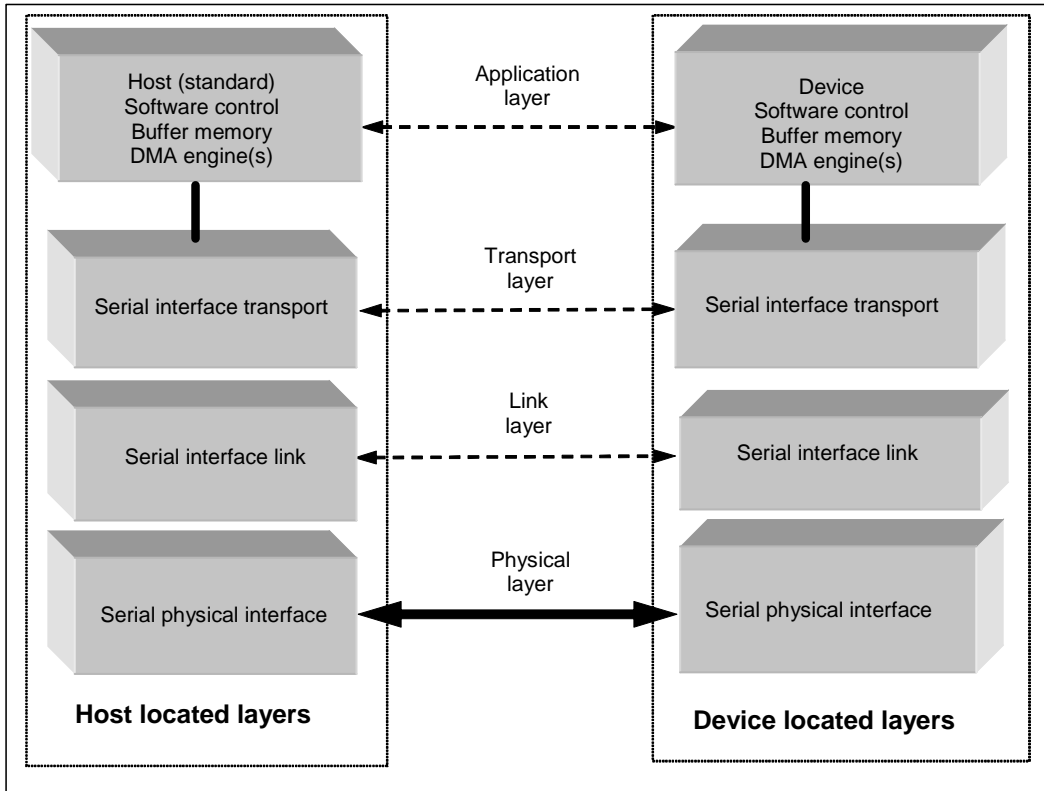


Figure 5 - Communication layers

5 Serial transport physical layer

5.1 Overview

This subclause describes the physical layer of the serial transport. 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 subclauses and should be taken as examples only. Exact implementations may vary.

5.1.1 List of services

- a) Transmit a 1.5 Gb/sec differential NRZ serial stream at specified voltage levels
- b) Provide matched termination at the transmitter
- c) Serialize a 10, 20, 40, or other width parallel input from the Link for transmission
- d) Receive a 1.5 Gb/sec differential NRZ serial stream
- e) Provide a matched termination at the receiver
- f) Extract data (and, optionally, clock) from the serial stream
- g) Deserialize the serial stream
- h) Detect the K28.5 comma character and provide a bit and word aligned 10, 20, 40, or other width parallel output
- i) Provide specified OOB signal detection and transmission
- j) Perform proper power-on sequencing and speed negotiation
- k) Provide interface status to link layer
- l) Host/device present
- m) Host/device absent
- n) Host/device present but failed to negotiate communications
- o) Optionally support power management modes
- p) Optionally perform transmitter and receiver impedance calibration
- q) Handle the input data rate frequency variation due to a spread spectrum transmitter clock
- r) Accommodate request to go into Far-End retimed loopback test mode of operation when commanded

5.2 Connectors specifications

5.2.1 Overview

This subclause covers the serial transport connectors and cable assemblies. It defines the

- a) Connector mating interfaces
- b) Connector location on the device
- c) Electrical, mechanical and reliability requirements of the connectors and cable assemblies
- d) Connector and cable testing procedures

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

- a) The mounting feature of the connectors
- b) The cabling and cable terminations
- c) The methods on how the PCB connects to other components of the system

5.2.1.1 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 6, is inserted directly into a host receptacle connector, illustrated as (g) in Figure 6. 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 6, mates with the signal cable receptacle connector on one end of the cable, illustrated as (c) in Figure 6. 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 6. The signal cable wire consists of two twinax subclauses 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 6, on one end and may be directly connected to the host power supply 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 6. The host end of the power cable is not covered in this standard.

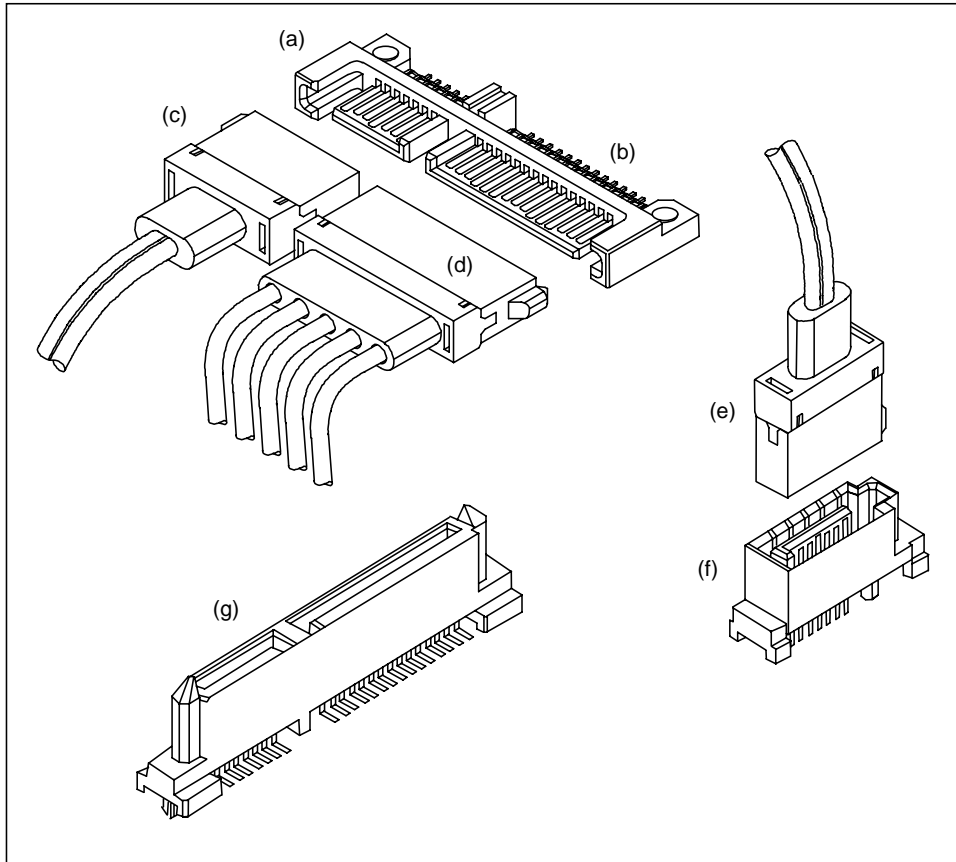


Figure 6 - Serial implementation connector examples

Illustrated above: (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) & (b).

5.2.2 Connector drawings
5.2.2.1 Device plug connector

Figure 7, Figure 8 and Table 5 show the interface dimensions for the device plug connector with both signal and power segments.

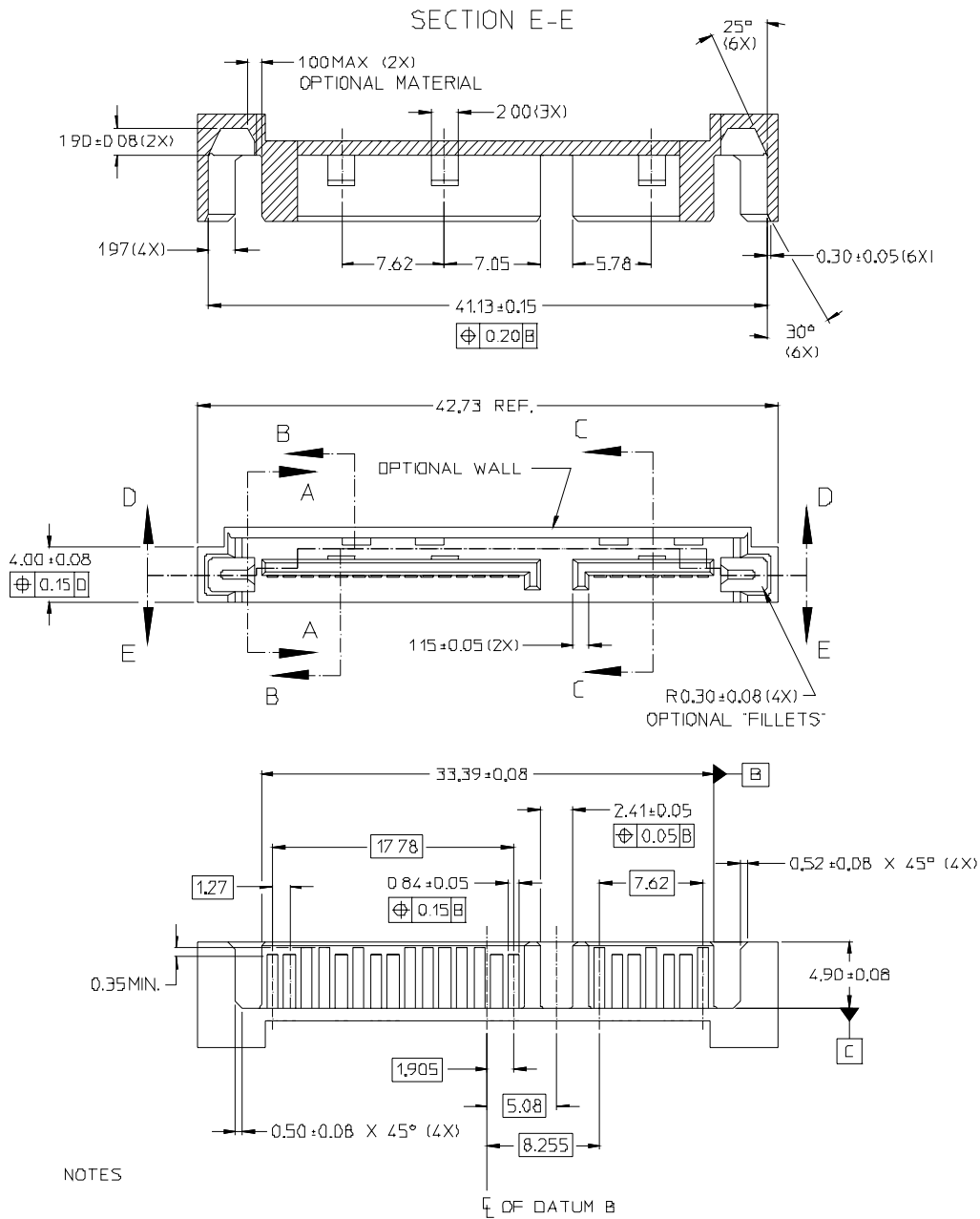


Figure 7 - Device plug connector part 1 of 2

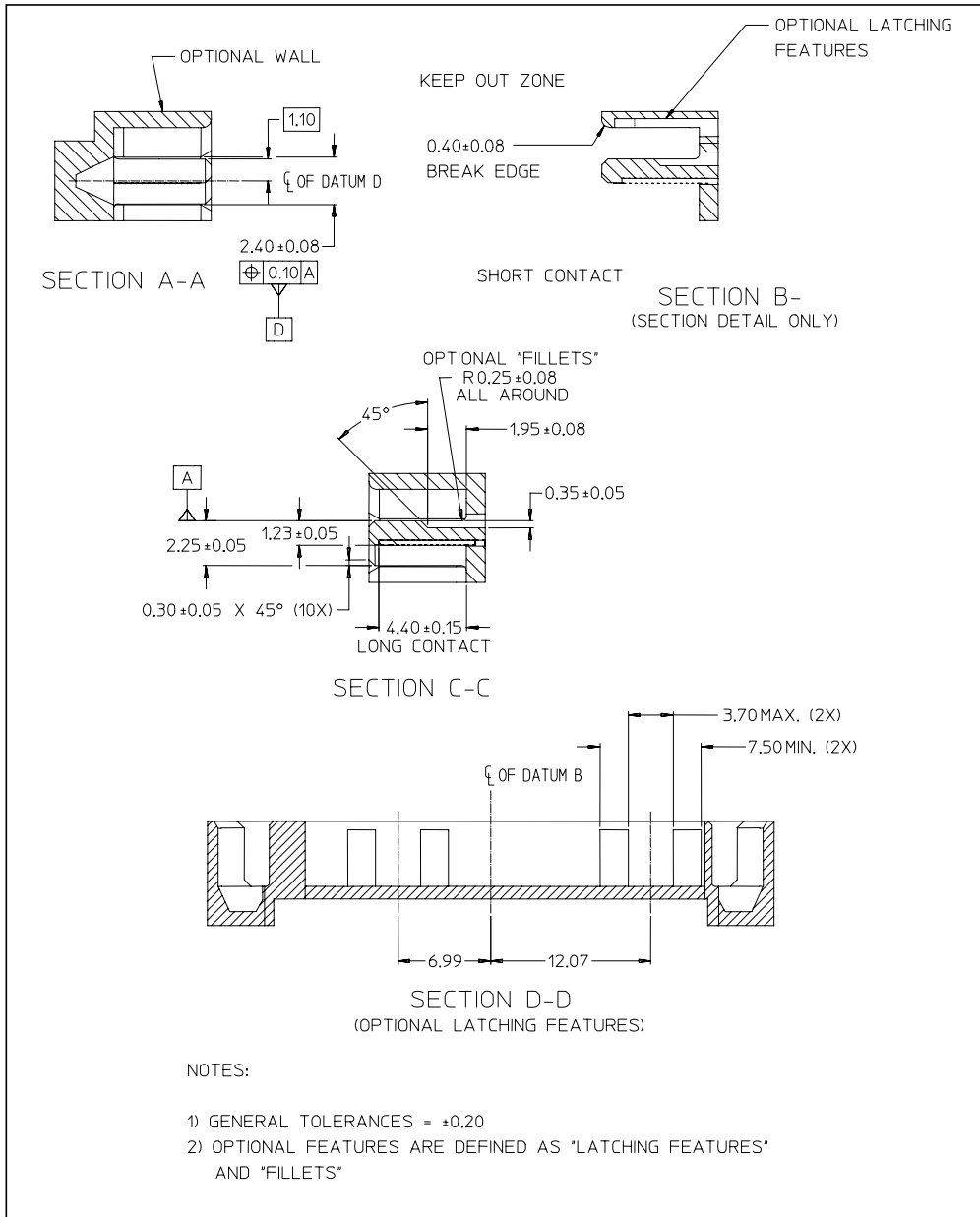


Figure 8 - Device Plug Connector part 2 of 2

5.2.2.2 Signal cable receptacle connector

Figure 9 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.

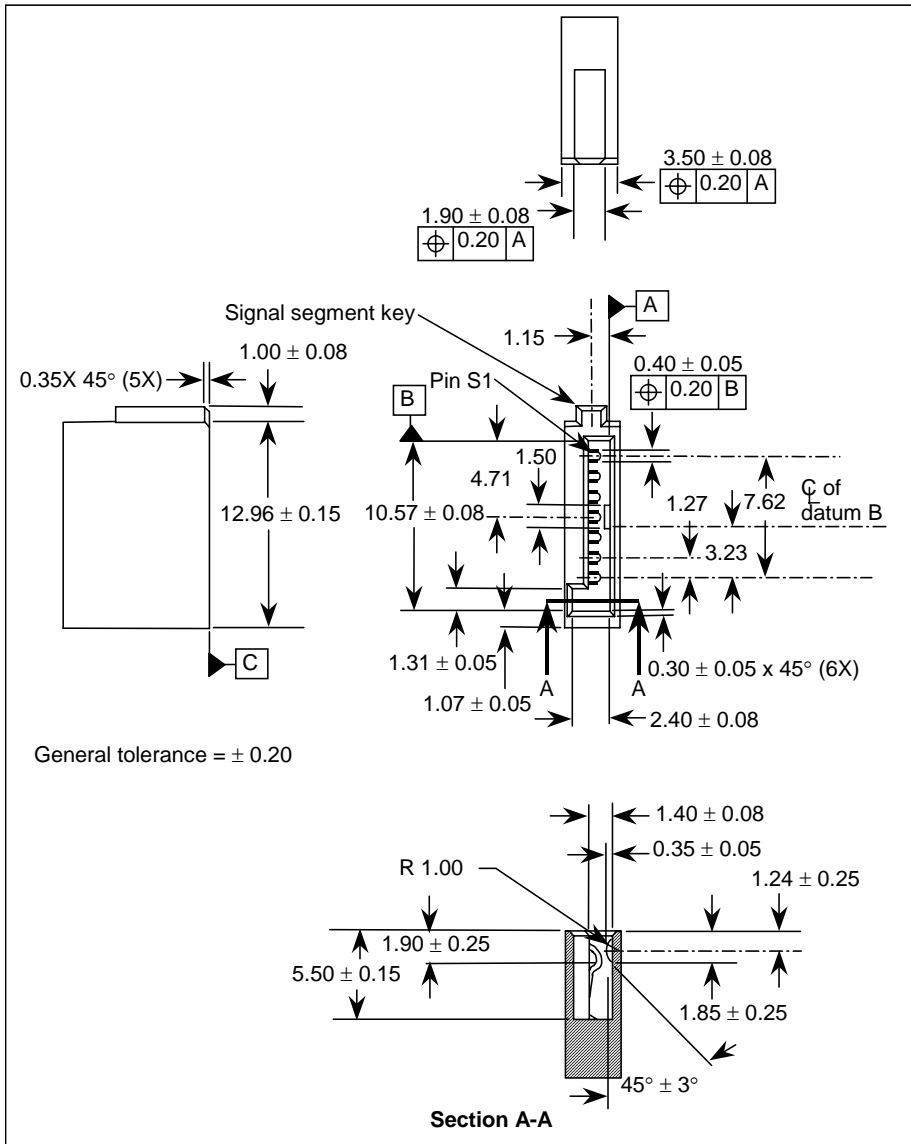


Figure 9 - Non-Latching Signal Cable receptacle connector interface dimensions

Figure 10 shows the additional dimensions for the optional Latching Signal Cable Receptacle.

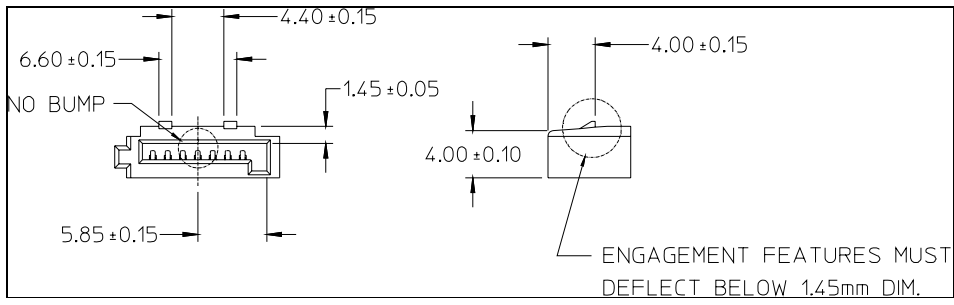


Figure 10 - Optional Latching Signal Cable Receptacle connector interface dimensions

5.2.2.3 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 11 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 19 shows the recommended clearance or spacing.

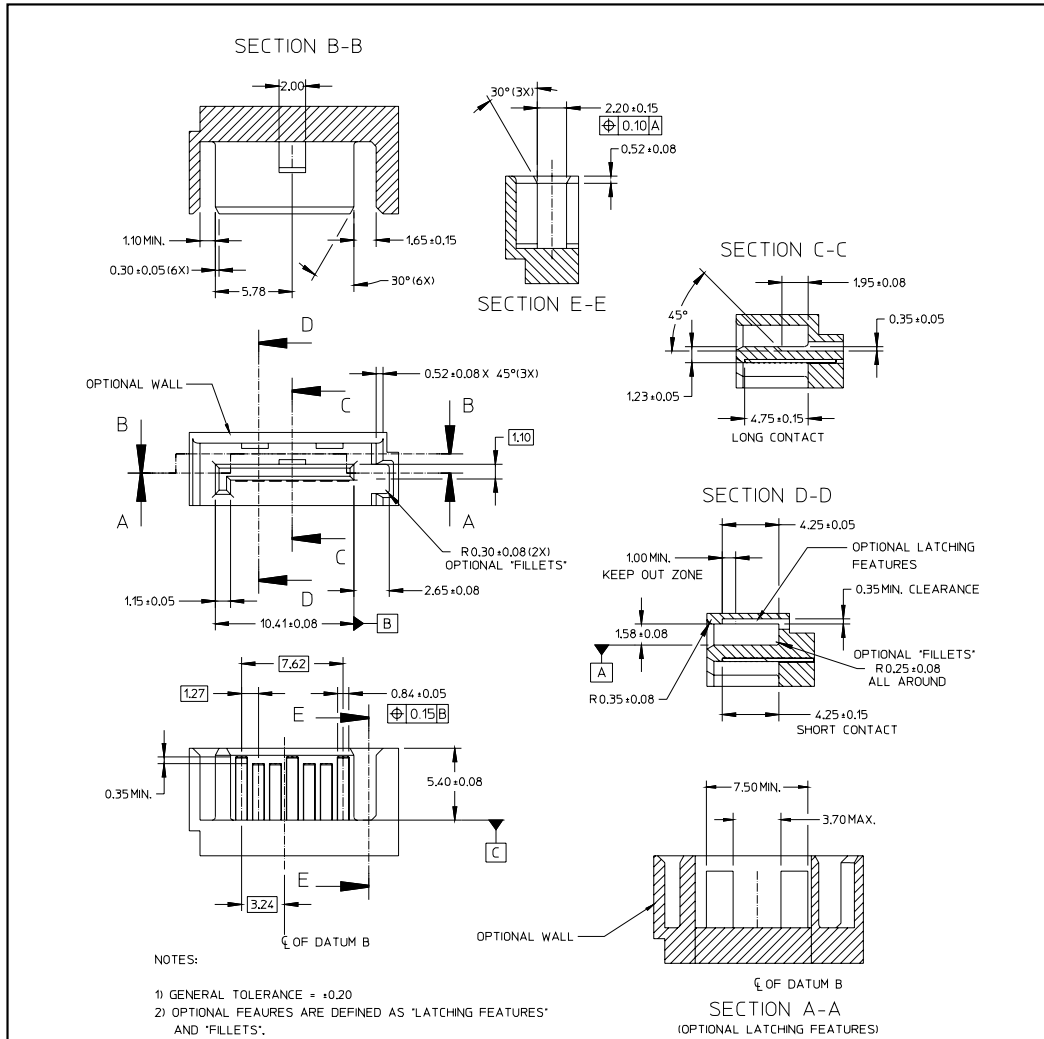


Figure 11 - Host plug connector interface dimension

5.2.2.4 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 12. Note that dimension B allows two values: 8.15mm and 14.15mm. 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.

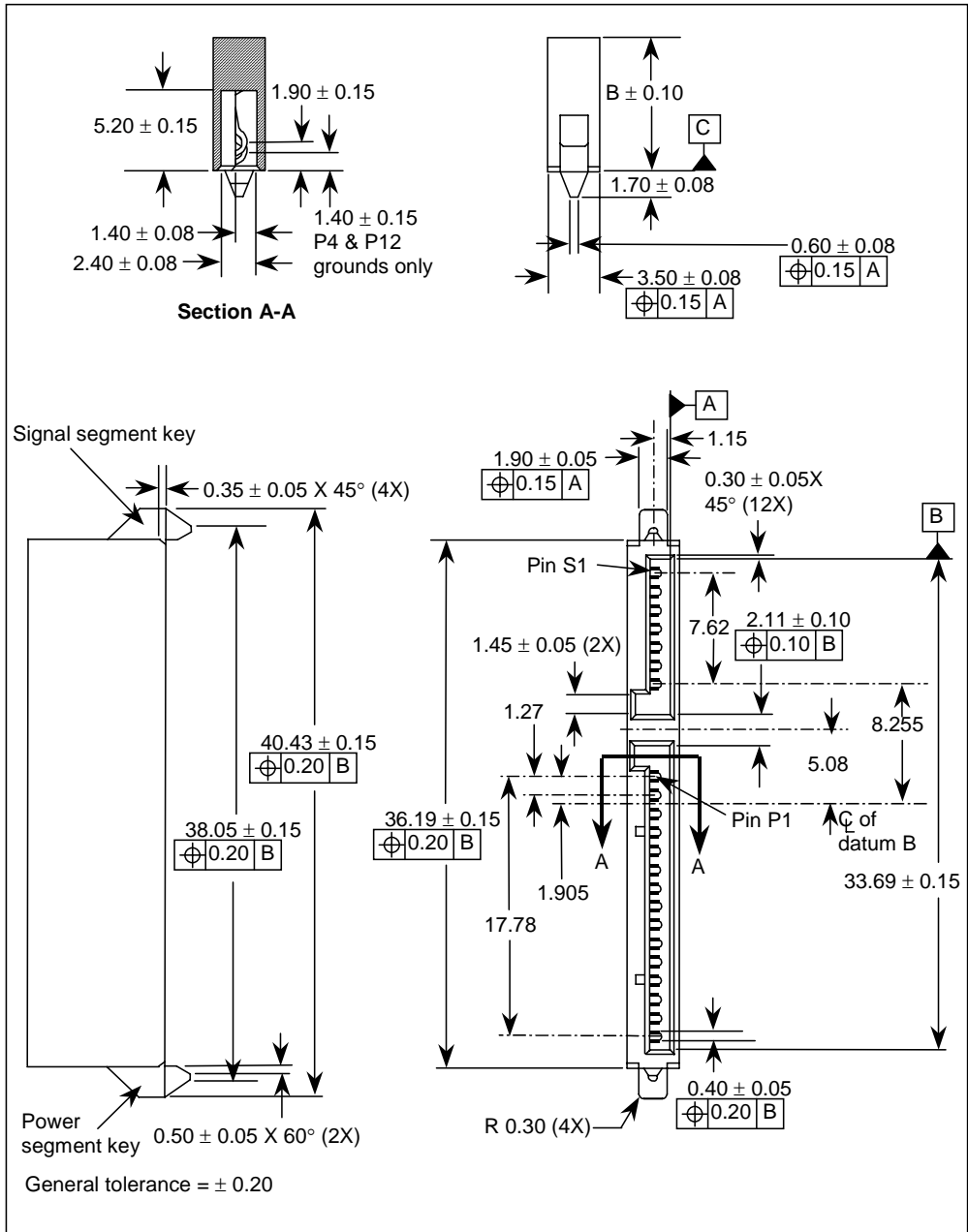


Figure 12 - Host receptacle connector interface dimensions

5.2.2.5 Power cable receptacle connector

The power cable receptacle connector mates with the power segment of the device plug, bringing power to the device. Figure 13 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 5.

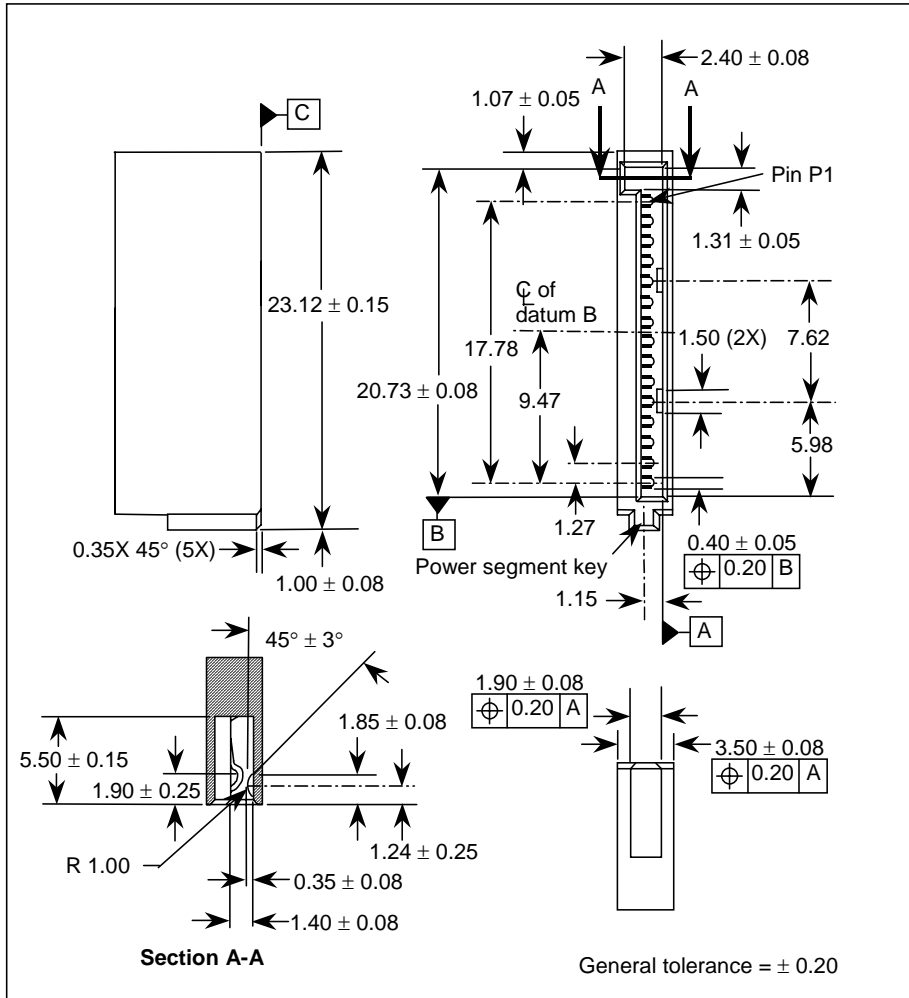


Figure 13 - Non-latching power cable receptacle connector interface dimensions

5.2.2.6 Optional latching power cable receptacle

Figure 14 shows the dimensions for the optional latching feature for the power cable receptacle.

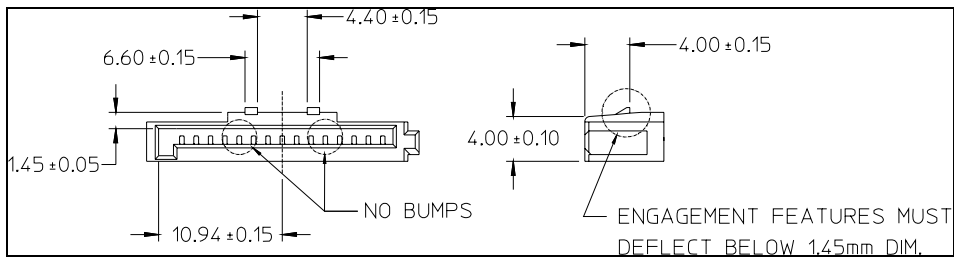


Figure 14 - Optional latching power cable receptacle

5.2.3 Connector pinouts

There are total of 7 pins in the signal segment and 15 pins in the power segment. The pin definitions are shown in Table 5.

Table 5 - Device plug connector pin definition

| Signal segment key | | | |
|--|-----|-----------------|---|
| Signal segment | S1 | Gnd | 2 nd mate |
| | S2 | A+ | Differential signal pair A from Phy |
| | S3 | A- | |
| | S4 | Gnd | 2 nd mate |
| | S5 | B- | Differential signal pair B from Phy |
| | S6 | B+ | |
| | S7 | Gnd | 2 nd mate |
| Signal segment "L" | | | |
| Central connector polarizer | | | |
| Power segment "L" | | | |
| Power segment | P1 | V ₃₃ | 3.3 V power |
| | P2 | V ₃₃ | 3.3 V power |
| | P3 | V ₃₃ | 3.3 V power, pre-charge, 2 nd mate |
| | P4 | Gnd | 1 st mate |
| | P5 | Gnd | 2 nd mate |
| | P6 | Gnd | 2 nd mate |
| | P7 | V ₅ | 5 V power, pre-charge, 2 nd mate |
| | P8 | V ₅ | 5 V power |
| | P9 | V ₅ | 5 V power |
| | P10 | Gnd | 2 nd 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 | 1 st mate |
| | P13 | V ₁₂ | 12 V power, pre-charge, 2 nd mate |
| | P14 | V ₁₂ | 12 V power |
| | P15 | V ₁₂ | 12 V power |
| Power segment key | | | |
| Notes: | | | |
| <ul style="list-style-type: none"> - 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. - There are three power pins for each voltage. One pin from each voltage is used for pre-charge in the backplane blind-mate situation. - V₃₃ pins shall be connected together in the device. - V₅ pins shall be connected together in the device. <p style="text-align: center;">V₁₂ pins shall be connected together in the device.</p> | | | |

5.2.4 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 15. Any skew angle of the plug, with respect to the receptacle, reduces the blind-mate tolerances.

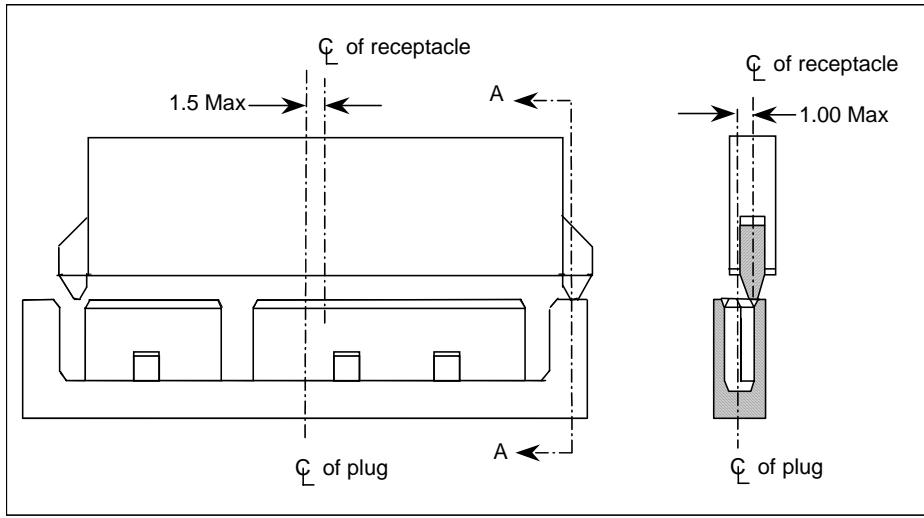


Figure 15 - Connector pair blind-mate misalignment tolerance

The device-to-backplane mating configuration is shown in Figure 16. Note that two values (8.45 and 14.45 mm) are allowed for dimension A

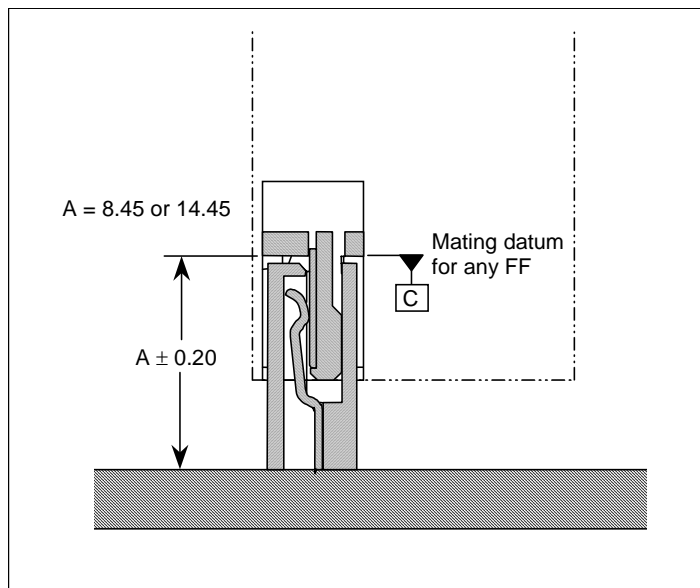


Figure 16 - Device-backplane mating configuration

5.2.5 Connector locations

The device connector location is defined to facilitate blind mating. Figure 17 and Figure 18 define the connector locations on 3.5" and 2.5" devices, respectively. Figure 19 shows the recommended host plug connector clearance and orientation for optional latching connectors.

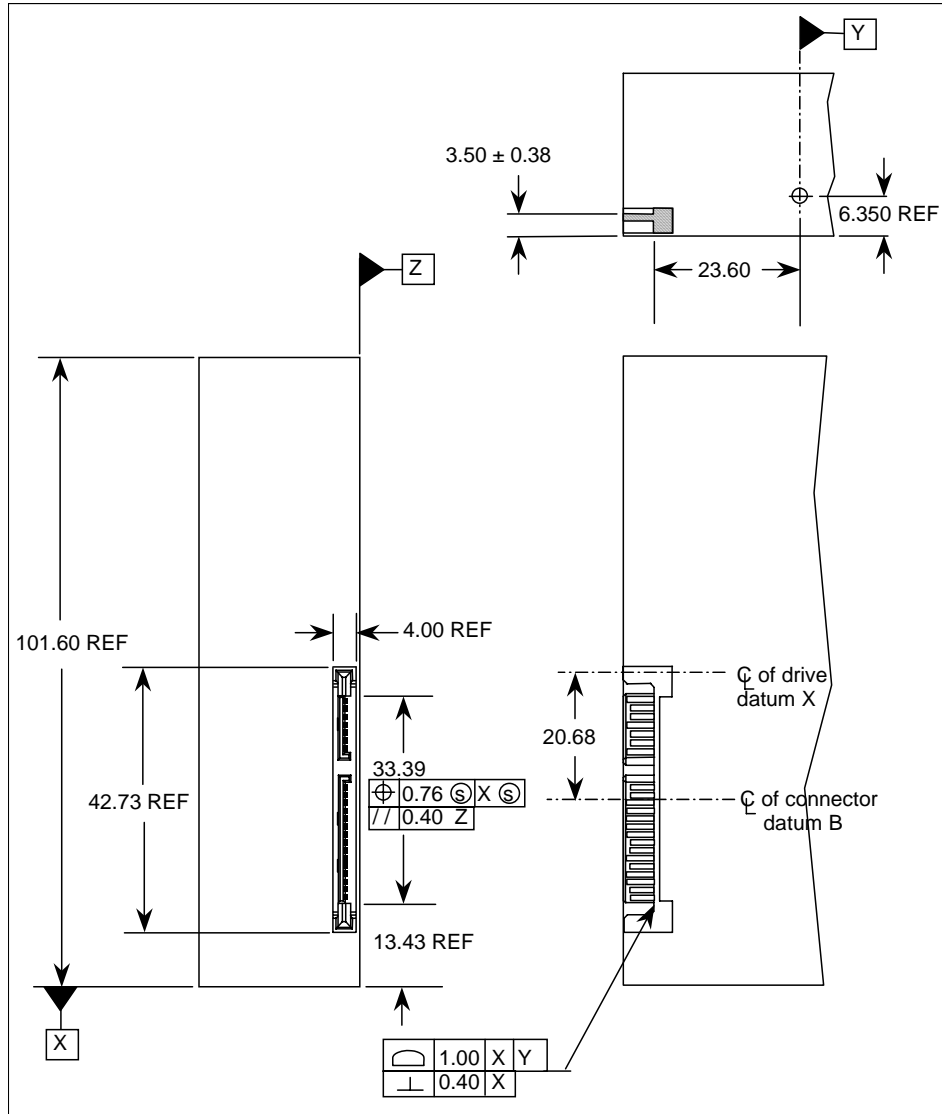


Figure 17 - Device plug connector location on 3.5" device

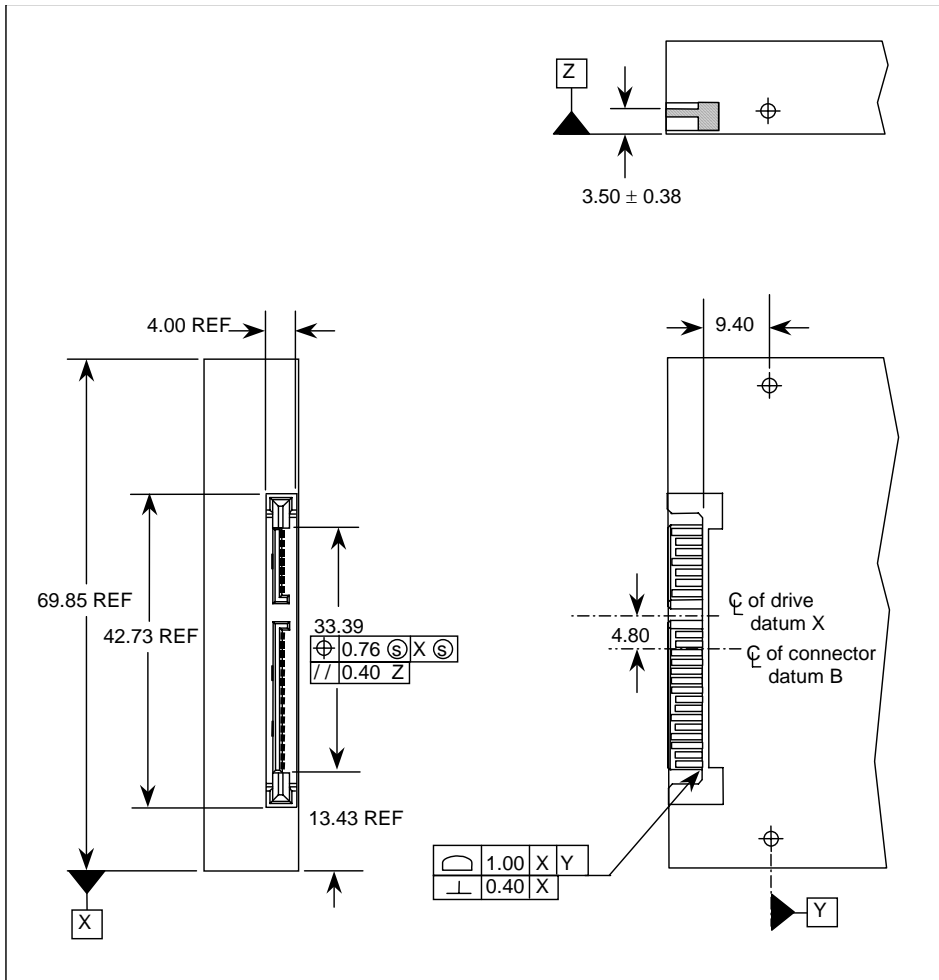


Figure 18 - Device plug connector location on 2.5" device

Recommended host plug connector clearance is shown in Figure 19.

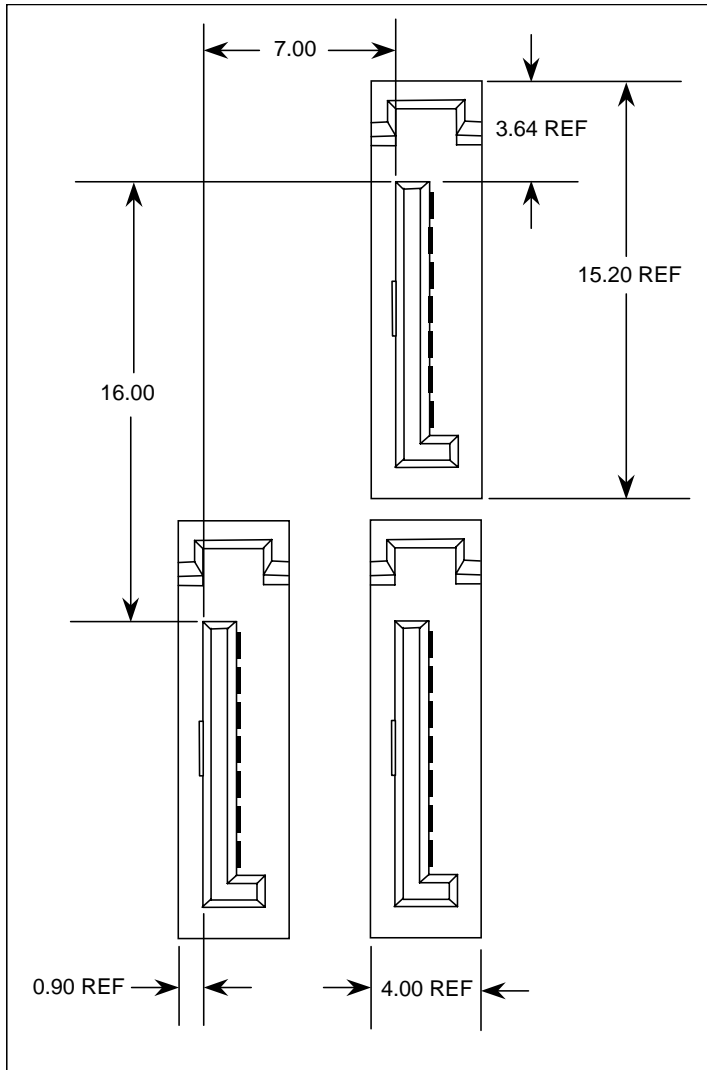


Figure 19 - Recommended host plug spacing for Non-Latching Connectors

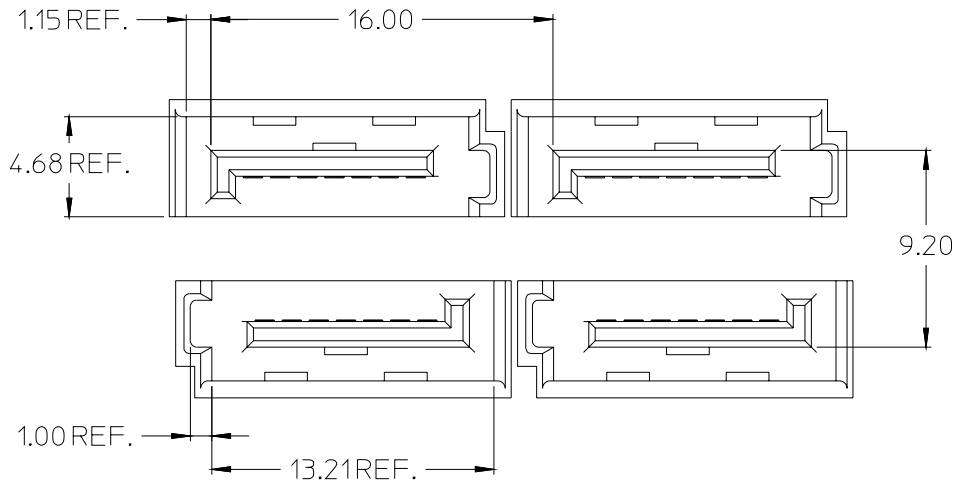


Figure 20 - Recommended host plug connector clearance & Orientation for Optional Latching Connectors

5.2.6 Connector conformance requirements

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

- a) Mated
- b) Temperature: 15° to 35° C
- c) Relative Humidity: 20% to 80%
- d) 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.

5.2.6.1 Signal

The test board shall consist of differential traces (100 ohm \pm 5 ohm) over a ground plane (single ended 50 ohm \pm 2.5 ohm).

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 6 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 6 electrical signaling parameters and requirements when tested with the above specified test fixture or equivalent.

Table 6 - Signal integrity requirements and test procedures

| Parameter | Procedure | Requirements |
|---------------------------|---|-----------------|
| Mated connector impedance | <ol style="list-style-type: none"> 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%-80% Vp). 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 ohm +/- 15% |
| Cable absolute impedance | <ol style="list-style-type: none"> 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%-80% Vp). 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 ohm +/- 10% |

(continued)

Table 6 - Signal integrity requirements and test procedures (continued)

| Parameter | Procedure | Requirements |
|-----------------------|---|---------------|
| Cable pair matching | <ol style="list-style-type: none"> 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%-80% Vp). 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 single-ended 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 $\text{Line1}_{imp} - \text{Line2}_{imp}$. | +/- 5 ohm |
| Common mode impedance | <ol style="list-style-type: none"> 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 20-80%) 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 to 40 ohms |

(continued)

Table 6 - Signal integrity requirements and test procedures (continued)

| Parameter | Procedure | Requirements |
|-----------------|---|--------------|
| Insertion loss | <ol style="list-style-type: none"> 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 to 4500 MHz. 4. Measure and record the IL of the sample, which includes fixturing IL, over a frequency range of 10 to 4500 MHz. 5. The IL of the sample is then the results of procedure 4 minus the results of Procedure 3. | 6 dB max |
| Crosstalk: NEXT | <ol style="list-style-type: none"> 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 to 4500 MHz. 8. Verify that the sample crosstalk is out of the noise floor. | -26 dB |

(continued)

Table 6 - Signal integrity requirements and test procedures (continued)

| Parameter | Procedure | Requirements |
|---------------------------|---|---------------|
| Rise time | <ol style="list-style-type: none"> 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 25-35 ps (measured 20%-80% Vp) 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%-80% Vp) and record the output risetime. | 85 ps |
| Inter-symbol Interference | K - 28.5 signal source running at 1.5 Gb/sec. The average position of zero crossing should not move more than specified value. | 50 ps maximum |

(continued)

Table 6 - Signal integrity requirements and test procedures (continued)

| Parameter | Procedure | Requirements |
|--|---|--------------|
| Intra-Pair Skew | <ol style="list-style-type: none"> 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%-80% Vp). Filtering may be used to slow the system down (See Note 2). 3. Measure propagation delay (50% of Vp) of each line in a pair single-endedly. The skew equals the difference between each single ended propagation delay. | 10 ps max |
| <p>NOTES –</p> <ol style="list-style-type: none"> 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 shall be verified to be <1 ps on a TDR. 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 $(t_{r(\text{observed})})^2 - (t_{r(\text{stimulus})})^2$. After filtering, verify the risetime is achieved using the risetime reference traces on the PCB fixture. 3: Calibrate the system by substituting either precision 50 ohm loads or precision air lines (also terminated in 50 ohm loads) for the test fixture. This places the calibration plane directly at the input interface of the test fixture. 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. | | |

(concluded)

5.2.6.2 Housing and contact electrical requirements

Table 7 is the connector housing and contact electrical requirements.

Table 7 - Housing and contact electrical parameters, test procedures, and requirements

| Parameter | Procedure | Requirement |
|--|---|--|
| Insulation resistance | EIA 364-21 After 500 VDC for 1 minute, measure the insulation resistance between the adjacent contacts of mated and unmated connector assemblies. | 1000 MΩ minimum |
| Dielectric withstanding voltage | EIA 364-20 Method B Test between adjacent contacts of mated and unmated connector assemblies. | The dielectric shall withstand 500 VAC for 1 minute at sea level. |
| Low level contact resistance (LLCR) | EIA 364-23 Subject mated contacts assembled in housing to 20 mV maximum open circuit at 100 mA maximum | <ul style="list-style-type: none"> • Initially 30 mΩ maximum. • Resistance increase 15 mΩ maximum after stress |
| Contact current rating (Power segment) | <ul style="list-style-type: none"> • 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. |

5.2.6.3 Mechanical and environmental requirements

Table 8 lists the mechanical parameters and requirements, while Table 9 the environmental and reliability tests and requirements:

Table 8 - Mechanical test procedures and requirements

| Test description | Procedure | Requirement |
|------------------------------------|--|---|
| Visual and dimensional inspections | EIA 364-18 Visual, dimensional and functional per applicable quality inspection plan. | Meets product drawing requirements. |
| Cable pull-out | EIA 364-38 Condition A Subject a Serial ATA cable assembly to a 40 N axial load for a min of one minute while clamping one end of the cable plug. | No physical damage |
| Cable flexing | For round cable: EIA 364-41 Condition I Dimension $x=3.7 \times$ 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 μ s during flexing. |
| Insertion force | EIA 364-13 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 | EIA 364-13 Measure the force necessary to unmate the connector assemblies at maximum rate of 12.5 mm (0.492") per minute. | 10 N minimum |
| Durability | EIA 364-09 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 5.2.6.5. |

Table 9 - Environmental parameters, test procedures, and requirements

| Parameter | Procedure | Requirement |
|--|---|---|
| Physical shock | EIA 364-27 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 μ s or longer duration. No physical damage. |
| Random vibration | EIA 364-28 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 μ s longer duration. |
| Humidity | EIA 364-31 Method II Test Condition A. Subject mated connectors to 96 hours at 40° C with 90% to 95% RH. | See Note 1 |
| Temperature life | EIA 364-17 Test Condition III Method A. Subject mated connectors to temperature life at +85°C for 500 hours. | See Note 1. |
| Thermal shock | EIA 364-32 Test Condition I. Subject mated connectors to 10 cycles between -55° C and +85° C. | See Note 1. |
| Mixed Flowing Gas | EIA 364-65, 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. |
| <p>NOTE –</p> <ol style="list-style-type: none"> Shall meet EIA 364-18 Visual Examination requirements, show no physical damage, and shall meet requirements of additional tests as specified in the test sequence in 5.2.6.5. Shock and vibration test fixture is to be determined by each user with connector vendors. | | |

An additional requirement is listed in Table 10.

Table 10 - Additional requirement

| Parameter | Procedure | Requirement |
|--------------|-----------|--|
| Flammability | UL 94V-1 | Material certification or certificate of compliance required with each lot to satisfy the Underwriters Laboratories 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.

5.2.6.4 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.

5.2.6.5 Test sequence

Table 11 shows the connector test sequences for five groups of tests.

Table 11 - 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)

5.3 Cable assemblies

Figure 21 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 cable size may be 30 to 26 AWG. The cable maximum length is one meter.

This standard does not specify how to construct a standard cable for a serial transport. 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 Annex J.2 for informative purposes only.

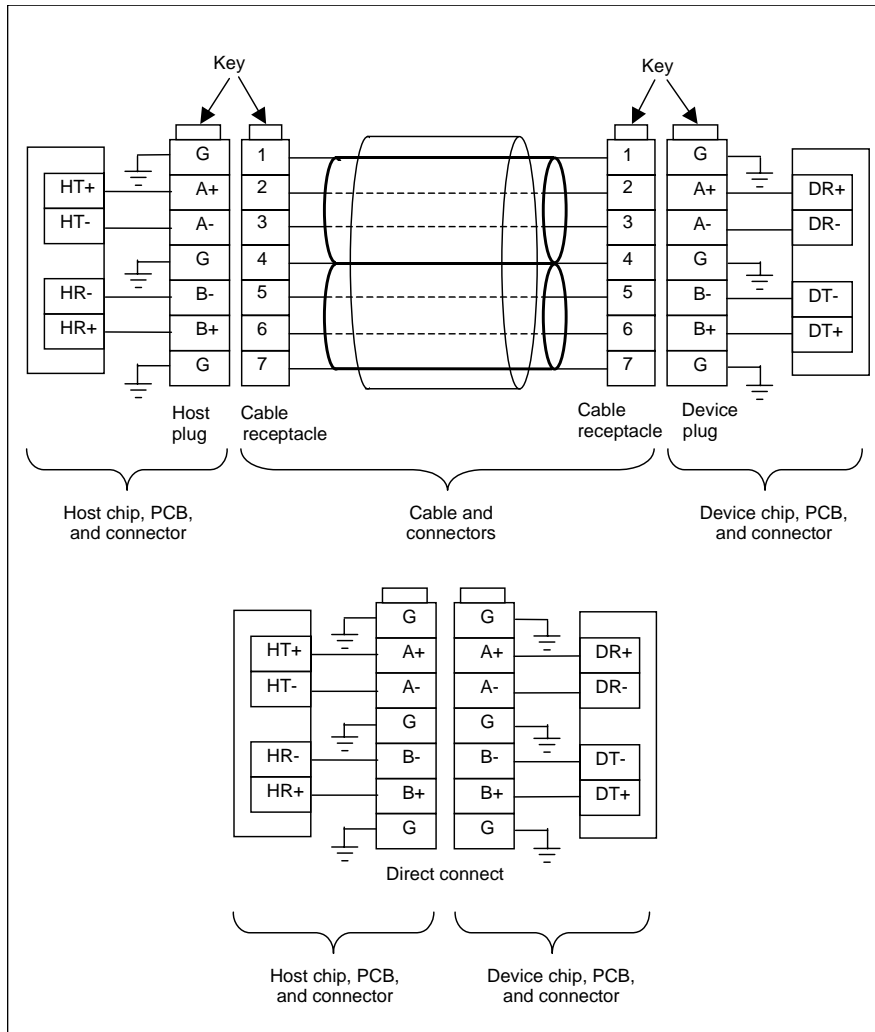


Figure 21 - Signals and grounds assigned in direct connect and cabled

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

5.4 Phy (physical layer electronics)

5.4.1 Block diagram (Informative)

The following block diagram is provided as a reference for the following subclauses of this document. Although informative in nature, the functions of the blocks described in Figure 22 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. Terms including "signal" in the name are used as defined terms in state diagrams and text descriptions to describe physical layer behavior and are not intended to require a specific implementation.

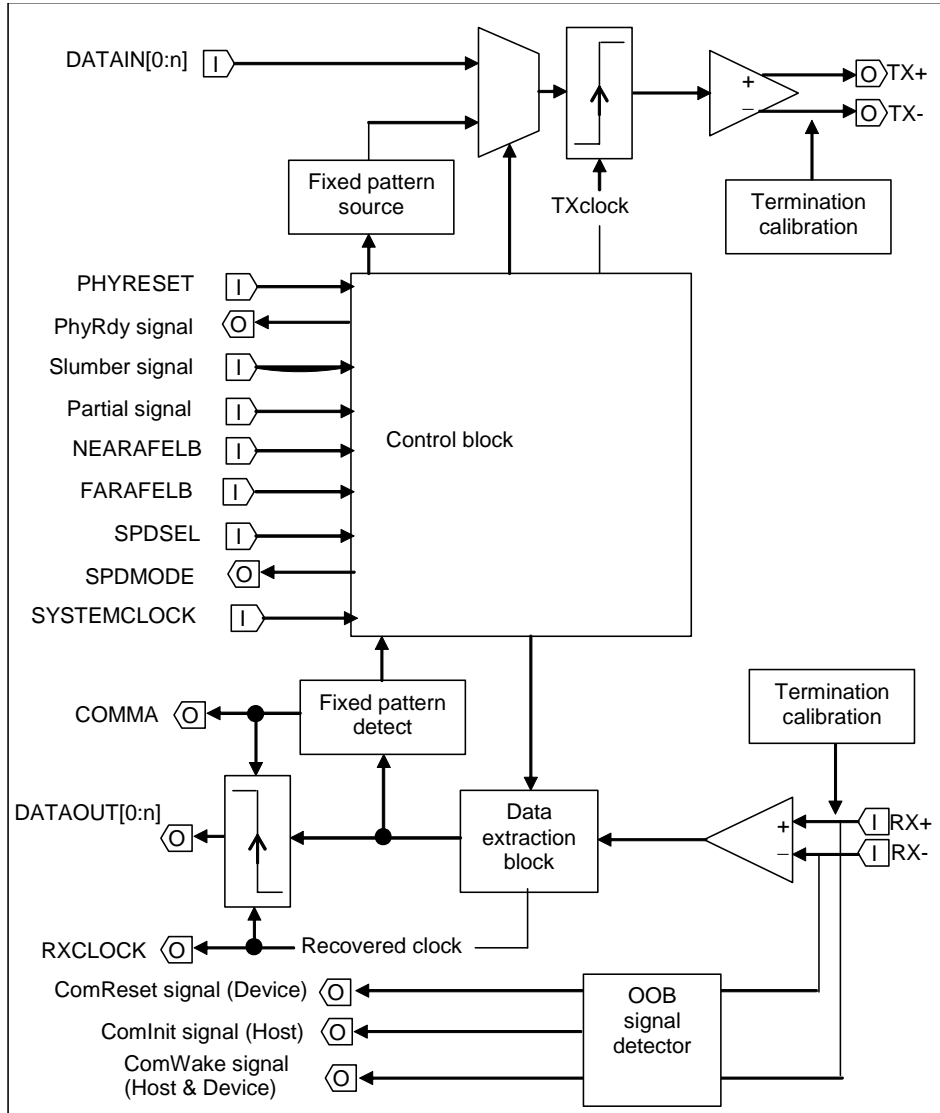


Figure 22 - Physical plant simplified block diagram (Informative)

| | |
|-----------------------|--|
| 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 signaling 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 | Indication 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 signal | Causes the physical layer to transition to the Slumber mode for power management. |
| Partial signal | Causes the physical layer to transition to the Partial mode for power management. |
| 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.

ComInit signal Indication from the host out of band (OOB) detector that ComInit sequence is being detected.

ComReset signal Indication from the device out of band (OOB) detector that the ComReset sequence is being detected.

ComWake signal Indication from the out of band detector (OOB) that the ComWake sequence is being detected.

5.4.2 Electrical specifications

The serial transport physical layer electrical requirements are depicted in Table 12.

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 25 or the transmitter schematic shown in Figure 24 shall be used to design a test fixture.

Table 12 - 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%-80% at transmitter |
| t _{fall} | 0.3 | 0.15 | 0.41 | UI | 80%-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 +2.0V or -2.0V 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 |

(continued)

Table 12 - General electrical specifications (continued)

| | Nom | Min | Max | units | Comments |
|--------------------------------|-----|-----|-----|-------------------|--|
| $V_{cm,ac}$ 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 +2.0V or -2.0V 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 quiescent bus condition. |
| $V_{diff,tx}$ | 500 | 400 | 600 | mV _{p-p} | +/- 250 mV differential nominal. Measured at Serial ATA connector on transmit side |
| $V_{diff,rx}$ | 400 | 325 | 600 | mV _{p-p} | +/- 200 mV differential nominal. Measured at Serial ATA connector on receive side |
| Tx pair differential impedance | 100 | 85 | 115 | Ohm | As seen by a differential TDR with 100 ps (max) edge looking into connector (20%-80%). Measured with TDR in differential mode. |
| Rx pair differential impedance | 100 | 85 | 115 | Ohm | As seen by a differential TDR with 100 ps (max) edge looking into connector (20%-80%). Measured with TDR in differential mode. |

(continued)

Table 12 - General electrical specifications (continued)

| | Nom | Min | Max | units | Comments |
|---|-----|-------|------|-------------------|--|
| Tx single-ended impedance | | 40 | | Ohm | As seen by TDR with 100 ps (max) edge looking into connector (20%-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 | | Ohm | As seen by TDR with 100 ps (max) edge looking into connector (20%-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. |
| C _{ACcoupling} | | | 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 1.500GHz. Excludes the +0/-5000ppm SSC downspread AC modulation per 5.5.9 |
| TX AC clock frequency skew | | -5000 | +0 | ppm | Specifies the allowed ppm extremes for the SSC AC modulation, subject to the "Downspread SSC" triangular modulation (30-33kHz) profile per 5.5.9. Note: Total TX Frequency variation around nominal 1.500G, includes [TXDC] + [TX AC] ppm variations. |
| TX differential skew | | | 20 | ps | (Nominal value architecture specific) |
| Squelch detector threshold | 100 | 50 | 200 | mV _{p-p} | Minimum differential signal amplitude |
| ComReset/ComInit Detector off threshold | | 175 | 525 | ns | Detector shall reject all bursts with spacings outside this spec. |

(continued)

Table 12 - General electrical specifications (continued)

| | Nom | Min | Max | units | Comments |
|---|------------|------------|------------|--------------|--|
| 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 100mV 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 spec. |
| 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 100mV differential crosspoints from last to first edges of bursts |
| U _{I_OOB} | | 646.67 | 686.67 | ps | Operating data period during OOB burst transmission |

(concluded)

5.4.3 Frame error-rate measurements

Frame Error Rates of a system shall be measured and computed, to be no greater than $(8.196 \cdot 10^8)$ frame errors when tested with any given 8b/10b pattern, but shall include the Frame-Error-Rate reference patterns cited in 5.5.4

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 5.5.4.1

5.4.4 Receiver Differential voltage

When subjected to the high-transition density patterns of 5.5.2.3, the Differential Voltage measured at the Receiver shall comply to the electrical specifications of 5.4.2

5.4.5 Receiver Common-mode voltage

References to peak-to-peak voltages are cited in 5.4.2.

The Receiver shall operate to within the bit error rates cited in 5.5.3, when subjected to a sinusoidal interfering signal with peak-to-peak voltage see Table 12 and swept from the frequency range extremes, at a sweep rate period no shorter than 33.33 μ s.

The receiver shall operate to within the frame error rates cited in 5.5.4, 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 μ s.

5.4.6 Transmitter Differential voltage

When subjected to the high-transition density patterns of 5.5.2.3, the Differential Voltage measured at the transmitter shall comply to the respective electrical specifications of 5.4.2.

5.4.7 Transmitter Common-mode voltage

The transmitter shall comply to the electrical specifications of 5.4.2, 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 μ s.

5.4.8 Rise/fall times

Output rise times and fall times are measured between 20% and 80% of the signal, see Signal Rise & Fall Times, Figure 23. Rise and fall time requirements apply to differential transitions, for both In-Band and Out-Of-Band signaling.

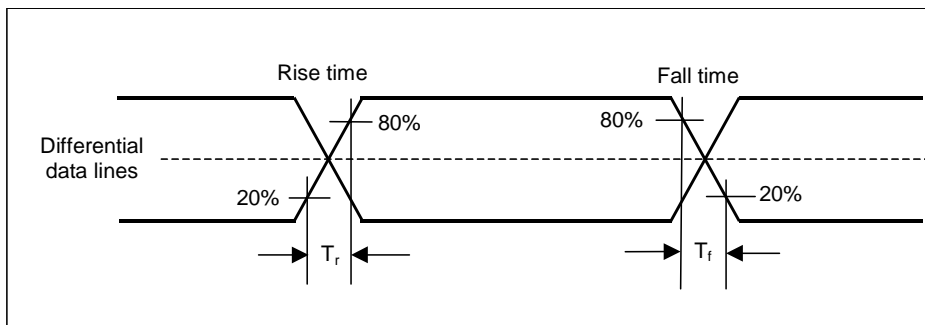


Figure 23 - Signal rise and fall times

The rise and fall times for transmitter differential buffer lines are measured with the load shown in Figure 24, at the transmitter mated connector pair, and shall comply to Electrical Specification paragraph 5.4.2 as well as be matched within $\pm 10\%$ of each other to minimize RFI emissions and signal skew.

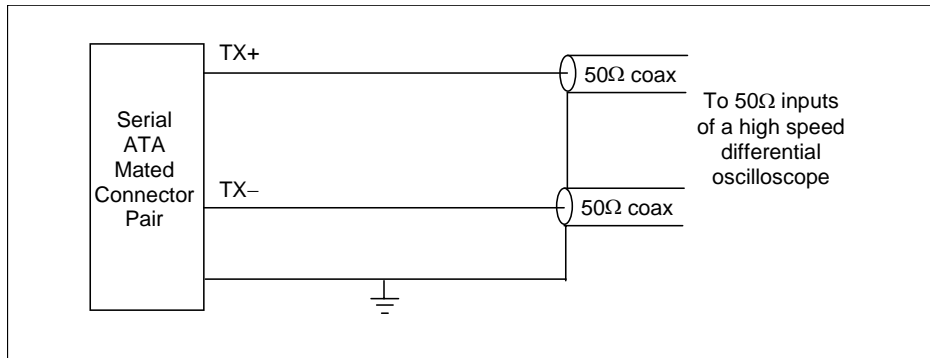


Figure 24 - Transmit test fixture

When testing the serial transport with the specified bit pattern sequences of 5.5.2, use the receiver Test fixture as shown in Figure 25.

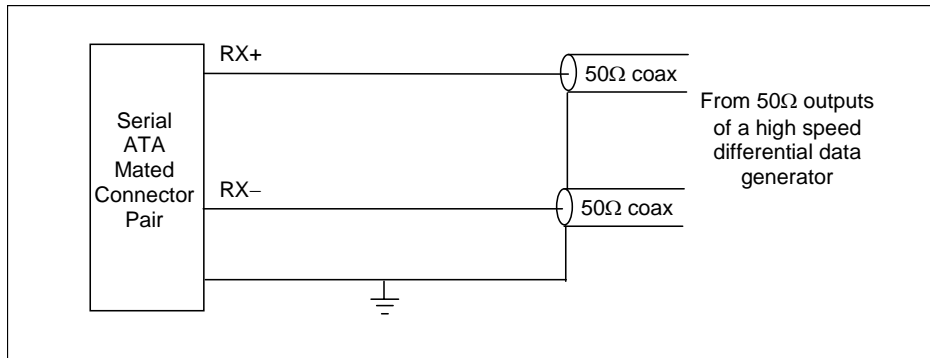


Figure 25 - Receive test fixture

5.5 Electrical features

5.5.1 Definitions

- RJ Random Jitter (peak to peak). Assumed to be Gaussian and equal to 14 times the 1σ rms 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 valid differential signaling starting from and ending with the idle condition on the interface. These are used for low-level signaling when the high-speed communication channel is not established.
- SSC Spread 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 EMI requirements.

5.5.2 Test Methodology and Test Patterns

5.5.2.1 Test bit patterns and sequence characteristics

A number of components of the system make up the system's performance, 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 subclause 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 is defined using a set of reference frames, including specific set of ordered test patterns within the frame.

Test bit sequences are those bit sequences that are transmitted onto a serial link, so as to test the serial transport 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.

Various types of bit sequence patterns emphasize low/high transition density patterns, as well as low/high frequency patterns.

- a) 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 signaling levels.
- b) High transition density patterns are those patterns containing short runs of ones and zeroes, also intended to create inter-symbol interference.
- c) 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.
- d) Simultaneous switching outputs 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 Ldi/dt noise into substrate supply, and are a good test of the receiver circuitry.
- e) 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 transport interfaces. The patterns are classified in two categories:

- a) Non-compliant patterns
- b) 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 subclauses 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.

5.5.2.2 Low transition density bit pattern sequences

Low transition density bit patterns, as shown in Figure 26 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 subclause.

| | | | | | | | |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 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 26 - Low transition density pattern

5.5.2.3 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:

- a) Half-rate high transition density bit pattern sequence
- b) Quarter-rate high transition density bit pattern sequence

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

| | | | | | | | | |
|---|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| - | D21.5(-) B5h | D21.5(-) B5h | D21.5(-) B5h | D21.5(-) B5h | D21.5(-) B5h | D21.5(-) B5h | D21.5(-) B5h | D21.5(-) B5h |
| | 101010 1010 | 101010 1010 | 101010 1010 | 101010 1010 | 101010 1010 | 101010 1010 | 101010 1010 | 101010 1010 |
| Repeat n times (n>12) | | | | | | | | |
| | | | | | | | | |
| - | | | | | | | | |
| | D24.3(-) 78h | D24.3(+) 78h | D24.3(-) 78h | D24.3(+) 78h | D24.3(-) 78h | D24.3(+) 78h | D24.3(-) 78h | D24.3(+) 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(-) 4Ah | D10.2(-) 4Ah | D10.2(-) 4Ah | D10.2(-) 4Ah | D10.2(-) 4Ah | D10.2(-) 4Ah | D10.2(-) 4Ah | D10.2(-) 4Ah |
| | 010101 0101 | 010101 0101 | 010101 0101 | 010101 0101 | 010101 0101 | 010101 0101 | 010101 0101 | 010101 0101 |
| Repeat n times (n>12) | | | | | | | | |
| | | | | | | | | |
| - | | | | | | | | |
| | D25.6(-) D9h | D6.1(-) 26h | D25.6(-) D9h | D6.1(-) 26h | D25.6(-) D9h | D6.1(-) 26h | D25.6(-) D9h | D6.1(-) 26h |
| | 100110 0110 | 011001 1001 | 100110 0110 | 011001 1001 | 100110 0110 | 011001 1001 | 100110 0110 | 011001 1001 |
| Repeat n times (n>12) | | | | | | | | |

Figure 27 - Half-rate / quarter-rate high transition density pattern

5.5.2.4 Low frequency spectral content bit pattern sequences

Bit patterns that contain low frequency spectral components 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 | D11.5(-) ABh | D11.5(-) ABh | D11.5(-) ABh | D11.5(-) ABh | D11.5(-) ABh | D11.5(-) ABh | D11.5(-) ABh |
| | 110100 1010 | 110100 1010 | 110100 1010 | 110100 1010 | 110100 1010 | 110100 1010 | 110100 1010 | 110100 1010 |
| Repeat n times (n>12) | | | | | | | | |
| | | | | | | | | |
| - | | | | | | | | |
| | D11.7(-) EBh | D20.2(+) 54h | D20.2(+) 54h | D20.2(+) 54h | D20.2(+) 54h | D20.2(+) 54h | D20.2(+) 54h | D20.2(+) 54h |
| | 110100 1110 | 001011 0101 | 001011 0101 | 001011 0101 | 001011 0101 | 001011 0101 | 001011 0101 | 001011 0101 |
| byte group (D30.3(+), D30.3(-)) repeat n times (n>12) | | | | | | | | |
| | | | | | | | | |
| - | | | | | | | | |
| | D20.2(+) 54h | D20.2(+) 54h | D20.2(+) 54h | D20.7(+) F4h | D11.5(-) ABh | D11.5(-) ABh | D11.5(-) ABh | D11.5(-) ABh |
| | 001011 0101 | 001011 0101 | 001011 0101 | 001011 0001 | 110100 1010 | 110100 1010 | 110100 1010 | 110100 1010 |

Figure 28 - Low frequency spectral content pattern

5.5.2.5 Simultaneous switching outputs bit pattern sequences

Simultaneous switching outputs bit pattern sequences induce Ldi/dt 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 | D31.3(+) 7Fh | D31.3(-) 7Fh | D31.3(+) 7Fh | D31.3(-) 7Fh | D31.3(+) 7Fh | D31.3(-) 7Fh | D31.3(+) 7Fh |
| | 101011 0011 | 010100 1100 | 101011 0011 | 010100 1100 | 101011 0011 | 010100 1100 | 101011 0011 | 010100 1100 |

Repeat n times (n>512)

Figure 29 - Simultaneous switching outputs patterns

5.5.2.6 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 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.

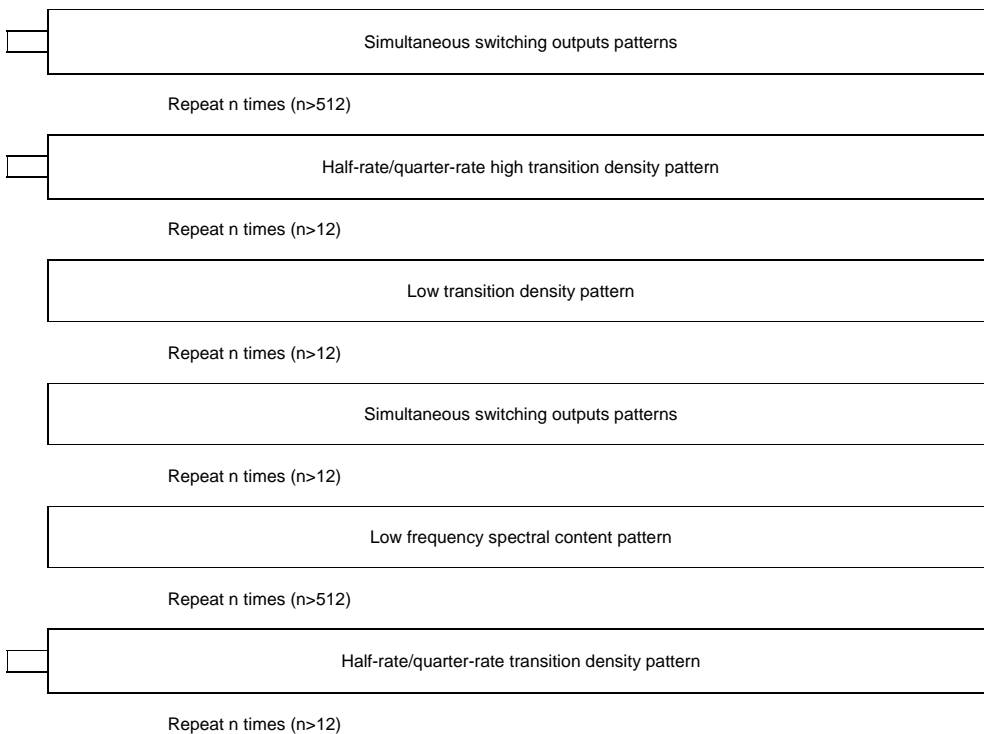


Figure 30- Composite patterns

5.5.3 Bit error rate testing - Informative

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 5.6), 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:

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

5.5.3.1 Error-burst-rate-thresholding measurement - Informative

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-minute period, when tested with the compliant frame patterns, defined by 5.5.2.2, 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-minute period. The required accuracy of the 5-minute period mentioned above is +/-0.3 seconds. Reaching the threshold error count of 15 in a 5-minute 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-minute period should therefore not be used as a measure of system compliance.

The error-burst-rate counting process will be restarted when the PhyRdy signal is asserted, 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-minute time period has expired even though an error-burst-rate threshold is not reached.

5.5.3.2 Bit-error-rate measurements - Informative

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 5.5.2.1

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

5.5.4 Frame error rate testing

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 the 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.

5.5.4.1 Frame error-rate patterns

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 transport 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 5.5.2.6, but with the N parameters extended so as to achieve the maximum frame length. Refer to subclause 5.7.1 for a description of Loopback Testing configurations.

5.5.4.1.1 Loopback test

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

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

5.5.5 Test requirements - non-compliant patterns

Electrical parameters of 5.4.2 shall be verified using the following set of test patterns, using the BIST Activate FIS, Far-End Transmit Mode, described in 7.5.7:

- a) Low transition density bit patterns, as per 5.5.2.2.
- b) High transition density bit patterns as per 5.5.2.3
- c) Low frequency spectral component bit patterns as per 5.5.2.4.
- d) Simultaneous switching outputs bit patterns as per 5.5.2.5.

5.5.6 Test requirements - compliant frame patterns

The frame error rates specified in 5.4.3 shall be tested for compliance when subjected to any implementation-determined worst-case compliant patterns, as well as the following set of compliant patterns:

- a) Compliant low transition density bit patterns, as per 5.5.2.2
- b) Compliant high transition density bit patterns as per 5.5.2.3
- c) Compliant low frequency spectral component bit patterns as per 5.5.2.4
- d) Compliant simultaneous switching outputs bit patterns as per 5.5.2.5
- e) Compliant composite patterns as per 5.5.2.6

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 transport 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 contain the necessary SOF leading primitive, the calculated CRC, and the trailing EOF primitive, as shown in the figure below:

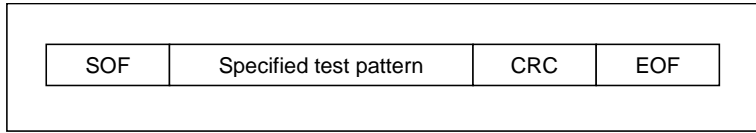


Figure 31- Compliant test patterns

5.5.7 Test requirements - loopback

As specified in 5.7.1, 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.

5.5.7.1 Test requirements - 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 Activate FIS described in 7.5.7.

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 1000 ms at Gen1 speed, using the following set of compliant reference frame patterns if the test pattern generation is supported by the initiating device:

- a) Compliant composite reference frame patterns as specified in 5.5.4.1.
- b) Compliant low transition density bit patterns, as per 5.5.2.2
- c) Compliant high transition density bit patterns as per 5.5.2.3
- d) Compliant low frequency spectral component bit patterns as per 5.5.2.4
- e) Compliant simultaneous switching outputs bit patterns as per 5.5.2.5
- f) Compliant composite patterns as per 5.5.2.6

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 transport operational transmission context.

The Far-End Interface shall remain in this Far-End Retimed Loopback, until receipt of the COMRESET or COMINIT..

5.5.7.2 Test requirements - loopback - far-end analog (vendor specific)

The test requirements for this Loopback scheme are vendor specific.

5.5.7.3 Test requirements - loopback - near-end analog (vendor specific)

The test requirements for this Loopback scheme, are vendor specific.

Deleted: <#>Test Method - Data Rate Frequency Variation - SSC Profile¶

Deleted: HFTP and MFTP (editors note: ?)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 Activate 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 Gen1 (TBC-250) (Editors note: what is this?), and Gen2 (TBC-250), 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 12.¶

¶ 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. ¶

5.5.8 Differential voltage/timing (EYE) diagram

The EYE diagram is more of a qualitative measurement than a spec. 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 t_3-t_1 is set equal to $DJ+6 \cdot RJ_{1\sigma}$, 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.

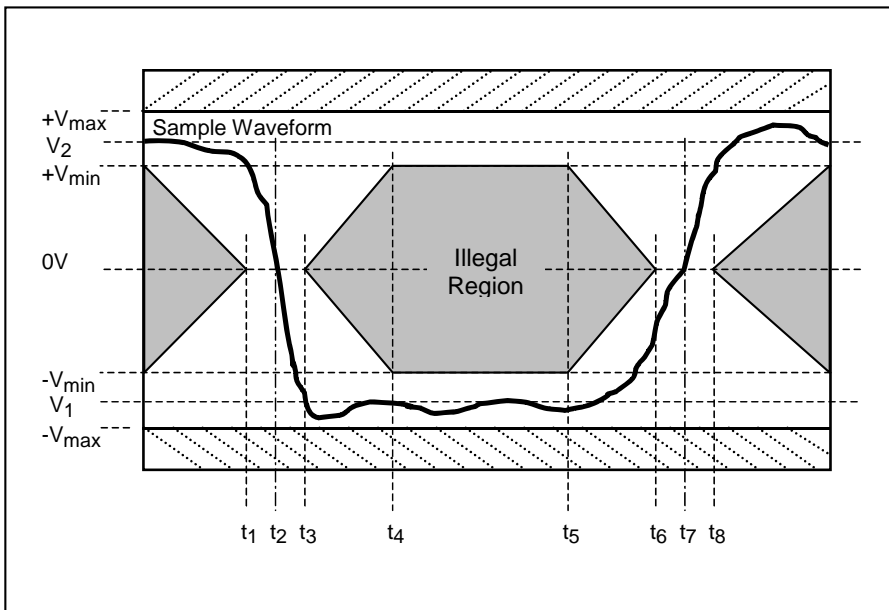


Figure 32 - Voltage / timing margin base diagram

Table 13 - 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} | $V_2 - V_1$ | |

5.5.8.1 Jitter output/tolerance mask

The spectral jitter shall comply to the requirements as indicated in the Jitter Output/Tolerance Graph, shown in Figure 33 with magnitudes defined in Table 14. 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 $n_x \cdot UI$ later (where x is 0,1, or 2). See annex C.4.

Transmitter output data edge to data edge timing variation from t_0 to $t_{y \text{ shall}}$ 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 $y \cdot UI$ 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 5.4. A receiver shall meet the error rate requirement under the maximum allowed jitter conditions found in Table 13 and Table 14.

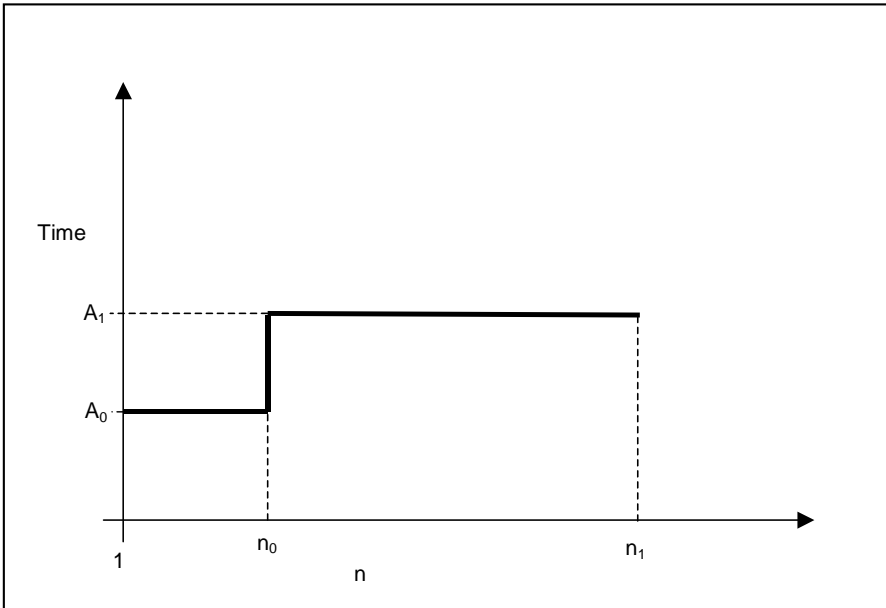


Figure 33 - Jitter output/tolerance mask

5.5.8.2 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 $V_{\text{differential}}$ 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). See 5.5.1.

The serial transport jitter characteristics should comply to within the jitter budget allocations of Table 14.

Table 14 - 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 | |
| $A_{0,p-p}$ (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 |
| $A_{1,p-p}$ (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 |
| NOTES – 1. The driver output is the maximum jitter that a driver may exhibit to guarantee operation. 2. This field is the maximum jitter that a receiver shall tolerate to guarantee operation. 3. Does not include UI error due to frequency skew (XTAL or SSC related) 4. Primarily determined by non-tracking architecture requirements 5. Primarily determined by tracking architecture requirements | | | | | | | | | |

5.5.8.3 Jitter output

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

5.5.8.3.1 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 34 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 34 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 .

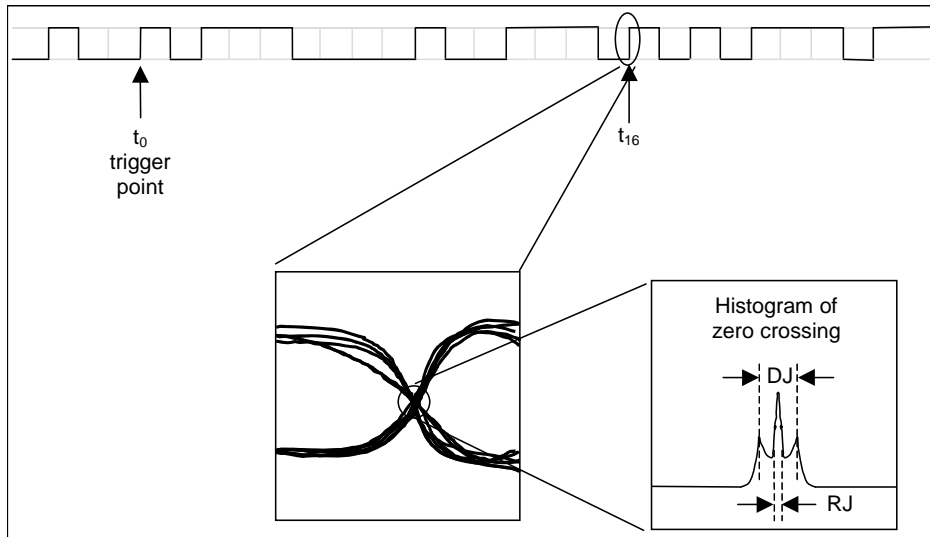


Figure 34 - Jitter measurement example

5.5.9 Spread spectrum clocking (SSC)

Spread Spectrum functionality define follows

- a) 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.
- b) 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 35. The modulation profile in a modulation period can be expressed as:

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

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.

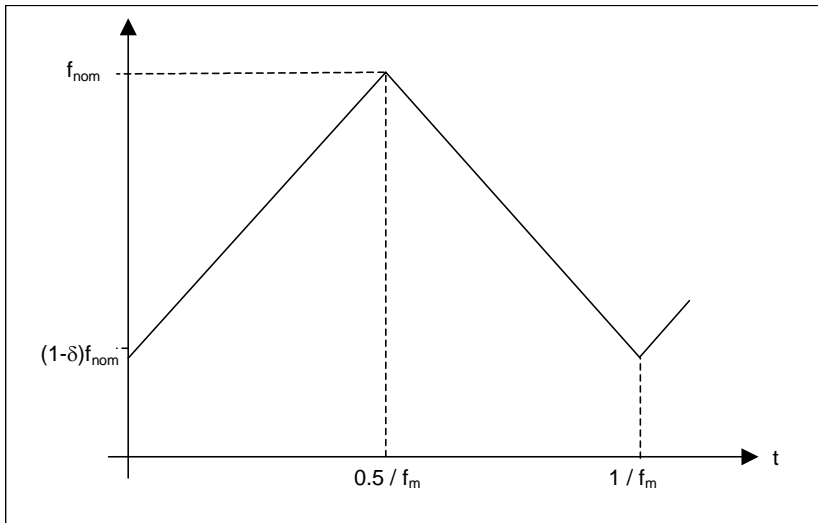


Figure 35 - Triangular frequency modulation profile

- c) 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%/-0.5%. The absolute spread amount at the fundamental frequency is shown in Figure 36, 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.

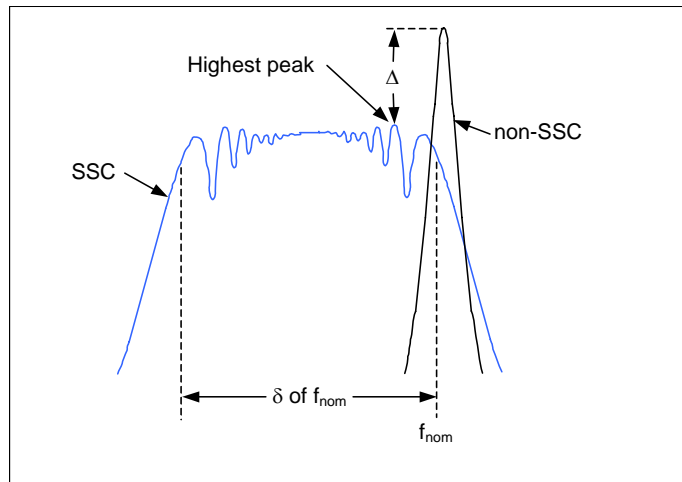


Figure 36 - Spectral fundamental frequency comparison

- d) To achieve 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 15. The peak reduction Δ is defined, as shown in Figure 36, as the difference between the spectral peaks in SSC and non-SSC modes at the specified measurement frequency

Table 15 - Desired peak amplitude reduction by SSC

| Clock freq. | Peak reduction Δ | Measurement freq. |
|--------------------|---|-------------------------------------|
| 66 MHz | 7 dB | 466 MHz (7 th harmonics) |
| 75 MHz | 7 dB | 525 MHz (7 th harmonics) |

NOTES –

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.

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 to 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.

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

5.5.10 Common-mode biasing

The serial transport 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 12) is not required to sustain the cable side of the its blocking capacitors. Similarly, an AC-coupled transmitter (See Table 12) is not required to sustain the common-mode level on the cable side of its blocking capacitors.

5.5.11 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 12.

Device peripherals shall provide impedance terminations, as per the specified parameters in Table 12, 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 COMWAKE, COMINIT, or COMRESET sequences. Signals generated for the impedance calibration process shall not exceed the normal operating voltage levels, cited in 5.4.2. See the power management subclause for suggested times to perform calibration during power-on.

5.5.12 Out of band signaling

See Figure 38 for a description of COMRESET, ComReset Sequence, and ComReset signal definitions; COMINIT, ComInIt Sequence, and ComInIt signal; COMWAKE, ComWake Sequence, and ComWake signal. Transmission of ComInIt Sequence, ComReset Sequence and ComWake Sequences are composed of bursts of ALIGN primitives, each burst being 160 U_{IOOB}. Each burst is followed by quiescent periods (at common-mode levels), having durations as depicted in Figure 37 and Table 16.

During OOB signaling transmissions, the differential and common mode levels of the signal lines shall comply to the same electrical specifications as for in-band data transmission, specified in 5.4.2. The ComReset Sequence, ComInIt Sequence, and ComWake Sequence are shown In Figure 37. OOB sequences 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 12 shall negate the ComWake signal output. The ComWake Sequence is used to bring the Phy out of a power-down state (Partial or Slumber) as described in 5.5.12.3.8. The interface shall be held inactive for at least the maximum ComWake detector off threshold in Table 12 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 228.3ns) at the end of a ComWake Sequence to prevent susceptibility to crosstalk.

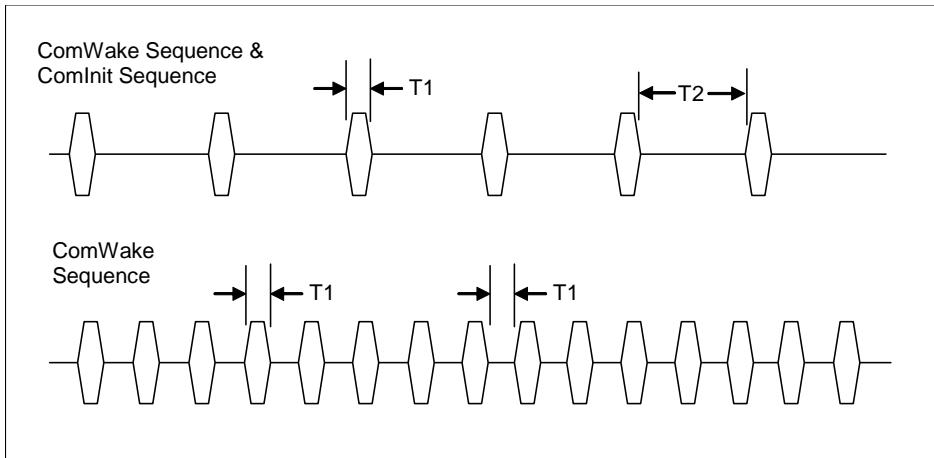


Figure 37 - Out of band signals

Table 16 – Out of band signal times

| Time | Value |
|-------------|----------------------------------|
| T1 | 160 U_{lob} (106.7 ns nominal) |
| T2 | 480 U_{lob} (320.0 ns nominal) |

5.5.12.1 Idle bus status

During the quiescent 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 5.4.2, and shall settle to within 25 mV of $V_{cm,D.C.}$ within $T_{settle,CM}$, cited in 5.4.2. Annex C shows several transmitter examples, and how the transition to and from the idle state may be implemented.

5.5.12.2 Phy Layer OOB Sequences (Informative)

Figure 38 shows the relationship between COMRESET, ComReset signal, COMINIT, ComInit Signal, COMWAKE, ComWake signal (host), and ComWake signal (Device). When the term COMRESET is used with all upper case letters, it refers to the host initiated hardware reset sequence from the issuance of a ComReset Sequence until the host PhyRdy signal is asserted (see 5.5.12.3). When the term COMINIT is used with all upper case letters, it refers to a device initiated hardware reset sequence from the issuance of ComInit Sequence until device the PhyRdy signal is asserted (see 5.5.12.3.1). When the term COMWAKE is used with all upper case letters, it refers to a host or device initiated wakeup sequence from the issuance of a ComWake Sequence until the device PhyRdy signal is asserted. When the terms ComReset Sequence, ComInit Sequence, or ComWake Sequence are used, they refer to the sequence of transmission bursts and quiescent gaps including negation times as defined in subclause 5.5.12. The terms ComWake signal, Cominit signal, and ComWake signal refer to the physical layer detector outputs (see Figure 22).

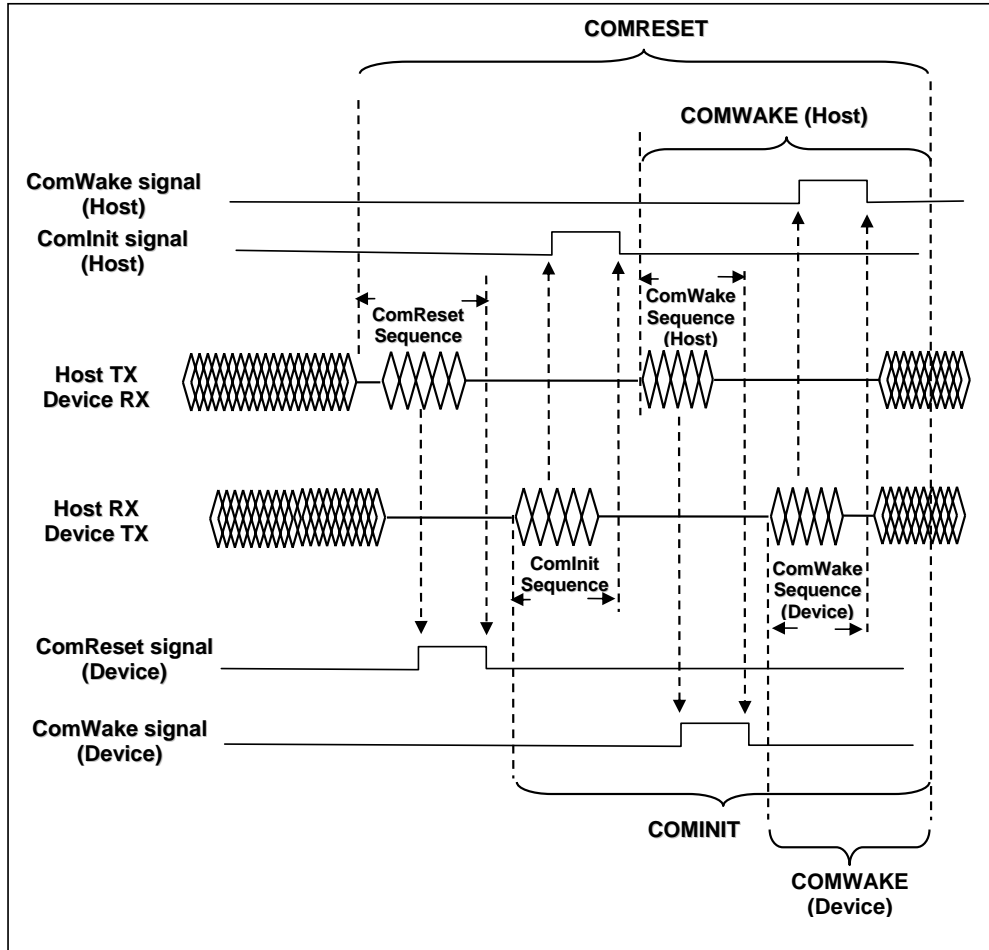


Figure 38 - OOB Sequences and Signals (Informative)

5.5.12.3 Host Phy initialization state machine

The following state diagrams specify the behavior of the host and device Phy from power-on, ComReset Sequence or ComInit Sequence (see 5.5.12) to the establishment of an active communication channel.

In those states where the Phy relies on detection of received ALIGN primitives see subclause 6.4.4. Reception of a ComInit signal (see Figure 22) from the Phy shall cause the host to reinitialize communications with the device and the host Phy shall transition to the HP3:HR_AwaitNoCOMINIT state regardless of other conditions. Reception of ComInit signal from the Phy is effectively an additional transition into the HP3:HR_AwaitNoCOMINIT state that appears in every host Phy Initialization state machine state. For the sake of brevity, this implied transition has been omitted from all the states.

(Editors Note: The state machine should be called physical layer State Machine, and is the same as Phy, many terms used host Phy state machine, etc)

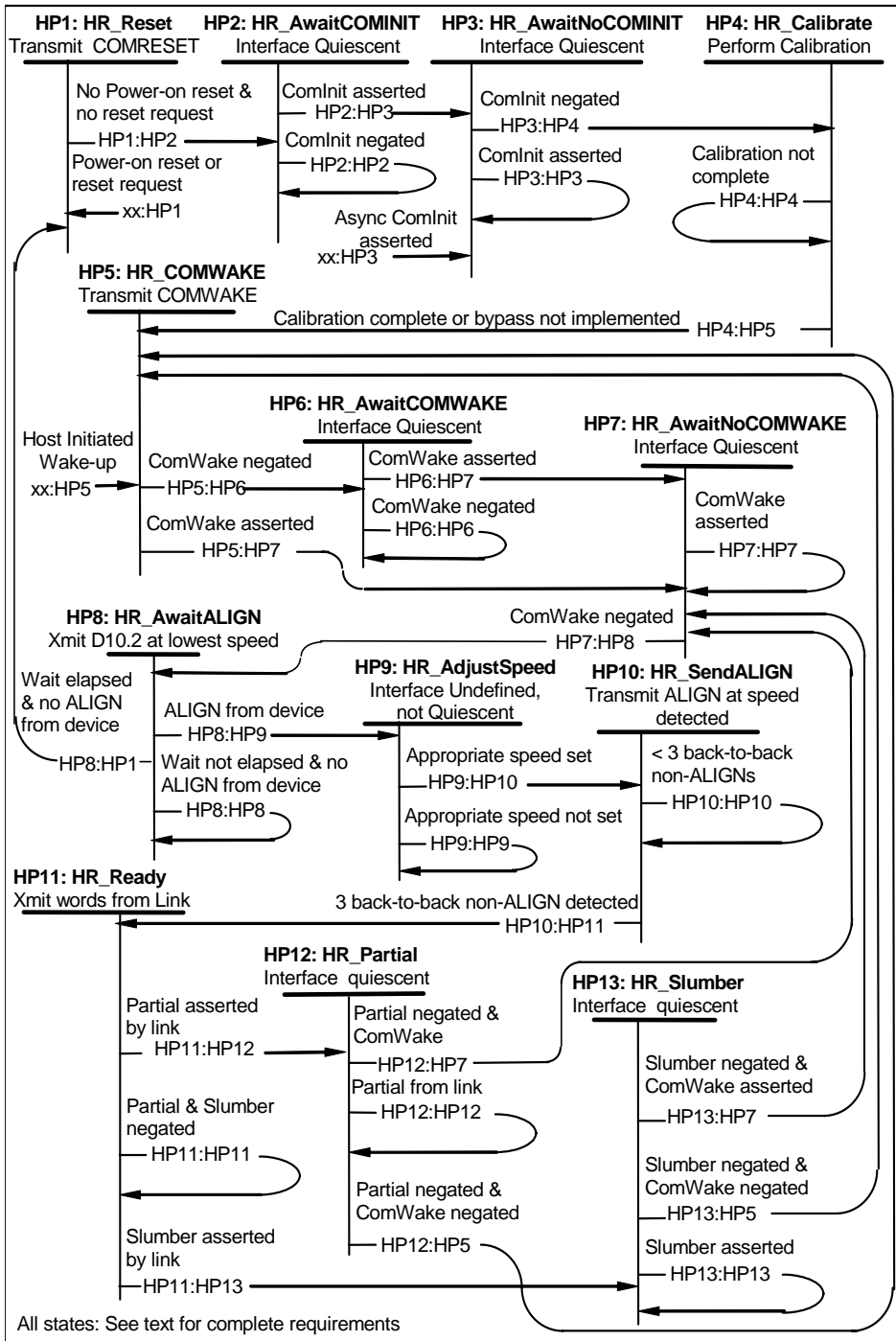


Figure 39 – Host Phy initialization state machine (States HP1-HP13)

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 (see 5.5.12) may be transmitted for the duration of this state, may be transmitted during this state and cease transmission 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 5.5.12.3.2.1.

Transition HP1:HP2: If 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: If 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 Sequence (see 5.5.12) to be detected.

Transition HP2:HP2: If the Phy does not detect a ComInit Sequence, the physical layer shall negate the ComInit signal and make a transition to the HP2: HR_AwaitCOMINIT state.

Transition HP2:HP3: If a ComInit Sequence is detected by the Phy, the physical layer shall assert the ComInit signal and make a transition to the HP3: HR_AwaitNoCOMINIT state.

HP3: HR_AwaitNoCOMINIT state: This state is entered when the ComInit signal (see Figure 22) is asserted while the physical layer is in the HR2: HR_AwaitCOMINIT state. It is also entered asynchronously at any time in response to the receipt of a valid ComInit Sequence that results in the ComInit signal being asserted, 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 the ComInit signal from the Phy to be negated.

Transition HP3:HP3: If the ComInit signal is asserted, the physical layer shall make a transition to the HP3: HR_AwaitNoCOMINIT state.

Transition HP3:HP4: If the ComInit signal is negated, 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 from Slumber or Partial.

When in this state the physical layer shall transmit the ComWake Sequence.

Transition HP5:HP6: After transmitting the ComWake Sequence (see 5.5.12), if the ComWake signal (see Figure 22) is negated, the physical layer shall make a transition to the HP6: HR_AwaitCOMWAKE state.

Transition HP5:HP7: After transmitting the ComWake Sequence, if 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 negated, 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: If 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 transmitting D10.2 characters at the lowest rate no later than 533ns (20 nominal Gen1 dword Times) after the ComWake signal is negated (See 5.5.12).

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 max, see Table 12) 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 max (See Table 12), such a host should wait at least 116.3 ns (the ComWake Detector off threshold max, plus 2 nominal Gen1 dword times, plus ComWake Detector on threshold max (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 us (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: If the ComWake signal is negated and less than 873.8 us (32768 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 transition has completed.

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. (editors note: the last sentence implies it is ok to only look for a k28.3 character, is that correct?)

Transition HP10:HP10: If zero, one, or two consecutive non-ALIGN primitives have 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 from HP10 when three 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. The PhyRdy signal 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: If the Partial signal (see Figure 22) is asserted from the link layer, the physical layer shall make a transition to the HP12: HR_Partial state.

Transition HP11:HP13: If the Slumber signal (see Figure 22) is asserted from the link layer, the physical layer shall make a transition to the HP13: HR_Slumber state.

Transition HP11:HP11: If the Partial and Slumber signals are negated 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 Partial. 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 negated, 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 link layer negation of the Partial signal, or from a Phy layer ComWake Sequence that caused the ComWake signal to be asserted. (See state DP4 and DP6 5.5.12.3.1). (editors note, fix link)

Transition HP12:HP7: If 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: If 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 if the Slumber signal is asserted from the link layer.

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

Transition HP13:HP5: If the Slumber signal is negated from the link layer and the ComWake signal is asserted, the physical layer shall make a transition to the HP5: HR_COMWAKE state. The host Phy layer shall remember if the ComWake signal was detected during Slumber to determine if the wakeup request originated from the link layer negation of the Slumber signal, or from a Phy layer ComWake Sequence that caused the ComWake signal to be asserted. (See state DP4 and DP6 5.5.12.3.1) (editors note: fix link).The host Phy layer may (editors note: shall?) take this transition only after it has recovered from Slumber mode and the Phy is prepared to initiate communications. If the Phy layer has not yet recovered from the Slumber mode it shall remain in this state.

Transition HP13:HP7: If the Slumber 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 layer shall remember if the ComWake signal was detected during Slumber to determine if the wakeup request originated from the link layer negation of the Slumber signal, or from a Phy layer ComWake Sequence that caused the ComWake signal to be asserted. (See state DP4 and DP6 5.5.12.3.1) (editors note: fix link).The host Phy layer 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: If the Slumber signal is asserted from the link layer, the physical layer shall make a transition to the HP13: HR_Slumber state.

5.5.12.3.1 Device power-up and Phy Layer state machine

Reception of a valid ComReset Sequence (see 5.5.12) results in the ComReset signal (see Figure 22) being asserted and 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.

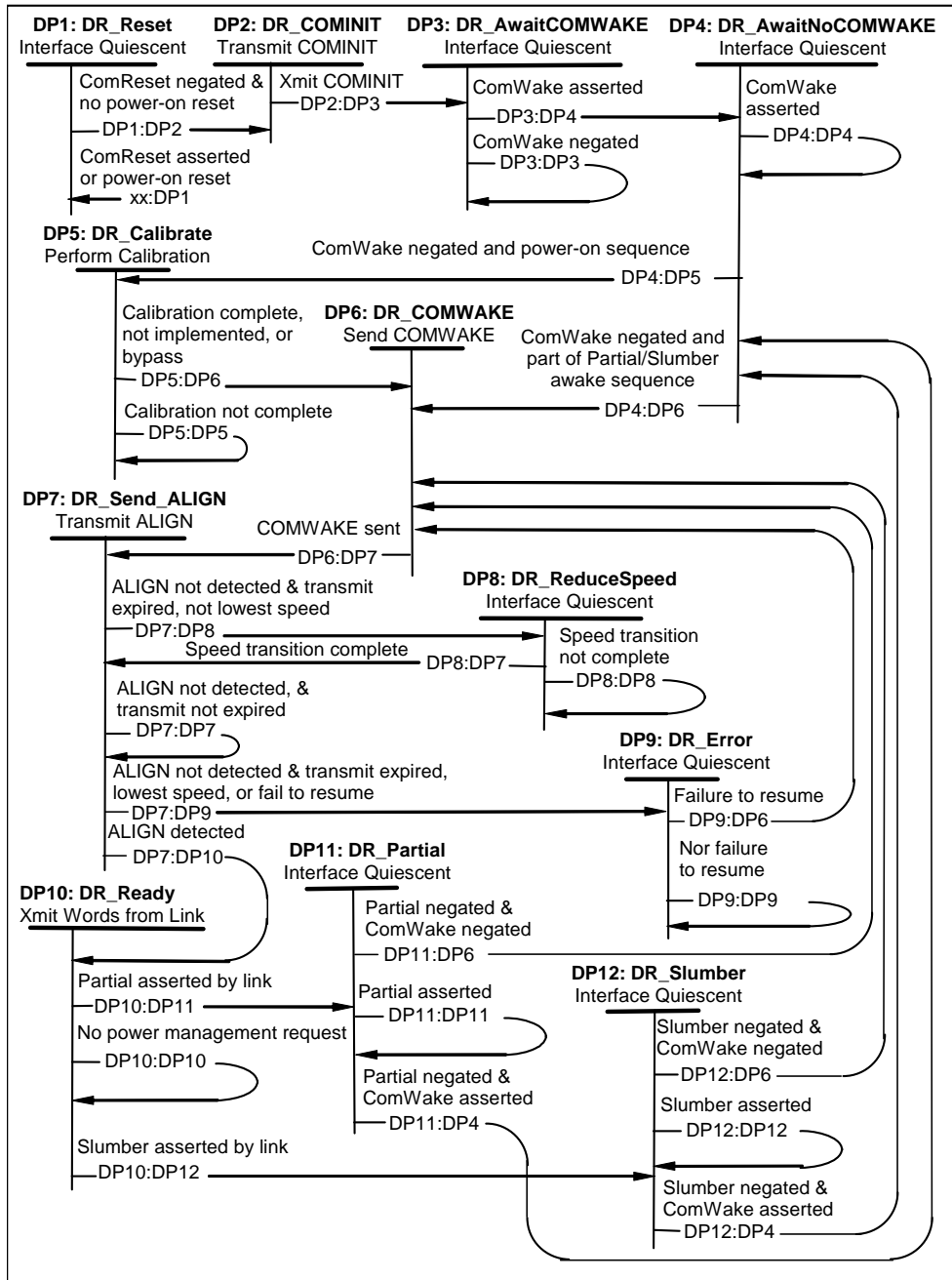


Figure 40 – Device initialization and Phy layer state machine (States DP1-DP12)

DP1: DR_Reset state: This state is entered asynchronously at power-on or when the ComReset Sequence (see 5.5.12) is detected from the host and the ComReset signal is asserted.

Transition DP1:DP2: If the ComReset signal (see Figure 22) is negated and power-on reset is negated, the physical layer shall make a transition to the DP2: HR_COMINIT state.

Transition xx:DP1: If 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 negated and power-on reset is negated.

When in this state the physical layer transmits the ComInit Sequence (see 5.5.12). 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 (see Figure 22) to be asserted.

Transition DP3:DP3: If the ComWake signal is negated, the physical layer shall make a transition to the DP3: DR_AwaitCOMWAKE state.

Transition DP3:DP4: If the ComInit signal is 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: If the ComWake signal has been negated and a 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 State or Slumber State for proper wakeup action. Calibration is optional. If bypassed or not implemented, proceed directly to DP6: DR_COMWAKE.

Transition DP4:DP6: If the ComWake signal has been negated and a return from Partial State or Slumber State 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 State or Slumber State 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 is 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 (see 5.5.12).

Transition DP6:DP7: After transmission of the ComWake Sequence, the physical layer shall 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 negated as specified in 5.5.12, 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 max (See Table 12) to negate the ComWake signal.

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 μ s (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 μ s (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 the ComWake signal. If this is part of device initiated resume from Slumber State or Partial 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 μ s (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 max (See Table 12) to negate the ComWake signal. If this is part of device initiated resume from the Partial State or Slumber 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 the ComWake signal 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 μ s (2048 nominal Gen1 dword Times), for a maximum total of 109.2 μ s, 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 μ s (2048 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 signaling 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 μ s (2048 nominal Gen1 ALIGN primitives) at the lowest speed, or the previous negotiated speed failed to resume from a power state.

Transition DP9:DP9: If the error is not due to failure to resume from Partial state or Slumber state, 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 from Slumber state or Partial state, 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. The PhyRdy signal 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: If 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: If 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: If the Partial signal and Slumber signal are negated 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 mode. The interface is quiescent in this state. (editors note, last sentence added)

Transition DP11:DP6: If the Partial signal is negated from the device link layer to initiate a resume from Partial mode, and the ComWake signal is asserted, the physical layer shall make a transition to the DP6: DR_COMWAKE state.

Transition DP11:DP4: If the Partial signal is negated from the device link layer to initiate a resume from Partial mode and the ComWake signal is asserted to initiate a resume from Partial State, the physical layer shall make a transition to the DP4: DR_AwaitNoCOMWAKE state.

Transition DP11:DP11: If 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 mode. The interface is quiescent in this state. (editors note, last sentence added)

Transition DP12:DP6: If the Slumber signal is negated from the device link layer to initiate a resume from Slumber mode and the ComWake signal is negated, the physical layer shall make a transition to the DP6: DR_COMWAKE state.

Transition DP12:DP4: If the Slumber signal is negated from the device link layer to initiate a resume from Slumber mode and the ComWake signal is negated, the physical layer shall make a transition to the DP4: DR_AwaitNoCOMWAKE state.

Transition DP12:DP12: If the Slumber signal is asserted from the link layer, the physical layer shall make a transition to the DP12: DR_Slumber state.

5.5.12.3.2 Power-up and COMRESET

5.5.12.3.2.1 COMRESET Description

COMRESET always originates from the host controller, and forces a hard reset in the device. It is indicated by transmitting a ComReset Sequence consisting of bursts (See Subclause 5.5.12) separated by gaps of quiescent bus condition.

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

- a) sustained/continued uninterrupted as long as the hard reset is asserted, or
- b) started during the system hard reset and ended some time after the deassertion of system hard reset, or
- c) 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 Sequence 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 min time (See Table 12) or greater than the ComReset/ComInit Detector off threshold max time shall negate the ComReset Detector output. The PHYRESET signal (editors note: changed from COMRESET signal, undefined) to the Phy layer will initiate the reset sequence shown Figure 41. The interface shall be held inactive for at least the ComReset/ComInit Detector off threshold max time after the last burst to ensure far-end detector detects the deassertion properly.

Figure 41 is provided for clarity and is informative.

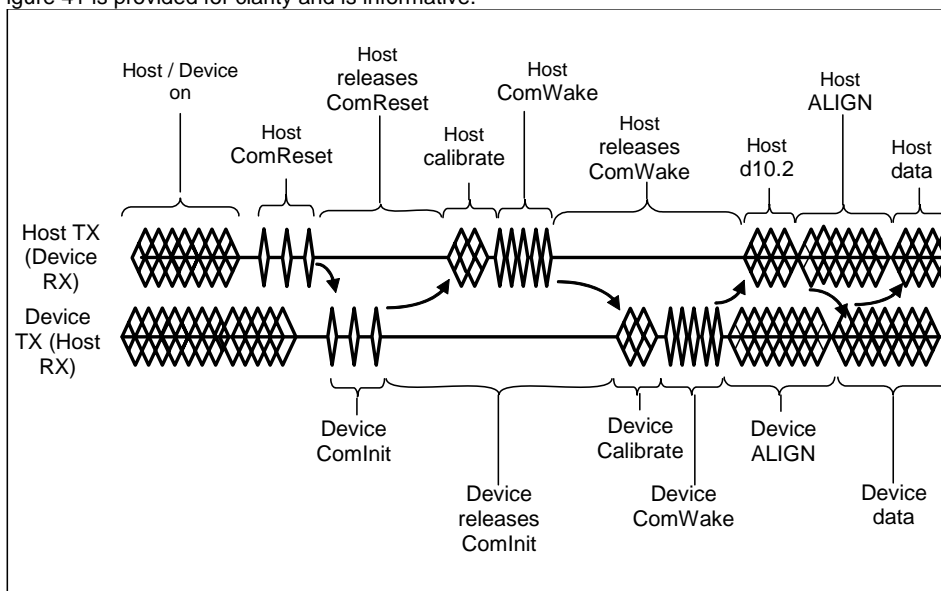


Figure 41 - COMRESET Description

Description:

1. Host/device are powered and operating normally with some form of active communication.
2. Some condition in the host causes the host to assert the PHYRESET signal.
3. Once the condition causing the PHYRESET is negated and the ComReset Sequence is completed, the host puts the bus (transmitter) in a quiescent condition.
4. When the device detects the negation of the ComReset signal, it responds with a ComInit Sequence. The device may initiate communications at any time by issuing a ComInit Sequence.
5. Host calibrates and transmits a ComWake Sequence.
6. Device responds - The device detects the ComWake Sequence on its RX pair, asserts the ComWake signal, and calibrates its transmitter (optional). Following calibration, the device sends a six burst ComWake Sequence and then sends a continuous stream of the ALIGN primitives starting at the device's highest supported speed. After ALIGN primitives have been sent for 54.6us (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 primitives at that rate for 54.6us (2048 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.
7. Host locks - After the assertion of the ComWake signal, the host starts transmitting D10.2 characters (See 5.6) at its lowest supported rate. Meanwhile, the host receiver locks to the ALIGN primitives and, when ready, transmits ALIGN primitives to the device at the same speed as received. A host shall be designed such that it can acquire lock in 54.6us (2048 nominal Gen1 dword times) at any given speed. The host should allow for at least 873.8us (32768 nominal Gen1 dword times) after detecting the negation of the ComWake signal to receive the first ALIGN primitive. This will ensure interoperability with multi-generational and synchronous devices. If no ALIGN is received within 873.8us (32768 nominal Gen1 dword times), the host restarts the power-on sequence - repeating until told to stop by the application layer.
8. Device locks - The device locks to the ALIGN sequence and, when ready, sends the SYNC primitive indicating it is ready to start normal operation.
9. Upon receipt of three back-to-back non-ALIGN primitives, the communication link is established and normal operation may begin.

5.5.12.3.3 COMINIT Description

A device initiated COMINIT originates from the drive and initiates a communication initialization by transmitting a ComInit Sequence. 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.

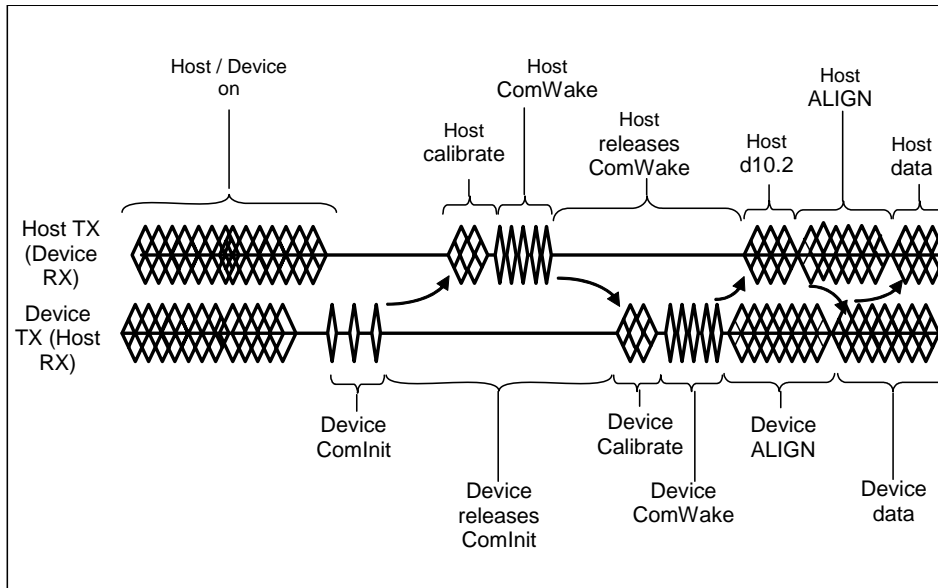


Figure 42 – Device Initiated Cominit Sequence

Description:

1. Host/device are powered and operating normally with some form of active communication.
2. Some condition in the device causes the device to transmit a ComInit Sequence.
3. Host calibrates and issues a ComWake Sequence.
4. Device responds - The device detects the ComWake Sequence on its RX pair, asserts the ComWake signal, and calibrates its transmitter (optional). Following calibration, the device sends a six burst ComWake Sequence and then sends a continuous stream of the ALIGN primitives starting at the device's highest supported speed. After ALIGN primitives have been sent for 54.6us (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 primitives at that rate for 54.6us (2048 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.
5. Host locks - After detecting the ComWake Sequence, the host physical layer asserts the ComWake signal and starts transmitting D10.2 characters (See 5.6) at its lowest supported rate. Meanwhile, the host receiver locks to the ALIGN primitives and, when ready, transmits ALIGN primitives to the device at the same speed as received. A host shall be designed such that it can acquire lock in 54.6us (2048 nominal Gen1 dword times) at any given speed. The host should allow for at least 873.8us (32768 nominal Gen1 dword times) after the negation of the ComWake signal to receive the first ALIGN primitive. This will ensure interoperability with multi-generational and synchronous devices. If no ALIGN is received within 873.8us (32768 nominal Gen1 dword times), the host restarts the power-on sequence - repeating indefinitely until told to stop by the application layer..
6. Device locks - The device locks to the ALIGN primitives and, when ready, sends the SYNC primitive indicating it is ready to start normal operation.
7. Upon receipt of three back-to-back non-ALIGN primitives, the communication link is established and normal operation may begin.

5.5.12.3.4 COMWAKE Description

The ComWake Sequence is used during ComInit Sequence, ComReset Sequence, and also separately as COMWAKE to bring the Phy out of a power-down state (Partial or Slumber) as described in subclause 5.5.12.3.5. COMWAKE begins with a ComWake Sequence that can originate from either the host controller or the device. It is signaled by transmitting bursts separated by quiescent bus spaces.

The ComWake Sequence shall consist of no less than six bursts, including inter-burst temporal spacing. Each burst shall be 160 Gen1 UI_{OOB}'s (approximately 106.7ns) long and each inter-burst space shall be 160 Gen1 UI_{OOB}'s (approximately 106.7ns) long. A ComWake Detector will look for four consecutive bursts with a ComWake transmit spacing time (approximately 106.7 ns nominal, see Table 12).

Any spacing less than ComWake Detector off threshold min or greater than ComWake Detector threshold max (See Table 12) shall negate the ComWake Detector output. The interface shall be held inactive (quiescent) for at least the ComWake Detector off threshold max time (See Table 12). after the last burst to ensure far-end detector detects the deassertion properly. The device may hold the interface inactive no more than 228.3ns (the ComWake Detector threshold max + 2 nominal Gen1 dword times) at the end of a ComWake burst to prevent susceptibility to crosstalk.

5.5.12.3.5 Interface power modes

In the serial transport, interface power modes 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 transport power modes are defined in Table 17.

Partial mode is defined as the host being in state HP12 (see Figure 39) or the device being in state DP11 (see Figure 40), and the TX transmitter is common mode as defined in 5.4.2.

Slumber mode is defined as the host being in state HP13 (see Figure 39) or the device being in state DP12 (see Figure 40), and the TX transmitter is common mode as defined in 5.4.2.

Active mode is defined as being in the host being in state HP11 (see Figure 39) or the device being in state DP10 (see Figure 40).

Table 17 - Interface power modes

| | |
|---------------------|---|
| Active mode | The Phy logic and main PLL are both on and active. The interface is synchronized and capable of receiving and sending data. Indicated by PhyRdy signal. |
| Partial mode | 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 10μs. |
| Slumber mode | 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 10ms. |

5.5.12.3.5.1 Quiescent bus condition

During power management modes (Partial and Slumber), the electrical interface shall maintain the proper common-mode levels, as cited in 5.4.2, 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 quiescent 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 Partial mode. When operating in the Slumber 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 5.4.2.

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

5.5.12.3.6 Power-on sequence timing diagram

The following timing diagrams and descriptions are provided for clarity and are informative. The state diagrams provided in subclause 5.5.12.2 comprise the normative behavior specification

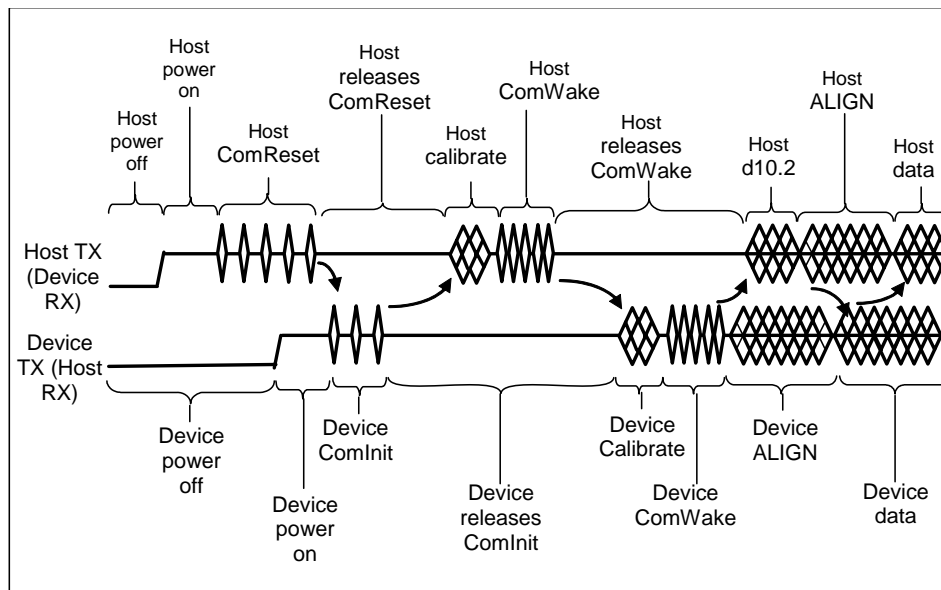


Figure 43 - Power-on Sequence

Description

1. Host/device power-off - Host and device power-off.
2. Power is applied - Host side signal conditioning pulls TX and RX pairs to neutral state (common mode voltage).
3. Host issues a ComReset Sequence.
4. When the Power-on reset is negated and ComReset Sequence is completed, the host puts the bus in a quiescent condition.
5. When the device detects the negation of the ComReset signal, 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 Sequence.
6. Host calibrates and issues a ComWake Sequence.
7. Device responds - The device detects the ComWake Sequence on its RX pair, asserts the ComWake signal, and calibrates its transmitter (optional). Following calibration, the device sends a six burst ComWake Sequence and then sends a continuous stream of the ALIGN primitives starting at the device's highest supported speed. After ALIGN primitives have been sent for 54.6us (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 primitives at that rate for 54.6us (2048 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.
8. Host locks - After the ComWake signal is asserted, the host starts transmitting D10.2 characters (See 5.6) at its lowest supported rate. Meanwhile, the host receiver locks to the ALIGN primitives and, when ready, transmits ALIGN primitives to the device at the same speed as received. A host shall be designed such that it can acquire lock in 54.6us (2048 nominal Gen1 dword times) at any given speed. The host should allow for at least 873.8us (32768 Gen1 dword

times) after detecting the negation of the ComWake signal to receive the first ALIGN. This will ensure interoperability with multi-generational and synchronous devices. If no ALIGN is received within 873.8us (32768 nominal Gen1 dword times), the host restarts the power-on sequence - repeating indefinitely until told to stop by the application layer.

9. Device locks - The device locks to the ALIGN primitives and, when ready, sends the SYNC primitive indicating it is ready to start normal operation.
10. Upon receipt of three back-to-back non-ALIGN primitives, the communication link is established and normal operation may begin.

5.5.12.3.7 Active to Partial or Slumber mode

Active to Partial or Slumber mode is initiated by the transmission of the PMREQ_P or PMREQ_S and PMACK primitives as described in 6.4.9.

5.5.12.3.8 Partial mode or Slumber mode to Active

The host initiates a wakeup from the Partial mode or Slumber mode by entering the initialization sequence at the HP5: HR_COMWAKE state in the host Phy initialization state 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 mode or Slumber mode by entering the initialization sequence at the DP6: DR_COMWAKE state in the device Phy initialization state 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.

5.5.12.4 On to Partial mode or Slumber mode

5.5.12.4.1 Host initiated

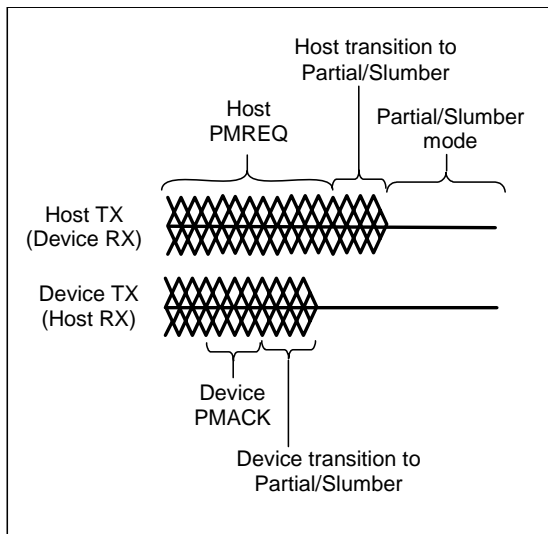


Figure 44 - On to Partial or Slumber mode - host initiated

5.5.12.4.2 Detailed sequence

1. Host Application layer sends request to host transport layer.
2. Host transport layer transmits request to host link layer.
3. Host link layer encodes request as PMREQ primitive and transmits it four times to host Phy layer.
4. Host Phy layer serializes PMREQ primitives and transmits them to device Phy layer.
5. Device Phy de-serializes PMREQ primitives and transmits them to device link layer.
6. Device link layer decodes PMREQ primitives and transmits request to device transport layer.
7. Device transport layer transmits request to device Application layer.
8. Device Application layer processes and accepts request. Issues accept to device transport layer.
9. Device transport layer transmits acceptance to device link layer.
10. Device link layer encodes acceptance as PMACK primitive and transmits it four times to device Phy layer.
11. Device Phy layer transmits four PMACK primitives to host Phy layer.
12. Device link layer places device Phy layer in Partial or Slumber mode.
13. Host Phy layer de-serializes PMACK primitives and transmits them to host link layer.
14. Host link layer decodes PMACK primitives and transmits acceptance to host transport layer.
15. Host link layer places host Phy layer in Partial or Slumber mode.
16. Host transport layer transmits acceptance to host Application layer.

5.5.12.4.3 Device initiated

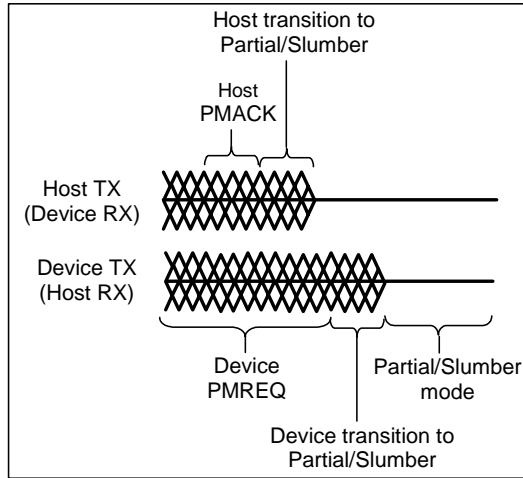


Figure 45 - ON to Partial or Slumber - device initiated

5.5.12.4.4 Detailed sequence

1. Device Application layer sends request to device transport layer.
2. Device transport layer transmits request to device link layer.
3. Device link layer encodes request as PMREQ primitive and transmits it to device Phy layer.
4. Device Phy layer serializes PMREQ primitives and transmits them to host Phy layer.
5. Host Phy de-serializes PMREQ primitives and transmits them to host link layer.
6. Host link layer decodes PMREQ primitives and transmits request to host transport layer.
7. Host transport layer transmits request to host Application layer.
8. 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 subclause of the chipset logic.
9. Host Application layer processes and accepts request. Issues accept to host transport layer.
10. Host transport layer transmits acceptance to host link layer.
11. Host link layer encodes acceptance as PMACK primitive and transmits it four times to host Phy layer.
12. Host Phy layer transmits four PMACK primitives to device Phy layer.
13. Host link layer asserts Partial signal or Slumber signal and places host Phy layer in Partial or Slumber mode.
14. Host Phy layer negates PhyRdy signal.
15. Device Phy layer de-serializes PMACK primitives and transmits them to device link layer.
16. Device link layer decodes PMACK primitives and transmits acceptance to device transport layer.
17. Device link layer asserts Partial signal or Slumber signal and places device Phy layer in Partial or Slumber mode.
18. Device Phy layer negates PhyRdy signal.
19. Device transport layer transmits acceptance to device Application layer.

5.6 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 but also 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 1024 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.

5.7 BIST (Built in self test)

BIST provides loopback testing of portions of the physical layer. BIST is initiated by the BIST Activate FIS (See 7.5.7).

5.7.1 Loopback testing

Three types of Loopback test schemes are defined.

- | | | |
|--|---|-----------|
| a) Far-End Retimed | - | Mandatory |
| b) Far-End Analog | - | Optional |
| c) Near-End Analog (Effectively Retimed) | - | Optional |

5.7.1.1 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 7.5.7.

The Far-End Interface shall remain in this Far-End Retimed Loopback, until receipt of a COMRESET or COMINIT 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.

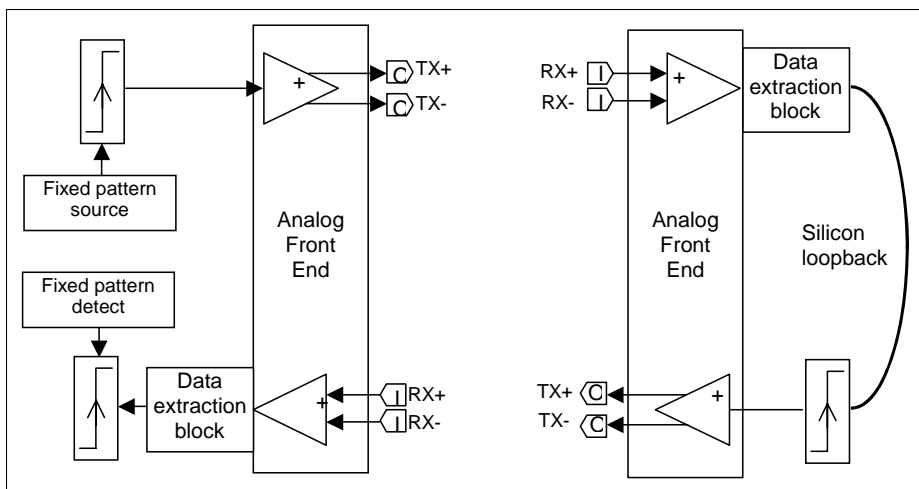


Figure 46 - Loopback far-end retimed

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 ALIGN's in the transmitted stream. The initiator shall transmit additional ALIGN sequences (Dual ALIGN primitives) in a single burst at the normal interval of every 256 dwords transmitted (as opposed to inserting ALIGN sequences at half the interval).

The loopback host/device may remove zero, one, or two ALIGN 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 ALIGN's are always in bursts of two and if the initiator transmitted dual ALIGN sequences (four consecutive ALIGN's), 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 ALIGN's returned by the loopback host/device and the relative position of the ALIGN's sent by the initiator.

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.

5.7.1.2 Loopback -- far-end analog (Optional)

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 7.5.7.

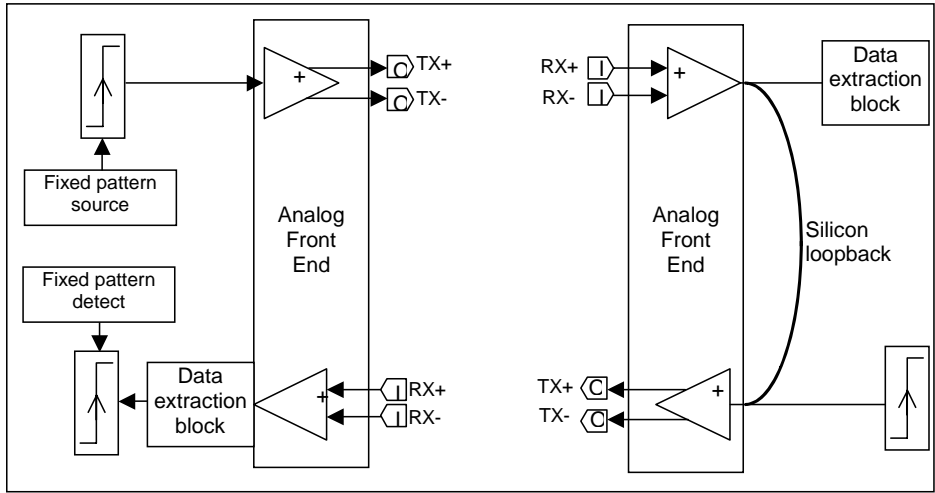


Figure 47 - Loopback far-end analog

Once entered, the Far-End Interface shall remain in this Far-End Analog Loopback mode, until receipt of the ComReset/ComInit 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.

5.7.1.2.1 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 mode, or Partial mode.

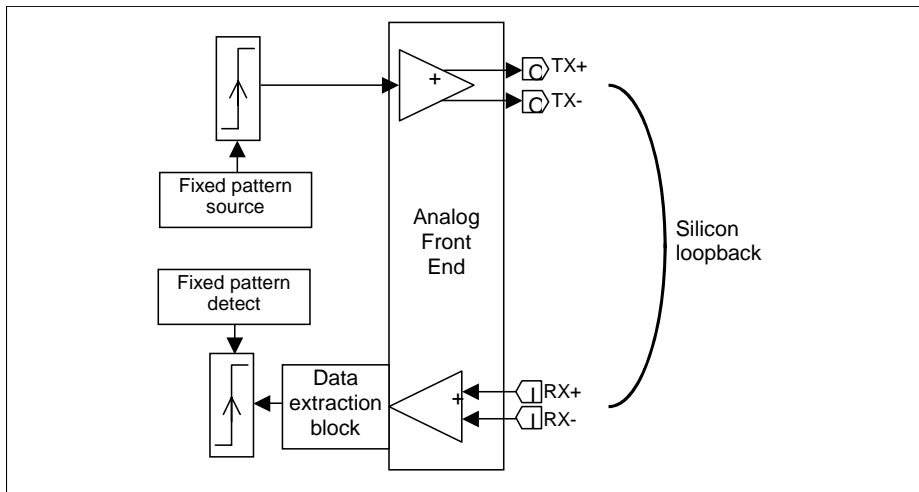


Figure 48 - Loopback - near-end analog

6 Serial transport link layer

6.1 Overview

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 – Link transmit state diagram).

6.1.1 Frame transmission

When requested by the transport layer to transmit a frame, the link layer provides the following services:

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

6.1.2 Frame receipt

When data is received from the physical layer, the link layer provides the following services:

- a) Acknowledges to the peer link layer readiness to receive a frame.
- b) Receives data in the form of encoded characters from the physical layer.
- c) Decodes the encoded 8b/10b character stream into aligned dwords of data.
- d) Removes the envelope around frames (i.e., SOF, CRC, EOF).
- e) Calculates CRC on the received dwords.
- f) Provides frame flow control in response to requests from the FIFO or the peer link layer.
- g) Compares the calculated CRC to the received CRC.
- h) Reports good reception or Link/physical layer errors to transport layer and the peer link layer.
- i) Descrambles (untransforms) the control and data dwords received from a peer link layer.

6.2 8b10 coding method

Information to be transmitted over the serial transport 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 transport 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 transport.

6.2.1 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 3.3.7.

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 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.

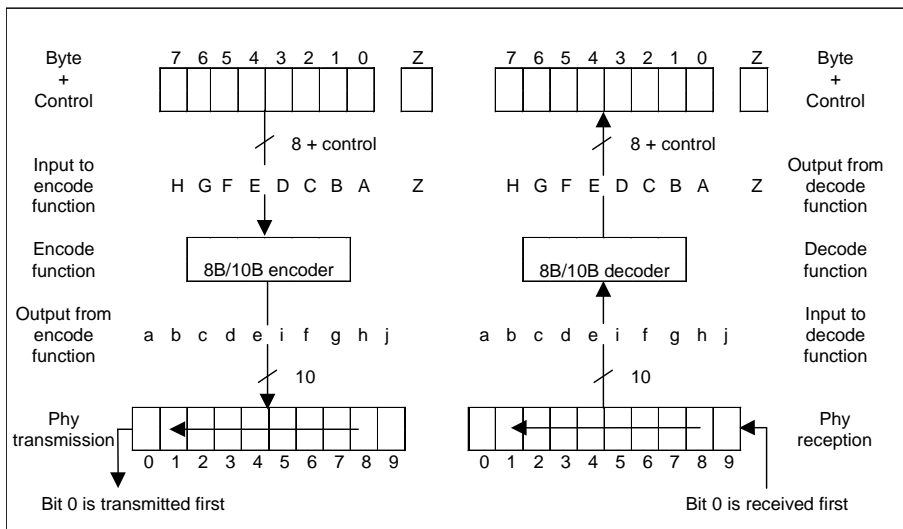


Figure 50 - Nomenclature reference

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

| byte notation | BCh, control character | | 4Ah, data character | |
|---|------------------------|------------------|---------------------|------------------|
| | 76543210 | Control variable | 76543210 | Control variable |
| Bit notation | 10111100 | K | 01001010 | D |
| | HGF EDCBA | Z | HGF EDCBA | Z |
| Unencoded bit notation | 101 11100 | K | 010 01010 | D |
| | Z | EDCBA HGF | Z | EDCBA HGF |
| Bit notation reordered to conform with Zxx.y convention | K | 11100 101 | D | 01010 010 |
| | character name | K 28 .5 | D 10 .2 | |

Figure 51 - Conversion examples

6.2.2 Character code

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 serial transport 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 6.2.4.1.

6.2.2.1 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.

6.2.2.2 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 zeros 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 zeros 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 zeros 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.

6.2.2.3 Data encoding

Table 18 - 5b/6b coding and Table 19 - 3b/4b coding 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 18 - 5b/6b coding

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

Table 19 - 3b/4b coding

| Inputs | | fghj outputs | | rd' |
|--------|-----|--------------|------|-----|
| Dx.y | HGF | rd+ | rd- | |
| Dx.0 | 000 | 0100 | 1011 | -rd |
| Dx.1 | 001 | 1001 | | rd |
| Dx.2 | 010 | 0101 | | |
| Dx.3 | 011 | 0011 | 1100 | -rd |
| Dx.4 | 100 | 0010 | 1101 | |
| Dx.5 | 101 | 1010 | | |
| Dx.6 | 110 | 0110 | | rd |
| Dx.P7 | 111 | 0001 | 1110 | -rd |
| Dx.A7 | 111 | 1000 | 0111 | |

NOTE –
A7 replaces P7 iff[(rd>0) and (e=i=0)] or [(rd<0) and (e=i=1)]

6.2.2.4 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 (1) 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 $[(rd>0) \text{ AND } (e=i=0)]$.

| Initial rd | character name | abcdei output | 6-bit sub-block rd | fgjh 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

6.2.2.5 8b/10b valid encoded characters

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.

6.2.2.5.1 Data characters

Table 20 - 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 |

(continued)

Table 20 - Valid data characters (continued)

| Name | byte | abcdei fghj output | | Name | byte | abcdei fghj output | |
|-------|------|--------------------|-------------|-------|------|--------------------|-------------|
| | | Current rd- | Current rd+ | | | Current rd- | Current rd+ |
| 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 |
| 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 |

(continued)

Table 20 - Valid data characters (continued)

| Name | byte | abcdei fghj output | | Name | byte | abcdei fghj output | |
|-------|------|--------------------|-------------|-------|------|--------------------|-------------|
| | | Current rd- | Current rd+ | | | Current rd- | Current rd+ |
| 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 | A Eh | 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 |
| 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 |

(continued)

Table 20 - Valid data characters (continued)

| Name | byte | abcdei fghj output | | Name | byte | abcdei fghj output | |
|-------|------|--------------------|-------------|-------|------|--------------------|-------------|
| | | Current rd- | Current rd+ | | | Current rd- | Current rd+ |
| 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 |
| 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 |

(concluded)

6.2.2.5.2 Control characters**Table 21 - 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 transport, 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.

6.2.3 Transmission summary

6.2.3.1 Transmission order

6.2.3.1.1 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

6.2.3.1.2 Bytes within a dword

For all transmissions and receptions, the serial transport 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.

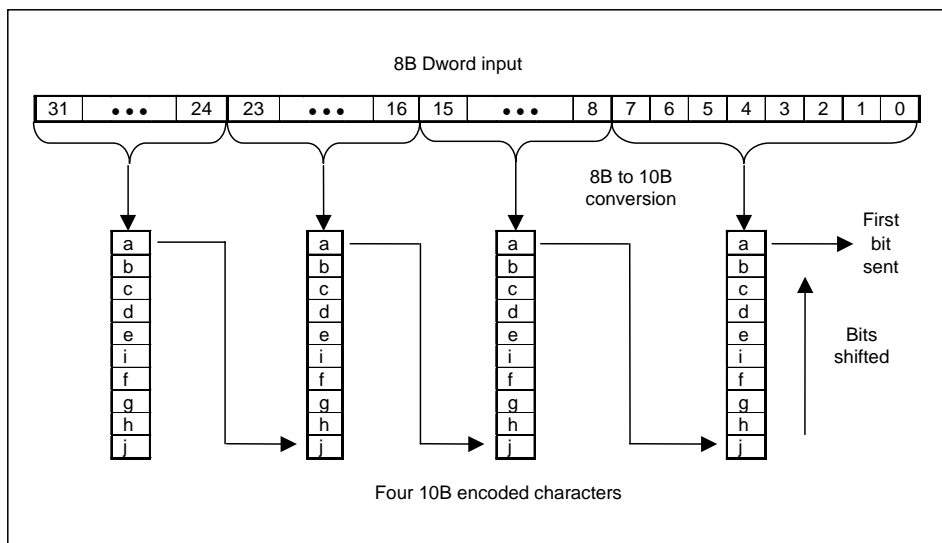


Figure 53 - Bit ordering and significance

6.2.3.1.3 dwords within a frame

A frame (as described in 7.3) 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 transport 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.

6.2.4 Reception

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.

6.2.4.1 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 error is introduced. The following examples illustrate this effect. The first example shows a bit error being propagated two characters before being detected. The second shows a single character of propagation.

It is important to note that the serial transport 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 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 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. Table notes 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) |

NOTES –

1. Bit error introduced: 1001 → 1011
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.
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) |

NOTES–

1. Bit error introduced: 1001 → 1011
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.
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

6.3 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 transport.

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 shows an example of a sequence of transmissions.

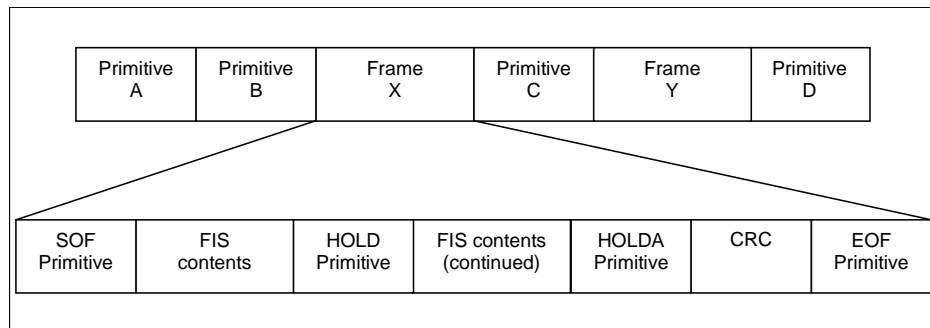


Figure 56 - Transmission structures

6.4 Primitives

6.4.1 Overview

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.

6.4.1.1 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.

6.4.1.2 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 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.

6.4.2 Primitive descriptions

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

Table 22 - 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 1 |
| HOLDA | Hold acknowledge | This primitive is sent by a transmitter as long the HOLD primitive is received by its companion receiver. | Repeated Note 1 |
| PMACK | Power management acknowledge | Sent in response to a PMREQ_S or PMREQ_P when a receiving node is prepared to enter Partial mode or Slumber mode. | 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 Partial mode or Slumber mode or when power management is not supported. | Repeated |
| PMREQ_P | Power management request to Partial mode | 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 mode. | Repeated Note 1 |
| PMREQ_S | Power management request to Slumber mode | 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 mode. | Repeated Note 1 |
| R_ERR | Reception error | Current node (host or device) detected error in received payload. | Repeated Note 1 |
| R_IP | Reception in Progress | Current node (host or device) is receiving payload. | Repeated Note 1 |
| R_OK | Reception with no error | Current node (host or device) detected no error in received payload. | Repeated Note 1 |
| R_RDY | Receiver ready | Current node (host or device) is ready to receive payload. | Repeated Note 1 |

(continued)

Table 22 - Description of primitives (continued)

| Primitive | Name | Description | Usage |
|---|----------------------------|--|--------------------|
| SOF | Start of frame | Start of a frame. Payload and CRC follow to EOF. | Single |
| SYNC | Synchronization | Synchronizing primitive - always idle. | Repeated Note 1 |
| 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 1 |
| X_RDY | Transmission data ready | Current node (host or device) has payload ready for transmission | Repeated Note 1 |
| Note 1: This primitive may be continued by use of the CONT primitive. | | | |

6.4.3 Primitive encoding

The following table defines the encoding for each primitive.

Table 23 - 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 |

6.4.4 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 23.

6.4.5 CONT primitive

The CONT primitive allows long strings of repeated primitives, as defined by Table 22 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 transport as described in 6.6. 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 2048 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 22) 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 24.

Table 24 - 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 22. 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 22.

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 5.5.12.3). 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 6.7.1.1 and 6.7.1.4), 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.

6.4.5.1 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 6.6.

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 6.6 for additional information on scrambling and repeated primitive suppression.

6.4.6 DMAT primitive

Note: No consistent use of the DMAT 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.

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 DMA context is assumed to remain stable after the EOF primitive has been sent.

In a Parallel Transport 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 transport a DMA read from device a device may terminate the transfer with an EOF, and send a Response FIS to the host, with Error and Status fields 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.

6.4.7 EOF primitive

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

6.4.8 HOLD/HOLDA primitives

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.

6.4.8.1 Flow Control Signaling Latency

There is a finite pipeline latency in a round-trip handshake across the serial transport. 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 25 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 - 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 26 - SRST write from host to device transmission breaking through a device to host Data FIS

(editors note: this is out of place, probably should be moved to 6.4.18)

| 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 HBA 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 host to device FIS 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 |

6.4.9 PMREQ_P, PMREQ_S, PMACK, and PMNAK primitives

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

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

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

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

6.4.10 R_ERR primitive

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

6.4.11 R_IP primitive

Current node (host or device) is receiving payload.

6.4.12 R_OK primitive

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

6.4.13 R_RDY primitive

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

6.4.14 SOF primitive

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

6.4.15 SYNC primitive

Synchronizing primitive - Indicates physical interface is idle.

6.4.16 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.

6.4.17 X_RDY primitive

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

6.4.18 Examples

The following examples illustrate basic primitive usage.

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

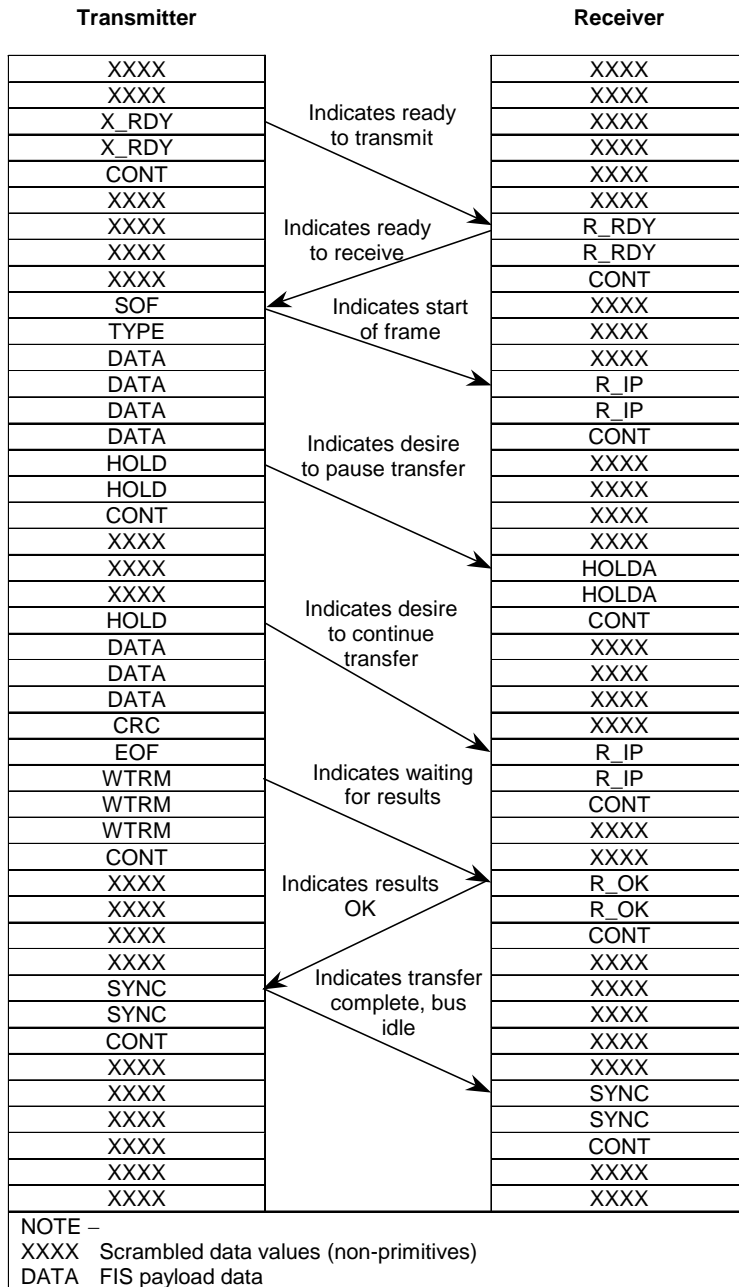


Figure 57 - CONT usage example

Table 27 – device to host FIS transmission example

| Host driver | Device driver | Description |
|-------------|---------------|--|
| SYNC | SYNC | Idle condition |
| SYNC | SYNC | Idle condition |
| X_RDY | SYNC | Host ready to send host to device FIS |
| 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 r FIS dword 0 / device decodes SOF |
| Hdr 1 | R_IP | Host sends FIS dword 1 / device stores Hdr 0 |
| ... | ... | ... |
| Hdr n | R_IP | Host sends 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 28 - 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 host to device FIS |
| 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 stores (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 29 - 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 |

6.5 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.

6.6 Scrambling

6.6.1 Frame content scrambling

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:

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

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.

6.6.1.1 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.

6.6.2 Repeated primitive suppression

A second linear feedback shift register shall be used to provide scrambled data for the suppression of repeated primitives with the CONT primitive (See 6.4.5). This LFSR shall implement the same polynomial as the frame scrambling LFSR but need not be reinitialized.

6.6.2.1 Relationship between scrambling of FIS data and repeated primitives

There are two separate scramblers used in the serial transport. 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.

6.7 link layer state diagrams

6.7.1.1 Link idle state diagram

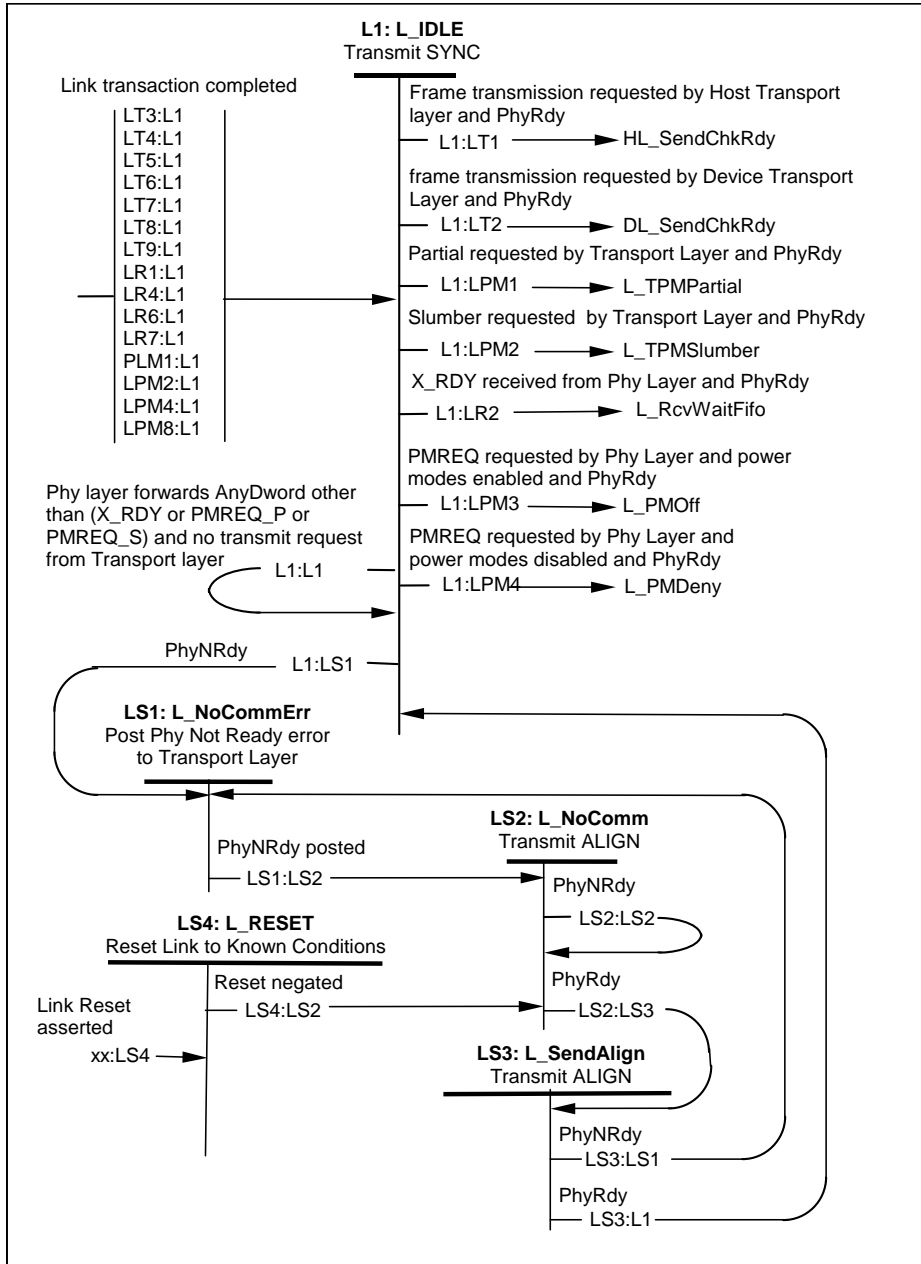


Figure 58 – Link idle state diagram (States L1, LS1-LS3)

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 6.4.5.

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 State 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 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 PhyRdy signal is negated, 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 negation of PhyRdy signal by 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: Transition 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 asserts the PhyRdy signal.

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

Transition LS2:LS2: If the the PhyRdy signal is negated, transition to LS2: L_NoComm.

Transition LS2:LS3: When the physical layer asserts PhyRdy, transition to LS3: L_SendAlign.

LS3: L_SendAlign state: This state is entered when the PhyRdy signal asserted. When in this state ALIGN is transmitted.

Transition LS3:LS1: If PhyRdy signal is negated, transition to LS1: L_NoCommErr.

Transition LS3:L1: If the Phy asserts the PhyRdy signal, transition to the L1: L_IDLE state.

LS4: L_RESET state: This state is entered whenever the Link Reset is asserted. 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 PHYRESET signal is also asserted (see Figure 22).

Transition xx:LS4: If Link Reset is asserted, transition back to the LS4: L_RESET state.

Transition LS4:LS2: When Link Reset is negated transition to the LS2: L_NoComm state.

6.7.1.2 Link transmit state diagram

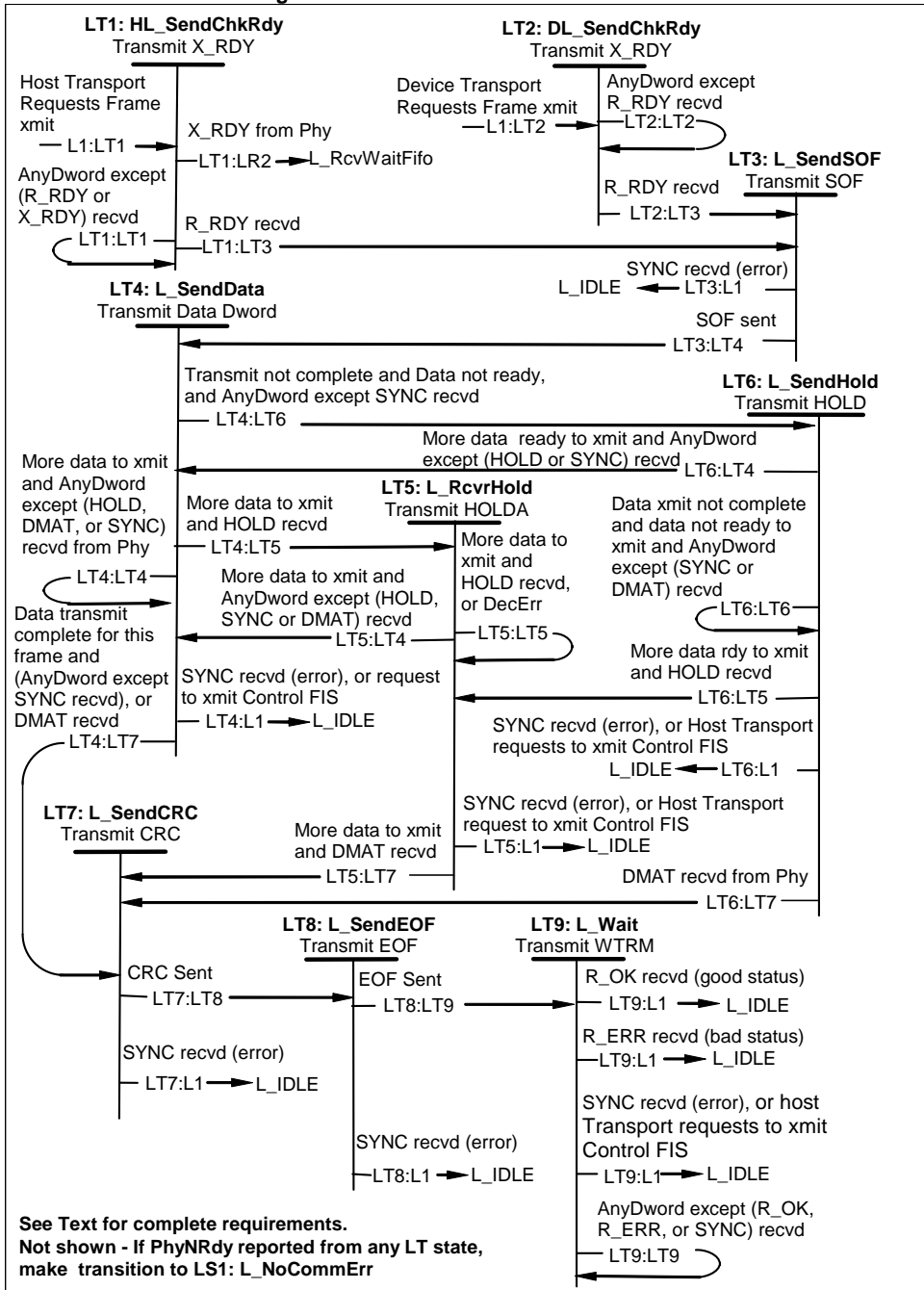


Figure 59 – Link transmit state diagram (States LT1-LT9)

In any state of the Link transmit state diagram, if the link layer detects that the PhyRdy signal is negated, 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: If 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: If 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: If 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: If the host link layer detects that the PhyRdy signal is negated, 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: If 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: If 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: If the device link layer detects that the PhyRdy signal is negated, 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: If 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: If the link layer detects that the PhyRdy signal is negated, 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: If 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 6.4.6)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: If 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: If 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: If 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 6.4.6) 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 6.4.6 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: If 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.

If the host link layer receives notification from the host transport layer that a Control FIS 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 Control FIS may then be transmitted. When this condition is true, the associated transition has priority over all other transitions exiting this state.

Transition LT4:LS1: If 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: If 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: If 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: If 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.

If the host link layer receives notification from the host transport layer that a Control FIS 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 Control FIS may then be transmitted. When this condition is true, the associated transition has priority over all other transitions exiting this state.

Transition LT5:LT7: If 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 6.4.6 and may transition to the LT7: L_SendCRC state. (Use of DMAT is not recommended see 6.4.6) 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: If the link layer detects that the PhyRdy signal is negated, 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: If 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: If 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: If 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 6.4.6 and transition to the LT7:L_SendCRC state. (Use of DMAT is not recommended see 6.4.6) 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: If 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 Control FISControl FIS 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 Control FISControl FIS may then be transmitted. When this condition is true, the associated transition has priority over all other transitions exiting this state.

Transition LT6:LS1: If the link layer detects that the PhyRdy signal is negated, 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: If the link layer detects that the PhyRdy signal is negated, 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 PhyRdy signal is asserted 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: If the link layer detects that the PhyRdy signal is negated, 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: If 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 Command FIS, 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: If the link layer detects that the PhyRdy signal is negated, 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.

6.7.1.3 Link receive state diagram

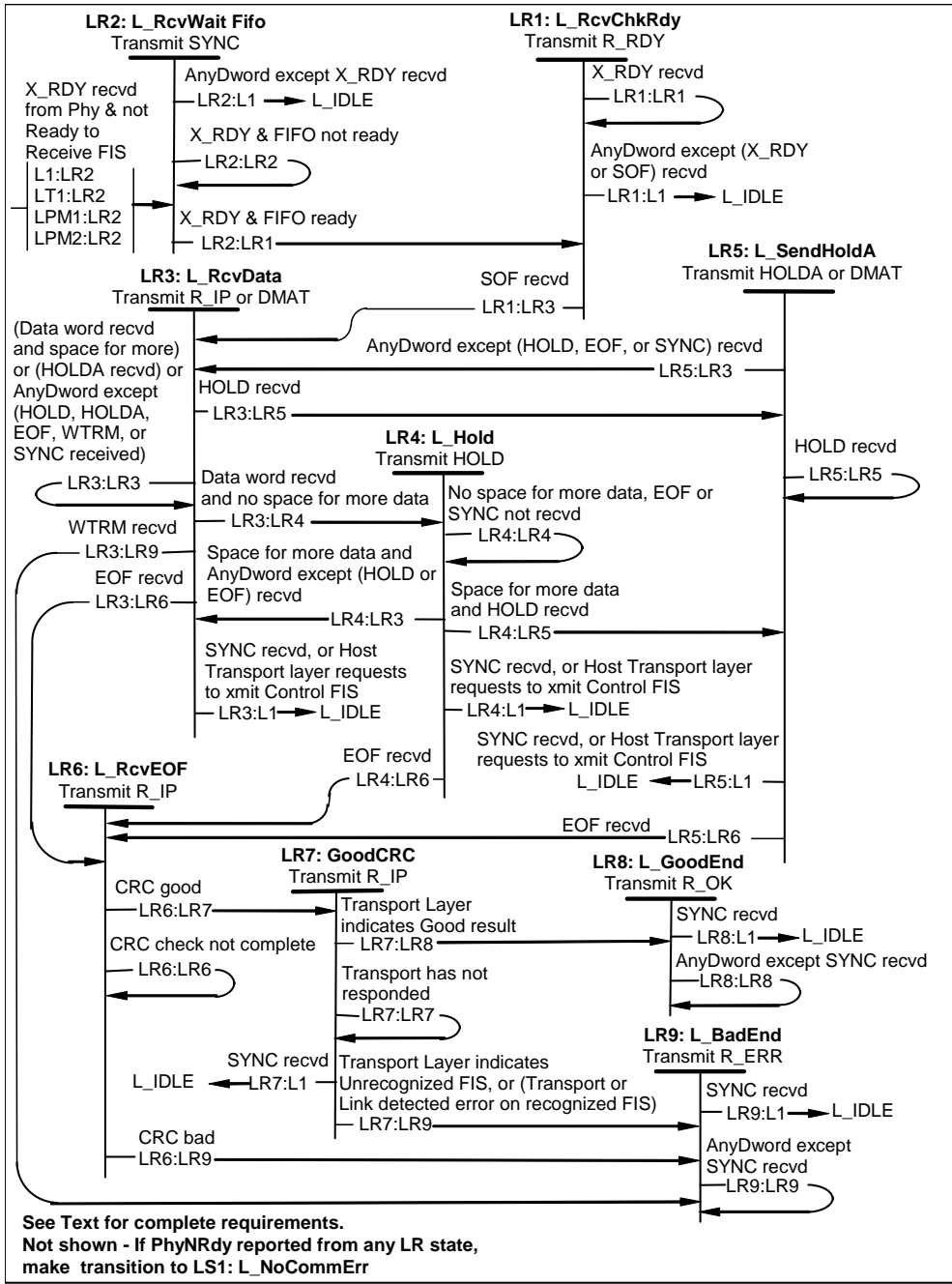


Figure 60 – Link receive state diagram (States LR1-LR9)

In any state of the Link receive state diagram, if the link layer detects the PhyRdy signal from the Phy is negated, 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: If 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: If 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: If 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: If the link layer detects that the PhyRdy signal is negated, 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: If 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: If 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: If 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: If the link layer detects that the PhyRdy signal is negated, 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 6.4.6) 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: If the link layer receives any dword other than a HOLD, HOLDA, EOF, or SYNC, or WTRM primitive from the physical layer, the link layer shall make a transition to the LR3: L_RcvData state.

Transition LR3:LR4: If 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: If 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: If 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: If 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: If 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 FIS 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 Control FIS may then be transmitted.

Transition LR3:LS1: If the link layer detects that the PhyRdy signal is negated, 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: If 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: If 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: If the link layer receives an 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 signal is asserted: If the transport layer indicates that there is not sufficient space available in its FIFO and the PhyRdy signal is negated, the link layer shall make a transition to the LR4: L_Hold state.

Transition LR4:LS1: If the link layer detects that the PhyRdy signal is negated, the link layer shall notify the transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR4:L1: If 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.

If the host link layer receives notification from the host transport layer that a Control FIS 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 Control may then be transmitted.

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 6.4.6) to signal the transmitter to terminate the transmission.

Transition LR5:LR3 AnyDword other than (HOLD or EOF or SYNC) received from Phy: If 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: If 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: If 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: If 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.

If the host link layer receives notification from the host transport layer that a Control FIS 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 Control FIS may then be transmitted.

Transition LR5:LS1: If the link layer detects that the PhyRdy signal is negated, 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: If 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: If 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: If the link layer detects that the PhyRdy signal is negated, 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: If the transport layer indicates a good result, the link layer shall transition to the LR8: L_GoodEnd state.

Transition LR7:LR9: If the transport layer indicates an unrecognized FIS, the link layer shall transition to the LR9: L_BadEnd state.

If 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: If the link layer detects that the PhyRdy signal is negated, the link layer shall notify the transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR7:L1: If 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: If 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: If 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: If the link layer detects that the PhyRdy signal is negated, 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: If 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: If 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: If the link layer detects that the PhyRdy signal is negated, the link layer shall notify the transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

6.7.1.4 Link power mode state diagram

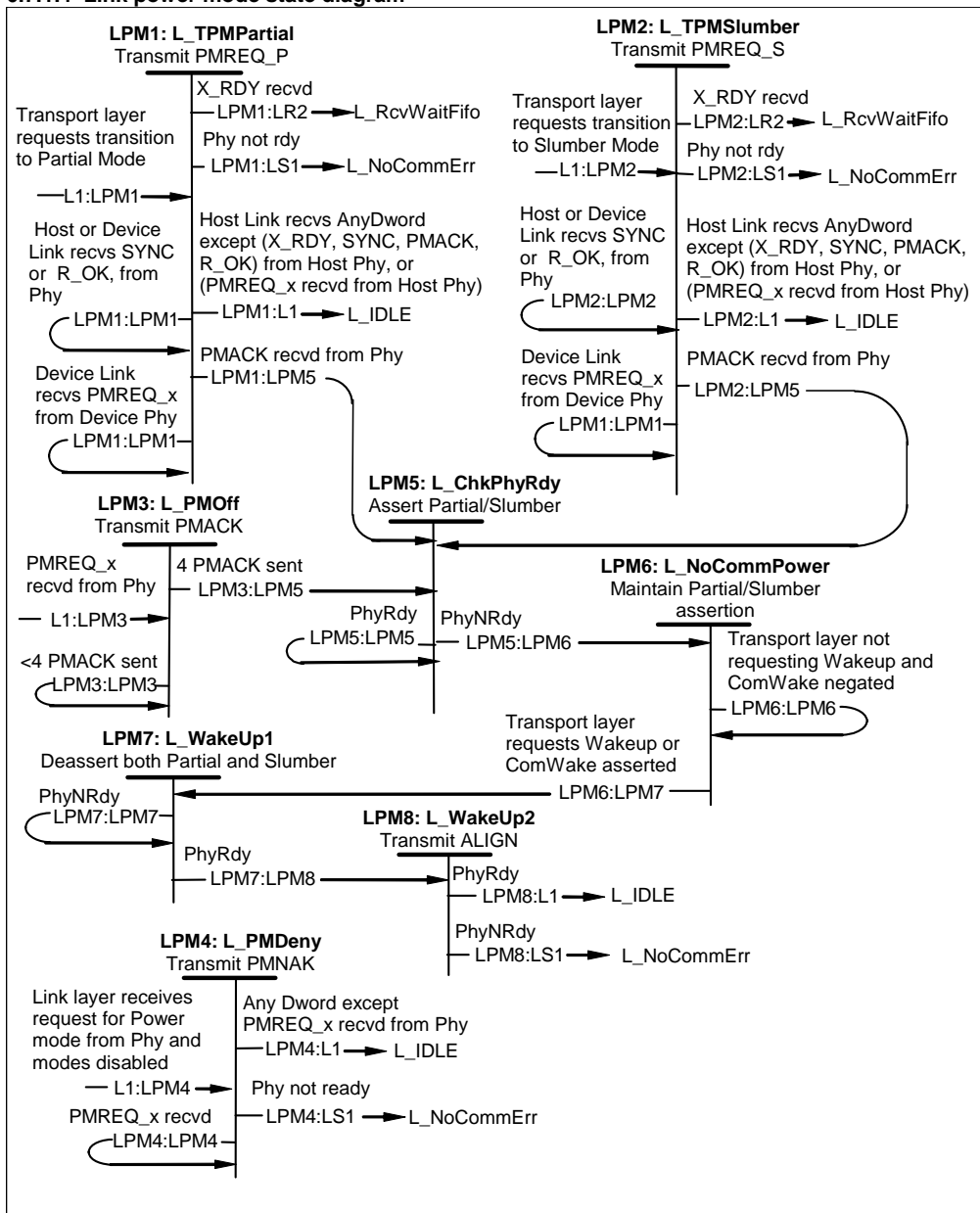


Figure 61 – Link power mode state diagram (States LPM1-LPM8)

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 PhyRdy signal is negated, 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: If 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 PhyRdy signal is negated, 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 than 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 PhyRdy signal is negated, 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 negate 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 the PhyRdy signal is negated, 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 negated 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 Phy indicates reception of the ComWake Sequence by the assertion 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 ComWake signal is negated, 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: If the PhyRdy signal is negated, 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 PhyRdy signal is asserted, a transition shall be made to the L1: L_IDLE state.

Transition LPM8:LS1 If the PhyRdy signal is negated, 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.

7 Transport layer

7.1 transport layer overview

The transport layer constructs Frame Information Structures (FISes) for transmission and decomposes received Frame Information Structures. The transport layer need not be cognizant of how frames are transmitted and received. 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.

7.1.1 FIS construction

When requested to construct an FIS by a higher layer, the transport layer provides the following services:

- a) Gathers FIS content based on the type of FIS requested.
- b) Places FIS content in the proper order.
- c) Notifies the link layer of required frame transmission and passes FIS content to Link.
- d) Manages Buffer/FIFO flow, notifies Link of required flow control.
- e) Receives frame receipt acknowledge from link layer.
- f) Reports good transmission or errors to requesting higher layer.

7.1.2 FIS decomposition

When an FIS is received from the link layer, the transport layer provides the following services:

- a) Receives the FIS from the link layer.
- b) Determines FIS type.
- c) Distributes the FIS content to the locations indicated by the FIS type.
- d) For the host transport layer, receipt of an FIS may also cause the construction of an FIS to be returned to the device.
- e) Reports good reception or errors to higher layer

7.2 Frame Information Structure (FIS)

7.3 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 - hereto referred as payload - and assumes the reader is aware of the primitives that are needed to support the information content.

7.4 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. See 7.5.

7.5 FIS types

The following subclauses define the structure of each individual FIS.

7.5.1 Command FIS

The Command FIS is a host to Device FIS, FIS Type 27h, with the C bit set to one. In the serial transport, there are two kinds of command set mappings into the Command FIS.

- a) 48-bit command set mapping
- b) 28-bit command set mapping

7.5.1.1 Command FIS - 48-bit command mapping

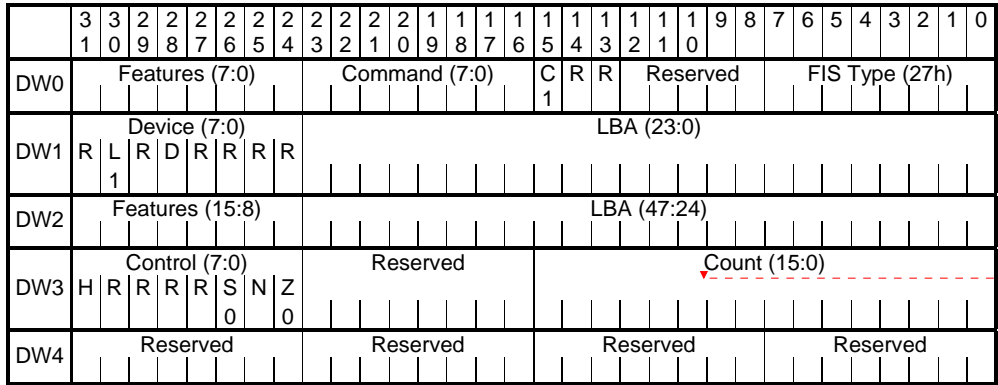


Figure 62 - Command FIS - 48-bit command transport FIS layout

Field Definitions:

FIS Type - Set to a value of 27h.

R - Reserved.

C - This bit shall be set to one to indicate that the FIS is a Command FIS. (See 7.5.2 for a Control FIS description when C bit is cleared to zero.)

Command (7:0) – Contains bits 7:0 of the Command field of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Command register of the HBA Command Block (see E.5).

Features (7:0) -Contains bits 7:0 of the Features field of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Features register (current) of the HBA Command Block.

LBA (23:0) – Contains bits 23:0 of the LBA field of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the LBA Low register (current), LBA Mid register (current), and LBA High register (current), in that order, of the HBA Command Block.

LBA (47:24) – Contains bits 47:24 of the LBA field of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the LBA Low register (previous), LBA Mid register (previous), and LBA High register (previous), in that order, of the HBA Command Block.

Device (7:0) – Contains bits 7:0 of the Device field of the command. These bits are individually assigned to specific functions as defined below. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Device register of the HBA Command Block.

D – Variable. In ATA/ATAPI-7 parallel emulation, this bit was mapped as the DEV bit. The bit is not used in the serial transport, and may be transmitted with a zero or a one value. It is recommended that it be cleared to zero.

L – Fixed for all commands in the 48-bit feature set. This bit shall be set to one. In ATA/ATAPI-7 parallel emulation, this bit was mapped as the LBA bit.

Features (15:8) – Contains bits 15:8 of the Features field of the Command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Features register (previous) of the HBA Command Block.

~~Count (15:0) – Contains bits 15:0 of the Count field of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Sector Count register of the HBA Command Block.~~

Deleted: Sector

Control (7:0) – Contains bits that are individually assigned to specific functions as defined below. In ATA/ATAPI-7 parallel emulation, these bits were mapped to the HBA Control Register of the HBA Command Block.

Z – Fixed. This bit shall be cleared to zero.

N – Variable. In ATA/ATAPI-7 parallel emulation, this bit corresponds to the nLEN bit. The bit is not used in the serial transport, and may be transmitted with a zero or a one value. It is recommended that it be cleared to zero.

S – Fixed. This bit shall be cleared to zero. In ATA/ATAPI-7 parallel emulation, this bit corresponds to the SRST bit. (See 7.5.2 for SRST description of the Control FIS when C bit is cleared to zero.)

H – Variable. In ATA/ATAPI-7 parallel emulation, this bit corresponds to the HOB bit. The bit is not used in the serial transport, and may be transmitted with a zero or a one value. It is recommended that it be cleared to zero.

7.5.1.2 Command FIS - 28-bit command and non-data command mapping

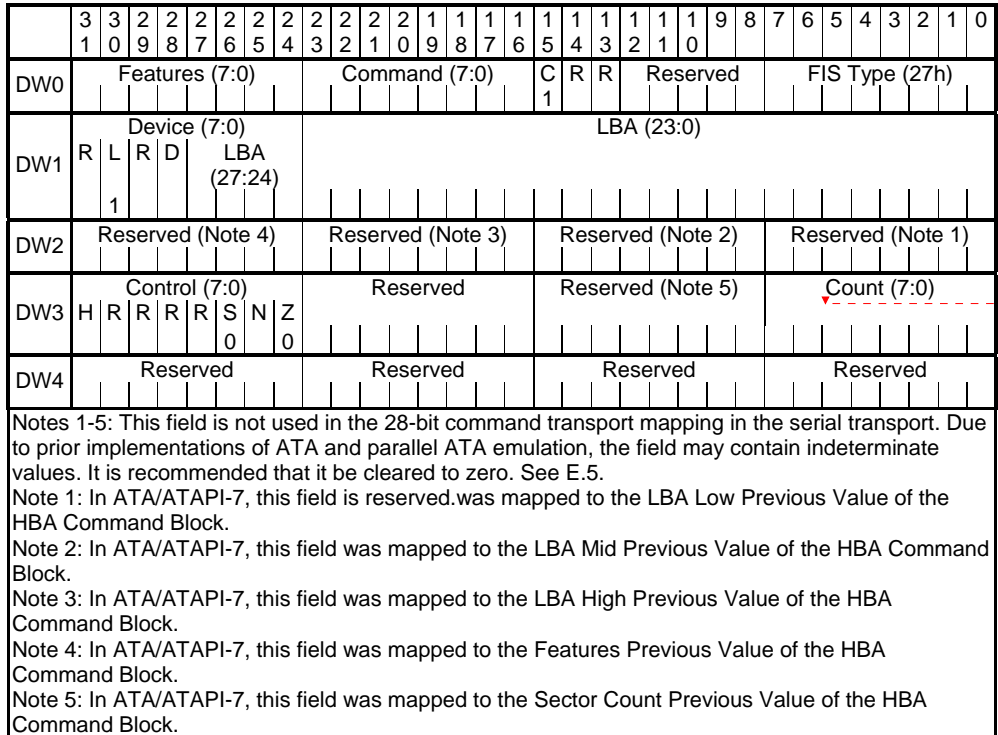


Figure 63 - Command FIS - 28-bit command transport and con-data command transport FIS layout

Field Definitions:

FIS Type - Set to a value of 27h.

R – Reserved.

C - This bit shall be set to one to indicate that the FIS is a Command FIS. (See 7.5.2 for a Control FIS description when C bit is cleared to zero.)

Command (7:0) – Contains the Command field bits 7:0 of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Command register of the HBA Command Block.

Features (7:0) – Contains 7:0 of the Features field of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Features register of the HBA Command Block.

LBA (23:0) – Contains bits 23:0 of the LBA field of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the LBA Low register, LBA Mid register, and LBA High register, in that order, of the HBA Command Block.

Device (7:0) – Contains bits 7:0 of the Device field of the command. These bits are individually assigned to specific functions as defined below. In ATA/ATAPI-7 parallel emulation, these bits correspond to the Device Register of the HBA Command Block.

LBA (27:24) – Contains bits 27:24 of the LBA field of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the Device Register bits 0:3 of the HBA Command Block.

L – Fixed for all commands in the 28-bit feature set. This bit shall be set to one. In ATA/ATAPI-7 parallel emulation, this bit corresponds to the LBA bit.

D – Variable. In ATA/ATAPI-7 parallel emulation, this bit corresponds to the DEV bit. The bit is not used in the serial transport, and may be transmitted with a zero or a one value. It is recommended that it be cleared to zero.

Count (7:0) - Contains bits 7:0 of the Count field of the command. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Sector Count register of the HBA Command Block.

Deleted: Sector

Control (7:0) – Contains bits that are individually assigned to specific functions as defined below. In ATA/ATAPI-7 parallel emulation, these bits were mapped to the HBA Control Register of the HBA Command Block.

Z – Fixed. This bit shall be cleared to zero.

N – Variable. In ATA/ATAPI-7 parallel emulation, this bit was mapped as the nIEN bit. The bit is not used in the serial transport, and may be transmitted with a zero or a one value. It is recommended that it be cleared to zero.

S – Fixed. This bit shall be cleared to zero. In ATA/ATAPI-7 parallel emulation, this bit was mapped as the SRST bit. (See 7.5.2 for SRST description of the Control FIS when C bit is cleared to zero.)

H – Variable. In ATA/ATAPI-7 parallel emulation, this bit was mapped as the HOB bit. The bit is not used in the serial transport, and may be transmitted with a zero or a one value. It is recommended that it be cleared to zero.

7.5.2 Control FIS

The Control FIS is a host to Device FIS, FIS Type 27h, with the C bit cleared to zero. In the serial transport, there is one control set mapping into the Control FIS.

7.5.2.1 Control FIS – control mapping

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|-------------------|---|---|---|---|---|---|-------------------|---|---|---|---|---|---|-------------------|---|---|----------|---|---|---|-------------------|---|---|----------------|---|---|---|---|--|--|
| | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | |
| | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | |
| DW0 | Reserved (Note 2) | | | | | | | Reserved (Note 1) | | | | | | | C | R | R | Reserved | | | | | | | FIS Type (27h) | | | | | | |
| DW1 | Reserved (Note 4) | | | | | | | Reserved (Note 3) | | | | | | | | | | | | | | | | | | | | | | | |
| DW2 | Reserved (Note 6) | | | | | | | Reserved (Note 5) | | | | | | | | | | | | | | | | | | | | | | | |
| DW3 | Control (7:0) | | | | | | | Reserved | | | | | | | Reserved (Note 8) | | | | | | | Reserved (Note 7) | | | | | | | | | |
| | H | R | R | R | R | S | N | Z | | | | | | | | | | | | | | | | | | | | | | | |
| DW4 | Reserved | | | | | | | Reserved | | | | | | | Reserved | | | | | | | Reserved | | | | | | | | | |

Notes 1-8: This field is not used in the serial transport. Due to prior implementations of ATA and parallel ATA emulation, this field may contain indeterminate values. It is recommended that it be cleared to zero(see E.5).
 Note 1: In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the HBA Command register of the Command Block.
 Note 2: In ATA/ATAPI-7 parallel emulation, these bits correspond to bits 7:0 of the Features register of the HBA Command Block.
 Note 3 - In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the to LBA Low register (current), LBA Mid register (current), and LBA High register (current), in that order, of the HBA Command Block.
 Note 4: In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Device register of the HBA Command Block.
 Note 5: In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the LBA Low register (previous), LBA Mid register (previous), and LBA High register (previous), in that order, of the HBA Command Block.
 Note 6: In ATA/ATAPI-7 parallel emulation, these bits correspond to bits 15:8 of the Features (previous) register.
 Note 7: In ATA/ATAPI-7 parallel emulation, these bits correspond to bits 7:0 of the Sector Count register of the HBA Command Block.
 Note 8: In ATA/ATAPI-7 parallel emulation, these bits correspond to bits 15:8 of the Sector Count (previous) register of the HBA Command B ock.

Figure 64 - Control FIS – control transport FIS layout

Field Definitions:

FIS Type - Set to a value of 27h.

C - This bit shall be set to zero to indicate that the FIS is a Control FIS. (See 7.5.2 for a Control FIS description when the C bit is set to one.)

R - Reserved.

Control (7:0) – Contains bits that are individually assigned to specific functions as defined below.

Z – Fixed. This bit shall be cleared to zero.

N – Variable. In ATA/ATAPI-7 parallel emulation, this bit was mapped as the nIEN bit. The bit is not used in the serial transport, and may be transmitted with a zero or a one value. It is recommended that it be cleared to zero.

S – SRST (Soft Reset). If set to one, the bit indicates that a Soft Reset operation has been requested by the host. (See 8.4 for description of the Soft Reset.)

H – Variable. In ATA/ATAPI-7 parallel emulation, this bit was mapped as the HOP bit. The bit is not used in the serial transport, and may be transmitted with a zero or a one value. It is recommended that it be cleared to zero.

In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the the HBA Control register of the HBA Command Block (see E.5).

7.5.3 Response FIS

The Response FIS is a device to host FIS, FIS Type 34h. In the serial transport, there are two kinds of command response mappings into the Response FIS.

- 48-bit command response mapping
- 28-bit command response mapping

7.5.3.1 Response FIS - 48-bit command response mapping

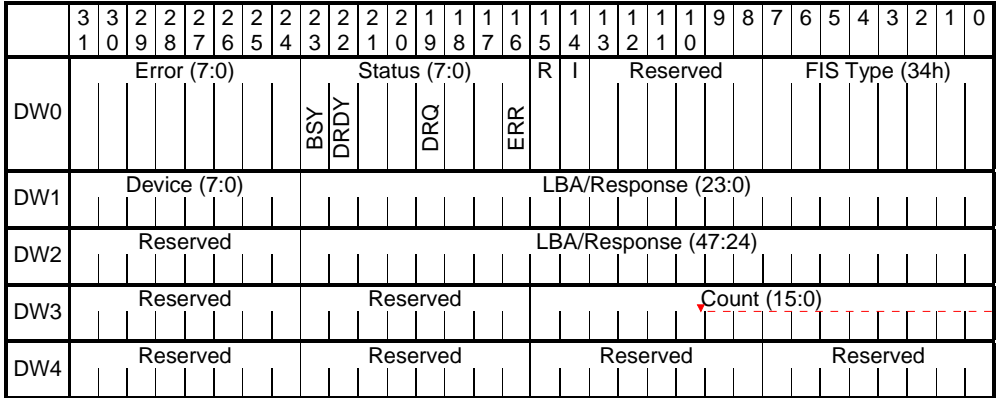


Figure 65 - Response FIS - 48-bit command response transport FIS layout

Field Definitions:

FIS Type - Set to a value of 34h.

I – Interrupt Pending Bit. Reflects the Interrupt Pending state of the device. The behavior of this bit is command dependent.

R – Reserved.

Status (7:0) – Contains bits 7:0 of the Status field of the Command response. The use of these bits are command dependent. Their function is defined by the device transport state machines (see 7.7) and host transport state machines (see 7.6) and the ATA command (see ATA8-ACS). In ATA/ATAPI-7 parallel emulation, Status bits 7:0 correspond to the content of the Status register of the command block.

Bit 0: ERR. Indicates that an error condition occurred during the execution of a command. .

Bit 1-2: Command dependent.

Bit 3: DRQ (Data Request). Indicates that the device is ready to transfer data between the host and the device.

Bit 4-5: Command dependent.

Bit 6: DRDY (Device Ready). Indicates that the device is ready to receive and (attempt to) execute a command.

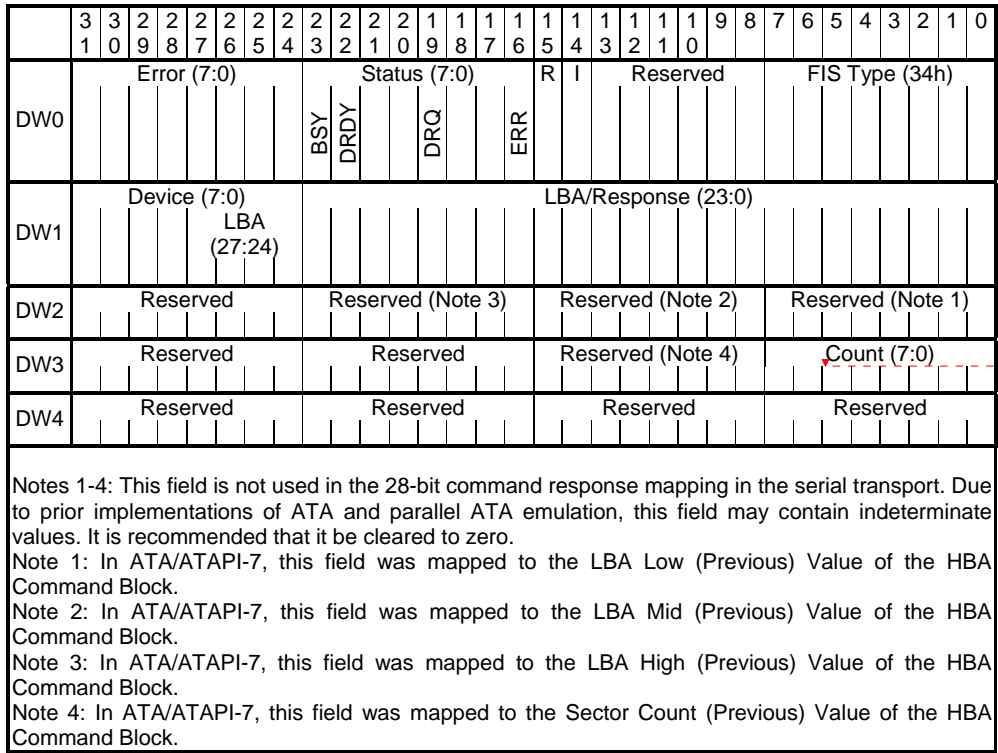
Bit 7: BSY (Busy). Indicate that the device is busy. This bit may be used in combination with DRQ bit (bit 3) to indicate that the device is ready to transfer data.

Error (7:0) – Contains bits 7:0 of the Error field of the command response. The content is command dependent as defined in the command error response (see ATA8-ACS) . In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Error register of the HBA Command Block.

- LBA/Response (23:0) – Contains bits 23:0 of the LBA/Response field of the command response (see ATA8-ACS). In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the LBA Low register (current), LBA Mid register (current), and LBA High register (current), in that order, of the HBA Command Block.
- Device (7:0) – Contains bits 7:0 of the Device field of the command response (see ATA8-ACS). The content is command dependent. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Device register of the HBA Command Block.
- LBA/Response (47:24) – Contains bits 24:47 of the LBA/Response field of the command response. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the LBA Low register (previous), LBA Mid register (previous), and LBA High register (previous), in that order, of the HBA Command Block.
- Count (15:0) – Contains bits 15:0 of the Count field of the command response. In ATA/ATAPI-7 parallel emulation, Sector Count bits 7:0 correspond to the content of the Sector Count register (current), and Sector Count bits 15:8 correspond to the content of the Sector Count register (previous) of the Command Block.

Deleted: Sector

7.5.3.2 Response FIS - 28-bit command and non-data command response mapping



Deleted: Sector

Figure 66 - Response FIS - 28-bit command response and non-data command response transport FIS layout

Field Definitions:

FIS Type - Set to a value of 34h.

I – Interrupt Pending Bit. Reflects the Interrupt Pending state of the device. The behavior of this bit is command dependent.

R - Reserved.

Status – Contains bits 7:0 of the Status field of the Command response. The use of these bits are command dependent. Their function is defined by the device transport state machines (see 7.7) and host transport state machines (see 7.6) and the ATA command structure (see ATA8-ACS). In ATA/ATAPI-7 parallel emulation, Status bits 7:0 correspond to the content of the Status register of the command block.

Bit 0: ERR. Indicates that an error condition occurred during the execution of a command.

Bit 1-2: Command dependent.

Bit 3: DRQ (Data Request). Indicates that the device is ready to transfer data between the host and the device.

Bit 4-5: Command dependent.

Bit 6: DRDY (Device Ready). Indicates that the device is ready to receive and (attempt to) execute a command.

Bit 7: BSY (Busy). Indicates that the device is busy. This bit may be used in combination with DRQ bit (bit 3) to indicate that the device is ready to transfer data.

Error (7:0) – Contains bits 7:0 of the Error field of the command response. The content is command dependent as defined in the command error response. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Error register of the Shadow Command Block.

LBA/Response (23:0) – Contains bits 23:0 of the command LBA/Response field of the response. In ATA/ATAPI-7 parallel emulation, these bits were mapped to LBA Low, LBA Mid, and LBA High Registers of the HBA Command block.

Device (7:0)– Contains bits 7:0 of the Device field of the command response. The content is command dependent. Bits 0:3 are mapped to LBA/response field bits 27:24 in 28-bit data commands responses. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Device register of the HBA Command Block.

~~Count (7:0) – Contains bits 7:0 of the Count field of the command response. In ATA/ATAPI-7 parallel emulation, bits 7:0 correspond to the content of the the Sector Count register (current) of the HBA Command Block.~~

Deleted: Sector

7.5.4 Set Device Bits FIS

The Set Device Bits FIS is a Device to host FIS, FIS Type A1h. The Set Device Bits FIS is used to update the host adapter with changes that occurred to the condition or the status of the device after BSY and DRQ have been cleared by a Response FIS. The FIS, if implemented by the host adapter, also provides a mechanism (bit 14) to signal to the host to generate an interrupt when the device has performed an operation that requires an interrupt.

7.5.4.1 Set Device Bits FIS

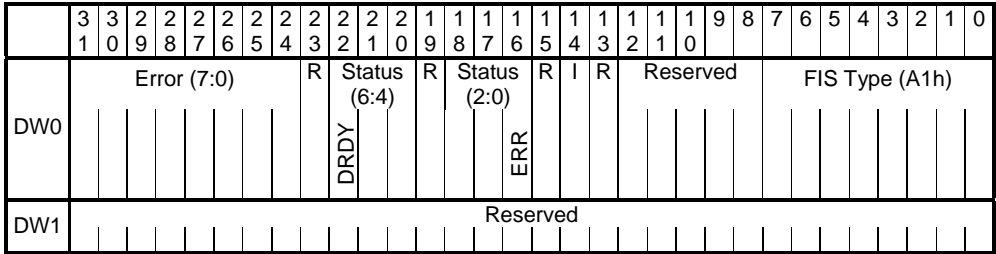


Figure 67 – Set Device Bits - device to host transport FIS layout

Field Definitions:

- FIS Type – Set to a value of A1h.
- I – Interrupt Pending. This bit signals to the host adapter that the device has performed an operation that requires the interrupt pending bit to be set. Prior implementation of ATA host adapters, which use parallel ATA emulation use this bit to set interrupts to the host when both BSY and DRQ bits have been cleared to zero by a Response FIS. A device shall not modify this bit based on the state of the nIEN bit received in a Command FIS or Control FIS.
- Error – Contains bits 7:0 of the Error field of the command response. The content of these bits is command dependent. In ATA/ATAPI-7 parallel emulation, these bits correspond to the bits in the Error register of the HBA Command Block.
- Status (2:0) - Contains bits 2:0 of the Status field of the command response. The use of these bits are command dependent. Their function is defined by the device transport state machines (see 7.7) and host transport state machines (see 7.6) and the ATA command structure (see ATA8-ACS). In ATA/ATAPI-7 parallel ATA emulation, these bits correspond to bits 2:0 in the Status register of the HBA Command Block.
- Bit 0: ERR/CHK (Error/Check). Indicates that an error or check condition occurred during the execution of the command.
- Bit 2:1: Use of these bits is command dependent.
- Status (4:6) - Contains bits 6:4 of the Status field of the command response. The use of these bits are command dependent. Their function is defined by the device transport state machines (see 7.7) and host transport state machines (see 7.6) and the ATA command structure (see ATA8-ACS). In ATA/ATAPI-7 parallel emulation, these bits correspond to bits 6:4 in the Status register of the HBA Command Block.
- Bit 4-5: Use of these bits is command dependent.
- Bit 6: DRDY (Device Ready) Indicates that the device is ready to receive and (attempt to) execute a command.

7.5.5 DMA Activate - device to host

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|--------------|---|---|---|---|---|---|---|--------------|---|---|---|---|---|--------------|---|---|---|----------------|---|---|---|---|---|---|---|---|---|---|
| | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DW0 | Reserved (0) | | | | | | | | Reserved (0) | | | R | R | R | Reserved (0) | | | | FIS Type (39h) | | | | | | | | | | |

Figure 68 - DMA Activate - device to host FIS layout

Field Definitions

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

7.5.5.1 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. This is similar to the DMARQ mechanism by which a Parallel Transport 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 FISes that are needed.

7.5.5.2 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.

7.5.5.3 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.

7.5.6 First Party DMA Setup (bidirectional)

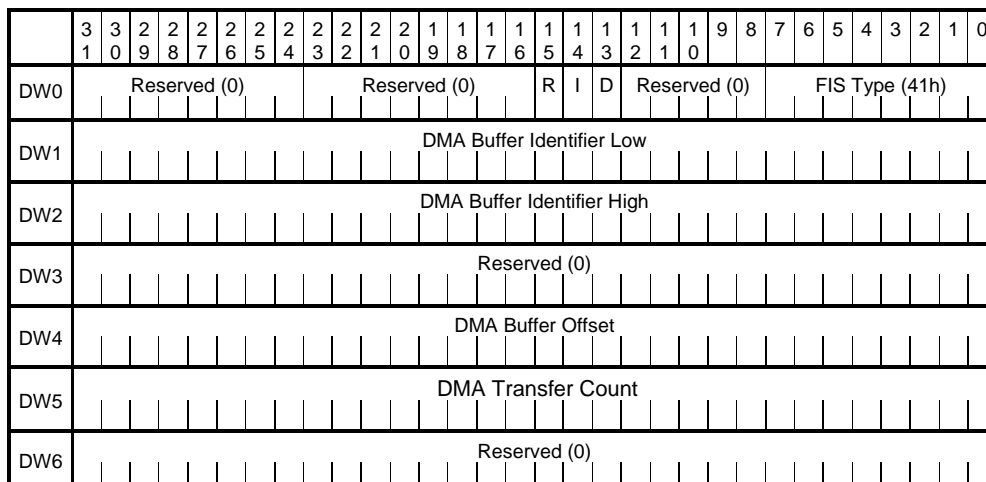


Figure 69 - First Party DMA Setup - device to host FIS layout

Field Definitions

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 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

7.5.6.1 Description

The First Party DMA Setup - device to host or host to device FIS (FPDMA Setup FIS) is the mechanism by which first party DMA access to memory is initiated. 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.

7.5.6.2 Transmission

A device or host transmits a FPDMA Setup FIS as the first step in performing a first party DMA access. The purpose of the FPDMA Setup FIS is to establish DMA hardware context for one or more data transfers.

A FPDMA Setup FIS is required only when the DMA context is to be changed. Multiple Data - host to device or device to host FISes 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 FPDMA Setup 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 FPDMA Setup FIS is used to alter the DMA context. This mechanism allows for the efficient streaming of data into a buffer.

7.5.6.3 Reception

Upon receiving a FPDMA Setup 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 FPDMA Setup FIS with the D bit cleared to 0, the recipient of the FPDMA Setup FIS responds with one or more Data - host to device or device to host FISes until the DMA count is exhausted. After a valid FPDMA Setup 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 FISes 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.

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 (DW1), and DW2. Note that pattern dwords DW1, and DW2 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 that 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 Activate 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 negated, 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 except FIS type are vendor specific in this mode.

7.5.7.2 Transmission

The initiator transmits a BIST Activate to the recipient in order to initiate the BIST mode of operation.

7.5.7.3 Reception

Upon receiving a BIST Activate, the recipient shall begin operations as per the BIST Activate FIS, described above.

7.5.8 PIO Setup FIS

The PIO Setup FIS is a device to host FIS, FIS Type 5Fh. The PIO Setup FIS is used by the device to provide the host adapter with the data transfer count and DRQ block status during the PIO data phase. This information allows the host adapter to control PIO data transfers. There are two command Response mappings into a Response FIS in the serial transport.

Deleted: structure

- a) 48-bit command response PIO Setup mapping
- b) 28-bit command response PIO Setup mapping

7.5.8.1 PIO Setup FIS - 48-bit command PIO setup mapping

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|----------------|----|----|----|----|----|----|-----------------------|----|----|----|----|----|----|--------------|----|----|----------|----|----|----|----|---|---|----------------|---|---|---|---|---|---|---|
| | 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 |
| DW0 | Error (7:0) | | | | | | | Status (7:0) | | | | | | | R | I | D | Reserved | | | | | | | FIS Type (5Fh) | | | | | | | |
| DW1 | Device (7:0) | | | | | | | LBA/Response (23:0) | | | | | | | | | | | | | | | | | | | | | | | | |
| DW2 | Reserved | | | | | | | LBA/Response (24-47) | | | | | | | | | | | | | | | | | | | | | | | | |
| DW3 | E_Status (7:0) | | | | | | | Reserved | | | | | | | Count (15:0) | | | | | | | | | | | | | | | | | |
| DW4 | Reserved | | | | | | | Transfer Count (15:0) | | | | | | | | | | | | | | | | | | | | | | | | |

Deleted: Sector

Figure 71 - PIO Setup – 48-bit command device to host transport FIS layout

Field Definitions:

FIS Type - Set to a value of 5Fh.

I – Interrupt Pending. This bit reflects the Interrupt Pending state of the device. It is defined by the device transport protocol (see 7.7).

D - Indicates whether the host memory is being written or read by the device. 1 = Write (device to host), 0 = Read (host to device).

R - Reserved.

Status (7:0) – Contains bits 7:0 of the Status field of the command response. It is used by the host adapter transport for ending status (Status bits BSY=0 and DRQ=0), or to prepare for a subsequent data transfer (Status bits BSY=1 or DRQ=1). If Status bits BSY=1 or DRQ=1, the operation of these bits is defined by the host and device transport state machines (see 7.6 and 7.7). Definition of these bits is the same as for the Response FIS - 48-bit command structure mapping (see 7.5.3.1). Note that Status field and the E_Status field contain different information.

Error (7:0) – Contains bits 7:0 of the Error field of the command response. These bits are command dependent. Values are only valid if E_Status bits BSY=0 and DRQ=0. If E_Status bit BSY=1 or DRQ=1, the content is indeterminate. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Error register of the HBA Command Block.

LBA/Response (23:0) – Contains bits 23:0 of the LBA/Response field of the command response. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the LBA Low register (current), LBA Mid register (current), and LBA High register (current), in that order, of the HBA Command Block.

Device (7:0) – Contains bits 7:0 of the Device field of the command response. These bits are command dependent. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Device register of the HBA Command Block. The values are only valid if both the BSY and DRQ bits in the E_Status field are cleared. If E_Status bits BSY=1 or DRQ=1, the content is indeterminate.

LBA/Response (47:24) – Contains bits 47:24 of the LBA/Response field of the command response. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the LBA Low register (previous), LBA Mid register (previous), and LBA High register (previous), in that order, of the HBA Command Block.

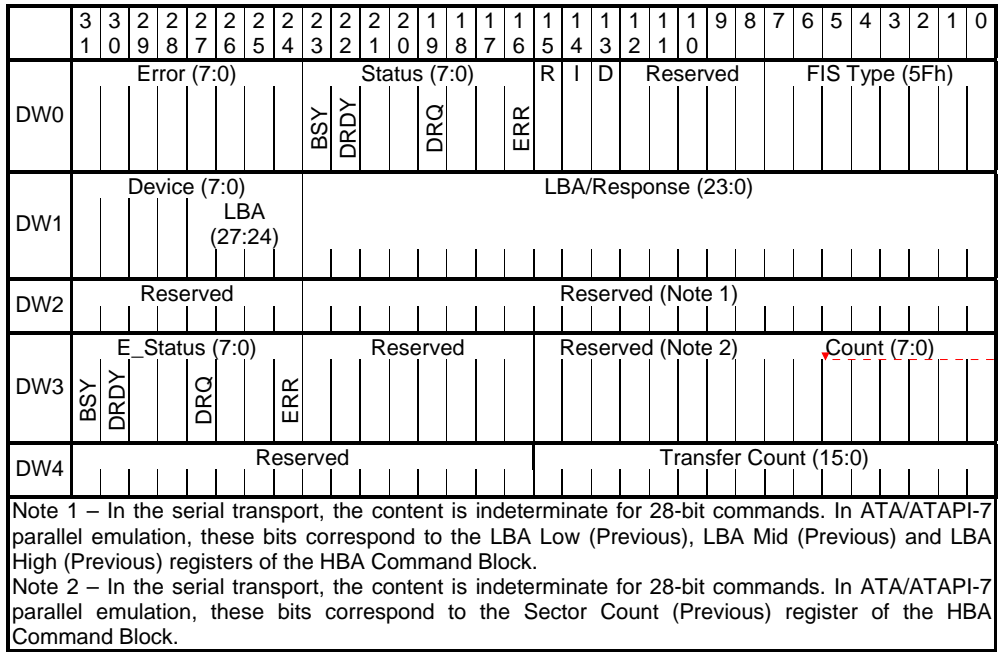
Count (15:0) – Contains bits 15:0 of the Count field of the command response. The values are only valid if E_Status bits BSY=0 and DRQ=0. If E_Status bits BSY=1 or DRQ=1, the content is indeterminate. In ATA/ATAPI-7 parallel emulation, Sector Count bits (7:0) correspond to the content of the Sector Count register (current), and Sector Count bits (15:8) correspond to the content of the Sector Count register (previous) of the HBA Command Block.

Deleted: Sector

E_Status (7:0)– Contains bits 7:0 of the E-Status field of the command response at the expiration of the Transfer Count for this PIO Setup, if this is the final PIO Setup for this command. If E_Status bits BSY=1 or DRQ=1, this is not the final PIO Setup for this command, and the content of this field is used by the host adapter ending status (see Status above) for data transfer control as defined by the host and device transport state machines (see 7.6 and 7.7). Definition of these bits is the same as for the Response FIS - 48-bit command structure Response Mapping (see 7.5.3.1).

Transfer Count (15:0) – Contains 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).

7.5.8.2 PIO Setup FIS - 28-bit command PIO setup mapping



Deleted: Sector

Figure 72 - PIO Setup – 28-bit command device to host transport FIS layout

Field Definitions:

FIS Type - Set to a value of 5Fh.

I – Interrupt Pending. This bit reflects the Interrupt Pending state of the device. It is defined by the device transport protocol (see 7.7).

D - Indicates whether the host memory is being written or read by the device. 1 = Write (device to host), 0 = Read (host to device).

R - Reserved.

Status (7:0) – Contains bits 7:0 of the Status field of the command response. It is used by the host adapter transport for ending status if (Status bits BSY=0 and DRQ=0), or to prepare for a subsequent data transfer if (Status bits BSY=1 or DRQ=1). If Status bits BSY=1 or DRQ=1, the operation of these bits is defined by the host and device transport state machines (see 7.6 and 7.7). Definition of these bits is the same as for the Response FIS – 28-bit command response mapping (see 7.5.3.1). Note that Status field and the E_Status field contain different information.

Error (7:0) – Contains bits 7:0 of the Error field of the command response. These bits are command dependent. Value are only valid if E-Status bits BSY=0 and DRQ=0. If E_Status bit BSY=1 or DRQ=1, the contents are indeterminate. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the Error register of the HBA Command Block.

LBA/Response (23:0) – Contains bits 23:0 of the LBA/Response field of the command response. In ATA/ATAPI-7 parallel emulation, these bits were mapped to LBA Low, LBA Mid, and LBA High registers, in that order, of the HBA Command Block.

Device (7:0) - Conatins bits 7:0 of the Device field of the command response. These bits is command dependent. In ATA/ATAPI-7 parallel emulation, these bits correspond to the content of the

Device register of the HBA Command Block. The value are only valid if E_Status bits BSY=0 and DRQ=0. If E_Status bit BSY=1 or DRQ=1) the content is indeterminate.

~~Count (7:0) – Contains bits 7:0 of the Count field of the command response. These value is only valid if E-Status bits BSY=0 and DRQ=0. If E_Status bits BSY=1 or DRQ=1, the content is indeterminate. In ATA/ATAPI-7 parallel emulation, this field was mapped to the Current Value in the Sector Count Register.~~

Deleted: Sector

E_Status (7:0) – Contains 7:0 of the E_Status field of the command response at the expiration of the transfer count (see below) for this PIO Setup, if this is the final PIO Setup for this command. If E_Status bits BSY=1 or DRQ=1, this is not the final PIO Setup for this command, and the content of this field are used by the host adapter ending status (see Status above) for data transfer control as defined by the host and device transport state machines (see 7.6 and 7.7). The definition of these bits is the same as for the Response FIS - 48-bit command response mapping (see 7.5.3.1).

Transfer Count (15:0) – Contains the number of bytes to be transferred in the subsequent Data FIS (see 7.5.9). The Transfer Count value shall be non-zero and the low order bit shall be zero (even number of bytes transferred).

7.5.9 Data (bidirectional)

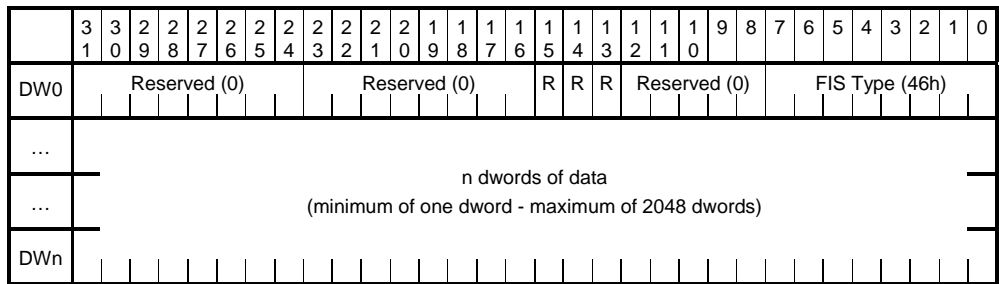


Figure 73 - Data FIS layout

Field Definitions

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 7.5.9.1.

R - Reserved

7.5.9.1 Description

The Data - host to device and the Data - device to host FISes (Data FISes) are used for transporting payload data, such as the data read from or written to a number of sectors on a hard drive. 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.

7.5.9.2 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 FISes 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.

7.5.9.3 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.

7.6 Host transport states

7.6.1 Host transport idle state diagram

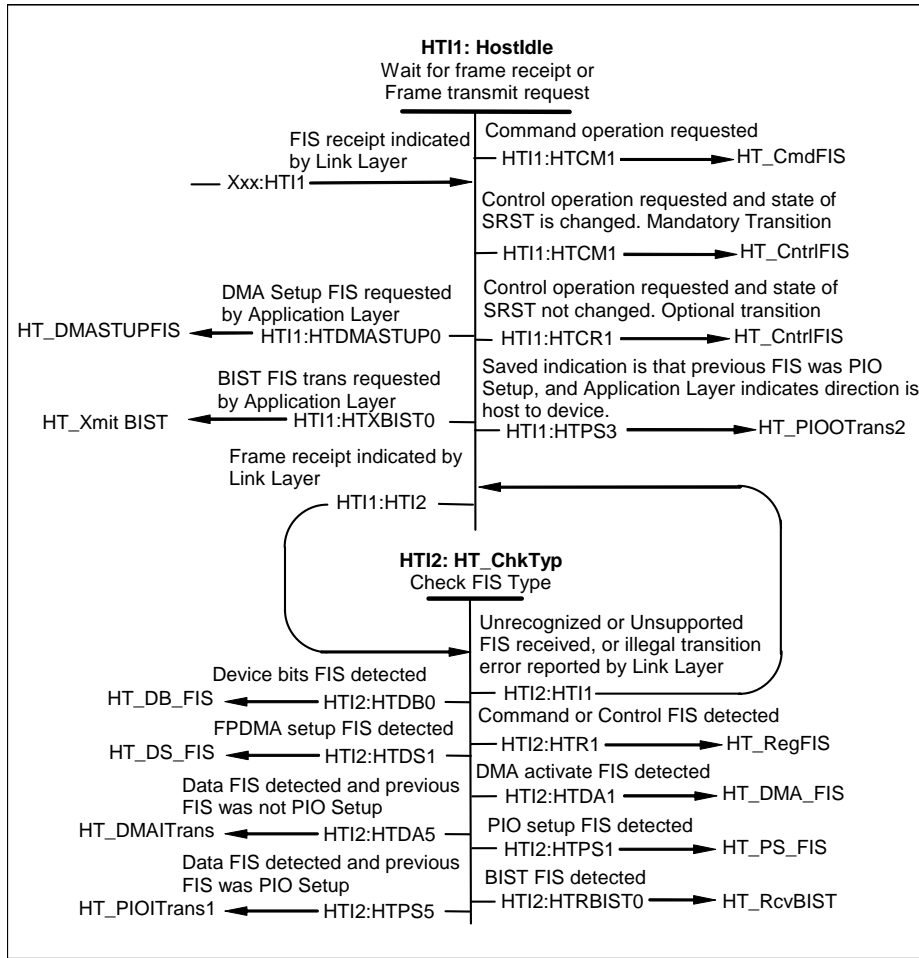


Figure 74 – Host transport idle state diagram (States HTI1-HTI2)

HTI1: HT_Hostidle state: This state is entered when a Frame Information Structure (FIS) transaction has been completed by the transport layer.

When in this state, the transport layer waits for a command operation to be requested, a control operation to be requested, or the link layer to indicate that an FIS is being received.

Transition HTI1:HTCM1: When a command operation is requested, the transport layer shall make a transition to the HTCM1: HT_CmdFIS state.

Transition HTI1:HTCR1: When a control operation is requested, 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 control operation request 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 Activate FIS 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. Data is transferred to the device with host to device data FISes.

HTI2: HT_ChkTyp state: This state is entered when the link layer indicates that an FIS is being received.

When in this state, the transport layer checks the FIS type of the incoming FIS.

Transition HTI2:HTR1: When the incoming FIS is a Response FIS 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_DMAlTrans 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.

7.6.2 Host Transport transmit Command FIS diagram

This protocol builds an FIS that contains a ATA8-ACS command and sends it to the device when the software driver or BIOS requests a command operation.

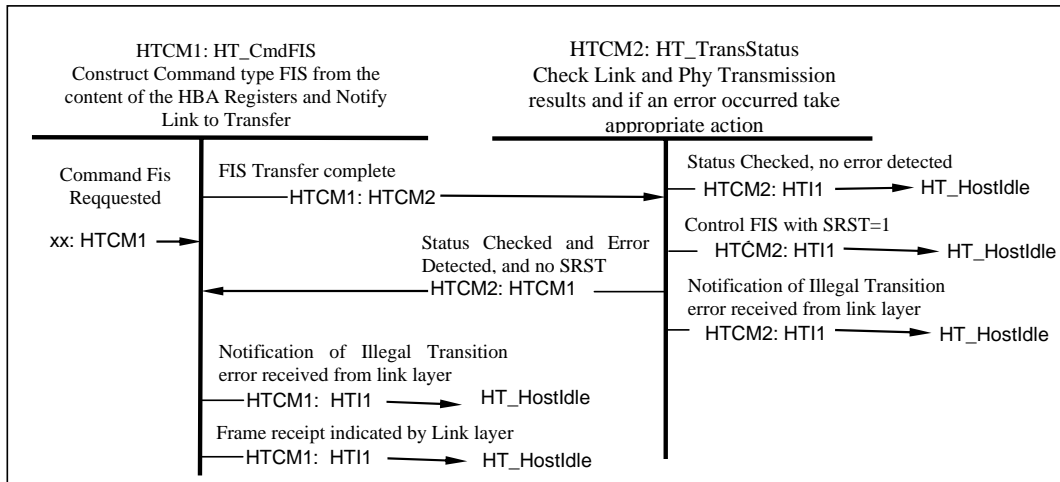


Figure 75 – Host transport transmit Command FIS diagram (States HTCM1-HTCM2)

HTCM1: HT_CmdFIS state: This state is entered when a command operation is requested.

When in this state, the transport layer shall construct a Command 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 make 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 (editors note: receipt response, define, global change?) for the FIS and take appropriate error handling action if required.

Transition HTCM2:HTI1: When the FIS status has been checked and no errors detected, the transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTCM2:HTCM1: When the FIS status has been checked and an error has been detected, and the host is not requesting a Control FIS with SRST=1, (editors note: changed wording), the transport layer shall transition to the HTCM1: HT_CmdFIS state.

Transition HTCM2:HTI1: When the FIS status has been checked, an error has been detected, and the host is requesting a Control FIS with SRST=1 (editors note: changed wording), 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.

7.6.3 Host Transport transmit control FIS diagram

This protocol transmits a FIS that requests a control operation.

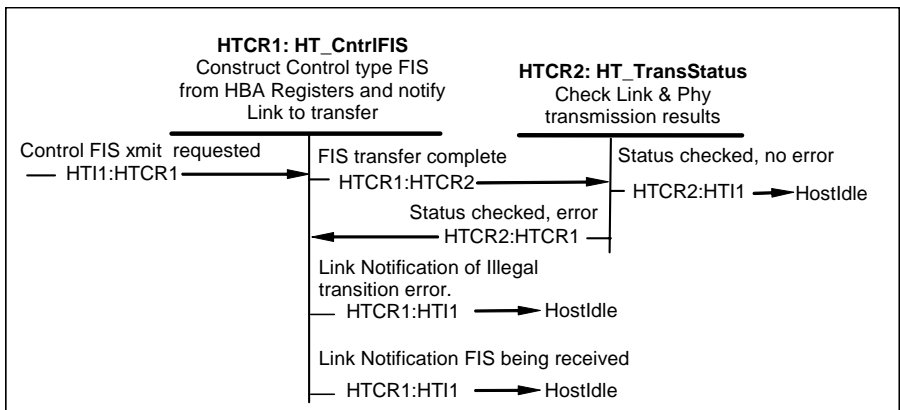


Figure 76 – Host transport transmit control FIS diagram (States HTCR1-HTCR2)

HTCR1: HT_Cntrl_FIS state: This state is entered when a control operation is requested.

When in this state, the transport layer shall construct a Control 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:HT11: When the transport layer receives notification from the link layer of an illegal state transition, the transport layer shall make a transition to the HT11: HT_HostIdle state.

When the link layer indicates that an FIS is being received, the transport layer shall make a transition to the HT11: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:HT11: When the FIS status has been checked and no errors have been detected, the transport layer shall transition to the HT11: HT_HostIdle state.

Transition HTCR2:HTCR1: When the FIS status has been checked and at least one error has been detected, the transport layer shall transition to the HTCR1: HT_Cntrl_FIS.

7.6.4 Host Transport transmit First Party DMA Setup - device to host or host to device FIS state diagram

This protocol 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 Pending bit.

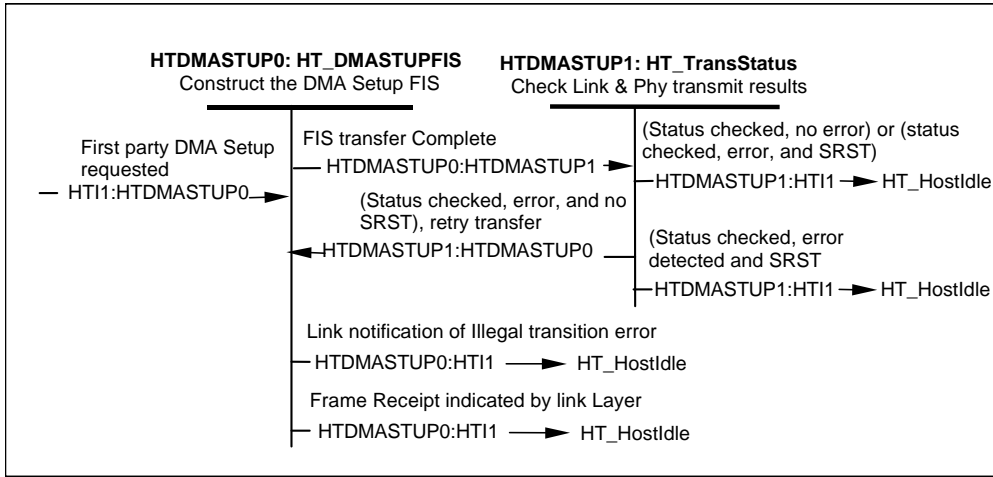


Figure 77 – Host transport transmit First Party DMA setup - device to host or host to device FIS (States HTDMASTUP0-HTDMASTUP1)

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.

HTDMASTUP1: 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 checked, and no error detected, the transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTDMASTERUP1:HTDMASTERUP0: When the FIS status has been checked, an error detected and the host is not requesting a Control FIS with SRST=1, the transport layer shall report status to the link layer, and retry this transfer by transitioning to the HTDMASTERUP0: HT_DMASTERUPFIS state.

Transition HTDMASTERUP1:HTI1: If the host has is requesting a Control FIS with SRST=1, or the DEVICE RESET command is requested, the transport layer shall inform the link layer to send SYNC primitives, and the transport layer shall transition to the HTI1: HT_HostIdle state.

7.6.5 Host Transport transmit BIST Activate FIS

This protocol 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.

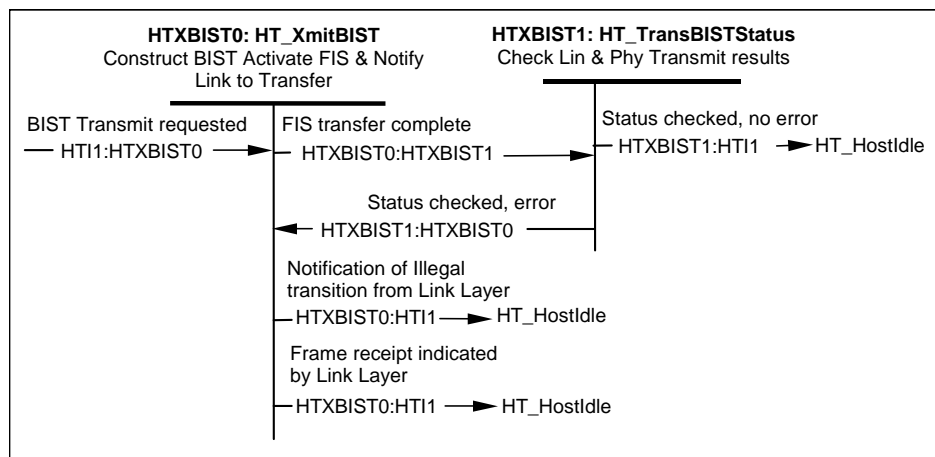


Figure 78 – Host transport transmit BIST activate FIS (States HTXBIST0-HTXBIST1)

HTXBIST0: HT_XmitBIST state: This state is entered to send a BIST Activate 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.

7.6.6 Host Transport decompose Response FIS diagram

This protocol receives a Response FIS from the device.

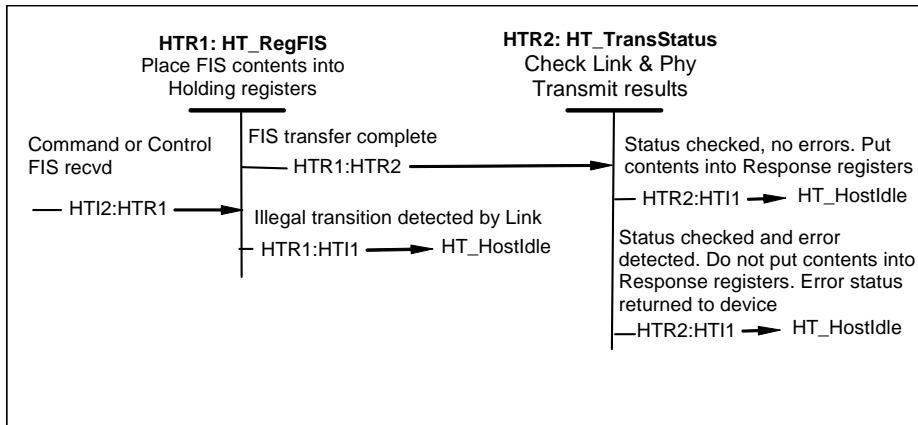


Figure 79 - Host transport decompose Response FIS diagram (States HTR1- HTR2)

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 Command FIS or Control FIS type.

When in this state, the transport layer shall decompose the Response 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 checked and no errors detected, the contents of the holding registers shall be placed in the HBA response registers 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 checked and at least one error detected, the contents of the holding registers shall not be transferred to the HBA response registers, error status shall be returned to the device, and the transport layer shall transition to the HTI1: HT_HostIdle state.

7.6.7 Host Transport decompose a Set Device Bits FIS state diagram

This protocol receives an FIS from the device containing bits of HBA response registers and places that content into the appropriate HBA response registers.

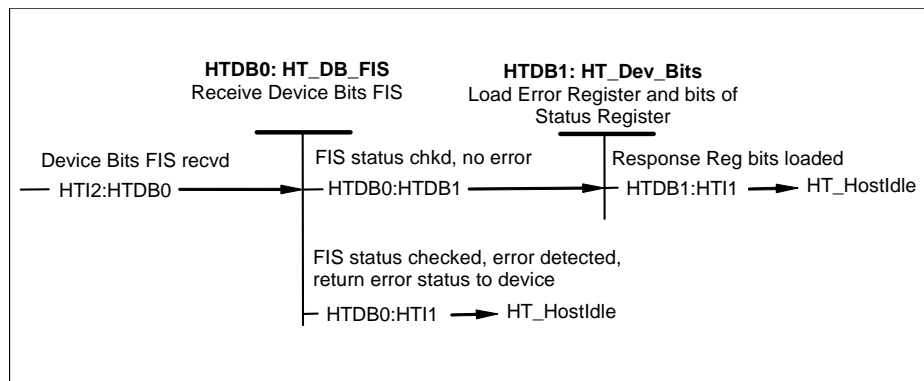


Figure 80 – Host transport decompose Set Device Bits FIS state diagram (States HTDB0-HTDB1)

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 content of the Set Device Bits FIS are loaded into the appropriate registers of the HBA response registers. If the I bit in the FIS is set to one and if both the BSY bit and the DRQ bit have been cleared to zero by a previous Response FIS, then the host adapter shall enter an Interrupt Pending state.

Note: In ATA-7 Parallel emulation, these bits were mapped into the host’s HBA Command and Control registers as follows: When in this state, the Error field of the received FIS shall be loaded into the appropriate host adapter’s Error register. The data in the Status-Hi field of the received FIS shall be loaded into bits 6, 5, and 4 of the HBA Status register. The data in the Status-Lo field of the received FIS shall be loaded into bits 2, 1, and 0 of the HBA Status register. Bit 7, BSY, and bit 3, DRQ, in the HBA 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 HBA 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.

7.6.8 Host Transport decompose a DMA Activate FIS diagram and DMA Data Transfer

This protocol receives an FIS that requests a 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.

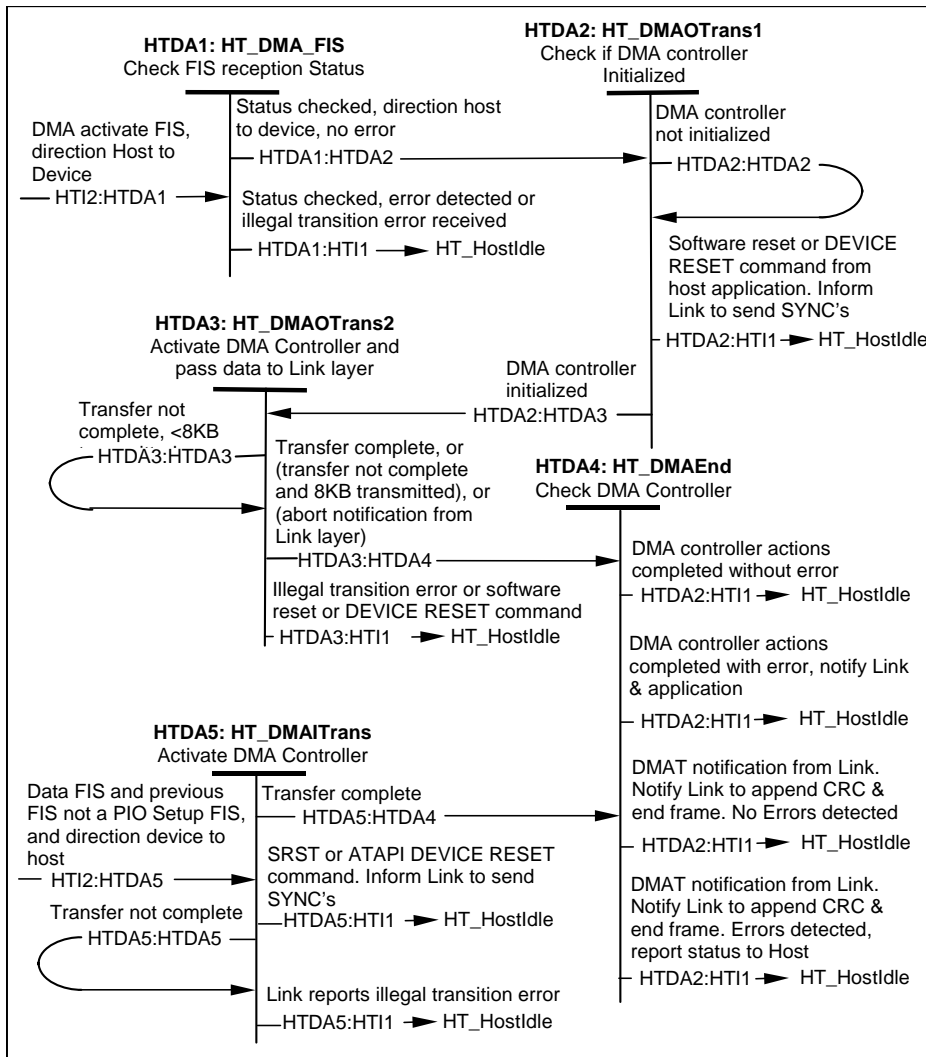


Figure 81 – Host transport decompose DMA activate FIS diagram (States HTDA1-HTDA5)

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 requested a Control FIS with the SRST =1, or the DEVICE RESET command operation is requested, 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 2048 dwords 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 2048 dwords 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 6.4.6.

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 requested a Control FIS with the SRST =1, or the DEVICE RESET command operation is requested, 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 2048 dwords 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 6.4.6)

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 host has requested a Control FIS with the SRST =1, 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.

7.6.9 Host Transport decompose a PIO Setup FIS state diagram

This protocol 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.

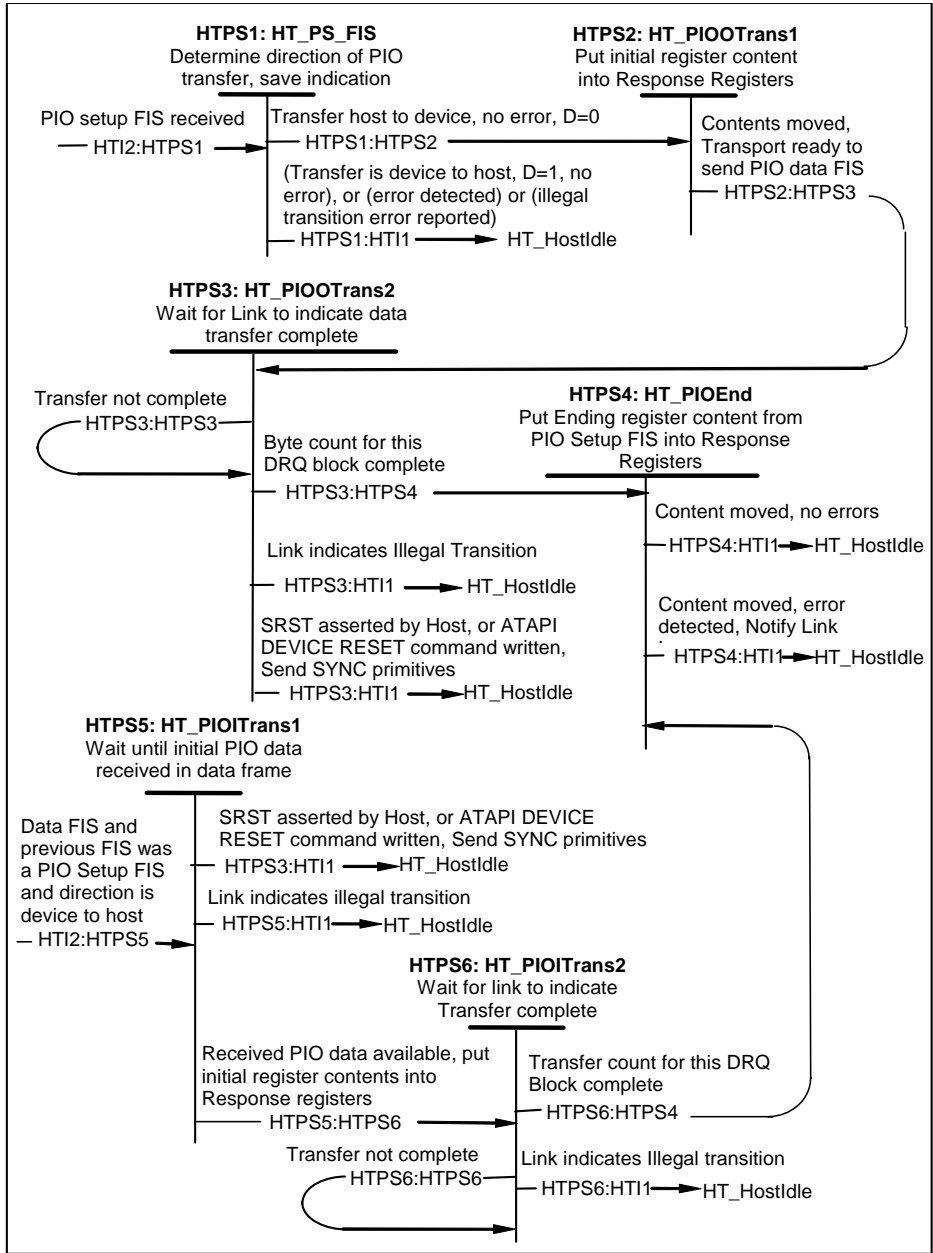


Figure 82 – Host transport decompose PIO setup FIS state diagram (States HTPS1-HTPS6)

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 contents (using Status field for HBA Status register) into the HBA response registers, including the FIS byte count into appropriate registers, and set the Interrupt Pending bit to one if the FIS indicates to do so.

Transition HTPS2:HTPS3: When the FIS initial contents have been placed into the HBA response registers, 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.

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.

When the host has requested a Control FIS with the SRST =1, or a DEVICE RESET command operation is requested, 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.

When in this state, the transport layer shall place the E_Status field from the previously received PIO Setup FIS into the HBA response registers.

Transition HTTPS4:HTI1: When the E_Status field for the previously received PIO Setup FIS has been placed into the HBA response registers 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 HBA response registers, 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 Response 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 Response FIS to the host.

HTTPS5: 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 HTTPS5:HTTPS6: When data is available for the host to read in the HBA Data register, the transport layer shall place the initial content received in the PIO Setup FIS (using Status field for HBA Status register) into the HBA response registers and transition to the HTTPS6: HT_PIOITrans2 state.

Transition HTTPS5:HTI1: When the host has requested a Control FIS with the SRST =1, or a DEVICE RESET command operation is requested, 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.

HTTPS6: 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 HBA response registers 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 HTTPS6:HTTPS6: If the transfer is not complete, the transport layer shall transition to the HTTPS6: HT_PIOITrans2 state.

Transition HTTPS6:HTTPS4: When the byte count for this DRQ data block is reached, the transport layer shall transition to the HTTPS4: HT_PIOEnd state.

Transition HTTPS6: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.

7.6.10 Host Transport decompose a First Party DMA Setup FIS state diagram

This protocol 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.

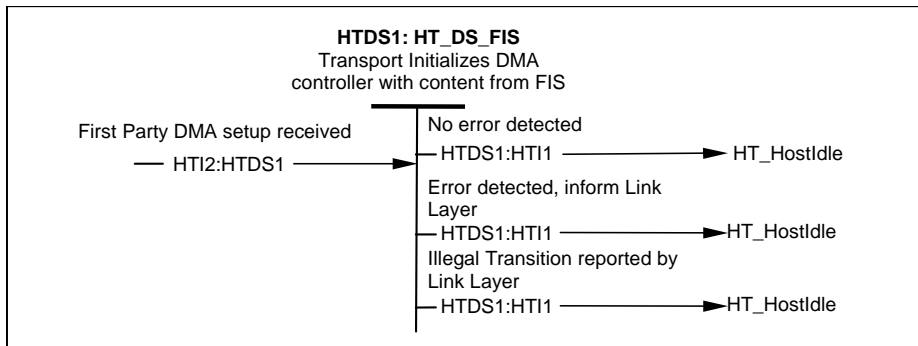


Figure 83 – Host transport decompose First Party DMA Setup FIS state diagram (State HTDS1)

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 Setup 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:HT_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.

7.6.11 Host transport decompose a BIST Activate FIS state diagram

This protocol 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.

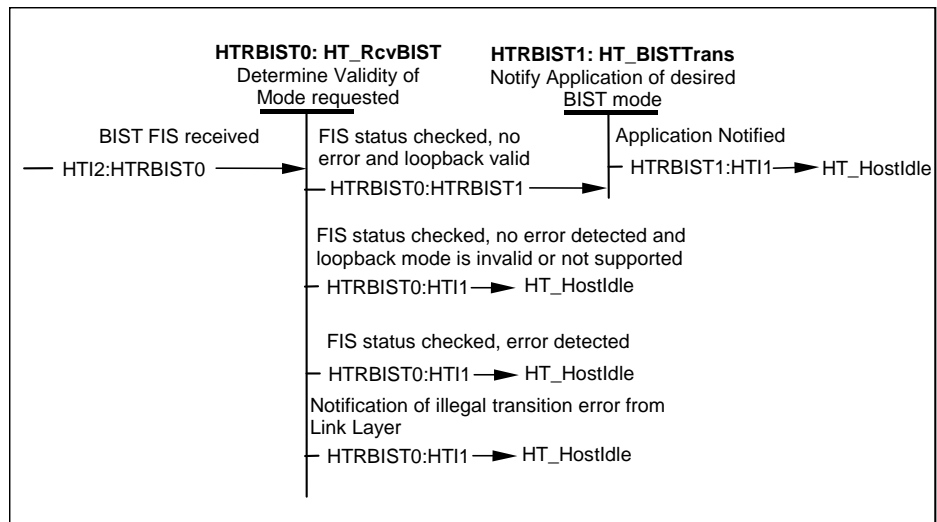


Figure 84 – Host transport decompose BIST activate FIS state diagram (State HTRBIST0-HTRBIST1)

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.

7.7 Device transport states

7.7.1 Device transport idle state diagram

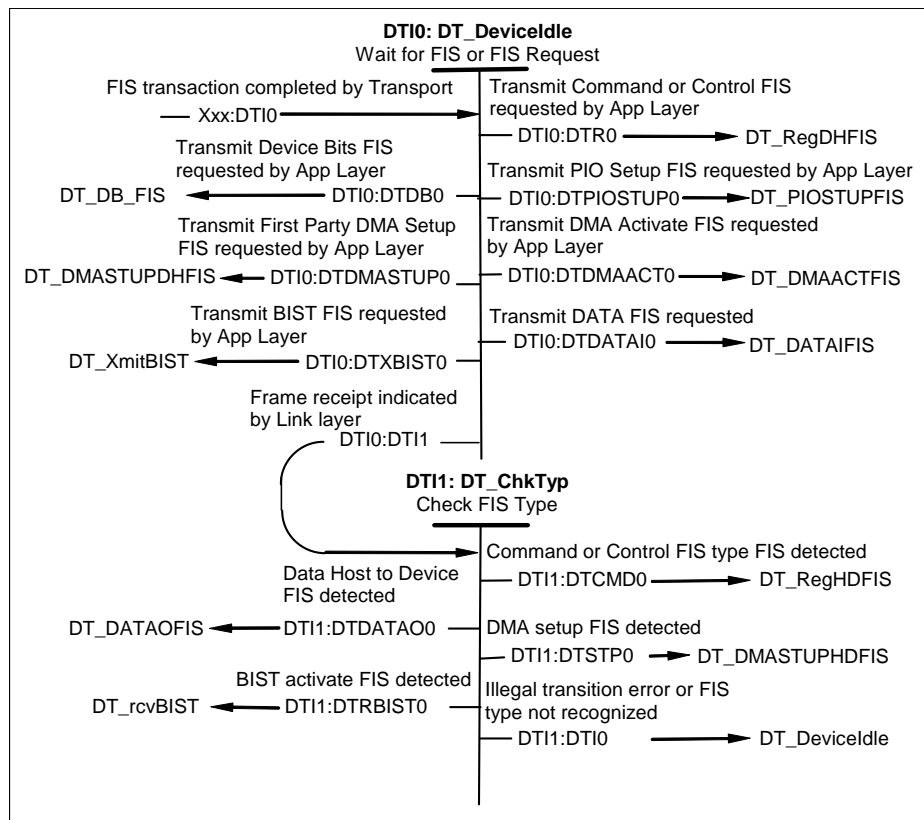


Figure 85 – Device transport idle state diagram (States DTI0-DTI1)

DTI0: DT_Deviceldle state: This state is entered when a Frame Information Structure (FIS) transaction has been completed by the transport layer.

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 Response 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.

When in this state, the transport layer checks the FIS type of the incoming FIS.

Transition DTI1:DTCMD0: When the incoming FIS is a Command FIS or Control 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.

7.7.2 Device Transport send Response FIS to host state diagram

This protocol builds a Response FIS that contains the response content (see 7.5.3) and sends it to the host when the device's Application layer requests the transmission.

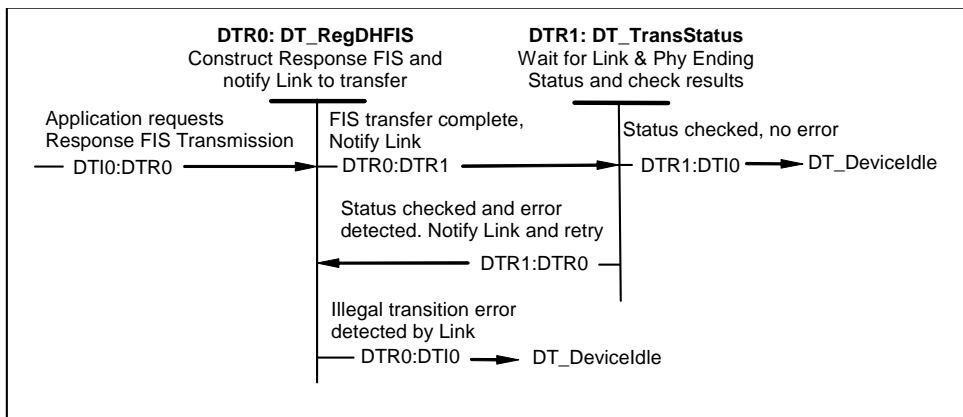


Figure 86 – Device transport send Response state diagram (DTR0-DTR1)

DTR0: DT_RegDHFIS state: This state is entered the Application requests the transmission of a Response FIS.

When in this state, the transport layer shall construct a Response 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 checked, and no error detected, the transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTR1:DTR0: When the FIS status has been checked, 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.

7.7.3 Device Transport send Set Device Bits FIS state diagram

This protocol sends a Set Device Bits FIS to the host adapter when the device's Application layer requests the transmission.

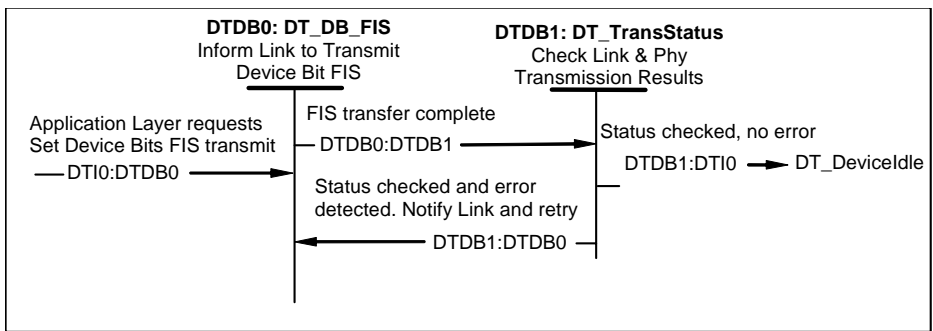


Figure 87 – Device transport send device bits FIS state diagram (DTDB0-DTDB1)

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 checked and no error detected, the transport layer shall transition to the DTI0:DT_DeviceIdle state.

Transition DTDB1:DTDB0: When the FIS status has been checked 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.

7.7.4 Device Transport transmit PIO Setup - device to host FIS state diagram

This protocol transmits a PIO Setup - device to host FIS to the host. Following this PIO Setup FIS, a single data frame containing PIO data shall be transmitted or received depending on the state of the D bit in the PIO Setup FIS.

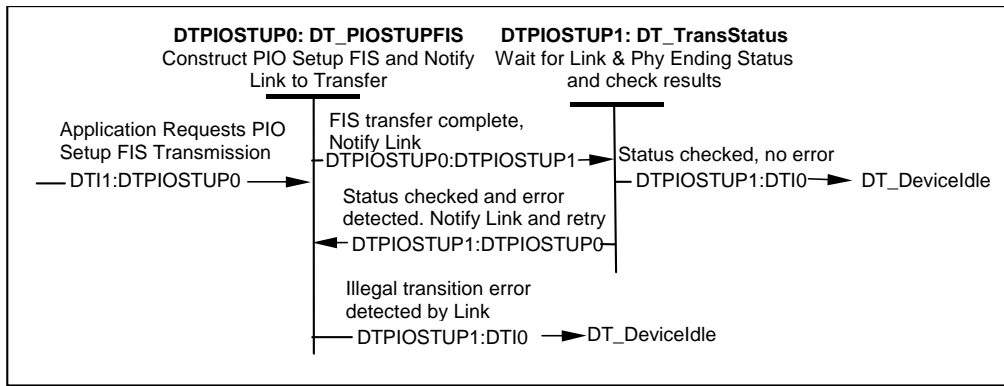


Figure 88 – Device transport transmit PIO setup - device to host FIS state diagram (States DTPIOSTUP0-DTPIOSTUP1)

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 checked and no error detected, the transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTPIOSTUP1:2: When the FIS status has been checked 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.

7.7.5 Device Transport transmit DMA Activate FIS state diagram

This protocol transmits a DMA Activate FIS to the host adapter. Following this DMA Activate FIS, a data FIS of DMA data shall be sent from the host to the device.

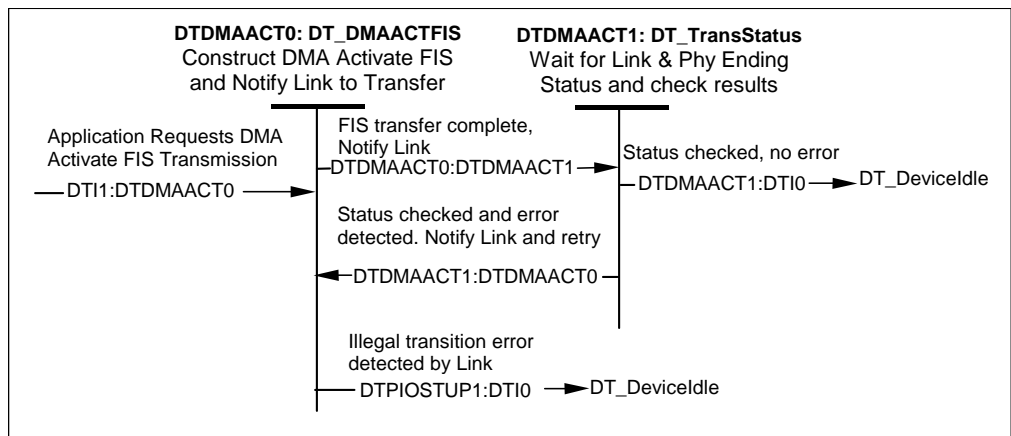


Figure 89 – Device transport transmit DMA activate FIS state diagram (States DTDMAACT0-DTDMAACT1)

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 checked and no error detected, the transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDMAACT1:DTDMAACT0: When the FIS status has been checked 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.

7.7.6 Device Transport transmit First Party DMA Setup - device to host FIS state diagram

This protocol 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 Pending bit.

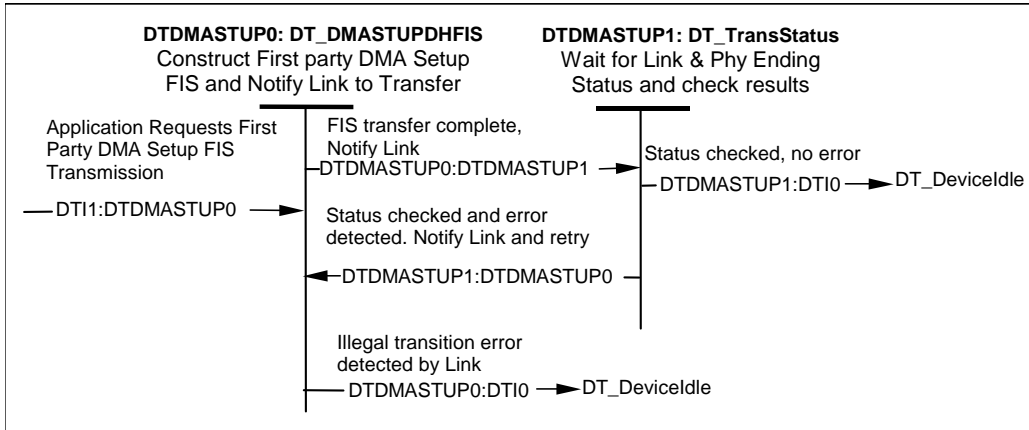


Figure 90 – Device transport transmit First Party DMA Setup - device to host state diagram (States DTDMASTUP0-DTDMASTUP1)

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 DTDMASTUP0: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.

DTDMASTUP1: 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 checked and no error detected, the transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDMASTUP1:DTDMASTUP0: When the FIS status has been checked and an error detected, the transport layer shall report status to the link layer and retry this transfer by transitioning to the DTDMASTUP0: DT_DMASTUPDHFIS state.

7.7.7 Device Transport transmit Data - device to host FIS diagram

This protocol builds a Data - device to host FIS.

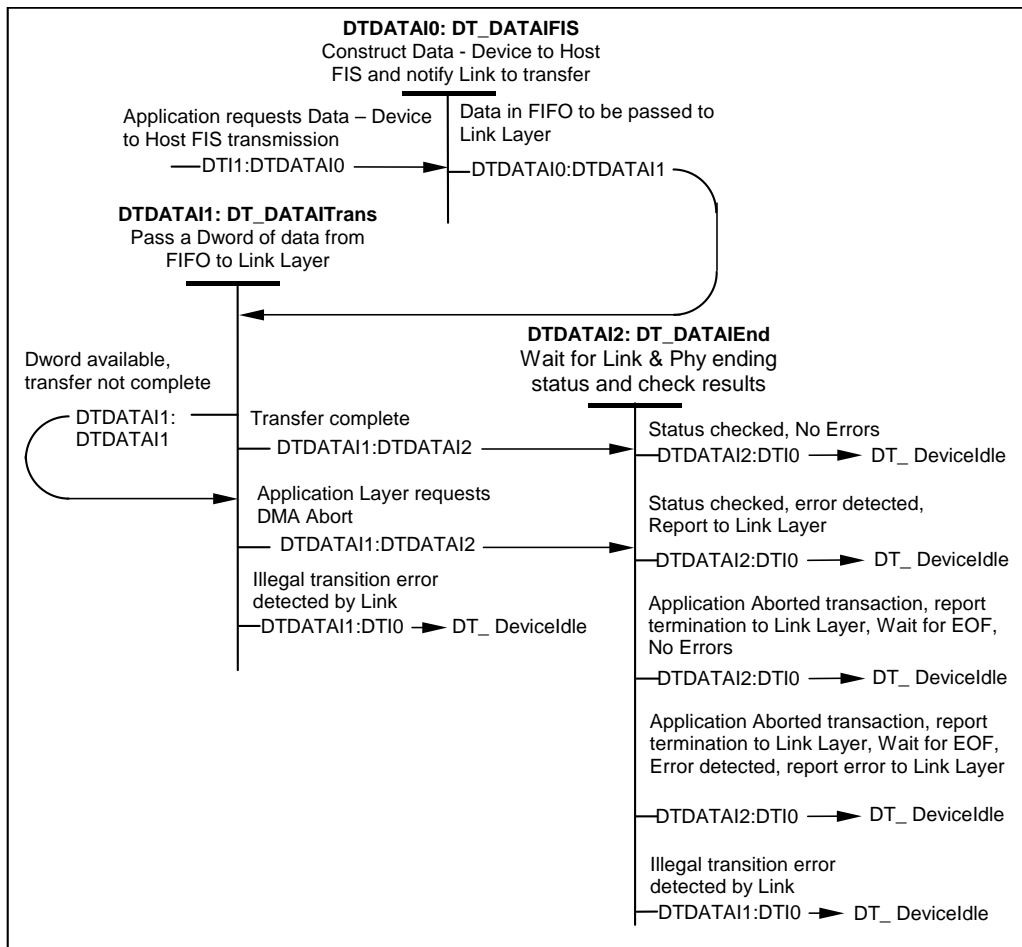


Figure 91 – Device transport transmit data - device to host FIS diagram (State DTDATAI0-DTDATAI2)

DTDATAI0: DT_DATAIFIS 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 checked and no error detected, the transport layer shall transition to the DTI1: DT_DeviceIdle state.

When the FIS status has been checked 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 6.4.6.

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 6.4.6.

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.

7.7.8 Device Transport transmit BIST Activate FIS diagram

This protocol 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.

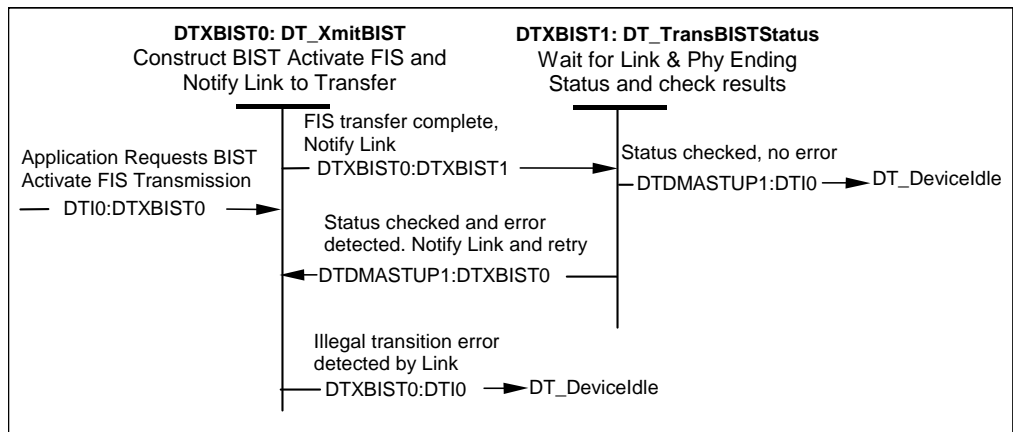


Figure 92 – Device transport transmit BIST activate FIS diagram (States DTXBI0-DTXBI1)

DTXBIST0: DT_XmitBIST state: This state is entered to send a BIST Activate FIS to the host.

Transition DTXBI0: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 DTXBI1:DT_TransBISTStatus state.

Transition DTXBI1: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 DTXBI1: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 DTXBI1:DTXBIST0: When the FIS transmission is completed and at least one error is detected, the transport layer shall transition to the DTXBI0: DT_XmitBIST state.

7.7.9 Device Transport decompose Command or Control FIS state diagram

This protocol receives a Command or Control FIS, places received FIS content into the appropriate vendor specific device registers, and notifies the device's Application layer of the FIS receipt.

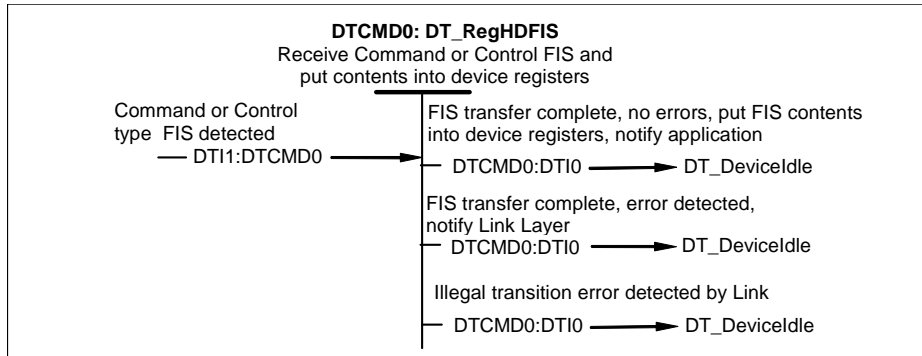


Figure 93 – Device transport decompose Command or Control FIS state diagram (State DTCMD0)

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 appropriate vendor specific 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.

7.7.10 Device Transport decompose Data (host to device) FIS state diagram

This protocol receives a Data - host to device FIS.

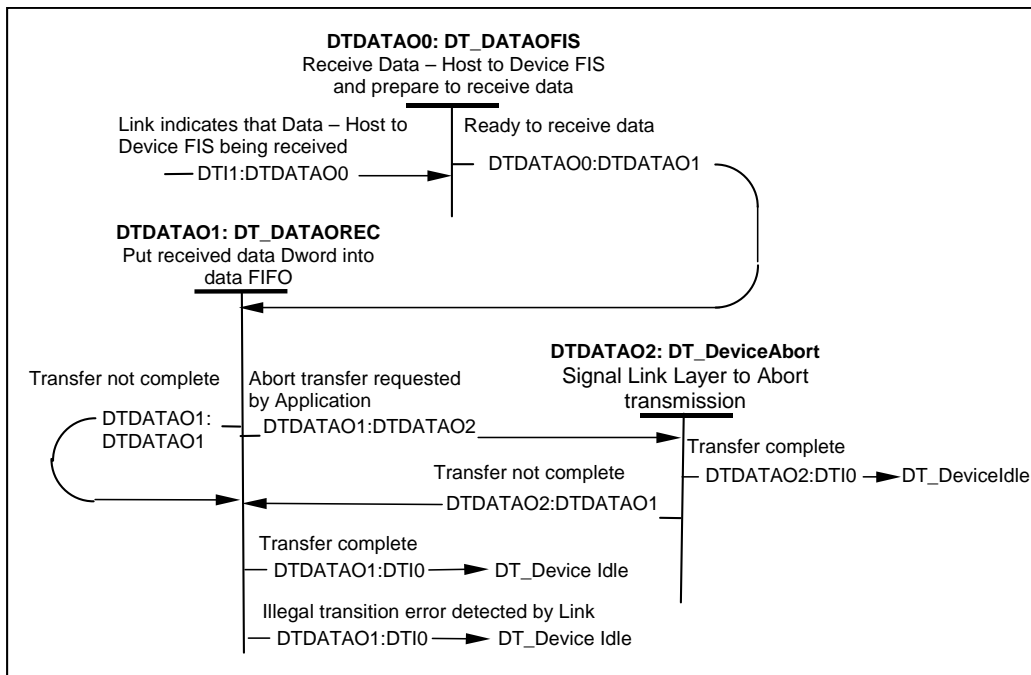


Figure 94 – Device transport decompose data (host to device) FIS state diagram (States DTDATAO0-DTDATAO2)

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 6.4.6)

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, transition to DTI0: DT_DeviceIdle. After signaling 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.

7.7.11 Device Transport decompose First Party DMA Setup FIS - host to device or device to host state diagram

This protocol 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.

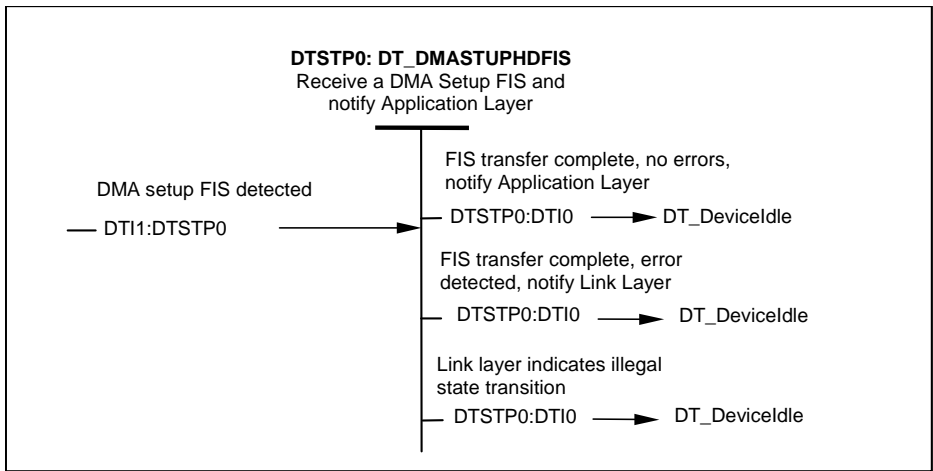


Figure 95 – Device transport decompose First Party DMA Setup FIS - host to device or device to host state diagram (State DTDMASTUP0)

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.

7.7.12 Device Transport decompose a BIST Activate FIS state diagram

This protocol 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.

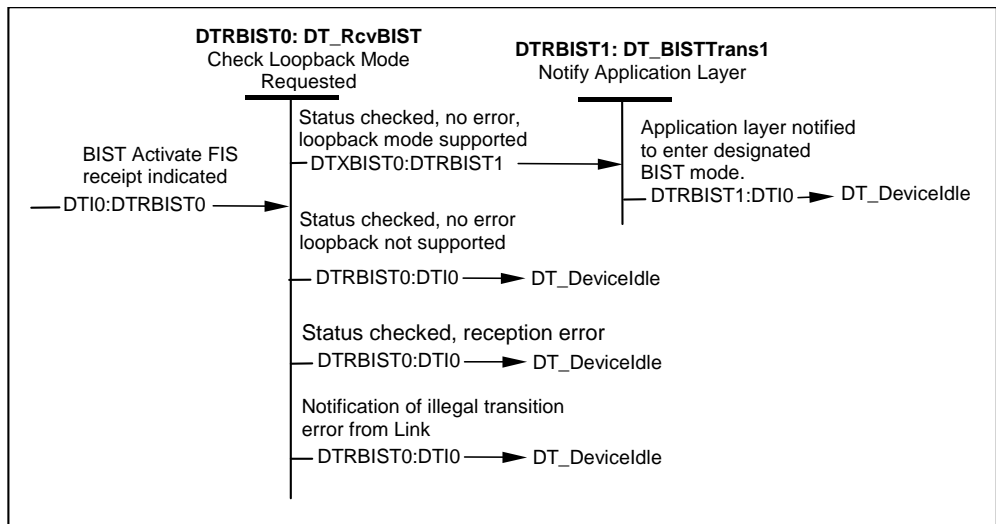


Figure 96 - Device transport decompose BIST activate FIS (States (DTRBIST0-DTRBIST1))

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 DT11:DT_DeviceIdle state.

Transition DTRBIST0:DTI0: If a reception error is indicated the transport layer shall make a transition to the DT11: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.

8 Serial Transport Device Command Layer Protocol

8.1 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 transmits a Control FIS with SRST=1 before the device has completed executing a command layer protocol, then the device shall immediately start executing its software reset protocol from the beginning.

8.2 Power-on and COMRESET protocol diagram

If the device receives a valid ComReset Sequence and the ComReset signal is asserted, the device shall execute the hardware reset protocol regardless of the power management mode or the current device command layer state.

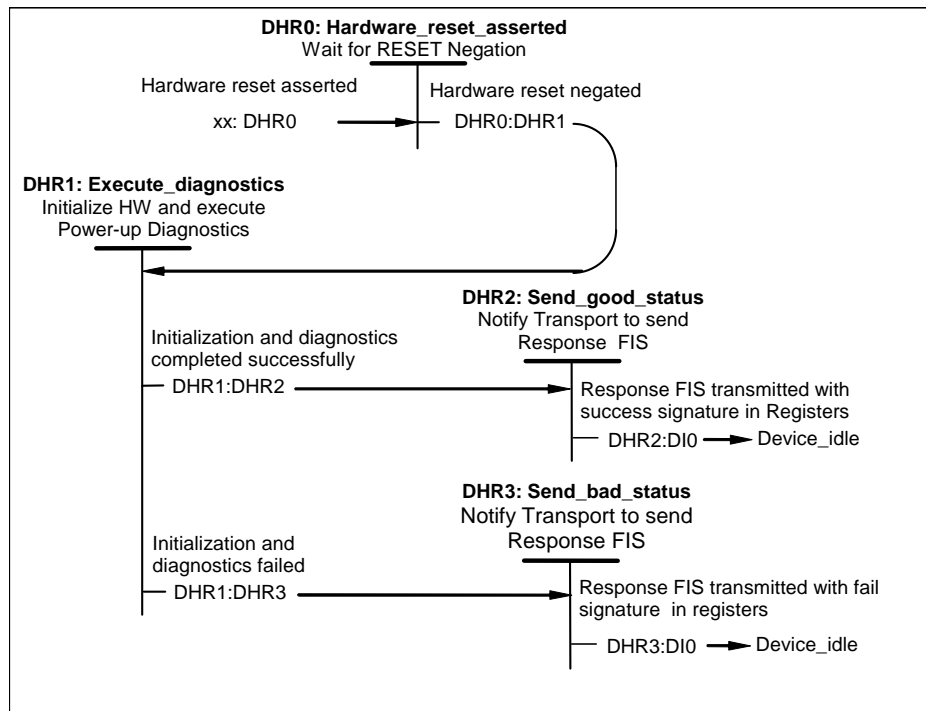


Figure 97 - Power on and COMRESET protocol (States DHR0-DHR3)

DHR0: Hardware_reset_asserted: This state is entered when the transport layer indicates that the ComReset signal is asserted.

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.

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.

When in this state, the device requests that the transport layer transmit a Response FIS to the host. The device signature shall be set in the FIS contents as defined in ATA8-ACS.

Transition DHR2:DIO: When the transport layer indicates that the Response FIS has been transmitted, the device shall transition to the DIO: Device_Idle state.

DHR3: Send_bad_status: This state is entered when the device hardware has been initialized and the power-up diagnostics failed.

When in this state, the device requests that the transport layer transmit a Response FIS to the host. The device signature shall be set in the FIS contents as defined in ATA8-ACS.

Transition DHR3:DIO: When the transport layer indicates that the Response FIS has been transmitted, the device shall transition to the DIO: Device_Idle state.

8.3 Device Idle protocol

The state diagram below describes the idle protocol for a device.

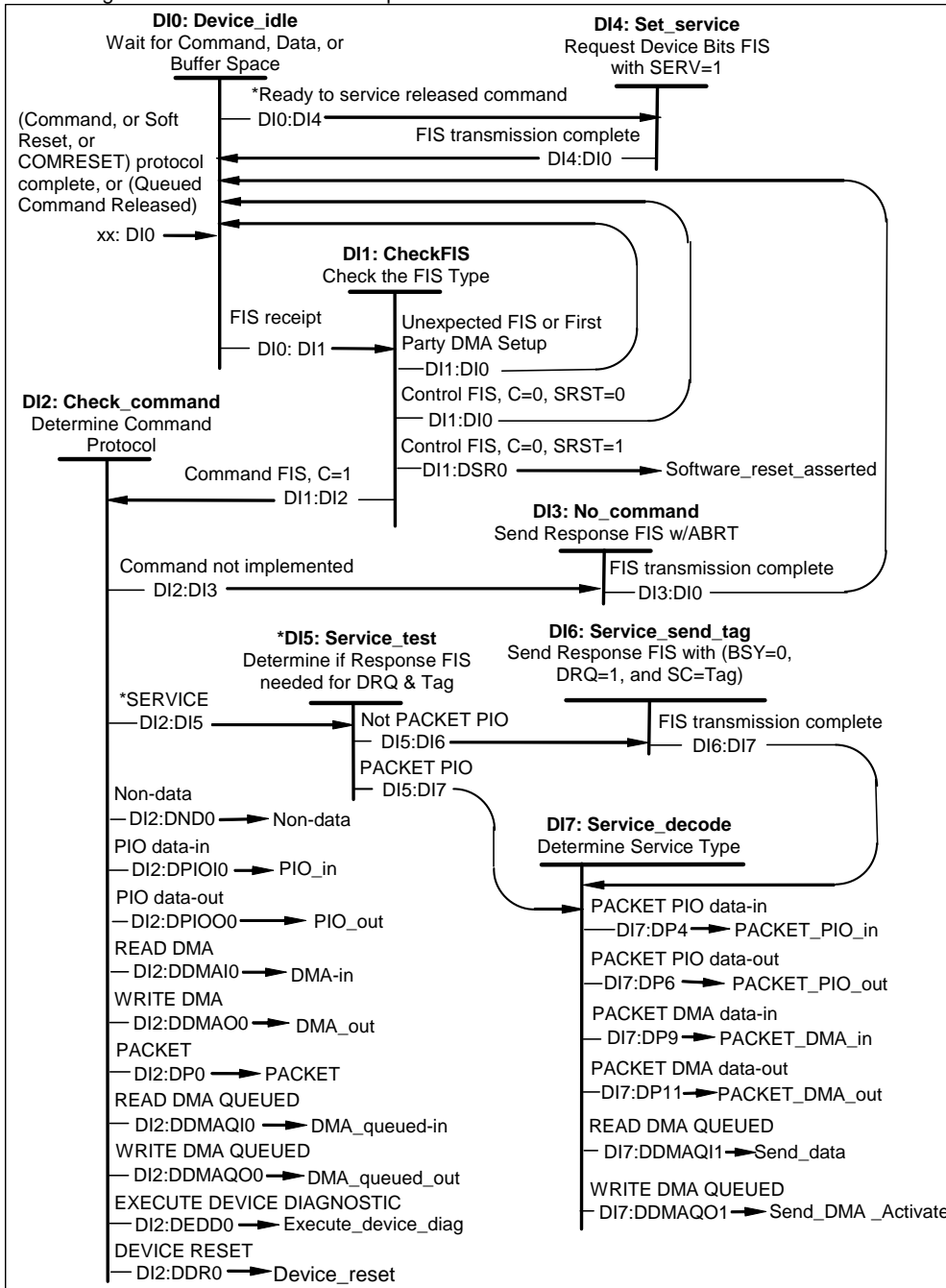


Figure 98 - Device idle protocol (States DI0-DI7)

D10: 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 D10:D11: When the device receives an FIS from the transport layer, the device shall transition to the D11: Check_FIS state.

Transition D10:D14: When the device is ready to complete the data transfer for a queued command, the device shall transition to the D14: Set_service state.

D11: 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 D11:DSR0: If the FIS type is a Command 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 D11:D10: If the FIS type is a Control FIS, the C bit in the FIS is cleared, and the SRST bit in the FIS is cleared, the device shall transition to the D10: Device_idle state.

Transition D11:D12: If the FIS type is a Command FIS and the C bit in the FIS is set, the device shall transition to the D12: Check_command state.

Transition D11:D10: 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 D10: Device_idle state.

Transition D11:D10: For any other FIS, the device shall transition to the D10: Device_idle state.

D12: Check_command state: This state is entered when the device recognizes that the received Command FIS contains a new command. NOTE: This state shows transitions for all commands. If a device does not implement any particular command, transition D12:11 to state D13:No_command shall be made.

When in this state, the device shall check the command protocol required by the received command.

Transition D12:DND0: When the received command is a non-data transfer command, the device shall transition to the DND0: Non-data state.

Transition D12:DPIOI0: When the received command is a PIO data-in command, the device shall transition to the DPIOI0: PIO_in state.

Transition D12:DPIOO0: When the received command is a PIO data-out command, the device shall transition to the DPIOO0: PIO_out state.

Transition D12:DDMAI0: When the received command is a READ DMA command, the device shall transition to the DDMAI0: DMA_in state.

Transition D12:DDMAO0: When the received command is a WRITE DMA command, the device shall transition to the DDMAO0: DMA_out state.

Transition D12:DP0: When the received command is a PACKET command, the device shall transition to the DP0: PACKET state.

Transition D12:DDMAQI0: When the received command is a READ DMA QUEUED command, the device shall transition to the DDMAQI0: DMA_queued_in state.

Transition DI2:DDMAQ00: When the received command is a WRITE DMA QUEUED command, the device shall transition to the DDMAQ00: 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 Response FIS with FIS content as described in the appropriate command description and the Interrupt Pending bit set to one.

Transition DI2:DI0: When the transport layer has transmitted the Response 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 field and with all other bits in the Error and Status fields the same as the current contents of the respective fields, 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 content update to set DRQ and send the command tag, but other queued commands require a Response 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 rResponse FIS is necessary for the command being serviced.

When in this state, the device shall request that the transport layer transmit a Response FIS including the desired command tag for the command being serviced.

Transition DI6:DI7: When the transport layer has transmitted the Response FIS, the device shall transition to the DI7: Service_decode state.

DI7: Service_decode state: This state is entered when a Response FIS has been transmitted, if necessary, 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:DDMAQ11: When the command to be serviced is a READ DMA QUEUED command, the device shall transition to the DDMAQ11: 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.

8.4 Software reset protocol

When the host sends a Control FIS with a one in the SRST bit position of the Control field, the device shall execute the software reset protocol regardless of the power management mode.

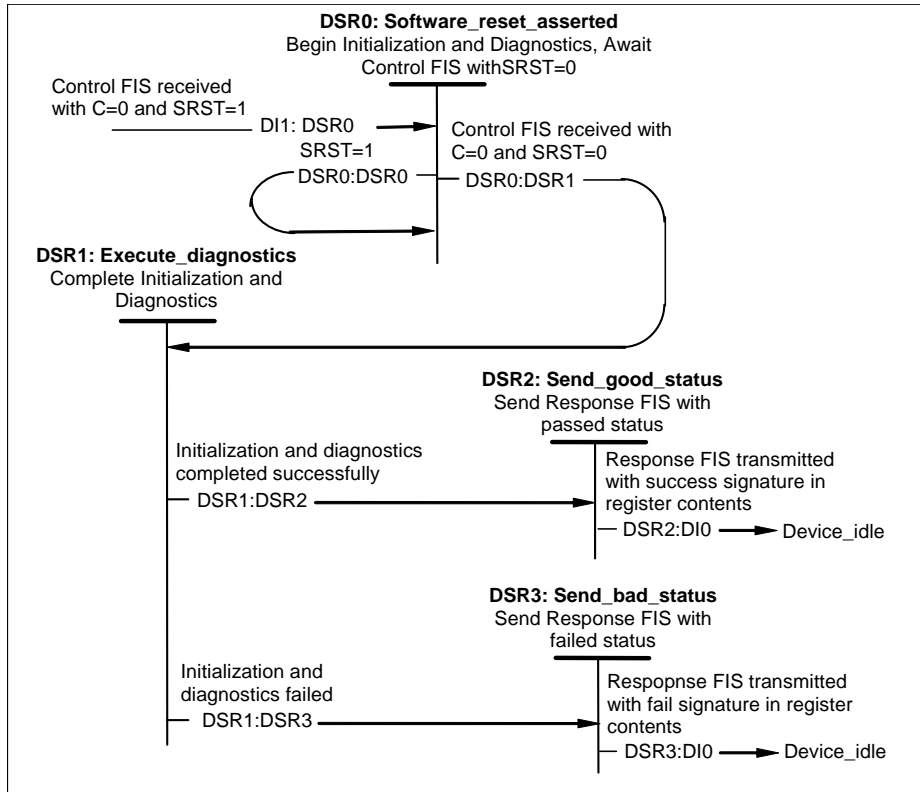


Figure 99 - Software reset protocol (States DSR0-DSR3)

DSR0: Software_reset_asserted: This state is entered when a Control FIS is received with the C bit in the FIS cleared to zero and the SRST bit set to one in the Control field.

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 Control FIS is received with the C bit in the FIS cleared to zero and the SRST bit cleared to zero in the Device field, the device shall transition to the DSR1: Execute_diagnostics state.

Transition DSR0:DSR0: If a Control FIS is received with the C bit in the FIS cleared to zero, and the SRST bit set to one in the Control field, the device shall transition to the DSR0: Software_reset_asserted state. (editors note: changed, error in ATA-7?)

DSR1: Execute_diagnostics: This state is entered when a Control FIS is received with the C bit in the FIS cleared to zero and the SRST bit cleared to zero in the Control field.

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 Response FIS to the host. The device signature shall be set in the appropriate fields as defined in ATA8-ACS.

Transition DSR2:DIO: When the transport layer indicates that the Response FIS has been transmitted, the device shall transition to the DIO: 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 Response FIS to the host. The device signature shall be set in the appropriate fields as defined in ATA8-ACS.

Transition DSR3:DIO: When the transport layer indicates that the Response FIS has been transmitted, the device shall transition to the DIO: Device_Idle state.

8.5 EXECUTE DEVICE DIAGNOSTIC command protocol

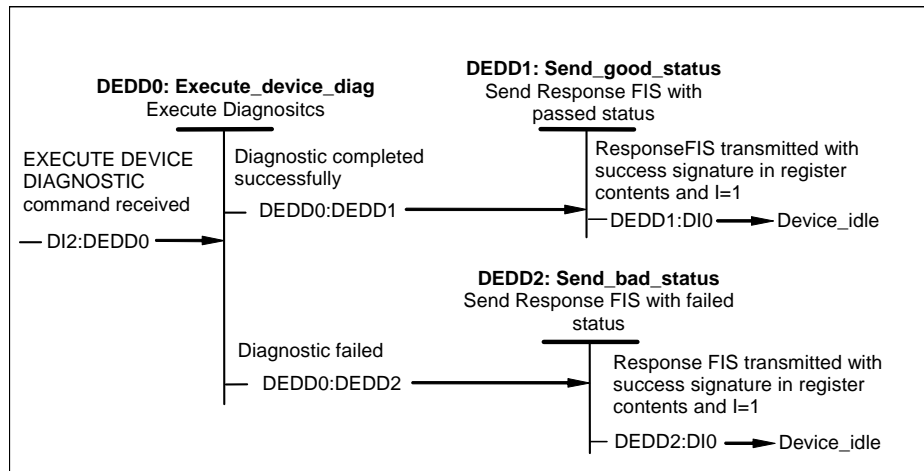


Figure 100 - EXECUTE DEVICE DIAGNOSTIC command protocol (States DEDD0-DEDD2)

DEDD0: Execute_device_diag: This state is entered when an EXECUTE DEVICE DIAGNOSTIC command is received.

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 Response FIS to the host, with the I bit set to one. The device signature shall be set in the appropriate fields as defined in ATA8-ACS.

Transition DEDD1:DI0: When the transport layer indicates that the Response 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 Response FIS to the host, with the I bit set to one. The device signature shall be set in the appropriate fields as defined in ATA8-ACS.

Transition DEDD2:DI0: When the transport layer indicates that the Response FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

8.6 DEVICE RESET command protocol

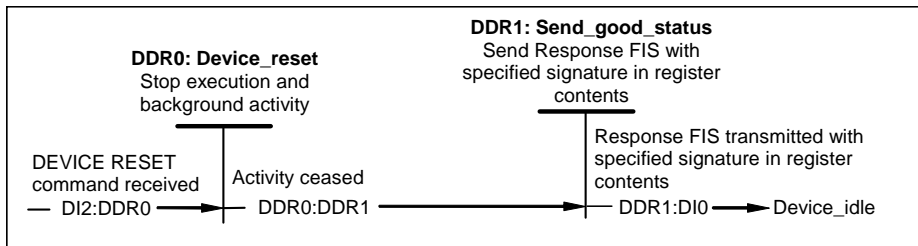


Figure 101 - DEVICE RESET command protocol (States DDR0-DDR1)

DDR0: Device_reset: This state is entered when an DEVICE RESET command is received.

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.

When in this state, the device requests that the transport layer transmit a Response FIS to the host. The device signature shall be set in the appropriate fields as defined in ATA8-ACS.

Transition DDR1:DI0: When the transport layer indicates that the Response FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

8.7 Non-data command protocol

Non-Data Commands are listed in ATA8-ACS.

Execution of these commands involves no data transfer. See ATA8-ACS for the NOP command description and the SLEEP command description for additional protocol requirements.

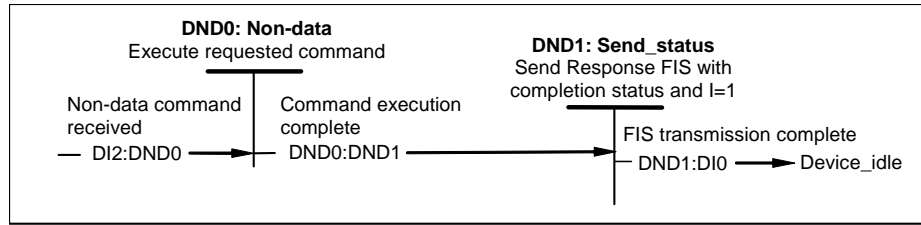


Figure 102 - Non-data command protocol (States DND0-DND1)

DND0: 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 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.

When in this state, the device shall request that the transport layer transmit a Response FIS with content as described in the appropriate command description and the I bit set to one.

Transition DND1:D10: When the FIS has been transmitted, then the device shall transition to the D10: Device_idle state.

8.8 PIO data-in command protocol

PIO Data In commands are listed in ATA8-ACS.

Execution of this class of command includes the PIO transfer of one or more blocks of data from the device to the host.

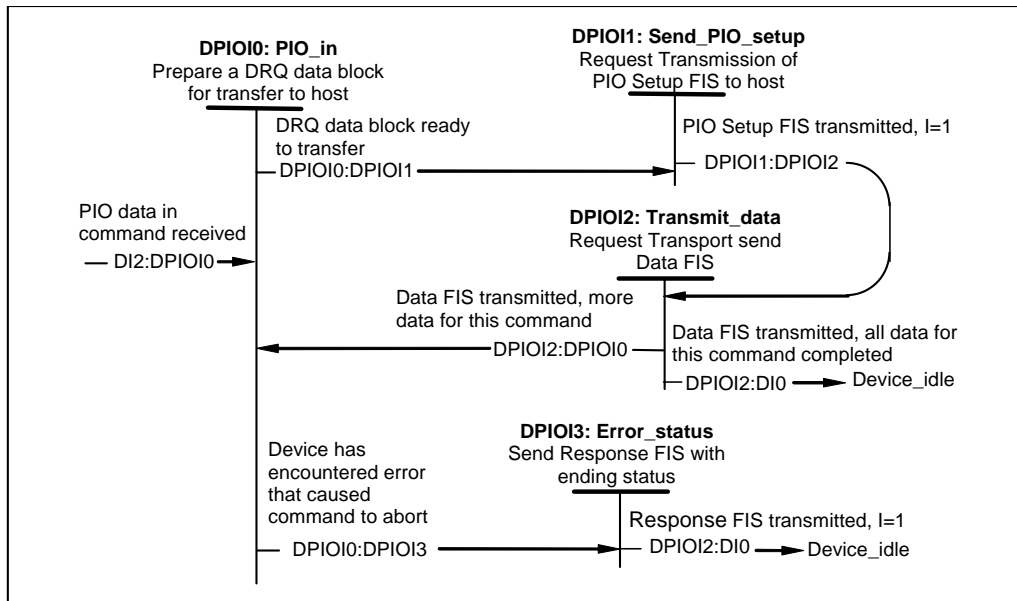


Figure 103 - PIO data-in command protocol (States DPIOI0-DPIOI3)

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.

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.

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 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.

When in this state, the device shall request that the transport layer transmit a Data FIS containing the DRQ data block.

Transition DPIOI2:DIO: When the Data FIS has been transferred and all data requested by this command have been transferred, the device shall transition to the DIO: 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 2048 dwords 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.

When in this state, the device shall request that the transport layer transmit a Response FIS with content as described in the appropriate command description and the I bit set to one.

Transition DPIOI3:DIO: When the FIS has been transmitted, the device shall transition to the DIO: Device_idle state.

8.9 PIO data-out command protocol

PIO Data-out commands are listed in ATA8-ACS. Execution of this class of command includes the PIO transfer of one or more blocks of data from the host to the device.

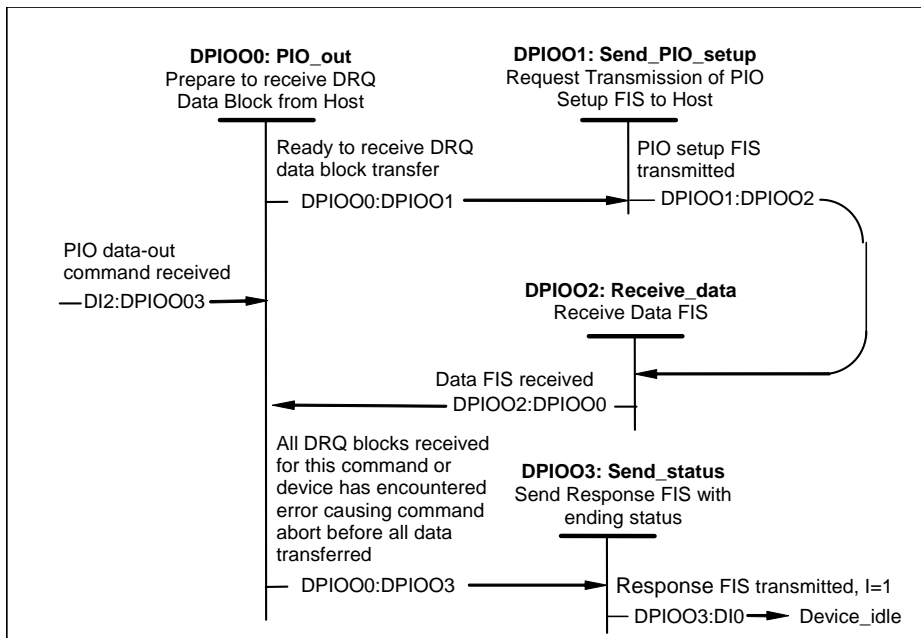


Figure 104 - PIO data-out command protocol (States DPIO00-DPIO03)

DPIO00: 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.

When in this state, device shall prepare to receive a DRQ data block transfer from the host.

Transition DPIO00:DPIO01: When the device is ready to receive a DRQ data block, the device shall transition to the DPIO01: Send_PIO_setup state.

Transition DPIO00:DPIO03: 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 DPIO03: Send_status state.

DPIO01: Send_PIO_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. 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 DPIO01:DPIO02: When the PIO Setup FIS has been transferred, the device shall transition to the DPIO02: Receive_data state.

DPIO02:Receive_data: This state is entered when the device has 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 DPIO02:DPIO00: When the Data FIS has been received, the device shall transition to the DPIO00: PIO_out state.

DPIO03: 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.

When in this state, the device shall request that the transport layer transmit a Response FIS with content as described in the appropriate command description and the I bit set to one.

Transition DPIO03:DIO: When the FIS has been transmitted, the device shall transition to the DIO: Device_idle state.

8.10 DMA data-in command protocol

DMA data-in commands are listed in ATA8-ACS.

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.

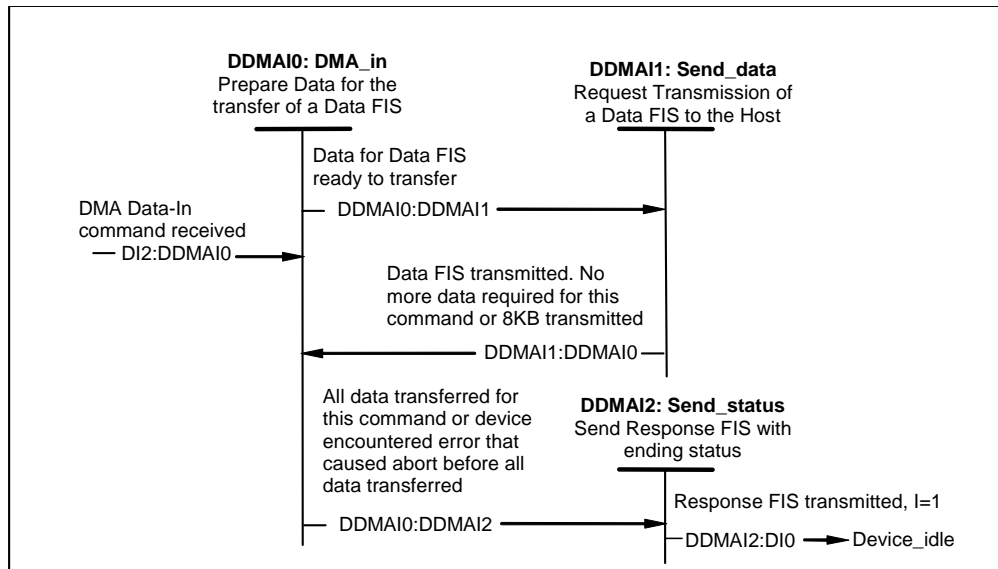


Figure 105 - DMA data-in command protocol (States DDMAI0-DDMAI1)

DDMAI0: DMA_in State: This state is entered when the device receives a DMA data-in command and 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 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.

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 2048 dwords.

Transition DDMAI1:DDMAI0: When the Data FIS has been transferred, the device shall transition to the DDMAI0: 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.

When in this state, the device shall request that the transport layer transmit a Response FIS with content as described in the appropriate command description and the I bit set to one.

Transition DDMAI2:DIO: When the FIS has been transmitted, the device shall transition to the DIO: Device_idle state.

8.11 DMA data out command protocol

DMA data out commands are listed in ATA8-ACS.

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. Interrupt Pending is set to one at the completion of the command.

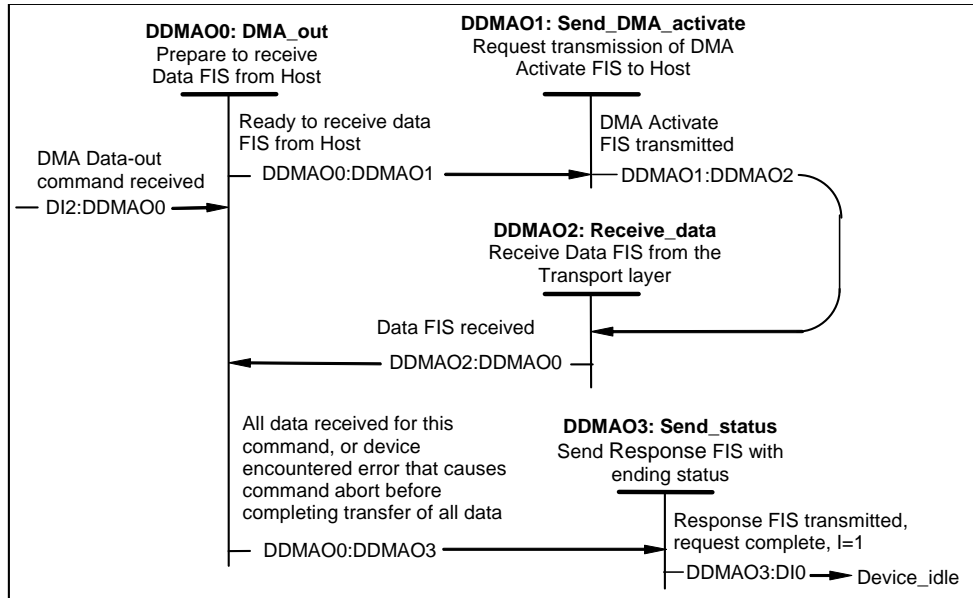


Figure 106 - DMA data-out command protocol (States DDMAO0-DDMAO3)

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.

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.

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.

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.

When in this state, the device shall request that the transport layer transmit a Response FIS with content as described in the appropriate command description and the I bit set to one.

Transition DDMAO3:DIO: When the FIS has been transmitted, the device shall transition to the DIO: Device_idle state.

8.12 PACKET protocol

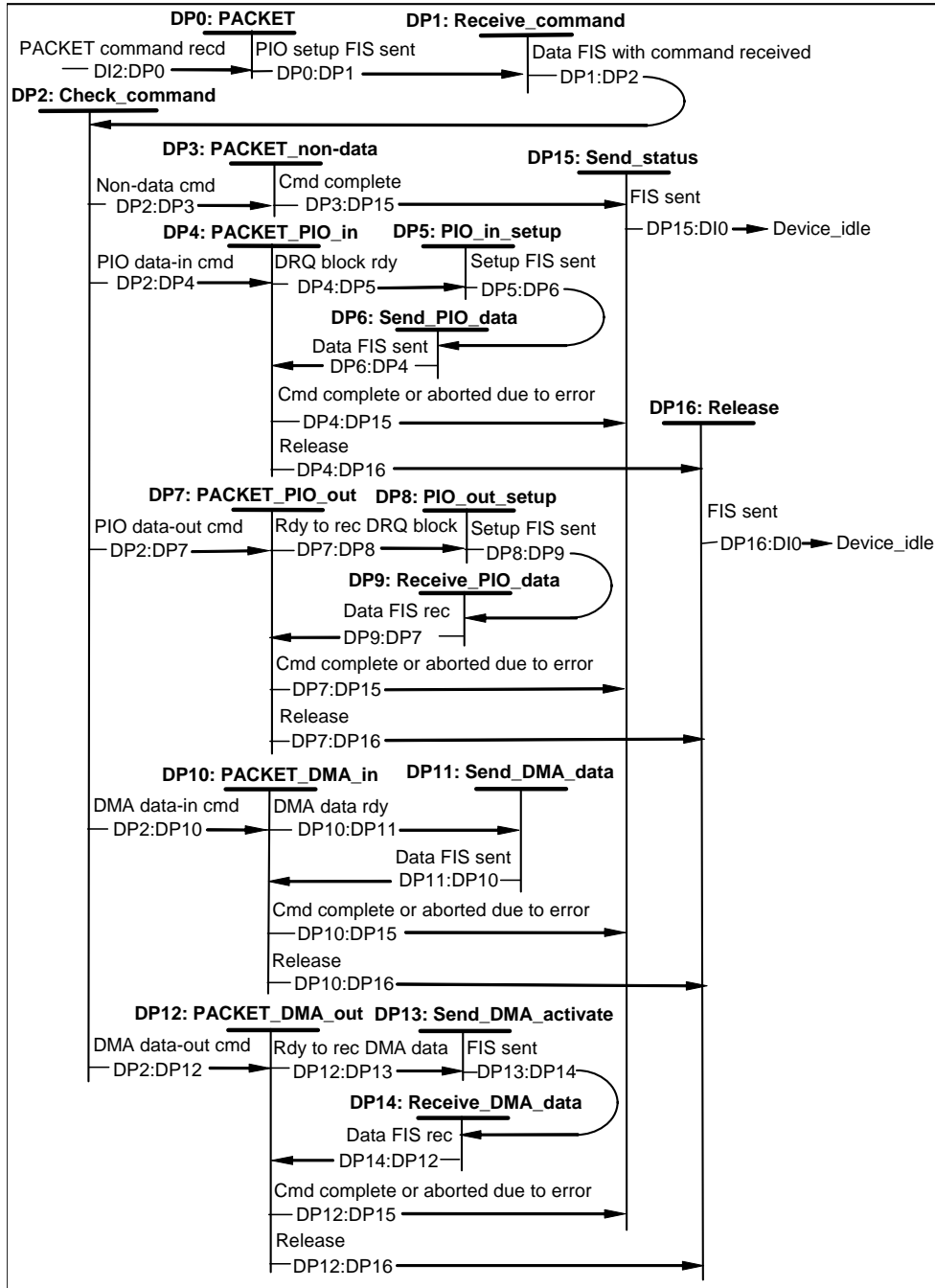


Figure 107 - PACKET command protocol (States DP0-DP16)

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 2048 dwords.

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 Response FIS with content as described in the appropriate command description. and the I bit set to one.

Transition DP15:DIO: When the FIS has been transmitted, then the device shall transition to the DIO: 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 Response FIS with 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:DIO: When the FIS has been transmitted, then the device shall transition to the DIO: Device_idle state.

8.13 READ DMA QUEUED command protocol

Read DMA Queued commands are listed in ATA8-ACS command set.

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 protocol may bus release before transferring data. Bus release is when the device sends (and host acknowledges) a Response FIS with with BSY=1 and DRQ=0 before the command has completed. When data transfer is begun, all data for the request shall be transferred without a bus release.

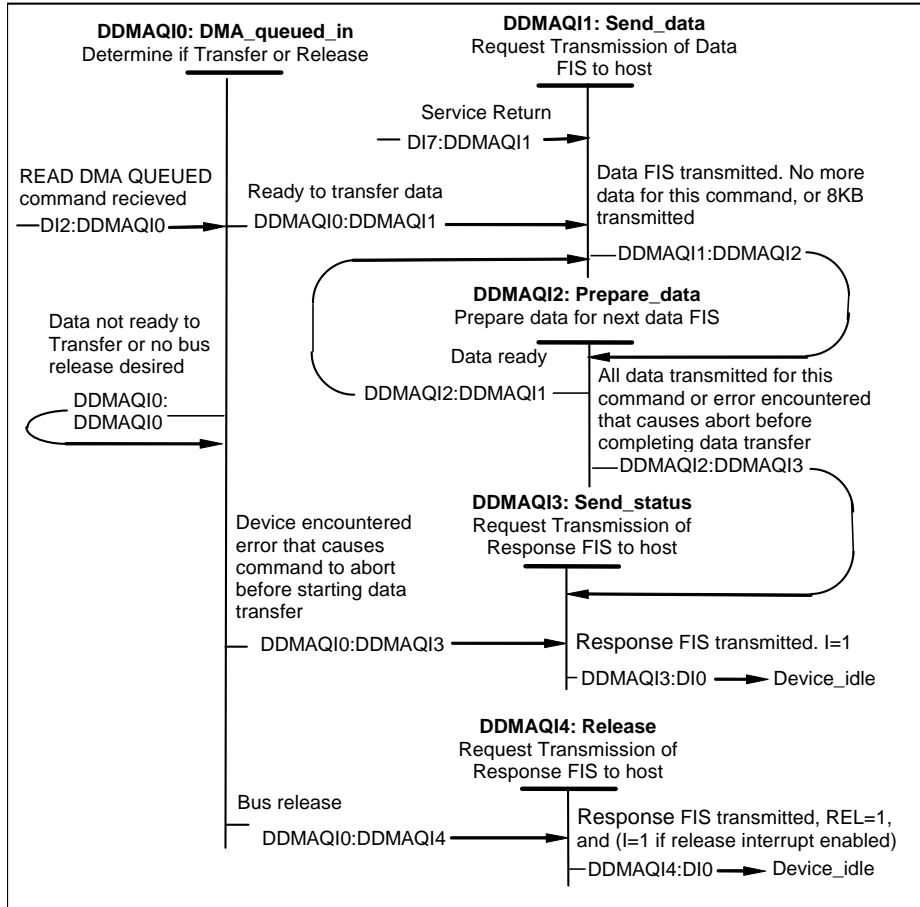


Figure 108 - READ DMA QUEUED command protocol (States DDMAQI0-DDMAQI4)

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 DDMAQ10:DDMAQ11: When the device has the requested data ready to transfer a Data FIS immediately, the device shall transition to the DDMAQ11: Send_data state.

Transition DDMAQ10:DDMAQ13: 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 DDMAQ13: Send_status state.

Transition DDMAQ10:DDMAQ14: 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 DDMAQ14: Release state.

Transition DDMAQ10:DDMAQ10: When the device does not have the requested data ready to transfer, or no bus release is desired, the device shall transition to the DDMAQ10: DMA_queued_in state.

DDMAQ11: 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 DDMAQ11:DDMAQ12: When the Data FIS has been transferred, the device shall transition to the DDMAQ12: Prepare_data state. The device command layer shall request a Data FIS size of no more than 2048 dwords.

DDMAQ12: 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 DDMAQ12:DDMAQ11: When data is ready for the Data FIS, the device shall transition to the DDMAQ11: Send_data state.

Transition DDMAQ12:DDMAQ13: 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 DDMAQ13: Send_status state.

DDMAQ13: 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 Response FIS with content as described in the appropriate command description and the I bit set to one.

Transition DDMAQ13:D10: When the FIS has been transmitted, the device shall transition to the D10: Device_idle state.

DDMAQ14: 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 Response FIS with the REL bit set to one, with 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 DDMAQ14:D10: When the FIS has been transmitted, then the device shall transition to the D10: Device_idle state.

8.14 WRITE DMA QUEUED command protocol

Write DMA Queued commands are listed in ATA8-ACS.

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.

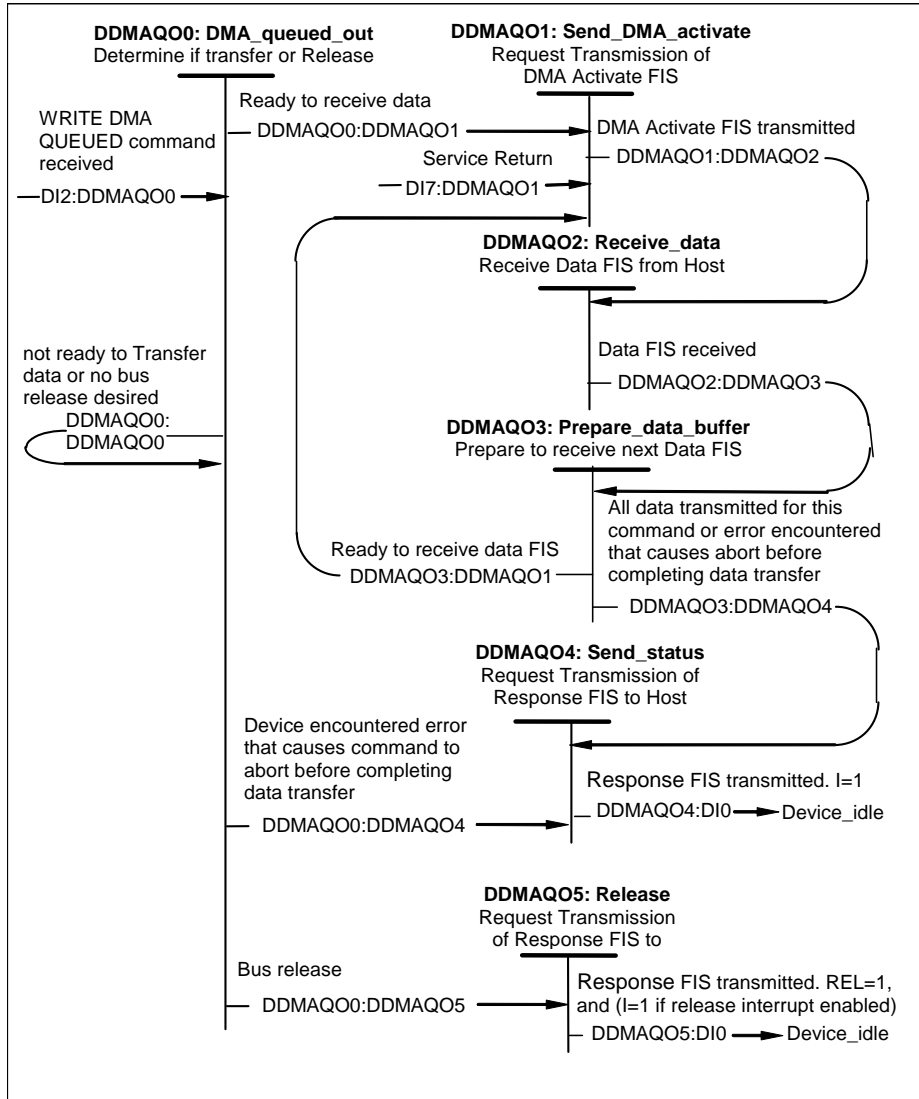


Figure 109 - WRITE DMA QUEUED command protocol (DDMAQ0-DDMAQ5)

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.

Transition DDMAQO0:DDMAQO5: When the device does not have the requested data ready to transfer a Data FIS immediately, the device shall transition to the DDMAQO5: Release state.

Transition DDMAQO0:DDMAQO0: When the device is not ready to transfer a Data FIS immediately, or no bus release is desired, the device shall transition to the DDMAQO0: DMA_queued_out state.

DDMAQO1: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 DDMAQO1:DDMAQO2: When the DMA Activate FIS has been transferred, the device shall transition to the DDMAQO2: Receive_data state.

DDMAQO2:Receive_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 DDMAQO2:DDMAQO3: When the Data FIS has been received, the device shall transition to the DDMAQO3: Prepare_data_buffer state.

DDMAQO3: Prepare_data_buffer: This state is entered when the device has completed receiving a Data FIS from the host.

When in this state, the device shall prepare for receipt of the next Data FIS.

Transition DDMAQO3:DDMAQO1: When ready to receive the Data FIS, the device shall transition to the DDMAQO1: Send_DMA_activate state.

Transition DDMAQO3:DDMAQO4: 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 DDMAQO4: Send_status state.

DDMAQO4: 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 Response FIS with content as described in the appropriate command description and the I bit set to one.

Transition DDMAQO4:DIO: When the FIS has been transmitted, then the device shall transition to the DIO: Device_idle state.

DDMAQO5: Release: This state is entered when the device cannot receive the requested data immediately.

When in this state, the device shall request that the transport layer transmit a Response FIS with REL set to one (see ATA8-ACS), with 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 DDMAQO5:DIO: When the FIS has been transmitted, then the device shall transition to the DIO: Device_idle state.

9 Serial transport host adapter register interface

9.1 Overview

Serial transport ATA host adapters include an additional block of registers mapped separately and independently from the ATA Command and Control Registers for reporting additional status and error information and to allow control of capabilities unique to the serial transport. These additional registers, referred to as the SStatus and SControl Registers (SCRs) are organized as 16 contiguous 32-bit registers.

The base address and mapping scheme for these registers is defined by the specific host adapter implementation - for example, PCI controller implementations may map the SCRs using the PCI mapping capabilities. Table 30 illustrates the overall organization of the SStatus and SControl registers. Parallel Transports' ATA software does not make use of the serial transport SStatus and SControl registers. The SStatus and SControl register are associated with the serial transport and are independent of any Device 0/Device 1 emulation the host adapter may implement. For Command Block and Control Block register definitions see ATA8-ACS.

Table 30 - SCR definition

| | | SStatus and SControl registers | |
|---------------|-----|--------------------------------|--|
| SATA register | 0 | SATA Status/Control | |
| SATA register | 1 | SATA Status/Control | |
| ... | ... | SATA Status/Control | |
| SATA register | 14 | SATA Status/Control | |
| SATA register | 15 | SATA Status/Control | |

9.2 SStatus, SError and SControl registers

The serial transport provides an additional block of registers to control the interface and to retrieve interface state information. There are 16 contiguous registers allocated of which the first three are defined and the remaining 13 are reserved for future definition. Table 31 defines the serial transport SStatus and SControl registers.

Table 31 - SCR Definition

| | |
|---------|-------------------|
| SCR[0] | SStatus register |
| SCR[1] | SError register |
| SCR[2] | SControl register |
| SCR[3] | Reserved |
| ... | ... |
| SCR[15] | Reserved |

9.2.1 SStatus register

The SStatus register is a 32-bit read-only register that conveys the current state of the interface and host adapter. The register conveys the interface state at the time it is read and is updated continuously and asynchronously by the host adapter. Writes to the register have no effect.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----|---|---|-----|---|---|-----|---|---|---|---|---|---|---|---|---|---|---|---|
| | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SCR0 | Reserved | | | | | | | | | | | | | | | | IPM | | | SPD | | | DET | | | | | | | | | | | | |

- DET The DET value indicates the interface device detection and Phy state.
 - 0000 No device detected and Phy communication not established
 - 0001 Device presence detected but Phy communication not established
 - 0011 Device presence detected and Phy communication established
 - 0100 Phy in offline mode as a result of the interface being disabled or running in a BIST loopback mode
 - All other values reserved

- SPD The SPD value indicates the negotiated interface communication speed established
 - 0000 No negotiated speed (device not present or communication not established)
 - 0001 Generation 1 communication rate negotiated
 - All other values reserved

- IPM The IPM value indicates the current interface power management state
 - 0000 Device not present or communication not established
 - 0001 Interface in Active mode
 - 0010 Interface in Partial mode
 - 0110 Interface in Slumber mode
 - All other values reserved

NOTE – The DET field is only valid if the interface is in the active state. When the interface is in the Partial mode or Slumber mode no communication between the host and target is established resulting in a DET value corresponding to no device present or no communication established. As a result the insertion or removal of a device may not be accurately detected under all conditions such as when the interface is quiescent as a result of being in Partial mode or Slumber mode. This field alone may therefore be insufficient to satisfy all the requirements for device attach or detach detection during all possible interface states.

9.2.2 SError register

The SError register is a 32-bit register that conveys supplemental Interface error information to complement the error information available in the HBA response registers. The register represents all the detected errors accumulated since the last time the SError register was cleared (whether recovered by the interface or not). Set bits in the error register are explicitly cleared by a write operation to the SError register, or a reset operation. The value written to clear set error bits shall have 1's encoded in the bit positions corresponding to the bits that are to be cleared. Host software should clear the Interface SError register at appropriate checkpoints in order to best isolate error conditions and the commands they impact.

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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 | | | | | | | | | | |
| SCR1 | DIAG | | | | | | | | | | | | | | | | ERR | | | | | | | | | | | | | | | | | | | | | | | | | |
| | R | | | | | | | | | | | | | | | | F | T | S | H | C | D | B | W | I | N | R | R | R | R | E | P | C | T | R | R | R | R | R | R | M | I |

ERR The ERR field contains error information for use by host software in determining the appropriate response to the error condition.

- C** Non-recovered persistent communication or data integrity error: A communication error that was not recovered occurred that is expected to be persistent. Since the error condition is expected to be persistent the operation need not be retried by host software. Persistent communications errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes.
- E** Internal error: The host bus adapter experienced an internal error that caused the operation to fail and may have put the host bus adapter into an error state. Host software should reset the interface before re-trying the operation. If the condition persists, the host bus adapter may suffer from a design issue rendering it incompatible with the attached device.
- I** Recovered data integrity error: A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action. This can arise from a noise burst in the transmission, a voltage supply variation, or from other causes. No action is required by host software since the operation ultimately succeeded, however, host software may elect to track such recovered errors in order to gauge overall communications integrity and step down the negotiated communication speed.
- M** Recovered communications error: Communications between the device and host was temporarily lost but was re-established. This can arise from a device temporarily being removed, from a temporary loss of Phy synchronization, or from other causes and may be derived from the PhyNRdy signal between the Phy and link layers. No action is required by the host software since the operation ultimately succeeded, however, host software may elect to track such recovered errors in order to gauge overall communications integrity and step down the negotiated communication speed.
- P** Protocol error: A violation of the serial transport protocol was detected. This can arise from invalid or poorly formed FISes being received, from invalid state transitions, or from other causes. Host software should reset the interface and retry the corresponding operation. If such an error persists, the attached device may have a design issue rendering it incompatible with the host bus adapter.
- R** Reserved
- T** Non-recovered transient data integrity error: A data integrity error occurred that was not recovered by the interface. Since the error condition is not expected to be persistent the operation should be retried by host software.

DIAG The DIAG field contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes.

- B** 10b to 8b Decode error: When set to a one, this bit indicates that one or more 10b to 8b decoding errors occurred since the bit was last cleared.
- C** CRC Error: When set to one, this bit indicates that one or more CRC errors occurred with the link layer since the bit was last cleared.

- D Disparity Error: When set to one, this bit indicates that incorrect disparity was detected one or more times since the last time the bit was cleared.
- F Unrecognized FIS type: When set to one, this bit indicates that since the bit was last cleared one or more FISes were received by the transport layer with good CRC, but had a type field that was not recognized.
- I Phy Internal Error: When set to one, this bit indicates that the Phy detected some internal error since the last time this bit was cleared.
- N PhyRdy change: When set to one, this bit indicates that the PhyRdy signal changed state since the last time this bit was cleared.
- H Handshake error: When set to one, this bit indicates that one or more R_ERR handshake response was received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 10b/8b decoding error, or other error condition leading to a negative handshake on a transmitted frame.
- R Reserved
- S Link Sequence Error: When set to one, this bit indicates that one or more link state machine error conditions was encountered since the last time this bit was cleared. The link layer state machine defines the conditions under which the link layer detects an erroneous transition.
- T Transport state transition error: When set to one, this bit indicates that an error has occurred in the transition from one state to another within the transport layer since the last time this bit was cleared.
- W COMWAKE: When set to one this bit indicates that a ComWake signal was detected by the Phy since the last time this bit was cleared (See Subclause 5.5.12.3.4).

10 Serial Transport error handling

10.1 Architecture

The layered architecture of the serial transport extends to error handling as well. As indicated in Figure 110, each layer in the serial implementation stack has as input error indications from the next lower layer (except for the physical layer which has no lower layer associated with it), data from the next lower layer in the stack, and data from the next higher layer in the stack. Each layer has its local error detection capability to identify errors specific to that layer based on the received data from the lower and higher layers. Each layer performs local recovery and control actions and may forward error information to the next higher layer in the stack.

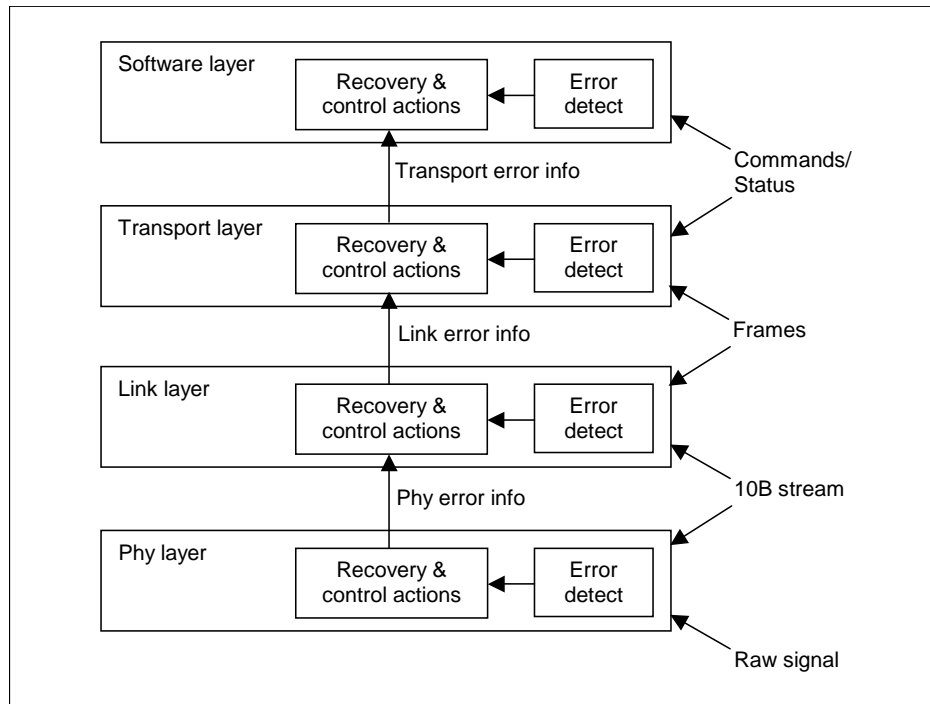


Figure 110 - Error handling architecture

Error responses are classified into four categories

Time out

- Abort
- Retry
- Track/ignore

The error handling responses described in this subclause are not comprehensive and are included to cover specific known error scenarios as well as to illustrate typical error control and recovery actions. This subclause is therefore informative and supplemental to the error reporting interface defined in clause 9 and implementations may vary in their internal error recovery and control actions.

Deleted: descriptive

Deleted: 10

For the most severe error conditions in which the state has been critically perturbed in a way that it is not recoverable, an appropriate error response is to time out and rely on a reset or similar operation to restore all necessary states to return to normal operation.

For error conditions that are expected to be persistent, the appropriate error response is to abort and fail the attempted operation. Such failures usually imply notification up the stack in order to inform host software of the condition.

For error conditions that are transient and not expected to persist, the appropriate response is to retry the failed operation. Only failed operations that have not perturbed system state are allowed to be retried. Such retries may either be handled directly by the recovery and error control actions in the relevant layer or may be handled by host software in response to error information being conveyed to it.

Non-critical recoverable conditions may either be tracked or ignored. Such conditions include those that were recovered through a retry or other recovery operation at a lower layer in the stack. Tracking such errors may often be beneficial in identifying a marginally operating component or other imminent failure.

10.2 Phy error handling overview

10.2.1 Error detection

There are three primary categories of error that the physical layer detects internally:

- a) No device present
- b) OOB signaling sequence failure
- c) Phy internal error (loss of synchronization of communications link)

A no device present condition results from a physical disconnection in the media between the host adapter and device, whether intermittent or persistent. The host adapter shall detect device presence as part of the interface reset sequence defined in 10.2.2.1.

An OOB signaling failure condition arises when the sequence of OOB signaling events cannot be completed, prior to assertion of the PhyRdy signal. OOB signaling sequences are required when emerging from a power management Partial mode or Slumber mode, from a "loopback/BIST" test state, or from initial power-up. OOB signaling sequences are used to achieve a specifically ordered exchange of ComReset Sequence, ComInit Sequence, ComWake Sequence, and ALIGN patterns to bring the communications link up between host adapter and device. The specific sequences are detailed in Subclause 5.

A Phy Internal Error can arise from a number of conditions, whether it is caused by the characteristics of the input signal, or an internal error unique to the implementation, it will always result in the loss of the synchronization of the communications link.

Fixed local receive PLL frequency architectures (oversampling, tracking/non-tracking) are sensitive to input data rate frequency variations from the nominal expected rate, thus they usually have elasticity buffers. Elasticity buffers are used to accommodate the difference between input data rate frequency, and the local receive PLL frequency -- overrun/underrun conditions will occur if the Tx difference in frequency is too high/low with respect to the Rx local PLL frequency, commonly declared as a Phy Internal Error.

VCO-based (PLL) clock recovery architectures are also sensitive to input data rate frequency variations, high frequency jitter, and may have trouble achieving lock -- which may be declared as a Phy Internal Error.

A number of state machine, impedance compensation, and serializer/deserializer (SerDes) circuits make up a typical the serial ATA interface Phy, and the various types of error conditions may be grouped together to make up the declared "Phy Internal Error". These errors are usually specific to each implementation.

10.2.2 Error control actions

10.2.2.1 No device present

Due to the nature of the physical interface, it is possible for the Phy to determine that a device is attached to the cable at various times. As a direct result, the Phy is responsible for detecting presence of an attached device and this presence shall be reported in the SStatus register such that host software can respond appropriately.

During the interface initialization sequence, an internal state bit "Device Detect" shall be cleared in the host adapter when the ComReset signal is issued. The "Device Detect" state bit shall be set in the host adapter when the host adapter detects a ComInit signal from the attached device. The "Device Detect" state bit corresponds to the device presence detect information in the SStatus register defined in 9.2.

Note that device presence and communications established are separately reported in the SStatus register in order to encompass situations in which an attached device is detected by the Phy, but the Phy is unable to establish communications with it.

10.2.2.2 OOB signaling sequence failure

The Phy does not have any timeout conditions for the interface reset signaling sequence as defined in 5.5.12.2. If a device is present, the Phy shall detect device presence within 10ms of a power-on reset (i.e. COMINIT shall be returned within 10ms of an issued COMRESET). If a device is not present, the Phy is not required to time-out and may remain in the reset state indefinitely until host software intervenes. Upon successful completion of the interface initialization sequence, the Phy is ready, active, and synchronized, and the SStatus register bits shall reflect this as defined in 9.2.1.

10.2.2.3 Phy internal error

As described in 10.2.1, there are several potential sources of errors categorized as "Phy Internal Errors." In order to accommodate a range of implementations without making the software error handling approach implementation dependent, all the different potential sources of internal Phy errors are combined for the purpose of reporting the condition in the SError register as defined in 9.2.2. This requirement does not preclude each vendor from implementing their own level of error diagnostic bits, but those shall reside in vendor specific register locations.

Phy internal errors that result in the Phy becoming not ready (the PhyRdy signal being negated) and the corresponding SStatus register and SError registers bits shall be updated as defined in 9.2.2.

The "Phy Ready Change" bit, as defined in the SError Register, is updated as per its definition in 9.2.2

10.2.3 Error reporting

Phy errors are reported to the link layer in addition to being reflected in the SStatus register and SError registers as defined in 9.2.2.

10.3 Link error handling overview

10.3.1 Error detection

There are two primary categories of errors that the link layer detects internally:

- a) Invalid state transitions
- b) Data integrity errors

Invalid state transition errors can arise from a number of sources and the link layer responses to many such error conditions are defined in Subclause 6. Data integrity errors generally arise from noise in the physical interconnect.

10.3.2 Error control actions

Errors detected by the Link during a transmission are handled by accumulating the errors until the end of the transmission and reflecting the reception error condition in the final R_ERR/R_OK handshake. Specific scenarios are listed in the following subclauses.

10.3.2.1 Invalid state transitions

Invalid state transitions are handled through the return to a known state, for example where the Link state diagrams in 6.7 define the responses for invalid state transition attempts. Returning to a known state is achieved through two recovery paths depending on the state of the system.

If the invalid state transitions are attempted during the transmission of a frame (after the receipt of an SOF), the Link shall signal negative acknowledgement (R_ERR) to the sender.

If the invalid state transition is not during a frame transmission, the Link shall go directly to the idle state, and await the next operation.

The following paragraphs outline the requirements during specific state transition scenarios, and their respective Link error control actions:

Following reception of one or more consecutive X_RDY at the receiver interface, if the next control character received is not SOF, the Link shall notify the transport layer of the condition and transition to the idle state.

Following transmission of X_RDY, if there is no returned R_RDY received, no Link recovery action shall be attempted. The higher-level layers will eventually time out, and reset the interface.

On receipt of an unexpected SOF, when the receiving interface had not yet signaled readiness to receive data with the R_RDY signal, that receiving interface shall remain in the idle state issuing SYNC primitives until the transmitting interface terminates the transmission and also returns to the idle state.

If the transmitter closes a frame with EOF and WTRM, and receives neither a R_OK nor an R_ERR within a predetermined timeout, no Link recovery action shall be attempted. The higher-level layers will eventually time out, and reset the interface.

If the transmitter signals EOF, and a control character other than SYNC, R_OK, or R_ERR is received, the link layer shall persistently continue to await reception of a proper terminating primitive.

10.3.2.2 Data integrity errors:

Data integrity errors are handled by signaling the transport layer in order to potentially trigger a transmission retry operation, or to convey failed status information to the host software. In order to return to a known state, data integrity errors are usually signaled via the frame acknowledgement handshake, at the end of a frame transmission, before returning to the idle state.

The following paragraphs outline the requirements during specific data integrity error scenarios, and the respective Link error control actions:

On detection of a CRC error at the end of receiving a frame (at EOF), the link layer shall notify the transport layer that the received frame contains a CRC error. Furthermore, the link layer shall issue the negative acknowledgement, R_ERR, as the frame status handshake, and shall return to the idle state.

On detection of a disparity error or other 8b/10b coding violation during the receipt of a frame, the link layer will retain this error information, and at the close of the received frame the link layer shall provide the negative acknowledgement, R_ERR, as the frame handshake, and shall notify the transport layer of the error.

The control actions are the same for coding violations as for CRC errors.

10.3.3 Error reporting

Link error conditions are reported to the transport layer in a vendor specific interface between the Link and transport layers. Additionally, Link errors are reported in the Interface SError register as defined 9.2.2.

10.4 Transport error handling

10.4.1 Overview

The transport layer communicates errors to the software and/or performs local error recovery, and initiates control actions, such as retrying FIS transmissions

The transport layer informs the link layer of detected errors so that the link layer can reflect Transport errors in the R_ERR/R_OK handshake at the end of each frame. Devices shall reflect any R_ERR frame handshakes in the command ending status for Data FISes reflected in the transmitted Response FIS that conveys the operation ending status. The transport layer also reflects any encountered error information in the Interface **S**Error register.

The Transport may retry any non-Data FIS transmission, provided the system state has not changed as a result of the corresponding failure, and may retry any number of times. For scenarios where repeated retry operations persistently fail, host software will time out the corresponding command and perform recovery operations.

10.4.2 Error detection

In addition to the error information passed to it by the link layer, the transport layer internally detects the following categories of errors:

- Internal errors
- Frame errors
- Protocol errors & state errors

There are several kinds of internal errors to the transport layer, including overflow/underflow of the various speed matching FIFOs. Internal errors are handled by failing the corresponding transaction, and returning to a state equivalent to a failed transaction (e. g. the state that would result from a bad CRC).

The transport layer detects several kinds of frame errors including reception of frames with incorrect CRC, reception of frames with invalid TYPE field, and reception of ill-formed frames (such as a frames that are not the correct length). Frame errors are handled by failing the corresponding transaction and returning to a state equivalent to a failed transaction (such as the state that would result from a bad CRC).

Protocol and state transition errors stem from devices not following the serial transport protocol, and include errors such as, the PIO count value not matching the number of data characters subsequently transferred, and errors in the sequence of events.

Protocol and state transition errors are handled by failing the corresponding transaction and returning to a state equivalent to a failed transaction (such as the state that would result from a frame being received with a bad CRC).

10.4.3 Error control actions

10.4.3.1 Internal errors

Internal errors are handled by failing the corresponding transaction and either re-trying the transaction or notifying host software of the failure condition in order to ultimately generate a host software retry response. The following are specific internal error scenarios and their corresponding Transport error control actions.

If the receive FIFO overflows, the transport layer shall signal frame reception negative acknowledgement, by signaling the link layer to return R_ERR during the frame acknowledgement handshake. Subsequent actions are equivalent to a frame reception with erroneous CRC.

If the transmit FIFO underruns, the transport layer shall close the transmitting frame with an EOF and CRC value that is forced to be incorrect in order to ensure the recipient of the corrupted frame also executes

appropriate error control actions. This scenario results only from a transmitter design problem and should not occur for properly implemented devices.

10.4.3.2 Frame errors

Frame errors may be handled in one of two ways depending on whether the error is expected to be transient or persistent and whether system state has been perturbed.

For error conditions expected to be transient (such as a CRC error), and for which the system state has not been perturbed, the transport layer may retry the corresponding transaction any number of times until ultimately a host timeout and software reset, or other error recovery attempt is made.

For error conditions that are not a result of a transient error condition (such as an invalid TYPE field in a received FIS), the error response is to fail the transaction and report the failure.

The following are specific frame error scenarios and their corresponding Transport error control actions.

If the Transport receives an FIS with an invalid CRC signaled from the link layer, the transport layer shall signal the link layer to negatively acknowledge frame reception by asserting R_ERR during the frame acknowledgement handshake.

The transmitter of a negatively acknowledged frame may retry the FIS transmission provided the system state has not been perturbed. Frame types that may be retransmitted are:

- Command FIS
- Response FIS
- DMA Activate - device to host
- First Party DMA Setup - device to host
- PIO Setup - device to host
- Set Device Bits - device to host
- BIST Activate **(Editors note: SATAIO reviewing for possible change)**

Formatted: Font: Bold

Because data transmission FISes result in a change in the host bus adapter's internal state, either through the DMA controller changing its state or through a change in the remaining PIO repetition count, data transmission FISes shall not be retried.

The transport layer is not required to retry those failed FIS transmissions that do not change system state, but the transport layer may attempt retry any number of times. For conditions that are not addressed through retries, such as persistent errors, host software will eventually time out the transaction and reset the interface.

If the transport layer detects reception of an FIS with unrecognized TYPE value, the transport layer shall signal the link layer to negatively acknowledge the frame reception by asserting R_ERR during the frame acknowledgement handshake.

If the transport layer detects reception of a malformed frame, such as a frame with incorrect length, the transport layer shall signal the link layer to negatively acknowledge the frame reception by asserting R_ERR during the frame acknowledgement handshake.

10.4.3.3 Protocol and state transition errors

Protocol and state errors stem from devices not following defined protocol. Such errors may be handled by failing the corresponding transactions and returning to a known state. Since such errors are not caused by an environmental transient, no attempt to retry such failed operations should be made. The following are specific frame error scenarios and their corresponding Transport error control actions.

If the PIO transfer count expires, and two symbols later is not the EOF control character (the CRC falls between the last data character and the EOF), the transfer count stipulated in the PIO Setup FIS did not

match the size of the subsequent data payload. For this data-payload/transfer-count mismatch, the transport layer shall signal the link layer to negatively acknowledge frame reception by asserting R_ERR during the frame acknowledgement handshake.

10.4.4 Error reporting

The transport layer reports errors to host software via the HBA response registers. Devices communicate Transport error information to host software via transmitting a response FIS to update the HBA response registers.

Transport error conditions that are not handled/recovered by the transport layer shall set the error bit in the Error field of the Response FIS, and update the value in the Error field through transmission of an appropriate Response FIS.

Host Transport error conditions shall result in the status and error values in the SStatus and SError register being updated with values corresponding to the error condition and shall result in the link layer being notified to negatively acknowledge the offending FIS during the final reception handshake.

10.5 Software error handling overview

The software layer error handling is in part defined by the error outputs of the specific command as defined in ATA8-ACS. In addition, there are superset error reporting capabilities supported by the transport layer through the SCRs, and software may take advantage of those error reporting capabilities to improve error handling for serial transport HBA's.

10.5.1 Error detection

There are three error detection mechanisms by which software identifies and responds to the serial transport errors

- a) Bad status in the HBA Status register
- b) Bad status in the serial transport SError register
- c) Command failed to complete (timeout)

Conditions that return bad status in the HBA Status register, but for which no serial implementation interface error information is available, correspond to the error conditions specified in the command descriptions. Such error conditions and responses are defined in the command descriptions and there is no unique handling of those in the serial transport. Errors in this category include command errors, such as attempts to read from an LBA past the end of the disk, as well as device-specific failures such as data not readable from the given LBA. These failures are not related to the serial transport interface, and thus no the serial transport specific interface status information is available for these error conditions. Only the status information returned by the device is available for identifying the source of the problem, plus any available SMART data that might apply.

Transport layer error conditions, whether recovered or not, are reflected in the SStatus and SError registers as defined in 9.2.2. The host transport layer is responsible for reflecting error information in the SStatus and SError registers, while the device transport layer is responsible for reflecting unrecovered errors in the HBA response registers through transmission of appropriate Response FISes.

Commands that fail to complete are detected by host driver software through a timeout mechanism. Such timeouts may not result in status or error information for the command being conveyed to host software and software may not be able to determine the source or cause of such errors.

10.5.2 Error control actions

Conditions that return bad status in the HBA response registers but for which no interface error information in the SError register is available shall be handled as defined in the Parallel Transport (ATA8-PT).

Conditions that return interface error information in the SError register are handled through four basic responses:

- Time out
- Abort/Fail
- Retry (possible after reset)
- Track/ignore

Error conditions that are not expected to be transient and which are not expected to succeed with subsequent attempts should result in the affected command being aborted and failed. Failure of such commands should be reported to higher software layers for handling. Scenarios in which this response is appropriate include attempts to communicate with a device that is not attached, and failure of the interface to successfully negotiate communications with an attached device.

Error conditions that are expected to be transient should result in the affected command being retried. Such commands may either be retried directly or may be retried after an interface and/or DEVICE RESET, depending on the particular error value reported in the SError register. Scenarios in which this response is appropriate include noise events resulting in CRC errors, 8b/10b code violations, or disparity errors.

Conditions that are recoverable and for which no explicit error handling is required may be tracked or ignored. Tracking such errors allows subsequent fault isolation for marginal components and accommodates possible recovery operations. Scenarios in which this response is appropriate include tracking the number of PHY synchronization losses in order to identify a potential cable fault or to accommodate an explicit reduction in the negotiated communications rate.

ANNEX A. SAMPLE CODE FOR SERIAL CRC SCRAMBLING (INFORMATIVE)

A.1 CRC calculation

A.1.1 Overview

The following subclause provides an informative implementation of the Serial ATA CRC polynomial. The example is intended as an aid in verifying a HDL implementation of the algorithm.

A.1.2 aximum frame size

The 32-bit CRC used by the serial transport can be shown to provide detection of two 10-bit errors up to a maximum frame size of 16384 bytes. This will provide for future expansion of the serial transport FISes to a maximum of 64 bytes of fixed overhead while still permitting a maximum user data payload of 8192 bytes.

A.1.3 Example code for CRC algorithm

The following code, written in C, illustrates an implementation of the CRC algorithm. A Command FIS containing a PIO write command is used as example input. The CRC calculated is the check dword appended to a transmitted serial stream immediately preceding the EOF primitive. The code displays both the resulting Command FIS and the intermediate CRC polynomial values. The GNU tool chain will compile the code using the following command:

```
"gcc -o crc.exe crc.c"
```

This code is supplied for illustrative purposes only.

```

/*****
/*
/* crc.c
/*
/* This sample code reads standard in for a sequence of 32 bit values
/* formatted in hexadecimal with a leading "0x" (e.g. 0xDEADBEEF). The
/* code calculates the serial ATA CRC for the input data stream. The
/* generator polynomial used is:
/*
/*      32  26  23  22  16  12  11  10  8  7  5  4  2
/* G(x) = x + x + x + x + x + x + x + x + x + x + x + x + x + 1
/*
/* This sample code uses a Parallel Transport of the CRC calculation
/* circuit that is suitable for implementation in hardware. A block
/* diagram of the circuit being emulated is shown below.
/*
/*
/*      +-----+      +-----+      +-----+
/*      |   +   |      |   *   |      |   R   |
/* Data_In ----->|----->|----->|   e   |-----+
/*      |   +   |      |   *   |      |   g   |
/*      +-----+      +-----+      +-----+
/*
/*
/*
/*
/*
*/

```

```

/* The CRC value is initialized to 0x52325032 */
/* */
/* */
/*****/

#include <stdlib.h>
#include <stdio.h>

main(argc,argv)
int argc;
char *argv[];
{
    int            i,
                  data_count;
    unsigned int   crc,
                  data_in;
    unsigned char  crc_bit[32],
                  new_bit[32];

    crc = 0x52325032;
    data_count = 0;

    while (scanf(" 0x%x", &data_in) == 1) {
        data_count++;
        /* Add the data_in value to the current value of the CRC held in the */
        /* "register". The addition is performed modulo two (XOR). */
        crc ^= data_in;
        /* Expand the value of the CRC held in the register to 32 individual */
        /* bits for easy manipulation. */
        for (i = 0; i < 32; ++i) {
            crc_bit[i] = (crc >> i) & 0x01;
        }

        /* The following 32 assignments perform the function of the box */
        /* labeled "*" in the block diagram above. The new_bit array is a */
        /* temporary holding place for the new CRC value being calculated. */
        /* Note that there are lots of shared terms in the assignments below. */
        new_bit[31] = crc_bit[31] ^ crc_bit[30] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[27] ^
        crc_bit[25] ^ crc_bit[24] ^
            crc_bit[23] ^ crc_bit[15] ^ crc_bit[11] ^ crc_bit[9] ^ crc_bit[8] ^
        crc_bit[5];
        new_bit[30] = crc_bit[30] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[26] ^
        crc_bit[24] ^ crc_bit[23] ^
            crc_bit[22] ^ crc_bit[14] ^ crc_bit[10] ^ crc_bit[8] ^ crc_bit[7] ^
        crc_bit[4];
        new_bit[29] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[26] ^
        crc_bit[25] ^ crc_bit[23] ^
            crc_bit[22] ^ crc_bit[21] ^ crc_bit[13] ^ crc_bit[9] ^ crc_bit[7] ^
        crc_bit[6] ^ crc_bit[3];
        new_bit[28] = crc_bit[30] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[25] ^
        crc_bit[24] ^ crc_bit[22] ^
            crc_bit[21] ^ crc_bit[20] ^ crc_bit[12] ^ crc_bit[8] ^ crc_bit[6] ^
        crc_bit[5] ^ crc_bit[2];
        new_bit[27] = crc_bit[29] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[24] ^
        crc_bit[23] ^ crc_bit[21] ^
            crc_bit[20] ^ crc_bit[19] ^ crc_bit[11] ^ crc_bit[7] ^ crc_bit[5] ^
        crc_bit[4] ^ crc_bit[1];
        new_bit[26] = crc_bit[31] ^ crc_bit[28] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[24] ^
        crc_bit[23] ^ crc_bit[22] ^
            crc_bit[20] ^ crc_bit[19] ^ crc_bit[18] ^ crc_bit[10] ^ crc_bit[6] ^
        crc_bit[4] ^ crc_bit[3] ^
            crc_bit[0];
        new_bit[25] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[22] ^ crc_bit[21] ^
        crc_bit[19] ^ crc_bit[18] ^
    }
}

```

```

        crc_bit[17] ^ crc_bit[15] ^ crc_bit[11] ^ crc_bit[8] ^ crc_bit[3] ^
crc_bit[2];
    new_bit[24] = crc_bit[30] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[21] ^ crc_bit[20] ^
crc_bit[18] ^ crc_bit[17] ^
        crc_bit[16] ^ crc_bit[14] ^ crc_bit[10] ^ crc_bit[7] ^ crc_bit[2] ^
crc_bit[1];
    new_bit[23] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[20] ^
crc_bit[19] ^ crc_bit[17] ^
        crc_bit[16] ^ crc_bit[15] ^ crc_bit[13] ^ crc_bit[9] ^ crc_bit[6] ^
crc_bit[1] ^ crc_bit[0];
    new_bit[22] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[24] ^
crc_bit[23] ^ crc_bit[19] ^
        crc_bit[18] ^ crc_bit[16] ^ crc_bit[14] ^ crc_bit[12] ^ crc_bit[11] ^
crc_bit[9] ^ crc_bit[0];
    new_bit[21] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[24] ^
crc_bit[22] ^ crc_bit[18] ^
        crc_bit[17] ^ crc_bit[13] ^ crc_bit[10] ^ crc_bit[9] ^ crc_bit[5];
    new_bit[20] = crc_bit[30] ^ crc_bit[28] ^ crc_bit[26] ^ crc_bit[25] ^ crc_bit[23] ^
crc_bit[21] ^ crc_bit[17] ^
        crc_bit[16] ^ crc_bit[12] ^ crc_bit[9] ^ crc_bit[8] ^ crc_bit[4];
    new_bit[19] = crc_bit[29] ^ crc_bit[27] ^ crc_bit[25] ^ crc_bit[24] ^ crc_bit[22] ^
crc_bit[20] ^ crc_bit[16] ^
        crc_bit[15] ^ crc_bit[11] ^ crc_bit[8] ^ crc_bit[7] ^ crc_bit[3];
    new_bit[18] = crc_bit[31] ^ crc_bit[28] ^ crc_bit[26] ^ crc_bit[24] ^ crc_bit[23] ^
crc_bit[21] ^ crc_bit[19] ^
        crc_bit[15] ^ crc_bit[14] ^ crc_bit[10] ^ crc_bit[7] ^ crc_bit[6] ^
crc_bit[2];
    new_bit[17] = crc_bit[31] ^ crc_bit[30] ^ crc_bit[27] ^ crc_bit[25] ^ crc_bit[23] ^
crc_bit[22] ^ crc_bit[20] ^
        crc_bit[18] ^ crc_bit[14] ^ crc_bit[13] ^ crc_bit[9] ^ crc_bit[6] ^
crc_bit[5] ^ crc_bit[1];
    new_bit[16] = crc_bit[30] ^ crc_bit[29] ^ crc_bit[26] ^ crc_bit[24] ^ crc_bit[22] ^
crc_bit[21] ^ crc_bit[19] ^
        crc_bit[17] ^ crc_bit[13] ^ crc_bit[12] ^ crc_bit[8] ^ crc_bit[5] ^
crc_bit[4] ^ crc_bit[0];
    new_bit[15] = crc_bit[30] ^ crc_bit[27] ^ crc_bit[24] ^ crc_bit[21] ^ crc_bit[20] ^
crc_bit[18] ^ crc_bit[16] ^
        crc_bit[15] ^ crc_bit[12] ^ crc_bit[9] ^ crc_bit[8] ^ crc_bit[7] ^
crc_bit[5] ^ crc_bit[4] ^
        crc_bit[3];
    new_bit[14] = crc_bit[29] ^ crc_bit[26] ^ crc_bit[23] ^ crc_bit[20] ^ crc_bit[19] ^
crc_bit[17] ^ crc_bit[15] ^
        crc_bit[14] ^ crc_bit[11] ^ crc_bit[8] ^ crc_bit[7] ^ crc_bit[6] ^
crc_bit[4] ^ crc_bit[3] ^
        crc_bit[2];
    new_bit[13] = crc_bit[31] ^ crc_bit[28] ^ crc_bit[25] ^ crc_bit[22] ^ crc_bit[19] ^
crc_bit[18] ^ crc_bit[16] ^
        crc_bit[14] ^ crc_bit[13] ^ crc_bit[10] ^ crc_bit[7] ^ crc_bit[6] ^
crc_bit[5] ^ crc_bit[3] ^
        crc_bit[2] ^ crc_bit[1];
    new_bit[12] = crc_bit[31] ^ crc_bit[30] ^ crc_bit[27] ^ crc_bit[24] ^ crc_bit[21] ^
crc_bit[18] ^ crc_bit[17] ^
        crc_bit[15] ^ crc_bit[13] ^ crc_bit[12] ^ crc_bit[9] ^ crc_bit[6] ^
crc_bit[5] ^ crc_bit[4] ^
        crc_bit[2] ^ crc_bit[1] ^ crc_bit[0];
    new_bit[11] = crc_bit[31] ^ crc_bit[28] ^ crc_bit[27] ^ crc_bit[26] ^ crc_bit[25] ^
crc_bit[24] ^ crc_bit[20] ^
        crc_bit[17] ^ crc_bit[16] ^ crc_bit[15] ^ crc_bit[14] ^ crc_bit[12] ^
crc_bit[9] ^ crc_bit[4] ^
        crc_bit[3] ^ crc_bit[1] ^ crc_bit[0];
    new_bit[10] = crc_bit[31] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[26] ^ crc_bit[19] ^
crc_bit[16] ^ crc_bit[14] ^
        crc_bit[13] ^ crc_bit[9] ^ crc_bit[5] ^ crc_bit[3] ^ crc_bit[2] ^
crc_bit[0];
    new_bit[9] = crc_bit[29] ^ crc_bit[24] ^ crc_bit[23] ^ crc_bit[18] ^ crc_bit[13] ^
crc_bit[12] ^ crc_bit[11] ^

```

```

        crc_bit[9] ^ crc_bit[5] ^ crc_bit[4] ^ crc_bit[2] ^ crc_bit[1];
    new_bit[8] = crc_bit[31] ^ crc_bit[28] ^ crc_bit[23] ^ crc_bit[22] ^ crc_bit[17] ^
crc_bit[12] ^ crc_bit[11] ^
        crc_bit[10] ^ crc_bit[8] ^ crc_bit[4] ^ crc_bit[3] ^ crc_bit[1] ^
crc_bit[0];
    new_bit[7] = crc_bit[29] ^ crc_bit[28] ^ crc_bit[25] ^ crc_bit[24] ^ crc_bit[23] ^
crc_bit[22] ^ crc_bit[21] ^
        crc_bit[16] ^ crc_bit[15] ^ crc_bit[10] ^ crc_bit[8] ^ crc_bit[7] ^
crc_bit[5] ^ crc_bit[3] ^
        crc_bit[2] ^ crc_bit[0];
    new_bit[6] = crc_bit[30] ^ crc_bit[29] ^ crc_bit[25] ^ crc_bit[22] ^ crc_bit[21] ^
crc_bit[20] ^ crc_bit[14] ^
        crc_bit[11] ^ crc_bit[8] ^ crc_bit[7] ^ crc_bit[6] ^ crc_bit[5] ^
crc_bit[4] ^ crc_bit[2] ^
        crc_bit[1];
    new_bit[5] = crc_bit[29] ^ crc_bit[28] ^ crc_bit[24] ^ crc_bit[21] ^ crc_bit[20] ^
crc_bit[19] ^ crc_bit[13] ^
        crc_bit[10] ^ crc_bit[7] ^ crc_bit[6] ^ crc_bit[5] ^ crc_bit[4] ^
crc_bit[3] ^ crc_bit[1] ^
        crc_bit[0];
    new_bit[4] = crc_bit[31] ^ crc_bit[30] ^ crc_bit[29] ^ crc_bit[25] ^ crc_bit[24] ^
crc_bit[20] ^ crc_bit[19] ^
        crc_bit[18] ^ crc_bit[15] ^ crc_bit[12] ^ crc_bit[11] ^ crc_bit[8] ^
crc_bit[6] ^ crc_bit[4] ^
        crc_bit[3] ^ crc_bit[2] ^ crc_bit[0];
    new_bit[3] = crc_bit[31] ^ crc_bit[27] ^ crc_bit[25] ^ crc_bit[19] ^ crc_bit[18] ^
crc_bit[17] ^ crc_bit[15] ^
        crc_bit[14] ^ crc_bit[10] ^ crc_bit[9] ^ crc_bit[8] ^ crc_bit[7] ^
crc_bit[3] ^ crc_bit[2] ^
        crc_bit[1];
    new_bit[2] = crc_bit[31] ^ crc_bit[30] ^ crc_bit[26] ^ crc_bit[24] ^ crc_bit[18] ^
crc_bit[17] ^ crc_bit[16] ^
        crc_bit[14] ^ crc_bit[13] ^ crc_bit[9] ^ crc_bit[8] ^ crc_bit[7] ^
crc_bit[6] ^ crc_bit[2] ^
        crc_bit[1] ^ crc_bit[0];
    new_bit[1] = crc_bit[28] ^ crc_bit[27] ^ crc_bit[24] ^ crc_bit[17] ^ crc_bit[16] ^
crc_bit[13] ^ crc_bit[12] ^
        crc_bit[11] ^ crc_bit[9] ^ crc_bit[7] ^ crc_bit[6] ^ crc_bit[1] ^
crc_bit[0];
    new_bit[0] = crc_bit[31] ^ crc_bit[30] ^ crc_bit[29] ^ crc_bit[28] ^ crc_bit[26] ^
crc_bit[25] ^ crc_bit[24] ^
        crc_bit[16] ^ crc_bit[12] ^ crc_bit[10] ^ crc_bit[9] ^ crc_bit[6] ^
crc_bit[0];

    /* The new CRC value has been calculated as individual bits in the */
    /* new_bit array. Re-assembled it into a 32 bit value and "clock" it */
    /* into the "register". */
    crc = 0;
    for (i = 31; i >= 0; --i) {
        crc = crc << 1;
        crc |= new_bit[i];
    }
    printf("Running CRC value is 0x%08X\n", crc);
}

printf("\n\nThe total number of data words processed was %d\n", data_count);
printf("The CRC is 0x%08X\n\n", crc);

return 0;
}

```

A.1.4 Example CRC implementation output

The following is the sample data used as input for the example stored in file *sample*:

```
0x00308027
0xE1234567
0x00000000
0x00000002
0x00000000
```

Executing the command `./crc < sample` yields the following output:

```
Running CRC value is 0x11E353FD
Running CRC value is 0x0F656DA7
Running CRC value is 0x3D14369C
Running CRC value is 0x92D0D681
Running CRC value is 0x319FFF6F
```

```
The total number of data words processed was 5
The CRC is 0x319FFF6F
```

A.2 Scrambling calculation

A.2.1 Overview

The following subclause provides an informative implementation of the scrambling polynomial. The example is intended as an aid in verifying a HDL implementation of the algorithm.

A.2.2 Example code for scrambling algorithm

The following code, written in C, illustrates an implementation of the scrambling algorithm. A Command FIS containing a PIO write command is used as example input. The code displays both the resulting scrambled data and the raw output of the scrambler. The GNU tool chain will compile the code using the following command:

```
"gcc -o scramble.exe scramble.c"
```

This code is supplied for illustrative purposes only.


```

/* Parallelized versions of the scrambler are initialized to a value */
/* derived from the initialization value of 0xFFFF defined in the */
/* standard. This implementation is initialized to 0xF0F6. Other */
/* Parallel Transports will have different initial values. The */
/* important point is that the first dword output of any implementation */
/* is equal to 0xC2D2768D. */
context = 0xF0F6;

for (i = 0; i < 65535; ++i) {
    /* Split the register contents (the variable context) up into its */
    /* individual bits for easy handling. */
    for (j = 0; j < 16; ++j) {
        now[j] = (context >> j) & 0x01;
    }

    /* The following 16 assignments implement the matrix multiplication */
    /* performed by the box labeled *M1. */
    /* Notice that there are lots of shared terms in these assignments. */
    next[31] = now[12] ^ now[10] ^ now[7] ^ now[3] ^ now[1] ^ now[0];
    next[30] = now[15] ^ now[14] ^ now[12] ^ now[11] ^ now[9] ^ now[6] ^ now[3] ^
now[2] ^ now[0];
    next[29] = now[15] ^ now[13] ^ now[12] ^ now[11] ^ now[10] ^ now[8] ^ now[5] ^
now[3] ^ now[2] ^ now[1];
    next[28] = now[14] ^ now[12] ^ now[11] ^ now[10] ^ now[9] ^ now[7] ^ now[4] ^
now[2] ^ now[1] ^ now[0];
    next[27] = now[15] ^ now[14] ^ now[13] ^ now[12] ^ now[11] ^ now[10] ^ now[9] ^
now[8] ^ now[6] ^ now[1] ^ now[0];
    next[26] = now[15] ^ now[13] ^ now[11] ^ now[10] ^ now[9] ^ now[8] ^ now[7] ^
now[5] ^ now[3] ^ now[0];
    next[25] = now[15] ^ now[10] ^ now[9] ^ now[8] ^ now[7] ^ now[6] ^ now[4] ^
now[3] ^ now[2];
    next[24] = now[14] ^ now[9] ^ now[8] ^ now[7] ^ now[6] ^ now[5] ^ now[3] ^
now[2] ^ now[1];
    next[23] = now[13] ^ now[8] ^ now[7] ^ now[6] ^ now[5] ^ now[4] ^ now[2] ^
now[1] ^ now[0];
    next[22] = now[15] ^ now[14] ^ now[7] ^ now[6] ^ now[5] ^ now[4] ^ now[1] ^
now[0];
    next[21] = now[15] ^ now[13] ^ now[12] ^ now[6] ^ now[5] ^ now[4] ^ now[0];
    next[20] = now[15] ^ now[11] ^ now[5] ^ now[4];
    next[19] = now[14] ^ now[10] ^ now[4] ^ now[3];
    next[18] = now[13] ^ now[9] ^ now[3] ^ now[2];
    next[17] = now[12] ^ now[8] ^ now[2] ^ now[1];
    next[16] = now[11] ^ now[7] ^ now[1] ^ now[0];

    /* The following 16 assignments implement the matrix multiplication */
    /* performed by the box labeled *M2. */
    next[15] = now[15] ^ now[14] ^ now[12] ^ now[10] ^ now[6] ^ now[3] ^ now[0];
    next[14] = now[15] ^ now[13] ^ now[12] ^ now[11] ^ now[9] ^ now[5] ^ now[3] ^
now[2];
    next[13] = now[14] ^ now[12] ^ now[11] ^ now[10] ^ now[8] ^ now[4] ^ now[2] ^
now[1];
    next[12] = now[13] ^ now[11] ^ now[10] ^ now[9] ^ now[7] ^ now[3] ^ now[1] ^
now[0];
    next[11] = now[15] ^ now[14] ^ now[10] ^ now[9] ^ now[8] ^ now[6] ^ now[3] ^
now[2] ^ now[0];
    next[10] = now[15] ^ now[13] ^ now[12] ^ now[9] ^ now[8] ^ now[7] ^ now[5] ^
now[3] ^ now[2] ^ now[1];
    next[9] = now[14] ^ now[12] ^ now[11] ^ now[8] ^ now[7] ^ now[6] ^ now[4] ^
now[2] ^ now[1] ^ now[0];
    next[8] = now[15] ^ now[14] ^ now[13] ^ now[12] ^ now[11] ^ now[10] ^ now[7] ^
now[6] ^ now[5] ^ now[1] ^ now[0];
    next[7] = now[15] ^ now[13] ^ now[11] ^ now[10] ^ now[9] ^ now[6] ^ now[5] ^
now[4] ^ now[3] ^ now[0];

```

```

    next[6] = now[15] ^ now[10] ^ now[9] ^ now[8] ^ now[5] ^ now[4] ^ now[2];
    next[5] = now[14] ^ now[9] ^ now[8] ^ now[7] ^ now[4] ^ now[3] ^ now[1];
    next[4] = now[13] ^ now[8] ^ now[7] ^ now[6] ^ now[3] ^ now[2] ^ now[0];
    next[3] = now[15] ^ now[14] ^ now[7] ^ now[6] ^ now[5] ^ now[3] ^ now[2] ^
now[1];
    next[2] = now[14] ^ now[13] ^ now[6] ^ now[5] ^ now[4] ^ now[2] ^ now[1] ^
now[0];
    next[1] = now[15] ^ now[14] ^ now[13] ^ now[5] ^ now[4] ^ now[1] ^ now[0];
    next[0] = now[15] ^ now[13] ^ now[4] ^ now[0];

    /* The 32 bits of the output have been generated in the "next" array. */
    /* Reassemble the bits into a 32 bit dword. */
    scrambler = 0;
    for (j = 31; j >= 0; --j) {
        scrambler = scrambler << 1;
        scrambler |= next[j];
    }
    /* The upper half of the scrambler output is stored backed into the */
    /* register as the saved context for the next cycle. */
    context = scrambler >> 16;
    printf("0x%08X\n", scrambler);

}

return 0;

}

```

A.2.3 Example scrambler implementation

The following lists the first 32 results generated by the scrambler and the C sample code listed above.

```

0xC2D2768D
0x1F26B368
0xA508436C
0x3452D354
0x8A559502
0xBB1ABE1B
0xFA56B73D
0x53F60B1B
0xF0809C41
0x747FC34A
0xBE865291
0x7A6FA7B6
0x3163E6D6
0xF036FE0C
0x1EF3EA29
0xEB342694
0x53853B17
0xE94ADC4D
0x5D200E88
0x6901EDD0
0xFA9E38DE
0x68DB4B07
0x450A437B
0x960DD708
0x3F35E698
0xFE7698A5
0xC80EF715
0x666090AF

```

0xFAF0D5CB
 0x2B82009F
 0x0E317491
 0x76F46A1E

A.3 Example frame

The following table shows the steps and values used in the transmission of a simple FIS.

For LBA = 1234567 and Count = 2

Deleted: Sector

Table 32 - CRC and scrambler calculation example - PIO Write Command

| FIS Data | Scrambler Value | Scrambled Data | Accumulated CRC Value | Comments |
|----------|-----------------|----------------|-----------------------|--|
| 3737B57C | N/A | 3737B57C | N/A | SOF, primitives not scrambled, scrambler reset |
| 00308027 | C2D2768D | C2E2F6AA | 11E353FD | RegH2D, Command = 30, Cbit set |
| E1234567 | 1F26B368 | FE05F60F | 0F656DA7 | LBA 1234567 |
| 00000000 | A508436C | A508436C | 3D14369C | Extended LBA = 0 |
| 00000002 | 3452D354 | 3452D356 | 92D0D681 | Control Reg = 0, 2 sectors |
| 00000000 | 8A559502 | 8A559502 | 319FFF6F | reserved = 0 |
| 319FFF6F | BB1ABE1B | 8A854174 | N/A | CRC |
| D5D5B57C | N/A | D5D5B57C | N/A | EOF, primitives not scrambled |

NOTE – All values in hexadecimal, shown 31:0.

The transmitted dwords for this FIS, prior to 8b/10b encoding, are :

SOF
 C2E2F6AA
 FE05F60F
 A508436C
 3452D356
 8A559502
 8A854174
 EOF

ANNEX B. FIS TYPE FIELD VALUE SELECTION (INFORMATIVE)

B.1 Overview

The values for the TYPE field of the FISes have been selected in order to provide additional robustness. In minimally buffered implementations that may not buffer a complete FIS, the state machines may begin acting on the received FIS TYPE value prior to the ending CRC having been checked. Because the TYPE value may be acted upon prior to the integrity of the complete FIS being checked against its ending CRC, the TYPE field values have been selected to maximize the Hamming distance between them.

Several considerations are made in selecting the TYPE field values. Since the value is 8b/10b encoded, the Hamming distance for the values after 10b encoding is maximized. Since the starting running disparity at the time the TYPE field is transmitted is not known, values have been selected that have the same encoding for both negative and positive starting running disparity.

B.2 Type field values

Table 33 lists the selected TYPE field values (with their 10b encoding) that have a Hamming distance of 4 or greater in the 10b space after scrambling and encoding and which have the same encoding for both positive and negative running disparity.

Table 33 - Type field values

| Type field value (hex) | Scrambler syndrome ¹ (hex) | 8b scrambled value (hex) | 10b encoding (binary) | Current assignment |
|---|---------------------------------------|--------------------------|-----------------------|-----------------------------------|
| 0x27 | 0x8D | 0xAA | 0101011010 | Command or Control host to device |
| 0x34 | 0x8D | 0xB9 | 1001101010 | Response - device to host |
| 0x39 | 0x8D | 0xB4 | 0010111010 | DMA activate |
| 0x41 | 0x8D | 0xCC | 0011010110 | DMA setup |
| 0x46 | 0x8D | 0xCB | 1101000110 | Data |
| 0x58 | 0x8D | 0xD5 | 1010100110 | BIST activate |
| 0x5F | 0x8D | 0xD2 | 0100110110 | PIO setup |
| 0xA1 | 0x8D | 0x2C | 0011011001 | Set device bits |
| 0xA6 | 0x8D | 0x2B | 1101001001 | Reserved |
| 0xB8 | 0x8D | 0x35 | 1010101001 | Reserved |
| 0xBF | 0x8D | 0x32 | 0100111001 | Reserved |
| 0xC7 | 0x8D | 0x4A | 0101010101 | Reserved |
| 0xD4 | 0x8D | 0x59 | 1001100101 | Reserved |
| 0xD9 | 0x8D | 0x54 | 0010110101 | Reserved |
| NOTE – | | | | |
| 1. The scrambler syndrome generator is reset on the SOF primitive and the type field value immediately follows the SOF primitive. The scrambler syndrome at the time the type field value is transmitted is therefore deterministic and is equal to the seed used for the scrambling generator. See the scrambler generator subclause A.2 for more information on the details of the scrambler and its generated syndromes. | | | | |

ANNEX C. PHYSICAL LAYER IMPLEMENTATION EXAMPLES (INFORMATIVE)

C.1 Cable construction example

Figure 111 shows a cable construction example.

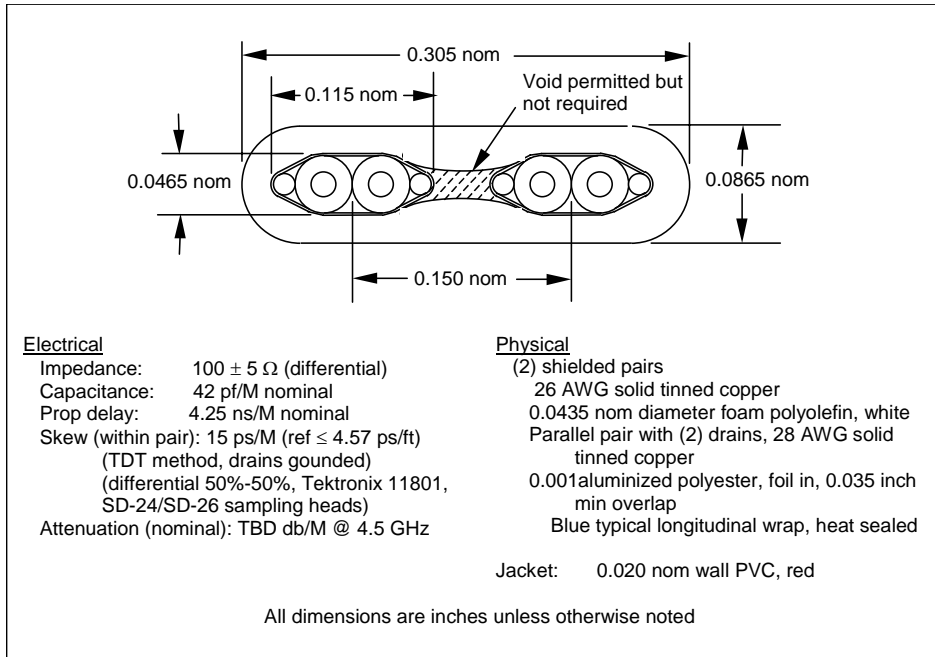


Figure 111 - Cable construction example

C.2 Contact material and plating

Table 34 shows an example for the contact material and plating.

Table 34 - Contact material and plating example

| Parameter | Recommendation | Comments |
|----------------|---|--|
| Material | Copper alloys, for example, brass for plug contacts and phosphor bronze for receptacle contact, the spring. | Material temper and thickness should be selected based on normal force and elastic deflection range consideration. |
| Mating Plating | Side For 50 durability cycles: 1.27 μm (50 μin) minimum Ni with either 0.38 μm (15 μin) minimum Au or 0.38 μm (15 μin) minimum 80/20 Pd/Ni with 0.051 μm (2 μin) minimum Au flash For 500 durability cycles: 1.27 μm (50 μin) minimum Ni with either 0.76 μm (30 μin) minimum Au or 0.76 μm (30 μin) minimum 80/20 Pd/Ni with 0.051 μm (2 μin) minimum Au flash | Exposed underplate or base material is not allowed in the mating area. |
| Solder Plating | Side Either Sn/Pb plating or Pd/Ni with Au flash: 1.27 μm (50 μin) minimum Ni with 3.18 μm (125 μin) minimum Sn/Pb. Or 1.27 μm (50 μin) minimum Ni with 0.76 μm (30 μin) minimum 80/20 Pd/Ni with 0.051 μm (2 μin) minimum Au flash. | Exposed base material is allowed in small areas where the contact is excised from its carrier strip or bandolier. |

C.3 Relationship of frequency to the jitter specification

It is often useful to look at the jitter specification as a function of frequency. This subclause is provided as clarifying information (informative, not normative). Figure 112 shows a plot of the maximum amplitude (in UI) sine wave at a given frequency that satisfies all the jitter specifications and the calculations leading to this graph. Two graphs are included - the second is a close-up of the high-frequency

requirements.

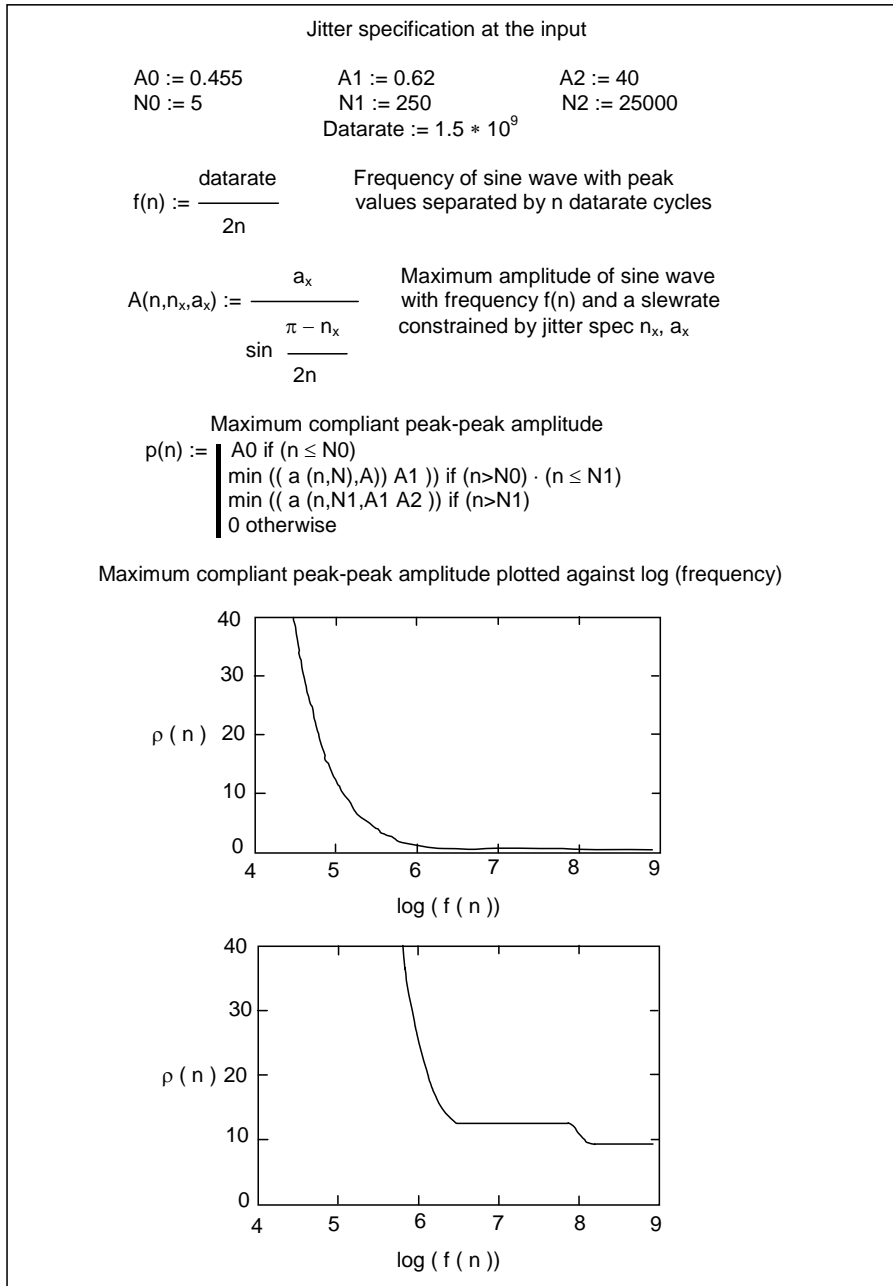


Figure 112 - Jitter as a function of frequency

C.4 Sampling BER and jitter formulas

The values for RJ and DJ above are calculated using the following relationship between BER and jitter. This is provided for reference (informative) and not intended for use as a compliance requirement (normative). The over-sampling formula assumes a worst-case oversampling ratio (osr) of three and sets the A_{0,n_0} jitter requirements. The tracking architecture sets the A_{1,n_1} requirement. A_{2,n_2} is set to guarantee SSC compliance. Also shown below are DJ_{RX} and RJ_{RX} numbers for the two architectures. These are example budgets for the local receiver that satisfy the 10^{-12} BER goal.

Gaussian distribution error function

$$G(x) := \frac{1}{\sqrt{2 \cdot \pi}} \int_0^x e^{-\frac{\xi^2}{2}} d\xi + 0.5 \quad x > 0$$

BER due to sampling (oversampling architecture)

osr := 3 $DJ_{RX} := 0.1$ $RJ_{RX} := 0.1$
 $DJ := 0.275 + DJ_{RX}$ $RJ := 0.18 + RJ_{RX}$

$$BER_{os} := 2 - G\left(\frac{\frac{1 - DJ}{osr - 2}}{\frac{RJ}{14}}\right) - G\left(\frac{\frac{osr - 1 - DJ}{osr - 2}}{\frac{RJ}{14}}\right)$$

$BER_{os} := 1.53 * 10^{-13}$

BER due to sampling (tracking architecture)

$DJ_{RX} := 0.15$ $RJ_{RX} := 0.18$
 $DJ := 0.37 + DJ_{RX}$ $RJ := 0.25 + RJ_{RX}$

$$BER_{tr} := 2 - G\left(\frac{\frac{1 - DJ}{2 - 2}}{\frac{RJ}{14}}\right) - G\left(\frac{\frac{1 - DJ}{2 + 2}}{\frac{RJ}{14}}\right)$$

$BER_{tr} := 2.665 * 10^{-15}$

Figure 113 - Sampling bit error rate formulas

C.5 DC and AC coupled transmitter examples

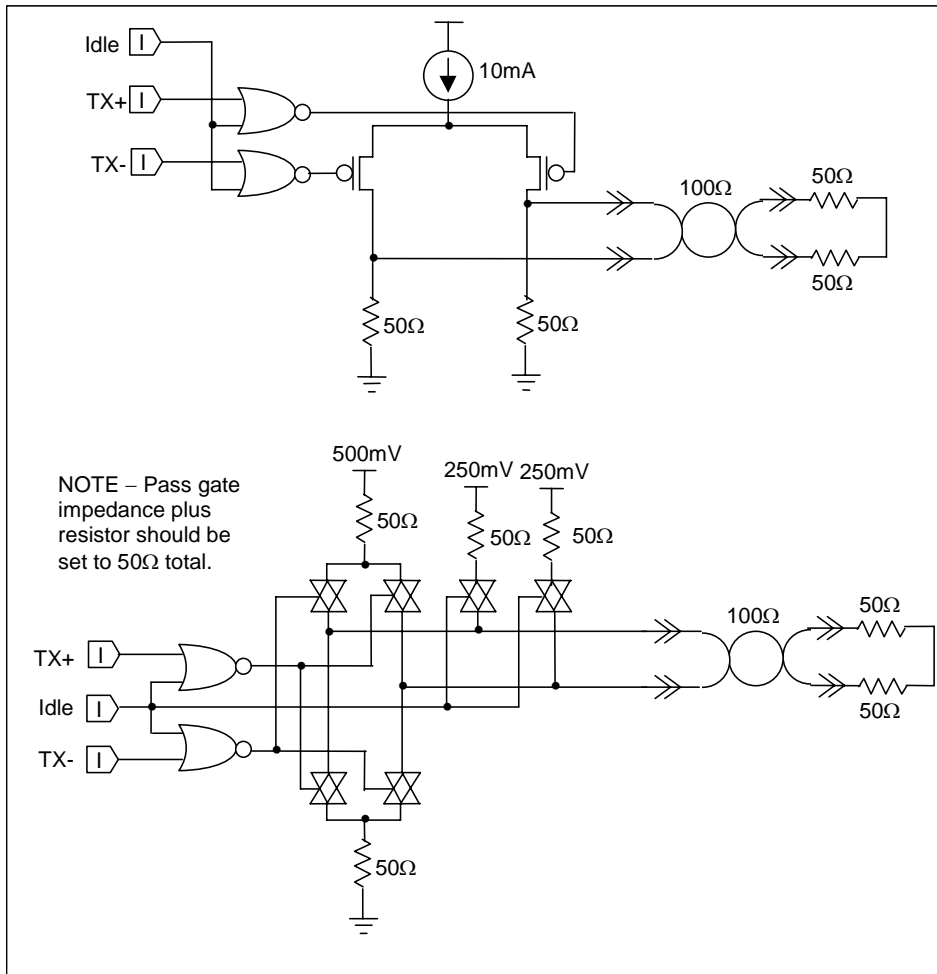
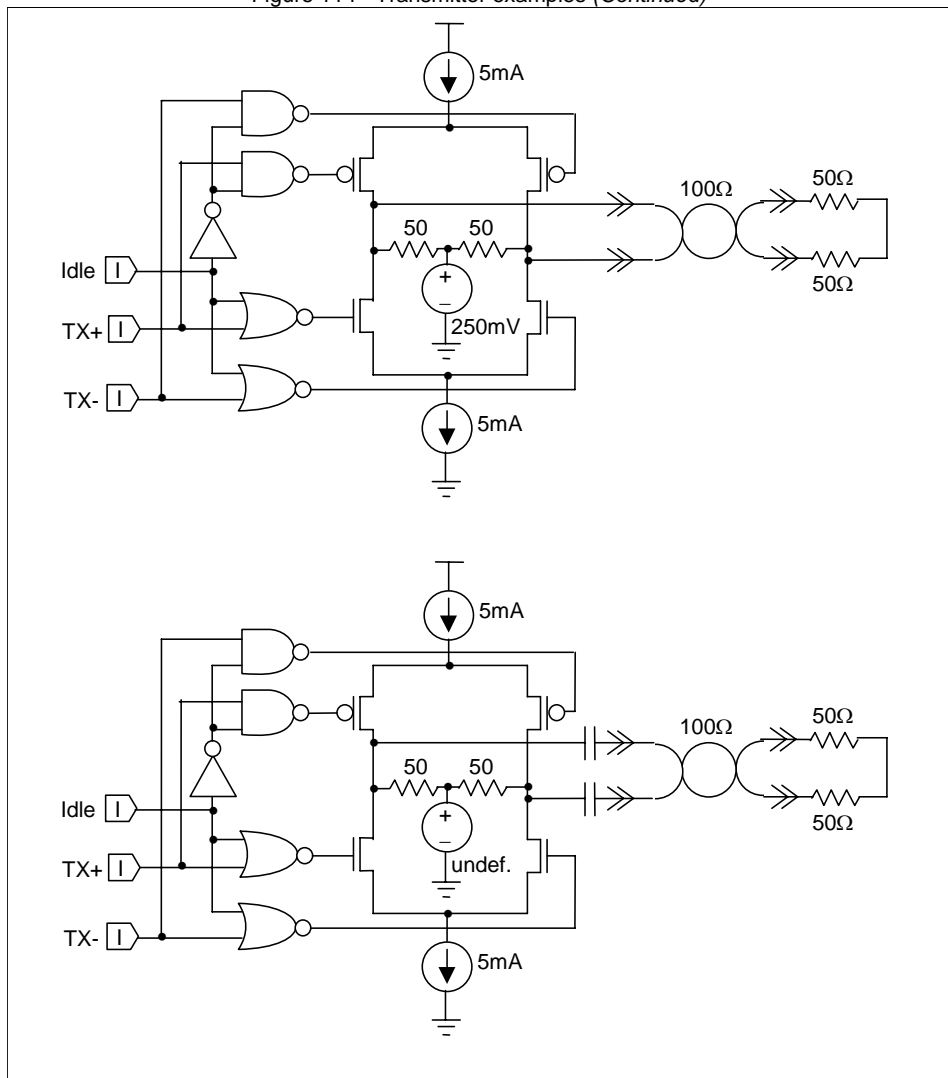


Figure 114 - Transmitter examples

(continued)

Figure 114 - Transmitter examples (Continued)



(concluded)

C.6 OOB signaling and squelch detector examples

This subclause is informative and represents one possible design example for detecting COMRESET/COMINIT and COMWAKE. Other design implementations are possible as long as they adhere to the requirements listed in this standard.

The output of the squelch detector is fed into four frequency comparators. When the period is within the window determined by the RC time constants for three consecutive cycles, the appropriate signal is asserted.

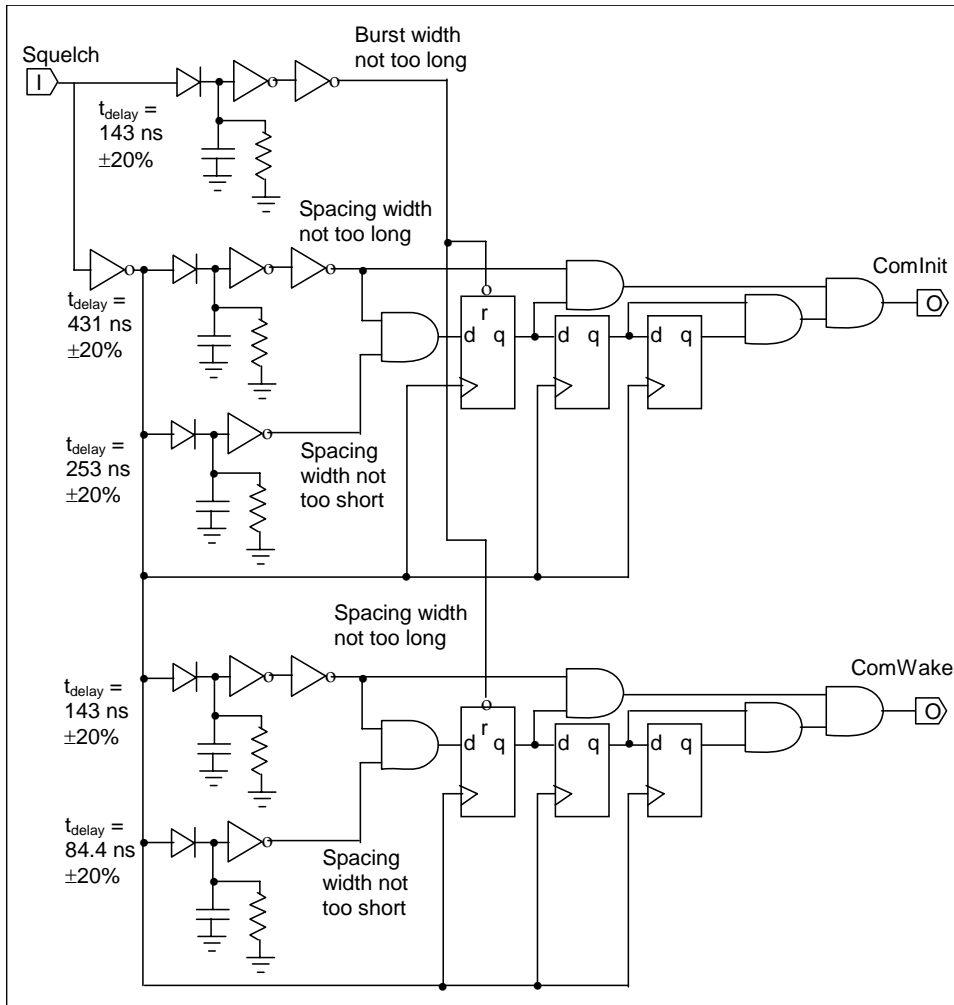


Figure 115 - OOB signaling detector

The Squelch detector example below makes use of a receiver with built-in hysteresis to filter out any signal not meeting the minimum amplitude. The squelch detector receiver should be true differential to ensure common-mode noise is rejected.

The full-swing output is fed into a pulse generator that charges up the capacitor through the diode. In the absence of signal, a resistor discharges the capacitor to ground. The circuit outputs a true signal when the capacitor voltage is below the turn-on threshold of the Schmitt trigger buffer - indicating insufficient signal level. This circuit shall be enabled in all power management states and should, therefore, be implemented with a small power budget.

Figure 116 is intended to show functionality (informative) only, and other solutions may be used to improve power consumption as long as they comply to the electrical specifications of Table 12, for the worst case noise environment (common-mode) conditions.

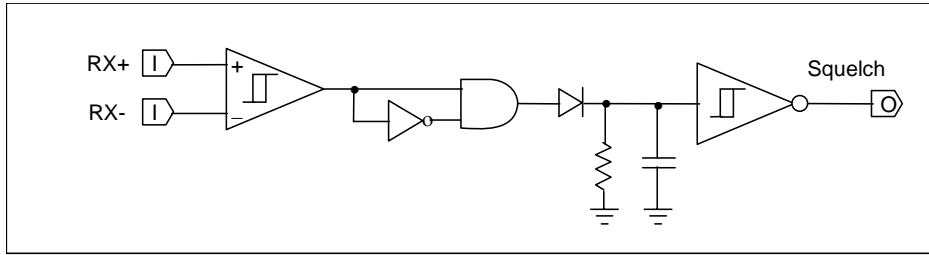


Figure 116 - Squelch detector

ANNEX D. COMMAND PROCESSING EXAMPLE (INFORMATIVE)

D.1 Non-data commands

When a command operation is requested by the application client, the host adapter enters a busy condition and transmits a Command FIS to the device.

When command actions are complete, the device transmits a Response FIS to set ending content of the HBA response registers.

D.1.1 DMA read by host from device

1. Prior to the command being issued to the device, the host driver software programs the host adapter's DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the "run" flag).
2. The host issues the command to the device.
3. In response to the command operation being requested, the host adapter enters a device busy condition and transmits a Command FIS to the device with the command as defined by ATA8-ACS.
4. When the device has processed the command and has data to transmit to the host, it transmits the read data to the host in the form of one or more Data FISes. This transfer proceeds in response to flow control signals/readiness.
5. The host adapter recognizes that the incoming frame is a Data FIS and the DMA controller is programmed, and directs the incoming data to the host adapter's DMA controller which forwards the incoming data to the appropriate host memory locations.
6. Upon completion of the transfer, the device transmits a Response FIS to indicate ending status for the command, clearing the BSY bit in the Response FIS Status field, and if the Interrupt Pending bit is set and an interrupt is asserted to the host.

In some error conditions there may be no Data FIS transmitted prior to the Response FIS being transmitted with the status information.

D.1.2 DMA write by host to device

1. Prior to the command being issued to the device, the application client programs the host adapter's DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the "run" flag). As a result the DMA controller becomes armed but remains paused pending a signal from the device to proceed with the data transfer.
2. The host issues the command.
3. In response to a command operation being requested, the host adapter enters a device busy condition and transmits a Command FIS to the device with the command as defined in ATA8-ACS.
4. When the device is ready to receive the data from the host, the device transmits a DMA Activate FIS to the host, which activates the armed DMA controller. The host adapter transmits the write data to the device in the form of one or more Data FISes. If more than one Data FIS is required to complete the overall data transfer request, a DMA Activate FIS will be sent prior to each of the subsequent Data FISes. The amount of data transmitted to the device is determined by the host adapter.
5. Upon completion of the data transfers, the device transmits a Response FIS to indicate ending status for the command, clearing the BSY bit in the Response FIS Status field, and if the Interrupt Pending bit is set and an interrupt is asserted to the host.

In some error conditions the device may signal an ending status by transmitting a Response FIS to the host without having transmitted a DMA Activate FIS.

D.1.3 PIO data read from the device

1. The host issues a PIO read command to the device.
2. In response to the command operation being requested, the host adapter enters a device busy condition and transmits a Command FIS to the device with the command as defined by ATA8-ACS.
3. When the device has processed the command and has data to transmit to the host, it first transmits a PIO Setup FIS to the host. The PIO Setup FIS has information needed to perform the transfer such as transfer count, beginning status, and intermediate or ending status used after the data has been transferred.
4. The device follows the PIO Setup FIS with a Data - device to host FIS. Upon receiving the Data FIS while holding the PIO Setup FIS contents, the host adapter transfers the register contents from the PIO Setup FIS into the HBA response registers including the initial status value, resulting in DRQ getting set and BSY getting cleared in the HBA Status register. Also, if the Interrupt Pending bit is set, an interrupt is generated to the host. The HBA response registers are a vendor specific implementation and actual implementations may use other means of utilizing the PIO Setup FIS information to control the data transfer.
5. The host adapter receives the incoming data.
6. When the number of words read reaches the value indicated in the PIO Setup FIS, the host adapter transfers the ending status value from the earlier PIO Setup FIS E_Status field into the HBA Status register resulting in DRQ being cleared and the ending status reported.
7. If there are more data blocks to be transferred, the ending status from the E_Status field will indicate BSY, and the process will repeat from the device sending the PIO Setup FIS to the host.

D.1.4 PIO data write to the device

1. The host issues a PIO write command to the device.
2. In response to the command operation being requested, the host adapter enters a device busy condition and transmits a Command FIS to the device with the command as defined by ATA8-ACS.
3. When the device is ready to receive the PIO write data, it transmits a PIO Setup FIS to the host to indicate that the device is ready to receive PIO data and the number of words of data that are to be transferred.
4. In response to a PIO Setup FIS with the D bit indicating a write to the device, the host adapter transfers the beginning HBA Status field contents from the PIO Setup FIS into the HBA Status register, resulting in DRQ getting set, BSY cleared. Also, if the Interrupt Pending bit is set, an interrupt is generated to the host. The HBA response registers are a vendor specific implementation and actual implementations may use other means of utilizing the PIO Setup FIS information to control the data transfer.
5. The data is transmitted to the device as a Data - host to device FIS. The link flow control scheme handles data throttling to avoid underflow of the transmit data.
6. When the number of words indicated in the PIO Setup FIS have been transmitted, the host adapter transfers the final status value indicated in the PIO Setup FIS into the HBA Status register resulting in DRQ being cleared, and closes the frame with a CRC and EOF.
7. If there are more data blocks to be transferred, the ending status from the E_Status field will indicate BSY, and the process will repeat from the device sending the PIO Setup FIS to the host.
8. When the number of sectors indicated in the Count field have been transferred, the device shall send a Response FIS with the Interrupt Pending bit set and BSY cleared to zero status.
9. In the case of a write error, the device may, on any block boundary include error status and Interrupt Pending bit set in the PIO setup FIS, and there is no need to send the Response FIS.

D.1.5 READ DMA QUEUED example

1. Prior to the command being issued to the device, the host programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller.
2. The host issues the command to the device.
3. In response to a command operation being requested, the host adapter enters a device busy condition and transmits a Command FIS to the device with the command as defined by ATA8-ACS.
4. When the device has queued the command and wishes to release the bus, it transmits a Response FIS to the host resulting in the BSY bit being cleared and the REL bit being set in the HBA Status register.
5. When the device has data to transmit to the host to complete the transfer for the queued command, it transmits a Set Device Bits FIS to the host resulting in the SERV bit being set in the HBA Status register. If no other command is active (i.e. BSY set to one), then the Interrupt Pending bit is set to one. The HBA response registers are a vendor specific implementation and actual implementations may use other means of utilizing the Set Device Bits FIS information to control the data transfer.
6. In response to the service request, the host issues a SERVICE command to the device by writing the HBA Command and Control registers, resulting in the BSY bit getting set and a Command FIS being transmitted to the device.
7. In response to the SERVICE request, the device transmits a Response FIS to the host conveying the TAG value to the host and clearing the BSY bit and settings the DRQ bit.
8. When the DRQ bit is set, the host uses the TAG value from the HBA response registers and restores the DMA controller context appropriate for the command that is completing.
9. The device transmits the read data to the host in the form of one or more Data FISes. This transfer proceeds in response to flow control signals/readiness.
10. The host adapter recognizes that the incoming packet is a Data FIS and the DMA controller is programmed, and directs the incoming data to the host adapter's DMA controller that forwards the incoming data to the appropriate host memory locations.
11. Upon completion of the transfer, the device transmits a Response FIS to indicate ending status for the command, clearing the BSY bit in the HBA Status register, and if the Interrupt Pending bit is set an interrupt is asserted to the host.

D.1.6 WRITE DMA QUEUED example

1. Prior to the command being issued to the device, the host programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the "run" flag).
2. The host issues the command to the device.
3. In response to a command operation being requested, the host adapter enters a device busy condition and transmits a Command FIS to the device with the command as defined by ATA8-ACS.
4. When the device has queued the command and wishes to release the bus, it transmits a Response FIS to the host resulting in the BSY bit being cleared and the REL bit being set in the HBA Status register.
5. When the device has data to transmit to the host to complete the transfer for the queued command, it transmits a Set Device Bits FIS to the host resulting in the SERV bit being set in the HBA Status register. If no other command is active (i.e. BSY set to one), then an interrupt is also generated.
6. In response to the service request, the host issues a SERVICE command to the device by writing the HBA Command Block and HBA Control Block, resulting in the BSY bit getting set and a Response FIS being transmitted to the device. The HBA response registers are a vendor specific implementation and actual implementations may use other means of utilizing the PIO Setup FIS information to control the data transfer.
7. In response to the SERVICE request, the device transmits a Response FIS to the host conveying the TAG value to the host and clearing the BSY bit and setting the DRQ bit.
8. When the DRQ bit is set, the host uses the TAG value from the HBA Command Block and restores the DMA controller context appropriate for the command that is completing.
9. When the device is ready to receive the data from the host, the device transmits a DMA Activate FIS to the host, which activates the armed DMA controller. The DMA controller transmits the write data to the device in the form of one or more Data - host to device FISes. If more than one Data FIS is required to complete the overall data transfer request, a DMA Activate FIS shall be sent prior to each of the subsequent Data FISes. The amount of data transmitted to the device is determined by the transfer count programmed into the host's DMA controller by the host driver software during the command setup phase.
10. Upon completion of the transfer, the device transmits a Response FIS to indicate ending status for the command, clearing the BSY bit in the HBA Status register, and if the Interrupt Pending bit is set an interrupt is asserted to the host.

D.1.7 ATAPI PACKET commands with PIO data-in

1. The host issues a PACKET command to the device.
2. In response to the command operation being requested, the host adapter enters a device busy condition and transmits a Command FIS to the device with the command as defined by ATA8-ACS.
3. When the device is ready to receive the ATAPI command packet, it transmits a PIO Setup - device to host FIS to the host to indicate that the device is ready to receive PIO data and the number of words of data that are to be transferred.
4. The host bus adapter transfers the beginning HBA Status contents from the PIO Setup FIS into the HBA Status register, resulting in BSY getting negated and DRQ getting asserted. The HBA response registers are a vendor specific implementation and actual implementations may use other means of utilizing the PIO Setup FIS information to control the data transfer.
5. The data is transmitted to the device as a Data - host to device FIS. The link flow control scheme handles data throttling to avoid underflow of the transmit data.
6. When the number of words indicated in the PIO Setup FIS have been transmitted, the host adapter transfers the final status value indicated in the PIO setup FIS into the HBA Status register resulting in DRQ being cleared and BSY being set, and closes the frame with a CRC and EOF. This completes the transmission of the command packet to the device.
7. When the device has processed the command and has data to transmit to the host, it first transmits a PIO Setup - device to host FIS to the host. Upon receiving the PIO Setup - device to host FIS, the host adapter holds the FIS contents in a temporary holding buffer.
8. The device follows the PIO Setup - device to host FIS with a Data - device to host FIS. Upon receiving the Data FIS while holding the PIO Setup FIS context, the host adapter transfers the contents from the PIO Setup FIS into the HBA response registers including the initial status value, resulting in DRQ getting set and BSY getting cleared in the HBA Status register. Also, if the Interrupt Pending bit is set, an interrupt is generated to the host.
9. When the number of words received in the Data FIS reaches the value indicated in the PIO Setup FIS and the host FIFO is empty, the host adapter transfers the ending status from the E_Status field value from the earlier PIO Setup into the HBA Status register resulting in DRQ being cleared and BSY being set.
10. The device transmits final ending status by sending a Response FIS with BSY cleared to zero and the command completion status for the command and the Interrupt Pending bit set.
11. The host adapter detects a Response FIS that contains BSY cleared to zero and command completion status for the command and the Interrupt Pending bit set and places the FIS content into the HBA response registers to complete the command.

D.1.8 ATAPI PACKET commands with PIO data out

1. The host issues a PACKET command to the device.
2. In response to the command register operation being requested, the host adapter sets the BSY bit in the HBA Status register and transmits a Command FIS to the device. with the command as defined by ATA8-ACS.
3. When the device is ready to receive the ATAPI command packet, it transmits a PIO Setup - device to host FIS to the host to indicate that the device is ready to receive data and the number of words of data that are to be transferred.
4. The host adapter transfers the beginning HBA Status register contents from the PIO Setup - device to host FIS into the HBA Status register, resulting in BSY getting negated and DRQ getting asserted. The HBA response registers are a vendor specific implementation and actual implementations may use other means of utilizing the PIO Setup FIS information to control the data transfer.
5. When the number of words indicated in the PIO Setup FIS have been transmitted, the host adapter transfers the final status value indicated in the PIO Setup FIS into the HBA Status register resulting in DRQ being cleared and BSY being set, and closes the frame with a CRC and EOF. This completes the transmission of the command packet to the device.
6. When the device has processed the command and is ready to receive the PIO write data, it transmits a PIO Setup - device to host FIS to the host to indicate that the device is ready to receive data and the number of words of data that are to be transferred.
7. In response to a PIO Setup FIS with the D bit indicating a write to the device, the host adapter transfers the beginning Status field contents from the PIO Setup FIS into the HBA Status register, resulting in DRQ getting set. Also, if the Interrupt Pending bit is set, an interrupt is generated to the host.
8. When the number of words indicated in the PIO Setup FIS have been transmitted, the host adapter transfers the final status value indicated in the PIO Setup FIS into the HBA Status register resulting in DRQ being cleared, BSY being set, and closes the frame with a CRC and EOF.
9. The device transmits command completion status by sending a Response FIS with BSY cleared to zero and the ending status for the command and the Interrupt Pending bit set.
10. The host detects an incoming Response FIS that contains BSY cleared to zero and command completion status for the command and the Interrupt Pending bit set and places the FIS content into the HBA response registers to complete the command.

D.1.9 ATAPI PACKET commands with DMA data-in

1. The host issues a PACKET command to the device.
2. In response to the command operation being requested, the host adapter enters a device busy condition and transmits a Command FIS to the device with the command as defined by ATA8-ACS.
3. When the device is ready to receive the ATAPI command packet, it transmits a PIO Setup - device to host FIS to the host to indicate that the device is ready to receive data and the number of words of data that are to be transferred.
4. The host adapter transfers the beginning Status field contents from the PIO Setup FIS into the HBA Status register, resulting in BSY getting negated and DRQ getting asserted.
5. As a result of BSY getting cleared and DRQ being set in the HBA Status register, the host driver software writes the command packet to the HBA Data register. The HBA Data register is a vendor specific implementation and actual implementations may use other means of utilizing the PIO Setup FIS information to control the data transfer.
6. When the number of words indicated in the PIO Setup FIS have been transmitted, the host adapter transfers the final status value indicated in the PIO Setup FIS into the HBA Status register resulting in DRQ being cleared and BSY being set, and closes the frame with a CRC and EOF. This completes the transmission of the command packet to the device.
7. When the device has processed the command and is ready, it transmits the read data to the host in the form of a single Data FIS. This transfer proceeds in response to flow control signals/readiness.
8. The host adapter recognizes that the incoming packet is a Data FIS and the DMA controller is programmed, and directs the incoming data to the host adapter's DMA controller that forwards the incoming data to the appropriate host memory locations.
9. Upon completion of the transfer, the device transmits a Response FIS to indicate command completion status, clearing the BSY bit in the HBA Status register, and if the Interrupt Pending bit is set, an interrupt is asserted to the host.

D.1.10 ATAPI PACKET commands with DMA data-out

1. The host issues a PACKET command to the device.
2. In response to the command operation being requested, the host adapter enters a device busy condition and transmits a Command FIS to the device with the command as defined by ATA8-ACS.
3. When the device is ready to receive the ATAPI command packet, it transmits a PIO Setup - device to host FIS to the host to indicate that the device is ready to receive data and the number of words of data that are to be transferred.
4. The host adapter transfers the beginning Status field contents from the PIO Setup FIS into the HBA Status register, resulting in BSY getting negated and DRQ getting asserted.
5. As a result of BSY getting cleared and DRQ being set in the HBA Status register, the host driver software writes the command packet to the HBA Data register. The HBA Data register is a vendor specific implementation and actual implementations may use other means of utilizing the PIO Setup FIS information to control the data transfer.
6. When the number of words indicated in the PIO Setup FIS have been transmitted, the host adapter transfers the final status value indicated in the PIO Setup FIS into the HBA Status register resulting in DRQ being cleared and BSY being set, and closes the frame with a CRC and EOF. This completes the transmission of the command packet to the device.
7. When the device is ready to receive the data from the host, the device transmits a DMA Activate FIS to the host, which activates the armed DMA controller. The DMA controller transmits the write data to the device in the form of one or more Data FISes. The transfer proceeds in response to flow control signals/readiness. The amount of data transmitted to the device is determined by the transfer count programmed into the host's DMA engine by the host driver software during the command setup phase.
8. Upon completion of the transfer, the device transmits a Response FIS to indicate ending status for the command, clearing the BSY bit in the HBA Status register, and if the Interrupt Pending bit is set an interrupt is asserted to the host.

D.1.11 First party DMA read of host memory by device

This subclause only outlines the basic first party DMA read transaction and does not outline command sequences that utilize this capability.

1. As a result of some condition and the device having adequate knowledge of host memory (presumably some prior state has been conveyed to the device), the device signals a DMA read request for a given address and given transfer count to the host by transmitting a First Party DMA Setup FIS.
2. Upon receiving the First Party DMA Setup FIS, the host adapter transfers the appropriate memory address, count value, and transfer direction into the host-side DMA controller and arms and activates the DMA controller (enables the "run" flag).
3. In response to being set up, armed, and activated, the DMA controller retrieves data from host memory and transmits it to the device in the form of one or more Data - host to device FISes. The transfer proceeds in response to flow control signals/readiness.

D.1.12 First party DMA write of host memory by device

This subclause only outlines the basic first party DMA write transaction and does not outline command sequences which utilize this capability.

1. As a result of some condition and the device having adequate knowledge of host memory (presumably some prior state has been conveyed to the device), the device signals a DMA write request for a given address and given transfer count to the host by transmitting a First Party DMA Setup FIS.
2. Upon receiving the First Party DMA Setup FIS, the host adapter transfers the appropriate memory address, count value, and transfer direction into the host-side DMA controller and arms the DMA controller (enables the "run" flag).
3. The device then transmits the data to the host in the form of one or more Data - device to host FISes. This DMA transfer proceeds in response to flow control signals/readiness.
4. The host adapter recognizes that the incoming packet is a Data FIS and directs the incoming data to the host adapter's DMA controller which forwards the incoming data to the appropriate host memory locations.

D.1.13 Odd word count considerations

This subclause outlines special considerations required to accommodate data transfers of an odd number of 16-bit word quantities. The considerations are separately outlined for each of the data transaction types. No accommodation in the serial transport is made for the transfer of an odd number of 8-bit byte quantities.

D.1.13.1 DMA read from device for odd word count

1. Prior to the command being issued to the device, the host driver software programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the "run" flag). The count for the DMA transfer is an odd number of word (16-bit) quantities.
2. When the device has processed the corresponding command and is ready to transmit the data to the host, it does so in the form of one or more Data FIS. Because the transfer count is odd, the last 32-bit dword transmitted to the host has the high order 16 bits padded with zeroes. The CRC value

transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.

3. The host adapter receives the incoming data and the DMA controller directs the received data from the receive FIFO to the appropriate host memory locations. The DMA controller has a transfer granularity of a 16-bit word (consistent with the 8038i specification). (editor's note: what is 8038i?)
4. Upon receiving the final 32-bit dword of receive data, the DMA controller transfers the first half (low order 16 bits) to the corresponding final memory location at which point the DMA engine's transfer count is exhausted. The DMA controller drops the high-order 16 bits of the final received dword since it represents data received beyond the end of the requested DMA transfer. The dropped 16 high order bits corresponds with the 16 bits of transmission pad inserted by the sender.

D.1.13.2 DMA write by host to device for odd word count

1. Prior to the command being issued to the device, the host driver software programs the host-side DMA controller with the memory address pointer(s) and the transfer direction, and arms the DMA controller (enables the "run" flag). The count for the DMA transfer is an odd number of word (16-bit) quantities.
2. When the device has processed the corresponding command and is ready to receive the data from the host, it signals readiness with a DMA Activate FIS.
3. Upon receiving the DMA Activate signal, the host transmits the data to the device in the form of one or more Data FIS. Because the transfer count is odd, the DMA controller completes its data transfer from host memory to the transmit FIFO after filling only the low order 16-bits of the last dword in the FIFO, leaving the upper 16 bits zeroed. This padded final dword is transmitted as the final symbol in the data frame. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
4. Having awareness of the command set and having decoded the current command, the device that receives the transmitted data has knowledge of the expected data transfer length. Upon receiving the data from the host, the device removes the 16-bit pad data in the upper 16 bits of the final 32-bit dword of received data.

D.1.13.3 PIO data read from the device

1. In response to decoding and processing a PIO read command with a transfer count for an odd number of 16-bit words, the device transmits the corresponding data to the host in the form of a single Data FIS. The device pads the upper 16-bits of the final 32-bit dword of the last transmitted FIS in order to close the FIS. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
2. Host driver software responsible for retrieving the PIO data is aware of the number of words of data it expects to retrieve from the Data register and performs a IN operation operation for an odd number of repetitions.
3. Upon exhaustion of the IN operation operation by the host driver software, the receive FIFO that interfaces with the data register has one 16-bit word of received data remaining in it that corresponds to the pad that the device included at the end of the transmitted frame. This remaining word of data left in the data register FIFO is flushed upon the next write of the HBA Command register or upon the receipt of the next Data FIS from the device.

D.1.13.4 PIO data write to the device

1. In response to decoding and processing a PIO write command with a transfer count for an odd number of 16-bit words, the device transmits a PIO Setup FIS to the host indicating it is ready to receive the PIO data and indicating the transfer count. The conveyed transfer count is for an odd number of 16-bit word quantities.
2. Host driver software responsible for transmitting the PIO data is aware of the number of words of data it will write to the Data register and performs a OUT operation for an odd number of repetitions.
3. After the final write by the software driver to the Data register, the transfer count indicated in the PIO Setup packet is exhausted, which signals the host adapter to close the FIS. Since the transfer count was odd, the upper 16 bits of the final 32-bit dword of data to transmit remains zeroed (pad) and the host adapter closes the FIS after transmitting this final padded dword. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
4. Having awareness of the command set and having decoded the current command, the device that receives the transmitted data has knowledge of the expected data transfer length. Upon receiving the data from the host, the device removes the 16-bit pad data in the upper 16 bits of the final 32-bit dword of received data.

D.1.13.5 First party DMA read of host memory by device

1. When the device wishes to initiate a first party DMA read of host memory for an odd word count, it constructs a First Party DMA Setup FIS that includes the base address and the transfer count (for an odd number of words).
2. Upon receiving the First Party DMA Setup FIS, the host adapter transfers the starting address and (odd) transfer count to the DMA controller and activates the DMA controller to initiate a transfer to the device of the requested data.
3. The host transmits the data to the device in the form of one or more Data FIS. Because the transfer count is odd, the DMA controller completes its data transfer from host memory to the transmit FIFO after filling only the low order 16-bits of the last dword in the FIFO, leaving the upper 16 bits zeroed. This padded final dword is transmitted as the final symbol in the data frame. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
4. Having awareness of the amount of data requested, the device that receives the transmitted data has knowledge of the expected data transfer length. Upon receiving the data from the host, the device removes the 16-bit pad data in the upper 16 bits of the final 32-bit dword of received data.

D.1.13.6 First party DMA write of host memory by device

1. When the device wishes to initiate a first party DMA write to host memory for an odd word count, it constructs a First Party DMA Setup FIS that includes the base address and transfer count (for an odd number of words).
2. Upon receiving the First Party DMA Setup FIS, the host adapter transfers the starting address and (odd) transfer count to the DMA controller and activates the DMA controller.
3. The device transmits the data to the host in the form of one or more Data FIS. Because the transfer count is odd, the last 32-bit dword transmitted to the host has the high order 16 bits padded with zeroes. The CRC value transmitted at the end of the FIS is computed over the entire FIS including any pad bytes in the final transmitted symbol.
4. The host adapter receives the incoming data and the DMA controller directs the received data from the receive FIFO to the appropriate host memory locations. The DMA controller has a transfer granularity of a 16-bit word.
5. Upon receiving the final 32-bit dword of receive data, the DMA controller transfers the first half (low order 16 bits) to the corresponding final memory location at which point the DMA controller's transfer

count is exhausted. The DMA controller drops the high-order 16 bits of the final received dword since it represents data received beyond the end of the requested DMA transfer. The dropped 16 high order bits corresponds with the 16 bits of transmission pad inserted by the sender.

ANNEX E. PARALLEL ATA EMULATION (INFORMATIVE)

E.1 Parallel ATA Emulation (Optional)

Emulation of Parallel Transports of ATA device behavior as perceived by the host BIOS or software driver, is a cooperative effort between the device and a serial transport 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 contains intermediate holding registers (called Shadows in prior version of this standard) of ATA Command and Control block registers, referred to as the HBA Command Block and HBA 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 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 Response FISes, and it is the responsibility of the host adapter to gate transmission of Response 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 HBA Status register and HBA Alternate Status Register and shall place the value 0xFFh in all the other HBA Command Block registers (0xFFFFh in the Data register). In this state the host bus adapter shall not accept writes to the HBA Command Block and HBA Control Block. When the host Phy detects presence of an attached device, see 10.2.2.1 and 9.2.1, the host bus adapter shall set bit 7 in the HBA Status register yielding the value 0xFFh or 0x80h, and the host bus adapter shall allow writes to the HBA Command Block and HBA 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 HBA Status register. The serial transport time limit of 10 ms is different than the Parallel Transport of ATA. (See ATA8-PT). When the attached device establishes communication with the host bus adapter, it shall send a Response FIS to the host, resulting in the HBA Command Block and HBA 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.

E.1.1 Software reset

Issuing a software reset is performed by toggling the SRST bit in the HBA Control register. The toggle period is no shorter than a minimum timeout (See ATA8-ACS). As a result of the SRST bit changing in the HBA Device register, host adapters shall issue at least two Control FISes to the device (one with the SRST bit set and a subsequent one with the SRST bit cleared). See subclause 7.5.2 for a detailed definition of the Control FIS. Although host software is required to toggle the SRST bit no faster than specified in Parallel Transports of ATA, serial devices shall not rely on the inter-arrival time of received Control FISes also meeting this timing. Because of flow control, frame handshaking, and other protocol interlocks, serial devices may receive the resulting Control FISes back-to-back.

Due to flow control, protocol interlocks, power management state, or other transmission latencies, the subsequent Control frame transmission clearing the SRST bit during a software reset may be triggered prior to the previous Control 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 Transport (ATA8-PT), even if frame transmission latencies result in the first Control FIS transmission taking longer than specified in the

Parallel Transport. Host adapters are required to ensure transmission of the two resulting Control FISes to the device regardless of the transmission latency of each individual FIS.

E.1.2 Device 0-only emulation

A serial implementation of an ATA host adapter behaves the same as if a Parallel Transport 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 Transport 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 Transport. This includes not setting the BSY bit in the HBA Status register when Device 1 is selected, as described in the Parallel Transport. When Device 0 is selected, the host adapter shall execute the serial implementation protocols for managing the HBA Command Block and HBA Control Block contents as defined in subclause 8.

When Device 0 is selected and the HBA Command register is written in the HBA Command Block, the host adapter sets the BSY bit in its HBA Status register. The host adapter then transmits a FIS to the device containing the new HBA command or control structure contents as defined by ATA8-ACS. When the HBA Control register is written in the HBA Control Block with a change of state of the SRST bit, the host adapter sets the BSY bit in its HBA Status register and transmits a Control FIS to the device containing the new register contents. Transmission of register contents when the HBA 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 HBA 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 HBA Status register to one when a hard reset (COMRESET) is requested or the SRST bit is set to one in the HBA Control register, see Figure 99.

The device updates the contents of the host adapter HBA Command Block and HBA Control Block by transmitting a Response FIS 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 HBA Status register for PIO transfers.

Finally the host adapter provides an interrupt pending flag (Interrupt Pending bit) in the host adapter. This flag is set by the host adapter when the device sends a Response FIS including the request to set the interrupt to the host. The host adapter asserts the interrupt to the host processor any time the Interrupt Pending bit is set, the DEV bit is cleared to zero in the HBA Device register, and the nEN bit in the HBA Control register is cleared to zero. The host adapter clears the Interrupt Pending any time a COMRESET is requested, the SRST bit in the HBA Control register is set to one, a command operation is requested and DEV is cleared to zero, or the HBA Status register is read and DEV is cleared to zero. This allows the emulation of the host interrupt and its proper timing.

E.1.3 Device 0/Device 1 emulation (optional)

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 HBA Command Block and HBA 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 HBA Command Block and HBA Control Block (one set for each of the two devices) based on the value of the DEV bit in the HBA Device register. Based on the value of the DEV bit, the host adapter shall direct accesses to the HBA Command Block and HBA Control Block to the appropriate set of HBA Command Block and HBA 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 Transport.

When device 1 is selected and device 0 is responding for device 1 (See table 44 in vol 2), a host adapter with Device 0/Device 1 emulation generates the non-packet device response for both packet and non-packet devices.

E.1.3.1 Software reset

Host adapters that emulate Device 0/Device 1 behavior shall emulate Parallel Transport 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. Control Register writes, that have the SRST bit set to one, shall result in the associated HBA 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 Control FIS being transmitted to each attached device. Similarly, the subsequent write to the HBA Control register that clears the SRST bit shall result in a Control 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 fields for the attached devices, in accordance with the Parallel Transport, to produce the HBA Error and HBA Status register values visible to host software.

E.1.3.2 EXECUTE DEVICE DIAGNOSTICS

Host adapters that emulate Device 0/Device 1 behavior shall emulate the Parallel Transport behavior for EXECUTE DEVICE DIAGNOSTICS (See ATA8-ACS). 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 HBA Command Block and HBA Control Block being written for each port to which a device is attached. A Command 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 fields for the attached devices, in accordance with the Parallel Transport to produce the HBA Error and HBA Status register values visible to host software.

E.1.3.3 Restrictions and limitations

Superset capabilities that are unique to the serial transport and not supported by the Parallel Transport 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.

E.2 Host Command Layer state diagram for Parallel ATA Emulation

E.3 Overview

Figure 117 defines the protocol of the host adapter to emulate Device 0 only operation of the Parallel Transport of ATA devices as seen from the host BIOS or software driver. The Interrupt Pending flag (IPF) is an internal state bit in the host adapter that reflects whether or not the device has an Interrupt Pending to the host. See E.1 for additional information on parallel ATA emulation. Registers referred to in Figure 117 and

following descriptions refer to the host adapter's implementation of the parallel ATA Command and Control Registers as defined in the Parallel Transport (ATA8-PT). When emulating parallel ATA, these registers are also referred to as Shadow Command and Shadow Control registers.

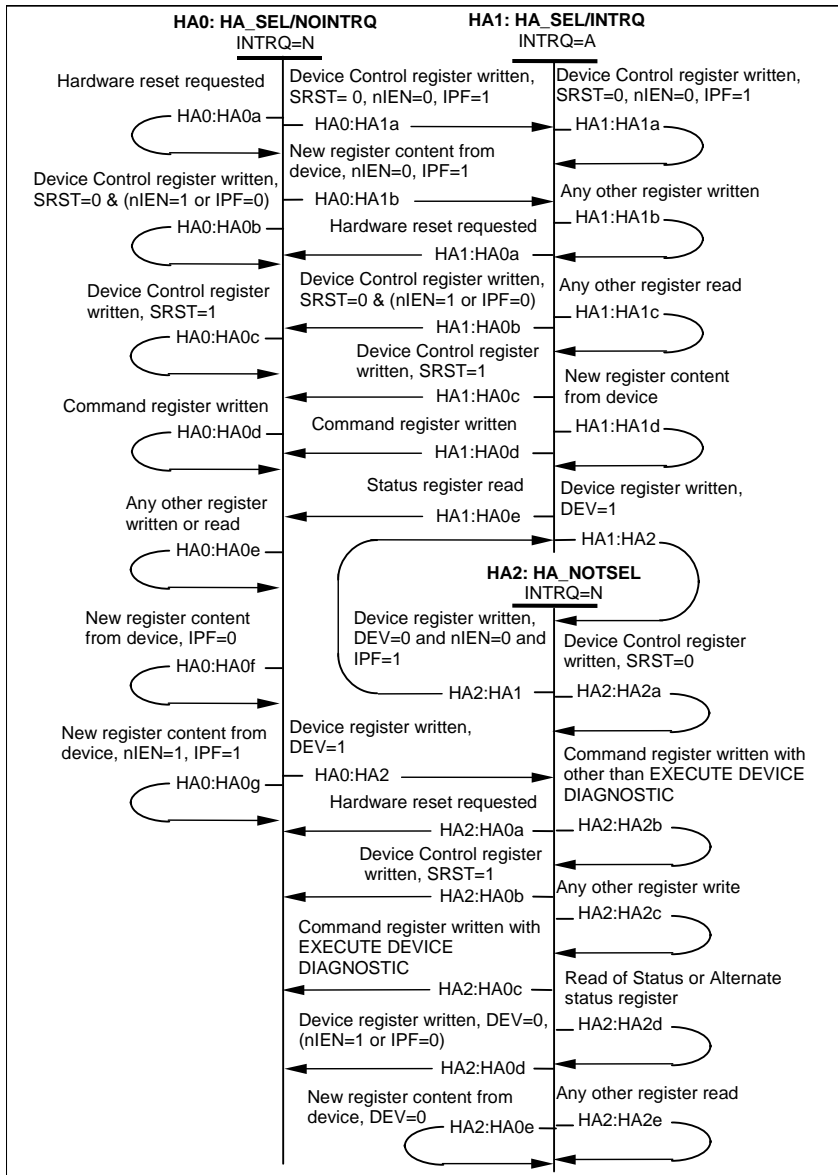


Figure 117 – Host adapter state diagram (States HA0-HA2)

HA0: HA_SEL/NOINTRQ state: This state is entered when Device 0 is selected and either IPF is cleared to zero or nIEN is set to one.

When in this state, the interrupt signal to the host shall not be asserted.

Transition HA0:HA0a: When a COMRESET is requested by the host, the host adapter shall set BSY to one in the HBA Status register, clear IPF to zero, notify the link layer to have a bus COMRESET asserted, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:HA0b Control register written by the host with SRST=0 and (nIEN=1 or IPF=0): When the HBA Control register is written by the host with SRST cleared to zero and either nIEN set to one or IPF is cleared to zero, the host adapter shall notify the transport layer to send a Register host to device FIS with the C bit cleared to 0 with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA0: HA_SEL/NOINTRQ state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA0:HA1a Control register written by the host with SRST=0 and (nIEN=0 and IPF=1): When HBA Control register is written by the host with SRST cleared to zero, nIEN cleared to zero, and IPF is set to one, the host adapter shall notify the transport layer to send a Register host to device FIS with the C bit cleared to 0 with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA1: HA_SEL/INTRQ state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA0:HA0c: When HBA Control register is written by the host with SRST set to one, the host adapter shall set BSY to one in the HBA Status register, clear IPF to zero, notify the transport layer to send a Register host to device FIS with the C bit cleared to 0 with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA0: HA_SEL/NOINTRQ state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA0:HA0d: When the Command register is written by the host, the host adapter shall set BSY to one in the HBA Status register, clear IPF to zero, notify the transport layer to send a Register host to device FIS with the C bit set to 1, with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:HA2: When the HBA Device register is written by the host with DEV set to one, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA0:HA0e: When any register is written by the host other than the HBA Control, Command, or Device Register, the host adapter shall make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:HA0f: When the transport layer indicates new register content from the device with no set IPF flag, the host adapter shall place the new register content into the HBA Command Block and HBA Control Block and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:HA0g: When the transport layer indicates new register content from the device with set IPF flag and nIEN is set to one, the host adapter shall set IPF to one, place the new register content into the HBA Command Block and HBA Control Block, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:HA1b0: When the transport layer indicates new register content from the device with set IPF flag and nIEN is cleared to zero, the host adapter shall set IPF to one, place the new register content into the HBA Command Block and HBA Control Block, and make a transition to the HA1: HA_SEL/INTRQ state.

HA1: HA_SEL/INTRQ state: This state is when DEVICE 0 is selected, nIEN is cleared to zero, and IPF is set to one.

When in this state, the interrupt signal to the host shall be asserted.

Transition HA1:HA0a: When a COMRESET is requested by the host, the host adapter shall set BSY to one in the HBA Status register, clear IPF to zero, notify the link layer to have a bus COMRESET asserted, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:HA0b Control register written by the host with SRST=0 and (nIEN=1 or IPF=0): When the HBA Control register is written by the host with SRST cleared to zero, with nIEN set to one or IPF cleared to zero, the host adapter shall notify the transport layer to send a Register host to device FIS with the C bit cleared to 0 with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA0: HA_SEL/NOINTRQ state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA1:HA1a Control register written by the host with SRST=0 and (nIEN=0 and IPF=1): When the HBA Control register is written by the host with SRST cleared to zero with nIEN cleared to zero and IPF set to one, the host adapter shall notify the transport layer to send a Register host to device FIS with the C bit cleared to 0 with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA1: HA_SEL/INTRQ state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA1:HA0c: When the HBA Control register is written by the host with SRST set to one, the host adapter shall set BSY to one in the HBA Status register, clear IPF to zero, notify the transport layer to send a Register host to device FIS with the C bit cleared to 0 with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA0: HA_SEL/NOINTRQ state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA1:HA0d: When the Command register is written by the host, the host adapter shall set BSY to one in the HBA Status register, clear IPF to zero, notify the transport layer to send a Register host to device FIS with the C bit set to 1, with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:HA2: When the HBA Device register is written by the host with DEV set to one, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA1:HA0e: When the Status register is read by the host, the host adapter shall clear IPF to zero and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:HA1b: When any register is written by the host other than those described above, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA1:HA1c: When any register is read by the host other than that described above, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA1:HA1d: When the transport layer indicates new register content from the device, the host adapter shall place the new register content into the HBA Command Block and HBA Control Block and make a transition to the HA1: HA_SEL/INTRQ state.

HA2: HA_NOTSEL state: This state is entered when DEVICE 1 is selected.

When in this state, the interrupt signal to the host shall not be asserted.

Transition HA2:HA0a: When a COMRESET is requested by the host, the host adapter shall set BSY to one in the HBA Status register, clear DEV to zero in the HBA Device register, clear IPF to zero, notify the link layer to have a bus COMRESET asserted, and make a transition to the HA0: SEL/NOINTRQ state.

Transition HA2:HA2a: When the HBA Control register is written by the host with SRST cleared to zero, the host adapter shall notify the transport layer to send a Register host to device FIS with the C bit cleared to 0 with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA2: HA_NOTSEL state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA2:HA0b: When the HBA Control register is written by the host with SRST set to one, the host adapter shall set BSY to one in the HBA Status register, clear DEV to zero in the HBA Device register, clear IPF to zero, notify the transport layer to send a Register host to device FIS with the C bit cleared to 0 with

the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA0: HA_SEL/NOINTRQ state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA2:HA2b: When the Command register is written by the host with any command code other than EXECUTE DEVICE DIAGNOSTIC, the host adapter shall not set BSY in the HBA Status register and shall make a transition to the HA2: HA_NOTSEL state.

Transition HA2:HA0c: When the Command register is written by the host with the EXECUTE DEVICE DIAGNOSTIC command code, the host adapter shall set BSY to one in the HBA Status register, clear DEV to zero in the HBA Device register, clear IPF to zero, notify the transport layer to send a Register host to device FIS with the C bit set to 1, with the current content of the HBA Command Block and HBA Control Block, and make a transition to the HA0: HA_SEL/INTRQ state.

Transition HA2:HA0d Device register written by the host with DEV = 0 and (nIEN=1 or IPF=0): When the HBA Device register is written by the host with DEV cleared to zero and either nIEN is set to one or IPF is cleared to zero, the host adapter shall make a transition to the HA0: HA_SEL/NOINTRQ state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA2:HA1 Device register written by the host with DEV = 0 and (nIEN=0 and IPF=1): When the HBA Device register is written by the host with DEV cleared to zero, nIEN cleared to zero, and IPF is set to one, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state. If the state of the SRST is unchanged from the prior state, the sending of a Register HD FIS with C=0 is optional.

Transition HA2:HA2c: When any register is written by the host other than the Command Register, Device Register or the Control register, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA2:HA2d: When the Status or Alternate Status register is read by the host, the host shall return register content 00h to the host and make a transition to the HA2: HA_NOTSEL state.

Transition HA2:HA2e: When any register is read by the host other than the Status or Alternate Status register, the host adapter shall return the current HBA Command Block and HBA Control Block content to the host and make a transition to the HA2: HA_NOTSEL state.

Transition HA2:HA0e: When the transport layer indicates new register content from the device, the host adapter shall place the new register content into the HBA Command Block and HBA Control Block and make a transition to the HA2: HA_NOTSEL state.

E.4 Device Emulation of nIEN with Interrupt Pending (Informative)

The informative behavior described in this subclause describes implementation considerations for designs that support parallel emulation.

The serial transport defines the I bit in Register device to host FISes as the Interrupt pending state of the device, and it is not modified by the state of nIEN in received host to device FISes. In the serial transport devices ignore the nIEN bit in received host to device FISes and always perform as if nIEN is cleared to zero. See subclause 7.5.1 and ATA8-ACS.

Prior to the development of this standard, some devices used the nIEN bit of the Register host to device FIS as a pre-condition to the setting the Interrupt Pending flag (I-bit) of the Register device to host, and Set Device Bits FISes.

The purpose of using nIEN to enable the I-bit was to emulate the operation of the Parallel Transport of ATA (See Subclause E.1). In the Parallel Transport, when nIEN is cleared to zero, the driver is enabled for the INTRQ line to the host. When nIEN is set to one, the INTRQ line is put into the high impedance state by the

device. This function is typically used in devices that support Device 0 and Device 1 operation (See E.1.3), and it is also required for Overlapped operation (See Volume 1).

If parallel emulation is supported with a serial transport, the implementation of Device 0/Device 1 emulation is performed exclusively by the Host (See E.1.3).

Since operation prior to this standard was not uniformly implemented, the system designer should be aware of the limitations of those implementations. In most cases, the prior implementations are compatible with this standard. Some commands, however, may not operate in the same manner, or may require new device drivers for compatible operation. One example of this is when Overlapped/Queued operations are implemented with host-parallel to device-serial bridges. With proper management of the nIEN and INTRQ bit to the parallel host, it is possible to emulate the parallel operation.

One serious side effect of device emulation of nIEN is the possibility of lost interrupts. In the Parallel Transport, a host can disable interrupts, and upon re-enabling interrupts (by clearing nIEN to zero) see the INTRQ line again asserted. If a serial device is performing I-bit masking based on the state of nIEN, a Register device to host FIS may be received with I=0 (since it is masked by nIEN). The device, however, may have an Interrupt Pending at that time. When the host writes the Control Register with nIEN cleared to zero, it will not see the pending interrupt reflected by the assertion of INTRQ as in the parallel case. There is no way for the device to "re-send" the I=1 condition to the host. In this instance, the host will have to resort to a polling operation to resume the operation.

The system designer should be aware of the following:

- 1) The "I" bit is the Interrupt Pending flag for Serial ATA devices.
- 2) The behavior of the "I" bit may be modified by the state of the nIEN bit in devices implemented prior to this standard. Devices implemented prior to this standard may change the behavior of the "I" bit based on the current state of the nIEN bit, as last written by a Register host to device FIS with C=0 or C=1. Some devices do not write the nIEN bit when C=1.
- 3) There is no defined behavior for a device when nIEN bit changes and the modification of the behavior of the "I" bit by nIEN is vendor specific.
- 4) The host should set nIEN=0 in all Register HD FISes. This will result in the highest compatibility with this standard.
- 5) If a device supports nIEN emulation, and the nIEN bit is set to one by the host, the host device drivers should accommodate the case of no interrupt generation when nIEN is cleared to zero and the device has a pending interrupt.

E.5 Frame Information Structure (FIS) Mapping for Parallel ATA Emulation

E.5.1 Description

The following subclauses define FIS mappings used in Parallel ATA Emulation. This methodology was used in ATA/ATAPI-7 to define mapping of register-based exchanges of the serial implementation of ATA. This mapping preserves concepts of parallel ATA such as 8-bit register definitions and bit mappings.

This mapping of the Command and Control registers into a FIS is an alternative way of naming and describing the contents of a FIS, but the contents of the FIS are not changed.

E.5.2 Register - host to device

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----------------|---|---|---|---|---|---|---|----------------|---|---|---|---|---|---|---|--------------------|---|---|--------------|---|---|---|---|----------------|---|---|---|---|--|--|--|
| | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | |
| | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 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 118 - Register - host to device FIS layout

Field Definitions

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 HBA Control register.

Setting C =1 and SRST=1 in the Control Field is invalid and will result in indeterminate behavior.

Command - Contains the contents of the Command register of the HBA Command Block.

LBA Low - Contains the contents of the LBA Low register of the HBA Command Block.

LBA Low (exp) - Contains the contents of the expanded address field of the HBA Command Block

Control - Contains the contents of the HBA Control register of the HBA Command Block.

LBA Mid - Contains the contents of the LBA Mid register of the HBA Command Block.

LBA Mid (exp) - Contains the contents of the expanded address field of the HBA Command Block

LBA High - Contains the contents of the LBA High register of the HBA Command Block.

LBA High (exp) - Contains the contents of the expanded address field of the HBA Command Block.

Device - Contains the contents of the Device register of the HBA Command Block.

Features - Contains the contents of the Features register of the HBA Command Block.

Features (exp) - Contains the contents of the expanded address field of the HBA Command Block.

R - Reserved.

Sector Count - Contains the contents of the Sector Count register of the HBA Command Block.

Sector Count (exp) - Contains the contents of the expanded address field of the HBA Command Block.

E.5.2.1 Description

The Register - host to device FIS is used to transfer the contents of the HBA Command Block and HBA Control Block from the host to the device. This is the mechanism for issuing the Parallel Transport of ATA commands to the device.

E.5.2.2 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 HBA Control register with a value different than is currently in the HBA Control register. There are BIOS and drivers that write the HBA 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 HBA Control Block. Upon initiating transmission, the current contents of the HBA Command Block and HBA 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 HBA Control register being written. The host adapter shall set the BSY bit in the HBA 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 HBA Status register to one within 400 ns after a write operation to the HBA Control register if the write to the HBA Control register changes the state of the SRST bit from 0 to 1 but shall not set the BSY bit in the HBA Status register for writes to the HBA 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 HBA Command Block and HBA Control Block as the Parallel Transport 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.

E.5.2.3 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.

E.5.3 Register - device to host

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------------|---|---|---|---|---|---|---|---|---|----------------|---|---|---|---|---|---|---|---|---|--------------------|---|---|--------------|---|---|---|---|----------------|--|-------------------|--|--|--|--|
| | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | |
| | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 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 119 - Register - device to host FIS layout

Field Definitions

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 HBA Command Block.

LBA Low (exp) - Contains the contents of the expanded address field of the HBA Command Block.

LBA Mid - Contains the contents to be placed in the LBA Mid register of the HBA Command Block.

LBA Mid (exp) - Contains the contents of the expanded address field of the HBA Command Block.

LBA High - Contains the contents to be placed in the LBA High register of the HBA Command Block.

LBA High (exp) - Contains the contents of the expanded address field of the HBA Command Block.

Device - Contains the contents to be placed in the Device register of the HBA Command Block.

Error - Contains the contents to be placed in the Error register of the HBA 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 subclause E.4.

R - Reserved

Sector Count - Contains the contents to be placed in the Sector Count register of the HBA Command Block.

Sector Count (exp) - Contains the contents of the expanded address field of the HBA Command Block.

Status - Contains the contents to be placed in the Status (and Alternate status) Register of the HBA Command Block.

E.5.3.1 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 HBA Command Block and HBA Control Block. This is the mechanism by which devices indicate command

completion status or otherwise change the contents of the host adapter's HBA Command Block and HBA Control Block.

E.5.3.2 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 HBA Command Block and HBA 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 HBA 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.

E.5.3.3 Reception

Upon reception of a valid Register - device to host FIS the received register contents are transferred to the host adapter's HBA Command Block and HBA Control Block.

If the BSY bit and the DRQ bit in the HBA 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 HBA Command Block or HBA Control Block.

E.5.4 Set Device Bits - device to host

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------------|---|---|---|---|---|---|---|---|-----------|---|---|---|-----------|---|---|---|---|---|--------------|---|---|---|---|----------------|---|---|---|---|--|
| | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
| | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 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 120 - Set Device Bit - device to host FIS layout

Field Definitions

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 HBA 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 subclause E.4.

Error- Contains the contents to be placed in the Error register of the HBA Command Block.

Status-Hi- Contains the contents to be placed in bits 6, 5, and 4 of the Status register of the HBA Command Block.

Status-Lo- Contains the contents to be placed in bits 2,1, and 0 of the Status register of the HBA Command Block.

R- Reserved

E.5.4.1 Description Set Device Bits device to host FIS

The Set Device Bits - device to host FIS is used by the device to load HBA Command Block bits for which the device has exclusive write access. 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 HBA 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 HBA Status Register are both cleared to zero when this FIS is received.

Some the serialto-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.

E.5.4.2 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 HBA 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 HBA 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.

E.5.4.3 Reception

Upon receiving a Set Device Bits - device to host, the host adapter shall load the data from the Error field into the HBA Error register, the data from the Status-Hi field into bits 6, 5, and 4, of the HBA Status register, and the data from the Status-Lo field into bits 2, 1, and 0 of the HBA Status register. Bit 7, BSY, and bit 3, DRQ, of the HBA 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 HBA Status register are cleared to zero when this FIS is received, then the host adapter shall enter an Interrupt Pending state.

E.5.5 PIO Setup - device to host

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--------------|---|---|---|---|---|---|---|----------------|---|---|---|---|---|---|---|--------------------|---|---|--------------|---|---|---|---|-------------------|---|---|---|---|--|--|--|
| | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | |
| | 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 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 121 - PIO Setup - device to host FIS layout

Field Definitions

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 HBA Command Block.

LBA Mid - Holds the contents of the LBA Mid register of the HBA Command Block.

LBA Mid (exp) - Contains the contents of the expanded address field of the HBA 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 HBA 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 HBA Command Block.

E_Status - Contains the value to be placed in the HBA 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).

E.5.5.1 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. 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 Transport 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.

E.5.5.2 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 HBA Status register at the initiation of the Data FIS and the E_Status value to be placed in the HBA 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.

E.5.5.3 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 HBA 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 HBA Status register, host software proceeds to perform a series of write operations to the HBA Data register, which the host adapter collects to produce a Data FIS to the device. Each write of the HBA 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 HBA 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.

E.5.5.4 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 HBA Status register at the beginning of the PIO Data FIS and the E_Status value to be placed in the HBA 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.

E.5.5.5 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 HBA Status and HBA Error registers and host proceeds to perform a series of read operations from the HBA Data register. Each read of the HBA Data register results in a countdown register being decremented accordingly. The E_Status value shall be transferred to the HBA 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.

E.5.6 Data - host to device or device to host (Bidirectional)

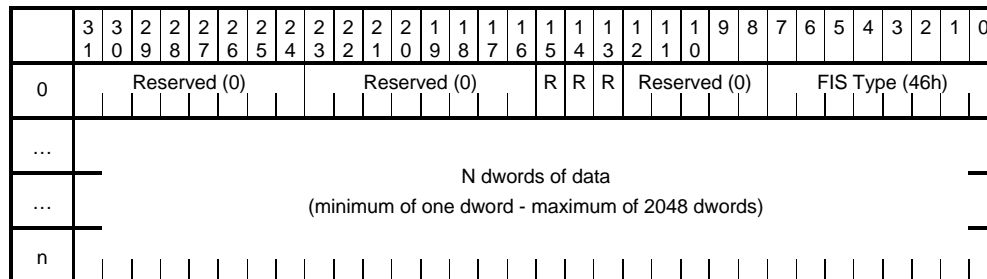


Figure 122 - Data - host to device or device to host FIS layout

Field Definitions

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 7.5.9.1.

R - Reserved

E.5.6.1 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. 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.

E.5.6.2 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 FISes 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.

E.5.6.3 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.