Serial ATA International Organization:

Serial ATA Revision 3.0

RELEASE CANDIDATE 1
1-March-2009

SATA-IO Board Members:

Dell Computer Corporation

Hewlett Packard Corporation

Hitachi Global Storage Technologies, Inc.

Intel Corporation

Maxim Integrated Products

Seagate Technology

Western Digital Corporation

Serial ATA International Organization: Serial ATA Revision 3.0 specification ("Final Specification") is available for download at www.sata-io.org.

SPECIFICATION DISCLAIMER

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

For more information about Serial ATA, refer to the Serial ATA International Organization website at www.sata-io.org.

Serial ATA International Organization contact information:

SATA-IO
3855 SW 153rd Drive
Beaverton, Oregon 97006 USA
Tel: +1 503-619-0572
Fax: +1 503-644-6708
E-mail: admin@sata-io.org

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Revision History

1.1 Revision 2.5 (Ratification Date October 27, 2005)

Release that integrates and consolidates the following previously published specifications including all erratum against those specifications:

- Serial ATA revision 1.0a
- Serial ATA II: Extensions to Serial ATA 1.0a revision 1.2
- Serial ATA II: Electrical Specification revision 1.0
- Serial ATA II: Cable and Connectors Volume 1 revision 1.0
- Serial ATA II: Cable and Connectors Volume 2 revision 1.0
- Serial ATA II: Port Multiplier revision 1.2
- Serial ATA II: Port Selector revision 1.0

1.2 Revision 2.6 (Ratification Date February 15, 2007)

Release that incorporates errata against Revision 2.5 and the following new features and enhancements:

- Internal Slimline cable and connector
- Internal Micro SATA connector for 1.8” HDD
- Mini SATA Internal Multilane cable and connector
- Mini SATA External Multilane cable and connector
- NCQ Priority
- NCQ Unload
- Enhancements to the BIST Activate FIS
- Enhancements for robust reception of the Signature FIS

1.3 Revision 3.0 (Release Candidate 1, March 1, 2009)

Release that incorporates errata against Revision 2.6:

- ECN001 - Slimline Bump Correction
- ECN002 - Bump Correction
- ECN003 - State Name Corrections
- ECN004 - ATA Log & Subcode reservations
- ECN006 - fbaud/10 Jitter Parameter Removal
- ECN008 - fbaud/500 Jitter Parameter Clarification
- ECN009 - Correcting LBP references in the COMP data pattern and other locations
- ECN010 - Power State Resume Speed
- ECN011 - Data validity clarifications
- ECN012 - Cable & Connector Retention
- ECN013 - Section 13 Corrections
- ECN014 - PACKET State Names
- ECN016 - Long Term Frequency Accuracy and SSC Profile Tests for Transmitters
- ECN017 - OOB Burst/Gap Duration Clarification
- ECN018 - Figure 69 Pin Location Correction
- ECN019 - Cable ISI Test Source Risetime
- ECN021 - External plug height
- ECN022 - Editorial cleanup – 8KB, IDENTIFY DEVICE
- ECN023 - L-Key Opening Correction (Slimline Host Receptacle Connector)
- ECN024 - Contact Current Rating procedure
- ECN025 - Rise Time Measurements

- ECN026 - Gen 3i TX TJ Measurement Location
- ECN027 - Gen3i Rx Differential Return Loss Text Description and Figure Change - Clarification Only
- ECN028 - Clarification of Test Patterns for Measurement Defined in 7.2 Electrical Specification
- ECN029 - Addition of Pattern to the TX AC Common Mode Voltage (Gen3i) Measurement
- ECN031 - Correction to ECN 004
- ECN032 - Correction to TX AC Common Mode Voltage Table Value 'Units'
- ECN033 - Definition of Terms
- ECN034 - Corrections to Technical Proposal 005
- ECN035 - Clarification of Words 76 to 79
- ECN036 - LIF-SATA Clarifications
- ECN037 - Changes made by Technical Integration Work Group

and the following new features and enhancements

- TP002 - Gen3 register assignments
- TP004 - NCQ Clarifications
- TP005 - SATA Speed Indicator in ID String
- TP007 - Automatic Partial to Slumber Transitions
- TP009 - LIF Connector for 1.8” HDD for SATA Revision 2.6
- TP010 - Serial ATA NCQ Streaming Command
- TP011 - Serial ATA NCQ QUEUE MANAGEMENT Command
- TP012 - Gen 1 Clock to Data Jitter Definition
- TP013 - Connector for 7 mm slimline drives
- TP014 - Allow READ LOG DMA EXT to Clear NCQ Error
- TP015 - Remove Device Register from Signature
- TP016 - Add Write-Read-Verify to SSP support
- TP017 - Align SATA 2.6 with ATA8-ACS
- TP018 - Specification Revisions For Gen3i

2 Scope

This specification defines a high-speed serialized ATA data link interface (specifying Phy, Link, Transport, and Application layers). The serialized interface uses the command set from the ATA8-ACS standard, augmented with Native Command Queuing commands optimized for the serialized interface. The serialized ATA interface is defined in a register-compatible manner with parallel ATA to enable backward compatibility with parallel ATA drivers. The physical interface is defined to ease integration (low pin count, low voltages) and enable scalable performance (with currently defined data rates of 1.5 Gbps, 3.0 Gbps and 6.0 Gbps).

Complementary components are also specified including interconnect solutions for various applications, port expansion devices, and failover devices.

Normative information is provided to allow interoperability of components designed to this specification. Informative information, when provided, may illustrate possible design implementation.

3 Normative references

The following standards contain provisions that, through 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: Approved ANSI standards, approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), and 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.

3.1 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.

AT Attachment with Packet Interface – 5 (ATA/ATAPI-5) [ANSI INCITS 340-2000]

AT Attachment with Packet Interface – 6 (ATA/ATAPI-6) [ANSI INCITS 361-2002]

ATA/ATAPI-8 Command Set (ATA8-ACS) [ANSI INCITS 452-2008]

Serial Attached SCSI – 1.1 (SAS-1.1) [ANSI INCITS 417-2006]

SCSI-3 Enclosure Services (SES) Command Set [ANSI INCITS 305-1998]

SCSI-3 Enclosure Services (SES) Amendment 1 [ANSI INCITS 305-1998/AM1-2000]

SCSI Enclosure Services - 2 (SES-2) [ANSI INCITS T10/1559-D]

ATA/ATAPI Host Adapters Standard [ANSI INCITS 370-2004]

ASME Y14.5M Dimensioning and Tolerancing

To obtain copies of these documents, contact Global Engineering or INCITS.

Global Engineering Documents, an IHS Company
15 Inverness Way East
Englewood, CO 80112-5704
USA
Web site: http://global.ihs.com
Telephone: (303) 397-7956 or (303) 792-2181 or (800) 854-7179

Document Distribution
INCITS Online Store
managed by Techstreet
1327 Jones Drive
Ann Arbor, MI 48105
USA

Web site: http://www.techstreet.com/incits.html
Telephone: (734) 302-7801 or (800) 699-9277

Additional material and draft versions available from http://www.T10.org and http://www.T13.org.

SAF-TE – SCSI Accessed Fault-Tolerant Enclosure version 1.00 [revision R041497, April 14, 1997]. Available for download at http://www.intel.com/design/servers/ipmi/pdf/sr041497.pdf.

I^2C-Bus Specification version 2.1. Available from Philips Semiconductors at http://www.semiconductor.philips.com/buses/i2c.

IPMB - Intelligent Platform Management Bus Communications Protocol Specification version 1.0. Available for download at http://www.intel.com/design/servers/ipmi/spec.htm.

IPMI - Intelligent Platform Management Interface Specification version 1.5. Available for download at http://www.intel.com/design/servers/ipmi/spec.htm.

JEDEC Standards (Located on http://www.jedec.com).

- JESD22-A114-B, Electrostatic Discharge (ESD) Sensitivity Testing Human Body Model (HBM)
- JESD22-C101-A, Field-Induced Charged-Device Model Test Method for Electrostatic-Discharge-Withstand Thresholds of Microelectronic Components"

The following standards published by the SFF Committee are referenced. These standards are available for download through http://www.sffcommittee.org.

- SFF-8086, Compact Multilane Series: Common Elements
- SFF-8087, Compact Multilane Series: Unshielded
- SFF-8088, Compact Multilane Series: Shielded
- SFF-8111, 1.8" drive form factor (60x70 mm)
- SFF-8144, 54 mm x 71 mm Form Factor w/micro SATA Connector
- SFF-8201, Form Factor of 2.5" Disk Drives
- SFF-8301, Form Factor of 3.5" Disk Drives
- SFF-8470, Shielded High Speed Serial Multilane Copper Connector
- SFF-8484, Multilane Internal Serial Attachment Connector
- SFF-8553, Form Factor of 5 1/4" 7 mm Height Optical Drives with SATA Interface

The following EIA-364-xx standards published by EIA are referenced. To obtain copies of these documents, contact Global Engineering.

- EIA-364-09, Durability Test Procedure for Electrical Connectors and Contacts
- EIA-364-13, Mating and Unmating Forces Test Procedure for Electrical Connectors
- EIA-364-17, Temperature Life with or without Electrical Load Test Procedure for Electrical Connectors and Sockets
- EIA-364-18, Visual and Dimensional Inspection for Electrical Connector
- EIA-364-20, 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 Shock (Specified Pulse) Test Procedure for Electrical Connectors
- EIA-364-28, 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, Cable Pull-Out Test Procedure for Electrical Connectors
- EIA-364-41, Cable Flexing Test Procedure for Electrical Connectors
- EIA-364-65, Mixed Flowing Gas

Many of the EIA-364-xx specifications are available for download from:
http://www.ecaus.org/engineering/Downloads/eia364.htm

The following form factor standards published by EIA are referenced. These standards may be obtained from Global Engineering.

- EIA-740, specification for small form factor 3.5" disk drives
- EIA-720, specification for small form factor 2.5" disk drives

3.2 References under development

The following ANSI standards under development are referenced. Draft versions of these standards are available from http://www.T10.org or http://www.T13.org.

ATA/ATAPI-8 Serial Transport (ATA8-AST) [ANSI INCITS T13/1697-D]
ATA/ATAPI-8 Parallel Transport (ATA8-APT) [ANSI INCITS T13/1698-D]
ATA Host Adapter Standards - 2 (HBA-2) [ANSI INCITS T13/2014D]

3.3 Other references

The 8b/10b code used in Serial ATA is based on the following published references:

[1] A.X. Widmer and P.A. Franaszek, "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code." *IBM Journal of Research and Development*, 27, no. 5: 440-451 (September, 1983)

[2] U.S. Patent 4,486,739. Peter A. Franaszek and Albert X. Widmer. *Byte Oriented DC Balanced (0,4) 8B/10B Partitioned Block Transmission Code*. (Dec. 4, 1984)

4 Definitions, abbreviations, and conventions

4.1 Definitions and abbreviations

4.1.1 Active Port

The active port is the currently selected host port on a Port Selector.

4.1.2 ATA (AT Attachment)

ATA defines the physical, electrical, transport, and command protocols for the attachment of storage devices.

4.1.3 ATAPI (AT Attachment Packet Interface) device

A device implementing the Packet Command feature set.

4.1.4 BER (bit error rate)

4.1.5 bitrate

Reciprocal of the unit interval. Bitrate = 1 / UI.

4.1.6 bit synchronization

The state in which a receiver has synchronized the internal receiver clock to the external transmitter and is delivering retimed serial data.

4.1.7 burst

A short pulse of data starting from and ending with the idle condition on the interface. These are used for OOB signaling.

4.1.8 byte

A byte is an ordered set of eight (8) bits. The least significant bit is bit 0 and the most significant bit is bit 7.

4.1.9 character

A character is a representation of a byte in the Zxx.y notation (see 9.2.1).

4.1.10 character alignment

Character alignment is a receiver action that resets the character boundary to that of the comma sequence found in the K28.5 control character of $ALIGN_P$, and establishes Dword synchronization of the incoming serial data stream.

4.1.11 character slipping

Character slipping is the receiver action that realigns the receiver's clock to the received bit stream by adding or removing bit times within the characters of $ALIGN_P$.

4.1.12 ClickConnect

An optional positive latch solution for internal single lane interconnects (see section 6.1.4).

4.1.13 CLTF (Closed Loop Transfer Function)

For a feedback system, the CLTF is the ratio of the magnitude and phase of the output variable to the magnitude and phase of the input variable, as a function of frequency for sinusoidal excitation. This term is used in the Reference Clock sections of this specification.

4.1.14 code violation

A code violation is an error that occurs in the reception process as a result of (1) a running disparity violation or (2) an encoded character that does not translate to a valid data or control character or (3) an encoded character that translates to a control character other than K28.5 or K28.3 in byte 0 of a Dword or (4) an encoded character that translates to any control character (valid or invalid) in bytes 1-3 of a Dword.

4.1.15 comma character

A comma character is a control character, that when encoded, contains the comma sequence. In Serial ATA the only comma character used is K28.5, and only $ALIGN_P$ contains the comma character. The comma sequence is the first seven bits of the encoded character.

4.1.16 comma sequence

The comma sequence is a seven-bit sequence of 0011111 or 1100000 in an encoded stream. The comma sequence is unique in that it appears only in a single encoded character, and furthermore, may not appear in any subset of bits in adjacent encoded characters. This unique property allows the comma sequence to be used for determining alignment of the received data stream.

4.1.17 command aborted

Command aborted is command completion with ABRT bit set to one in the Error register, and ERR bit set to one in the Status register.

4.1.18 command completion

Command completion describes the completion of an action requested by command, applicable to the device. Command completion also applies to the case where the command has terminated with an error, and the following actions occurred:

a) the appropriate bits of the Status Register have been updated
b) BSY & DRQ bits have been cleared to zero
c) assertion of INTRQ (if nIEN is active-low), assuming that the command protocol specifies INTRQ to be asserted

In Serial ATA, the register contents are transferred to the host using a Register - Device-to-Host FIS.

4.1.19 command packet

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

4.1.20 concentrator

A concentrator is a generic term used to describe a logical block that has multiple Serial ATA ports to connect to Serial ATA devices plus some small number of ports to connect to a host. In the simplest case a concentrator may be a host bus adapter (HBA) that is plugged into the host that connects to some number of Serial ATA devices (like a PCI Serial ATA controller card). A concentrator may also be an internal or external RAID controller such as a fibre-channel to Serial ATA RAID controller, or may be some element that expands the number of ports through a fan-out scheme, like a Port Multiplier.

4.1.21 Control Block registers

Control Block registers are interface registers used for device control and to post alternate status.

4.1.22 control character

A control character is a character in which Z is equal to K (See 9.2.1).

4.1.23 control port

The Port Multiplier has one port address reserved for control and status communication with the Port Multiplier itself. The control port has port address Fh.

4.1.24 control variable

The control variable, Z, is a flag that determines the code set to be used to interpret a data byte. The control variable has the value D (for data characters) or K (for control characters).

4.1.25 CRC (Cyclic Redundancy Check)

An error checking mechanism that checks data integrity by computing a polynomial algorithm based checksum. In Serial ATA a 32-bit CRC is calculated over the contents of a Frame Information Structure. The Serial ATA CRC is the Dword in a frame that immediately precedes EOF_P.

4.1.26 data character

A data character is a character in which Z is equal to D (See 9.2.1).

4.1.27 data signal source

An instrument which provides a Serial ATA data signal.

4.1.28 device

Device is 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 specification and to an ATA standard.

4.1.29 device port

The device port is a port on a Port Multiplier or a Port Selector that is connected to the device. Port Multipliers may have up to 15 device ports. Port Selectors have one device port.

4.1.30 DCB (DC block)

The DC block is defined as a device that passes frequencies from 10 MHz to at least 12 GHz with minimal effect on the amplitude or phase of the signal.

4.1.31 differential signal

The differential signal is the voltage on the positive conductor minus the voltage on the negative conductor (i.e. TX+ – TX-).

4.1.32 DJ (deterministic jitter – peak to peak)

All jitter sources that have bounded probability distribution functions (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.

4.1.33 DMA (direct memory access)

DMA is a means of data transfer between device and host memory without host processor intervention.

4.1.34 Dword

A Dword is an ordered set of thirty-two (32) bits. The least significant bit is bit 0 and the most significant bit is bit 31.

4.1.35 Dword synchronization

The state in which a receiver has recognized the comma sequence and is producing an aligned data stream of Dwords (four contiguous bytes) from the zero-reference of the comma character.

4.1.36 EMI (Electromagnetic Interference)

4.1.37 encoded character

An encoded character is the output of the 8b/10b encoder – the result of encoding a character. An encoded character consists of 10 bits, where bit 0 is the most significant bit and bit 9 is the least significant. The bits in an encoded character are symbolically referred to as “abcdeifghj” where “a” corresponds to bit 0 and “j” corresponds to bit 9. Case is significant. Note the out-of-order representation. See section 9.2 for a description of the relationship between bytes, characters and encoded characters.

4.1.38 endpoint device

An endpoint device is an ATA or ATAPI device, as reported by the device signature after power-on or reset. This may include hard disk drives, optical disk drives, and tape drives.

4.1.39 elasticity buffer

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

4.1.40 eSATA

The System to System Interconnects - External Desktop Applications usage model (see section 5.2.6).

4.1.41 Fbaud

The nominal rate of data through the channel, measured in GHz.

4.1.42 FER (frame error rate)

Refer to section 7.2.2.1.2.

4.1.43 First-party DMA Data Phase

The First-party DMA Data Phase is the period from the reception of a DMA Setup FIS until either the exhaustion of the associated data transfer count or the assertion of the ERR bit in the shadow Status register.

4.1.44 First-party DMA access

First-party DMA access is a method by which a device accesses host memory.

4.1.45 FIS (Frame Information Structure)

The user payload of a frame, does not include the SOF_P, CRC, and EOF_P delimiters.

4.1.46 frame

A frame is an indivisible unit of information exchanged between a host and device. A frame consists of SOF_P, a Frame Information Structure, a CRC calculated over the contents of the FIS, and EOF_P.

4.1.47 Gen1

Refers to first generation signaling characterized by a speed of 1.5 Gbps. Refer to section 7.1.

4.1.48 Gen1i

The internal electrical specifications at 1.5 Gbps with cable lengths up to 1 meter.

4.1.49 Gen1m

The electrical specifications used in Short Backplane Applications, External Desktop Applications, and Data Center Applications with cable lengths up to two meters, defined at 1.5 Gbps.

4.1.50 Gen1x

The electrical specifications used in Long Backplane Applications and Data Center Applications supporting cable lengths up to and greater than two meters, defined at 1.5 Gbps.

4.1.51 Gen2

Refers to second generation signaling characterized by a speed of 3.0 Gbps. Refer to section 7.1.

4.1.52 Gen2i

The internal electrical specifications at 3.0 Gbps with cable lengths up to 1 meter.

4.1.53 Gen2m

The electrical specifications used in Short Backplane Applications, External Desktop Applications, and Data Center Applications with cable lengths up to two meters, defined at 3.0 Gbps.

4.1.54 Gen2x

The electrical specifications used in Long Backplane Applications and Data Center Applications supporting cable lengths up to and greater than two meters, defined at 3.0 Gbps.

4.1.55 Gen3

Refers to third generation signaling characterized by a speed of 6.0 Gbps.

4.1.56 Gen3i

The internal electrical specifications at 6.0 Gbps with cable lengths up to 1 meter.

4.1.57 HBA (Host Bus Adapter)

HBA is a component that connects to the host system's expansion bus to provide connectivity for devices. HBAs are also often referred to as controller cards or merely controllers.

4.1.58 HBWS (High Bandwidth Scope)

An oscilloscope with an analog bandwidth of 10 GHz or greater in the measurement path.

4.1.58.1 Gen1/Gen2 Requirement

For Gen1/Gen2 measurements, an oscilloscope with an analog bandwidth of 10 GHz or greater shall be used in the measurement path.

4.1.58.2 Gen3 Requirement

For Gen3 measurements, an oscilloscope with an analog bandwidth of 12 GHz or greater shall be used in the measurement path.

4.1.59 HFTP (High Frequency Test Pattern)

This pattern provides the maximum frequency allowed within the Serial ATA encoding rules. Pattern: 1010101010 1010101010b = encoded D10.2. The pattern is repetitive.

4.1.60 hot plug

The connection of a SATA device to a host system that is already powered. The SATA device is already powered or powered upon insertion/connection. See section 7.2.5.1 for details on hot plug scenarios.

4.1.61 host port

The host port is the port that is used to connect the Port Multiplier or Port Selector to a host. Port Multipliers have one host port. Port Selectors have two host ports.

4.1.62 inactive port

The inactive port is the host port that is not currently selected on the Port Selector.

4.1.63 interrupt pending

Interrupt pending is an internal state of the device that exists when the device protocol requires the device to notify the host of an event by asserting INTRQ, given the condition where nIEN is asserted active-low to zero.

4.1.64 immediate NCQ command

An immediate NCQ command is a NCQ command that shall be processed:

a) After any command previously accepted by the device for which the device has transmitted a DMA Setup FIS and has not reached command completion; and
b) Before any NCQ command previously accepted by the device for which the device has not transmitted a DMA Setup FIS.

4.1.65 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 a high level of high-frequency, data-dependent, jitter.

4.1.66 JMD (jitter measuring device)

A device used to measure jitter. Examples are a bit error rate tester (BERT), a timing interval analyzer (TIA), a single shot capture oscilloscope and processing software, or a HBWS.

4.1.67 JTF (Jitter Transfer Function)

In general terms, the JTF of a system is the ratio of the jitter magnitude and phase of the output variable to the jitter magnitude and phase of the input variable, as a function of frequency for sinusoidal jitter excitation. In the case of a jitter definition, this defines the magnitude of the jitter, as a function of frequency allowed to be generated by the transmitter or tolerated by the receiver. In the case of a JMD, this defines the ratio of the reported jitter to the applied jitter, as a function of frequency for sinusoidal excitation.

4.1.68 junk

An 8b/10b encoded data Dword sent between $CONT_P$ and another primitive transmitted on the link. All junk Dwords shall be ignored by the receiver.

4.1.69 LBA (Logical Block Address)

As defined in the ATA8-ACS standard.

4.1.70 LBP (Lone Bit Pattern)

This pattern is defined in section 7.2.4.3.5. The pattern is repetitive.

4.1.71 LED (Light Emitting Diode)

4.1.72 legacy mode

Legacy mode is the mode of operation which provides software-transparent communication of commands and status between a host and device using the ATA Command Block and Control Block registers.

4.1.73 legal character

A legal character is one for which there exists a valid decoding, either into the data character or control character fields. Due to running disparity constraints not all 10-bit combinations result in a legal character. Additional usage restrictions in Serial ATA result in a further reduction in the SATA defined control character space.

4.1.74 LFSR (Linear Feedback Shift Register)

Refer to section 9.4.5 for details on using a LFSR in scrambling.

4.1.75 LFTP (low frequency test pattern)

This pattern provides a low frequency, which is allowed within the Serial ATA encoding rules. Pattern: 0111100011 1000011100b = encoded D30.3. The pattern is repetitive.

4.1.76 LL (laboratory load)

An electrical test system connected to the unit under test. The LL receives a signal from the UUT at the defined impedance level of 100 Ohms differential and 25 Ohms common mode.

4.1.77 LSS (laboratory sourced signal or lab-sourced signal)

An instrument and electrical test system connected to the unit under test. The LSS provides a signal to the UUT at the defined impedance level of 100 Ohms differential and 25 Ohms common mode.

4.1.78 MFTP (mid frequency test pattern)

This pattern provides a middle frequency which is allowed within the Serial ATA encoding rules. Pattern: 1100110011 0011001100b = encoded D24.3. The pattern is repetitive.

4.1.79 NCQ streaming command

An NCQ Streaming command is a command using the FPDMA QUEUED protocol for which the ICC field is non-zero.

4.1.80 NCQ Non-streaming command

An NCQ Non-Streaming command is a command using the FPDMA QUEUED protocol which the ICC field is zero.

4.1.81 OOB (Out-of-Band signaling)

OOB signaling is a pattern of $ALIGN_P$ primitives or Dwords composed of D24.3 characters and idle time and is used to initialize the Serial ATA interface. OOB signaling is also used to recover from low power states and to signal specific actions during test modes. Refer to section 8.

4.1.82 OS-aware hot plug

The insertion of a SATA device into a backplane that has power shutdown. The backplane is later powered, and both the device and the host power up, and the host-initiated OOB sequence determines the time that SATA operations begin.

4.1.83 OS-aware hot removal

The removal of a SATA device from a powered backplane, that has been first placed in a quiescent state.

4.1.84 Phy offline

In this mode the host Phy is forced off and the host Phy does not recognize nor respond to COMINIT or COMWAKE. This mode is entered by setting the DET field of the SControl register to 0100b. This is a mechanism for the host to turn off its Phy.

4.1.85 PIO (programmed input/output)

PIO is a means of accessing device registers. PIO is also used to describe one form of data transfers. PIO data transfers are performed by the host processor utilizing PIO register accesses to the Data register.

4.1.86 port address

The control port and each device port present on a Port Multiplier have a port address. The port address is used to route FISes between the host and a specific device or the control port.

4.1.87 PRD (Physical Region Descriptor)

A PRD table is a data structure used by DMA engines that comply with the ATA/ATAPI Host Adapters standard. The PRD describes memory regions to be used as the source or destination of data during DMA transfers. A PRD table is often referred to as a scatter/gather list.

4.1.88 primitive

A primitive is a special Dword used by the Link layer for the transport control. Byte 0 of each primitive is a control character.

4.1.89 protocol-based port selection

Protocol-based port selection is a method that may be used by a host to select the host port that is active on a Port Selector. Protocol-based port selection uses a sequence of Serial ATA OOB Phy signals to select the active host port.

4.1.90 quiescent power condition

Entering a quiescent power condition for a particular Phy is defined as the Phy entering the idle bus condition as defined in section 7.5.2.

4.1.91 RJ (random jitter)

Random jitter is Gaussian. Random jitter is equal to the peak to peak value of 14 times the 1σ standard deviation value given the 10^{-12} BER requirement.

4.1.92 sector

A set of data bytes accessed and referenced as a unit.

4.1.93 SEMB (Serial ATA Enclosure Management Bridge)

A SEMB is a logical block that translates Serial ATA transactions into I^2C transactions to communicate enclosure services commands to a Storage Enclosure Processor.

4.1.94 SEP (Storage Enclosure Processor)

A SEP is a logical block that interfaces with the various enclosure sensors and actuators in an enclosure and is controlled through an I^2C interface to the Serial ATA Enclosure Management Bridge.

4.1.95 Shadow Register Block registers

Shadow Register Block registers are interface registers used for delivering commands to the device or posting status from the device.

4.1.96 side-band port selection

Side-band port selection is a method that may be used by a host to select the host port that is active on a Port Selector. Side-band port selection uses a mechanism that is outside of the Serial ATA protocol for determining which host port is active. The port selection mechanism used in implementations that support side-band port selection is outside the scope of this specification.

4.1.97 SMART

Self-Monitoring, Analysis, and Reporting Technology for prediction of device degradation and/or faults.

4.1.98 SSC (spread spectrum clocking)

The technique of modulating the operating frequency of a signal slightly to spread its radiated emissions over a range of frequencies. This reduction in the maximum emission for a given frequency helps meet radiated emission requirements.

4.1.99 surprise hot plug

The insertion of a SATA device into a backplane that has power present. The device powers up and initiates an OOB sequence.

4.1.100 surprise hot removal

The removal of a SATA device from a powered backplane, without first being placed in a quiescent state.

4.1.101 SYNC Escape

The condition when $SYNC_P$ is used to escape from the present FIS transmission on the interface and resynchronize the link back to an IDLE state. The most common use of the SYNC Escape mechanism is to bring the link to an IDLE condition in order to send a Control Register FIS with a software reset to the device, but may be used at other times to recover link communication from erroneous conditions. A SYNC Escape is requested by the Transport layer to the Link layer.

4.1.102 TDR (time domain reflectometer)

An instrument used to test the impedance of the unit under test.

4.1.103 TIA (timing interval analyzer)

Timing interval analyzer with Duty Cycle Distortion and ISI noise floor performance of better than 5% of a UI for K28.5 with less than 67 ps rise and fall times.

4.1.104 TJ (total jitter)

Peak to peak value of (14 * RJ_{σ}) + DJ.

4.1.105 UI (unit interval)

Equal to the time required to transmit one bit (e.g. 666.667 ps for Gen1).

4.1.106 unrecoverable error

An unrecoverable error is defined as having occurred at any point when the device sets either the ERR bit or the DF bit to one in the Status register at command completion.

4.1.107 UUT (unit under test)

The product under test and the other half of the “mated“ connector (which is physically on the laboratory load but considered part of the UUT).

4.1.108 VNA (vector network analyzer)

An instrument used to test the impedance of the unit under test.

4.1.109 warm plug

Device connection with host controller powered and power at connector pins off (un-powered). This mechanism is used in Slimline applications, refer to section 6.3.

4.1.110 word

A word is an ordered set of sixteen (16) bits. The least significant bit is bit 0 and the most significant bit is bit 15.

4.1.111 xSATA

The System to System Interconnects - Data Center Applications usage model (see section 5.2.5).

4.1.112 zero crossing

To locate the zero crossing of a Data Eye, turn on the horizontal histogram function to horizontally enclose all waveforms associated with the “edge” and vertically limit to +/-5% of the waveform voltage. The “zero crossing” is the location of the mean of the waveforms.

4.2 Conventions

Lowercase is used for words having the normal English meaning. Certain words and terms used in this document have a specific meaning beyond the normal English meaning. These words and terms are defined either in section 4.1 or in the text where they first appear.

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

Names of device registers begin with a capital letter (e.g., Command).

Primitive names are followed by a 'P' subscript (e.g., R_OK_P).

4.2.1 Precedence

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

4.2.2 Keywords

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

4.2.2.1 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.

4.2.2.2 mandatory

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

4.2.2.3 may

A keyword that indicates flexibility of choice with no implied preference.

4.2.2.4 na

A keyword that indicates that a field or value is not applicable and has no defined value and should not be checked by the recipient.

4.2.2.5 obsolete

A keyword used to describe bits, bytes, fields, and code values that no longer have consistent meaning or functionality from one implementation to another. However, some degree of functionality may be required for items designated as "obsolete" to provide for backward compatibility. An obsolete bit, byte, field, or command shall never be reclaimed for any other use in any future standard. Bits, bytes, fields, and code values that had been designated as "obsolete" in previous standards may have been reclassified as "retired" in this standard based on the definitions herein for "obsolete" and "retired".

4.2.2.6 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.

4.2.2.7 retired

A keyword indicating that the designated bits, bytes, 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, fields, or code values are utilized before they are reclaimed, they shall have the meaning or functionality as described in previous standards.

4.2.2.8 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.

4.2.2.9 shall

A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other standard conformant products.

4.2.2.10 should

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

4.2.3 Numbering

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

4.2.4 Dimensions

All dimensions are shown in millimeters unless otherwise noted.

4.2.5 Signal conventions

Signal names are shown in all uppercase letters.

4.2.6 State machine conventions

For each function to be completed a state machine approach is used to describe the sequence requirements. Each function is composed of several states to accomplish a set goal. Each state of the set is described by an individual state table. Table 1 below shows the general layout for each of the state tables that comprise the set of states for the function.

Table 1 – State Table Cell Description

State Designator: State name	Action list[P | W]	
Transition condition 0	→	**Next state 0**
Transition condition 1	→	**Next state 1**

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

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. 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.

It is assumed that all actions are executed within a state and that transitions from state to state are instantaneous.

4.2.7 Byte, word and Dword Relationships

The most significant bit in a byte (i.e., bit 7) is shown on the left (see Figure 1).

A word may be represented as an ordered set of two (2) bytes. The least significant byte (lower byte) is byte 0 and the most significant byte (upper byte) is byte 1. The most significant byte is shown on the left (see Figure 1).

A Dword may be represented as an ordered set of two (2) words. The least significant word (lower word) is word 0 and the most significant word (upper word) is word 1. The most significant word is shown on the left (see Figure 1).

A Dword may be represented as an ordered set of four (4) bytes. The least significant byte is byte 0 and the most significant byte is byte 3. The most significant byte is shown on the left (see Figure 1).

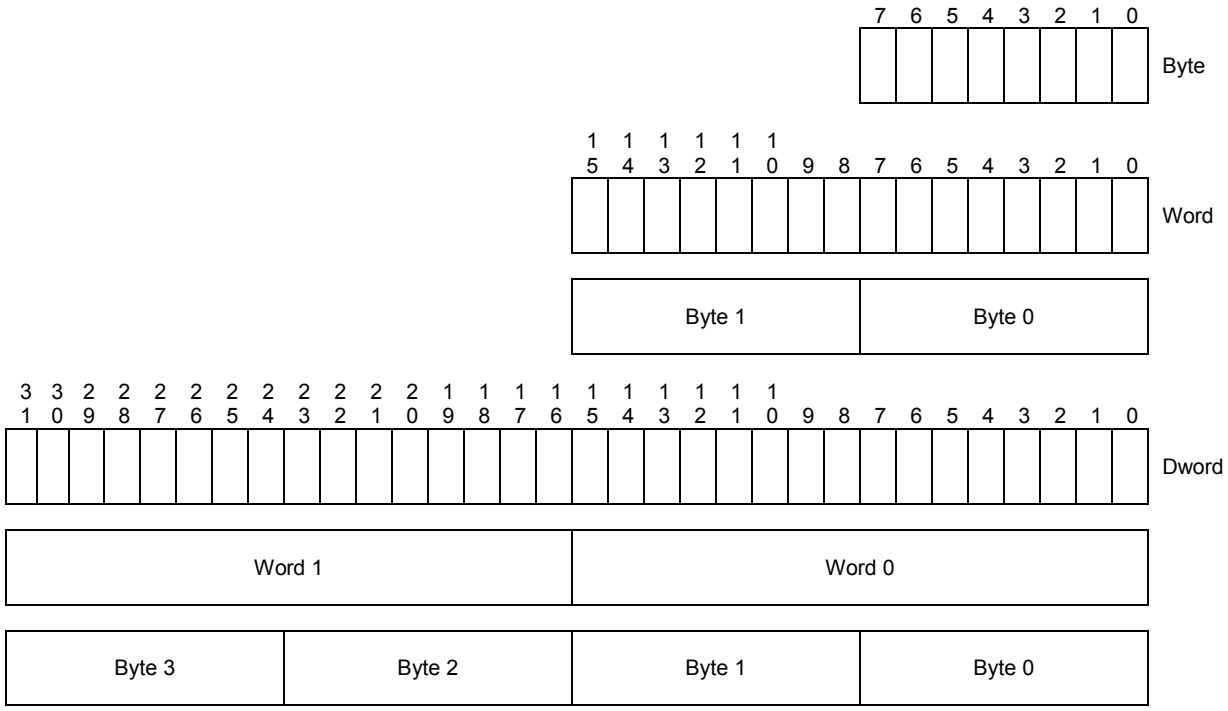


Figure 1 – Byte, word and Dword relationships

5 General overview

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

Figure 2 illustrates how two devices are connected to a parallel ATA host adapter. This method allows up to two devices to be connected to a parallel ATA bus using a Master/Slave communication technique. Each device is connected via a ribbon cable that “daisy chains” the devices.

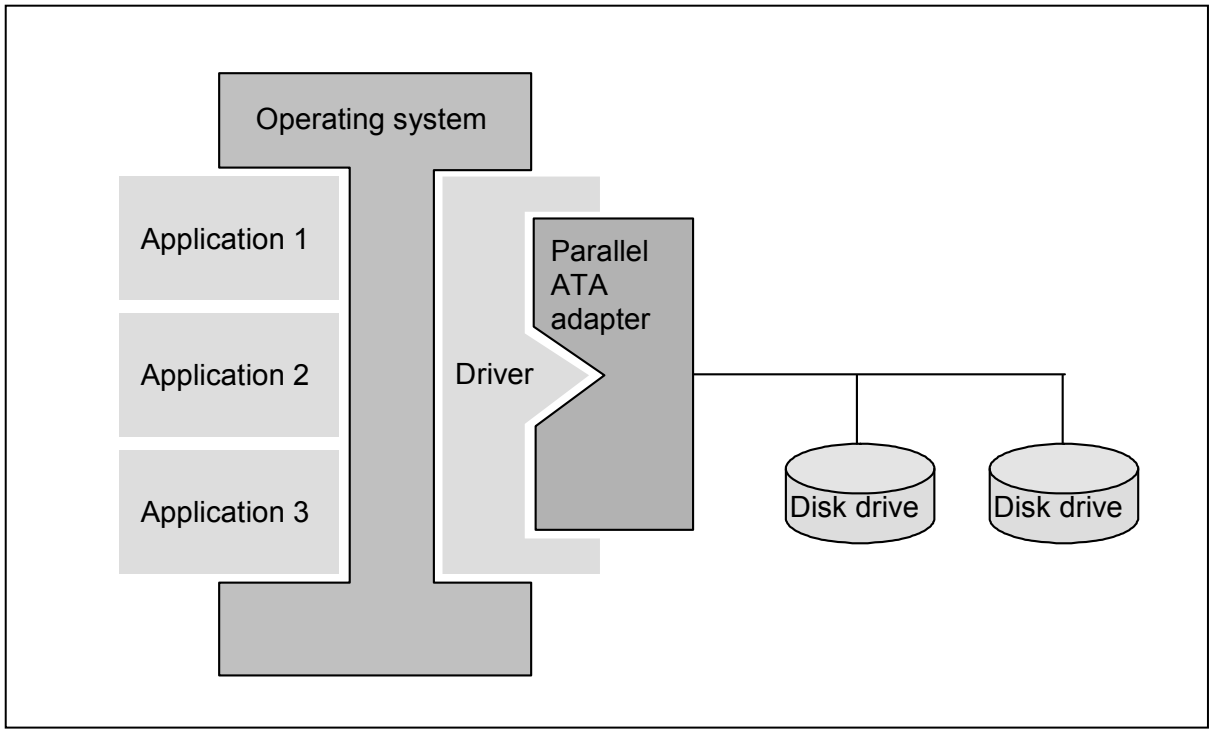


Figure 2 – Parallel ATA device connectivity

Figure 3 shows an example of how the same two devices are connected using a Serial ATA HBA. In this figure the dark grey portion is functionally identical to the dark grey portion of Figure 2. Host software that is only parallel ATA aware accesses the Serial ATA subsystem in exactly the same manner and functions correctly. In this case, however, the software views the two devices as if they were “masters” on two separate ports. The right hand portion of the HBA is of a new design that converts the normal operations of the software into a serial data/control stream. The Serial ATA structure connects each of the two devices with their own respective cables in a point-to-point fashion.

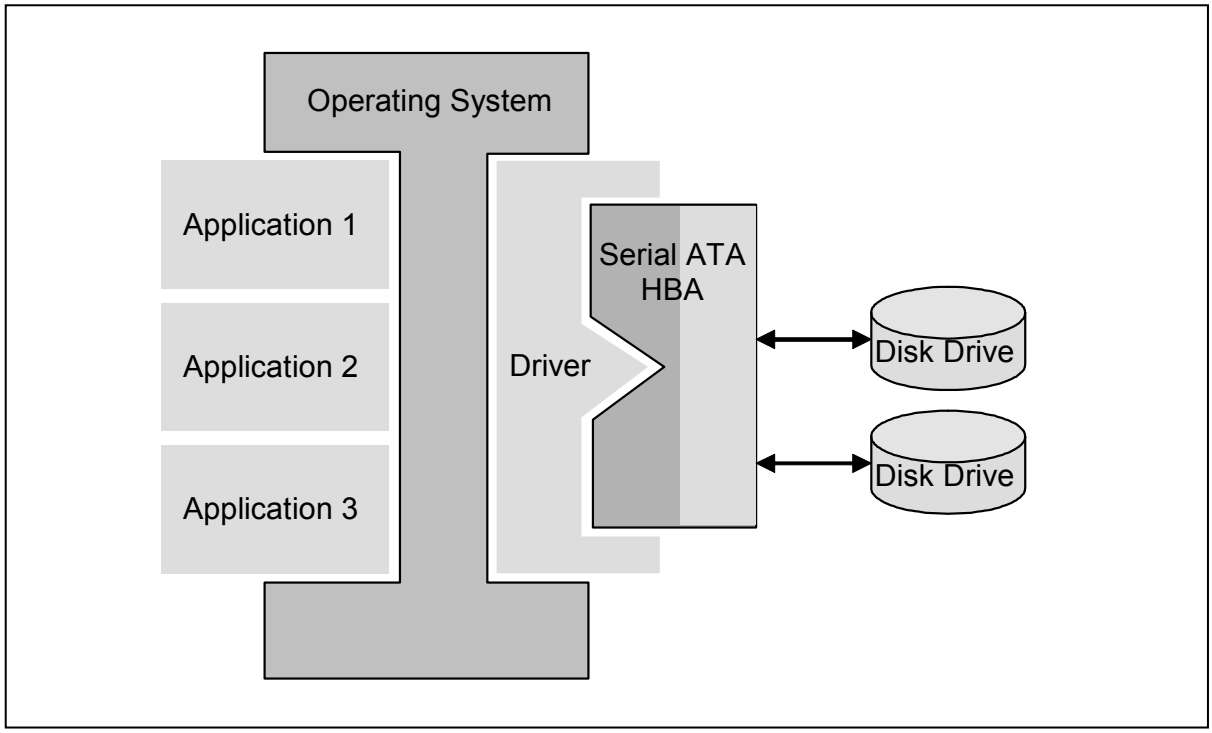


Figure 3 – Serial ATA connectivity

5.1 Architecture

There are four layers in the Serial ATA architecture: Application, Transport, Link, and Phy. The Application layer is responsible for overall ATA command execution, including controlling Command Block Register accesses. The Transport layer is responsible for placing control information and data to be transferred between the host and device in a packet/frame, known as a Frame Information Structure (FIS). The Link layer is responsible for taking data from the constructed frames, encoding or decoding each byte using 8b/10b, and inserting control characters such that the 10-bit stream of data may be decoded correctly. The Physical layer is responsible for transmitting and receiving the encoded information as a serial data stream on the wire.

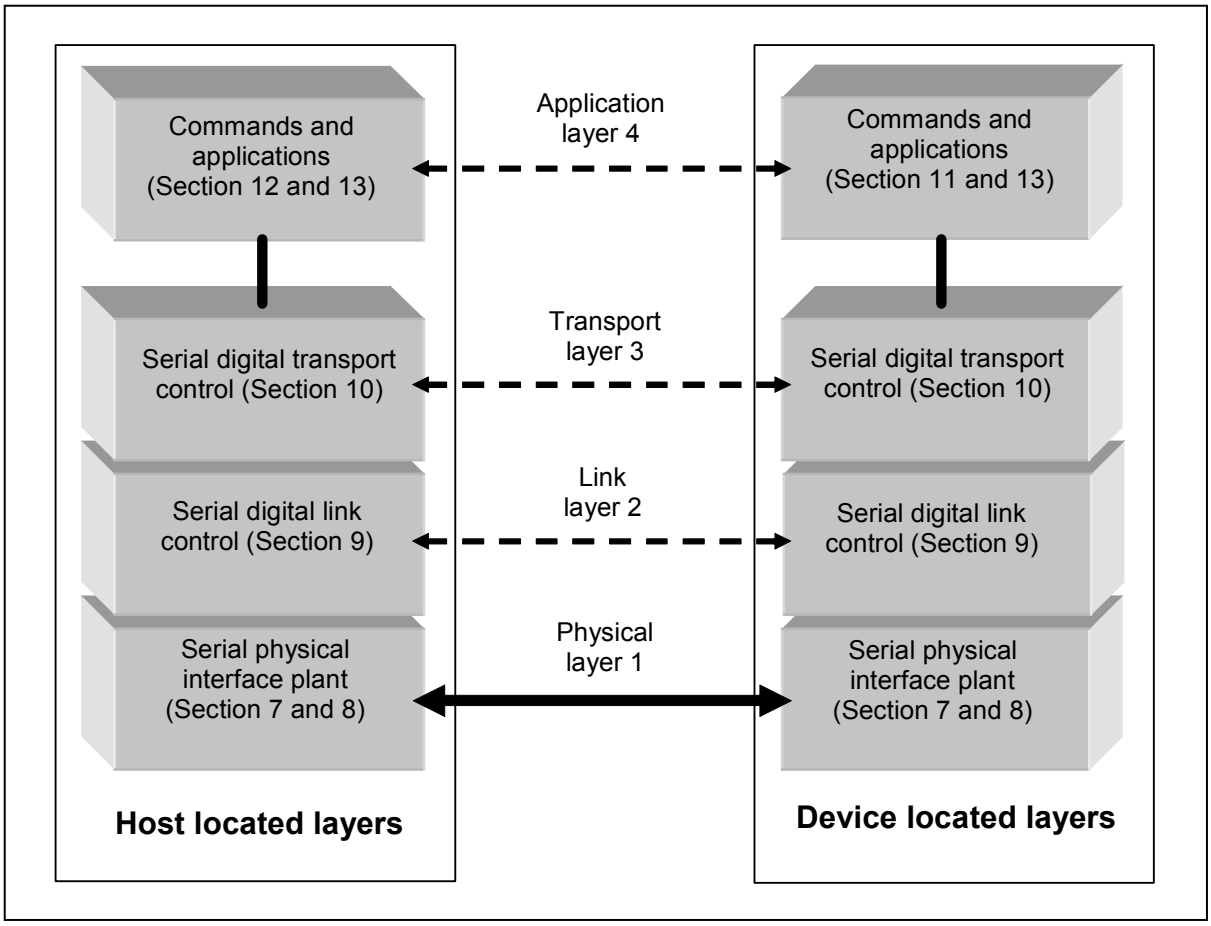


Figure 4 – Communication layers

The host may interact with the Application layer through a register interface that is equivalent to that presented by a traditional parallel ATA host adapter. When using parallel ATA emulation, the host software follows existing ATA/ATAPI-6 standards and conventions when accessing the register interface and follows standard command protocol conventions.

5.2 Usage Models

This section describes some of the potential applications of Serial ATA, including usage models that take advantage of features such as Native Command Queuing, enclosure management, Port Multipliers, and Port Selectors. Table 2 outlines the different usage models described throughout the section as well as highlights the relative requirements applicable to those usage models. Table 2 shows which characteristics in the first column are necessary to support the usage models in the top row.

Feature specific (FS) is intended to indicate that Gen1 is required but higher data rates are optional.

NOTE: The exact position of compliance points associated with the transmitter and the receiver are defined in sections 7.2.2.4 and 7.2.2.5.

Table 2 – Usage Model Descriptions

Characteristic	Internal 1 meter Cabled Host to Device	Short Backplane to Device	Long Back plane to Device	Internal 4-lane Cabled Disk Arrays	System to System Inter connects – Data Center Applications xSATA	System to System Inter connects – Data Center Applications xSATA	System to System Inter connects – External Desktop Applications eSATA	Proprietary Serial ATA Disk Arrays	Serial ATA and SAS	LIF-SATA
Use model section number	5.2.1	5.2.2	5.2.3	5.2.4	5.2.5	5.2.5	5.2.6	5.2.7	5.2.8	5.2.10
Cable and/or backplane type	Int SL	BP	BP	Int ML	Ext ML	Ext ML	Ext SL	BP and cable	BP	P
Cable length	<= 1 m			<= 1 m	<= 2 m	*subject to electrical requirements	<= 2 m			P
Cable Electrical	Table 18	P	P	Table 18	Table 20	Table 21	Table 19	Table 18	P	Table 18
Attenuation at 4.5GHz	-6dB	P	-16dB	-6dB	-8dB	-16dB	-8dB	-6dB	P	-6dB
Host-side connector	6.1.5	P	P	6.1.11 or 6.1.12	6.5.3 (key 7)	6.5.2 or 6.5.3 (key 1)	6.5.1	6.1.5 or P	SAS	6.4.3
Device-side connector	6.1.3.1	6.1.3.1	6.1.3.1 (inside canister)	6.1.3.1	6.5.3 (key 7)	6.5.2 or 6.5.3 (key 1 or 4)	6.5.1	6.1.3.1	6.1.3.1	6.4.2
Gen1i 1.5 Gbps	R	D (host to provide received signal)	D	R	NS	NS	NS	R	D	R
Gen1m 1.5 Gbps	NS	H	NS	NS	R (key 7)	NS	R	NS	NS	NS

Characteristic	Internal 1 meter Cabled Host to Device	Short Backplane to Device	Long Back plane to Device	Internal 4-lane Cabled Disk Arrays	System to System Inter connects – Data Center Applications xSATA	System to System Inter connects – Data Center Applications xSATA	System to System Inter connects – External Desktop Applications eSATA	Proprietary Serial ATA Disk Arrays	Serial ATA and SAS	LIF-SATA
Gen2i 3.0 Gbps	FS	D (host to provide received signal)	D	FS	NS	NS	NS	FS	D	FS
Gen2m 3.0 Gbps	NS	H	NS	NS	FS (key 7)	NS	FS	NS	NS	NS
Gen1x 1.5 Gbps	NS	NS	H, D	NS	NS	R	NS	NS	NS	NS
Gen2x 3.0 Gbps	NS	NS	H, D	NS	NS	FS	NS	NS	NS	NS
Gen3i 6.0 Gbps	FS	NS	NS	FS	NS	NS	NS	FS	D	FS
Hot plug support	NS	R	R	NS	R	R	R	R	R	NS

Key:
R – Required : configuration requires appropriate capabilities
FS – Feature specific : configuration is supported by specification but may be tied to an optional capability
NOTE: Feature specific is intended to indicate that Gen1 is required but higher data rates are optional.
NS – Not supported : configuration is not supported by definition in specification
P – Proprietary : implementation is vendor specific and not defined in specification
H – Host
D – Device
SL – single lane
ML – multi-lane
Int – Internal
Ext – External
BP – Backplane
NOTE : Many of the references in the table are section numbers or notations of clarification which do not require Key values

5.2.1 Internal 1 meter Cabled Host to Device

In this application, Gen1i, Gen2i or Gen3i electrical specifications compliant points are located at the Serial ATA mated connectors on both the host controller and device. The cable, specified in section 6, operates at 1.5 Gbps, 3.0 Gbps and 6.0 Gbps. The application may comply with the electrical hot plug specification described in section 7.2.5.

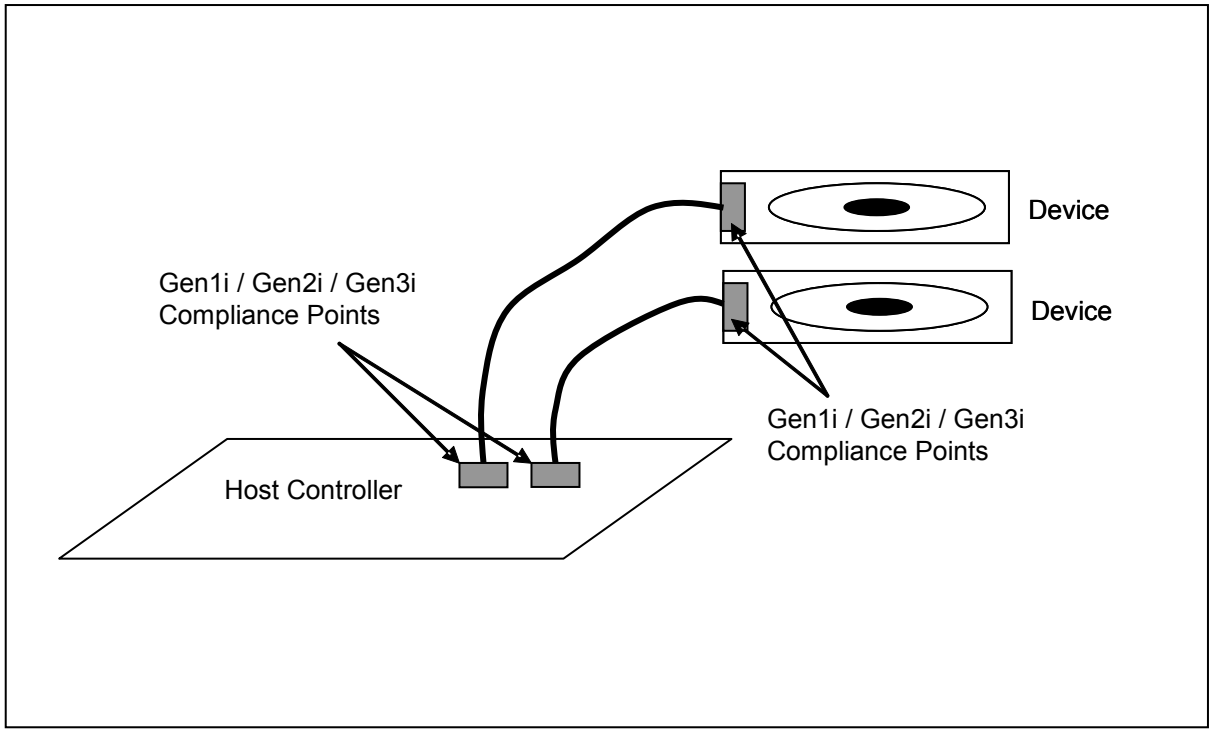


Figure 5 – Internal 1 meter Cabled Host to Device Application

5.2.2 Short Backplane to Device

In this application, Gen1i/2i disk devices within a small disk array are installed into drive canisters which are plugged into a “short” backplane. The host controller connecting to the Backplane shall contain a Host component which exceeds Gen1i/2i transmitter and receiver Differential Swings specifications so that the signals at the host controller connector complies with or exceeds the Gen1m/2m electrical specifications. All other electrical specifications at this compliance point shall meet Gen1i/2i specifications. Compliance points are located at the Serial ATA mated connectors on both the device (Gen1i/2i) and the host controller (Gen1m/2m). The signaling at the host controller connector may exceed the Gen1m/Gen2m transmit maximum providing the Gen1i/Gen2i receiver maximum is not exceeded at the device connector. The application does not contain a cable but routes Serial ATA signals on printed circuit board at 1.5 Gbps and 3.0 Gbps where it is anticipated that the backplanes attenuate signals more than the compliant copper cable described in section 6. The burden falls on the host controller to increase transmit signal swing and accommodate smaller receive swings at the host controller connector. The Gen1i/2i devices are not required to comply with the electrical hot plug specification described in section 7.2.5. However, there are practical application benefits from complying with the hot plug electrical specifications.

NOTE: At Gen2 speeds the designer faces significant challenges regarding signal integrity issues. Validation/feasibility data at Gen2 speeds has not been provided. Making this work is up to the system designer. Using a laboratory load to measured the host controller amplitude or jitter at the device connector and comparing with

Table 33 is not appropriate. Additional margin is required due to the non-ideal impedance match of transmitter and receiver. The receiver is tested to work at these amplitudes and jitter levels when the signal is applied from a lab-sourced signal.

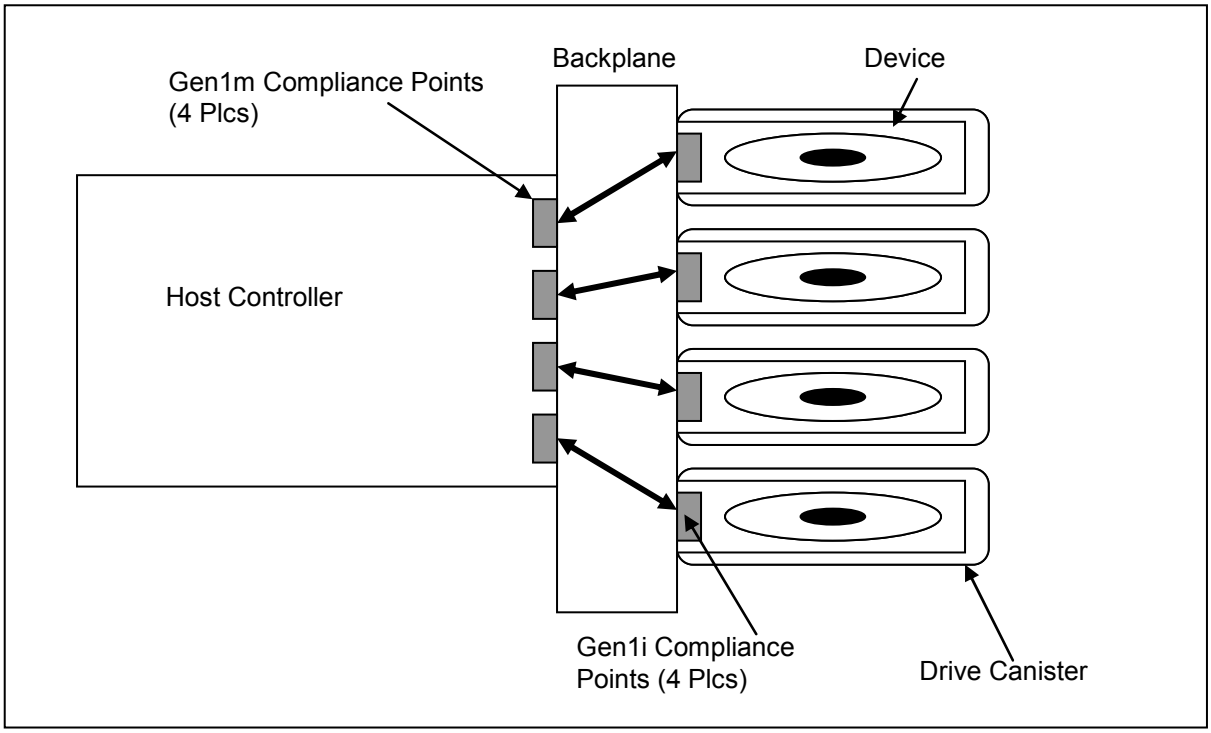


Figure 6 – Short Backplane to Device Application

5.2.3 Long Backplane to Device

In this 1.5 Gbps or 3.0 Gbps application, the length of the backplane is longer than in the previous example so attenuation of signals reduce their amplitude beyond usable levels. Therefore an IC may be placed between the device and the canister's connector to the backplane to convert the disk's Gen1i/Gen2i levels to Gen1x/Gen2x levels. A typical circuit might be a Serial ATA Port Selector. Likewise, the host controller complies with Gen1x/Gen2x levels at the backplane connector to the host controller. This allows reliable transmission of Serial ATA data over backplanes longer than in the Short Backplane Application described above. The burden of determining whether a Backplane Application is Short or Long falls upon the system designer based on the system's attenuation. Compliance points are located at the Serial ATA mated connectors on both the canister (Gen1x/Gen2x) and the host controller (Gen1x/Gen2x). The Gen1x/Gen2x Electrical Specifications require Hot Plug capability so that the disk/canister may be plugged and unplugged without damage.

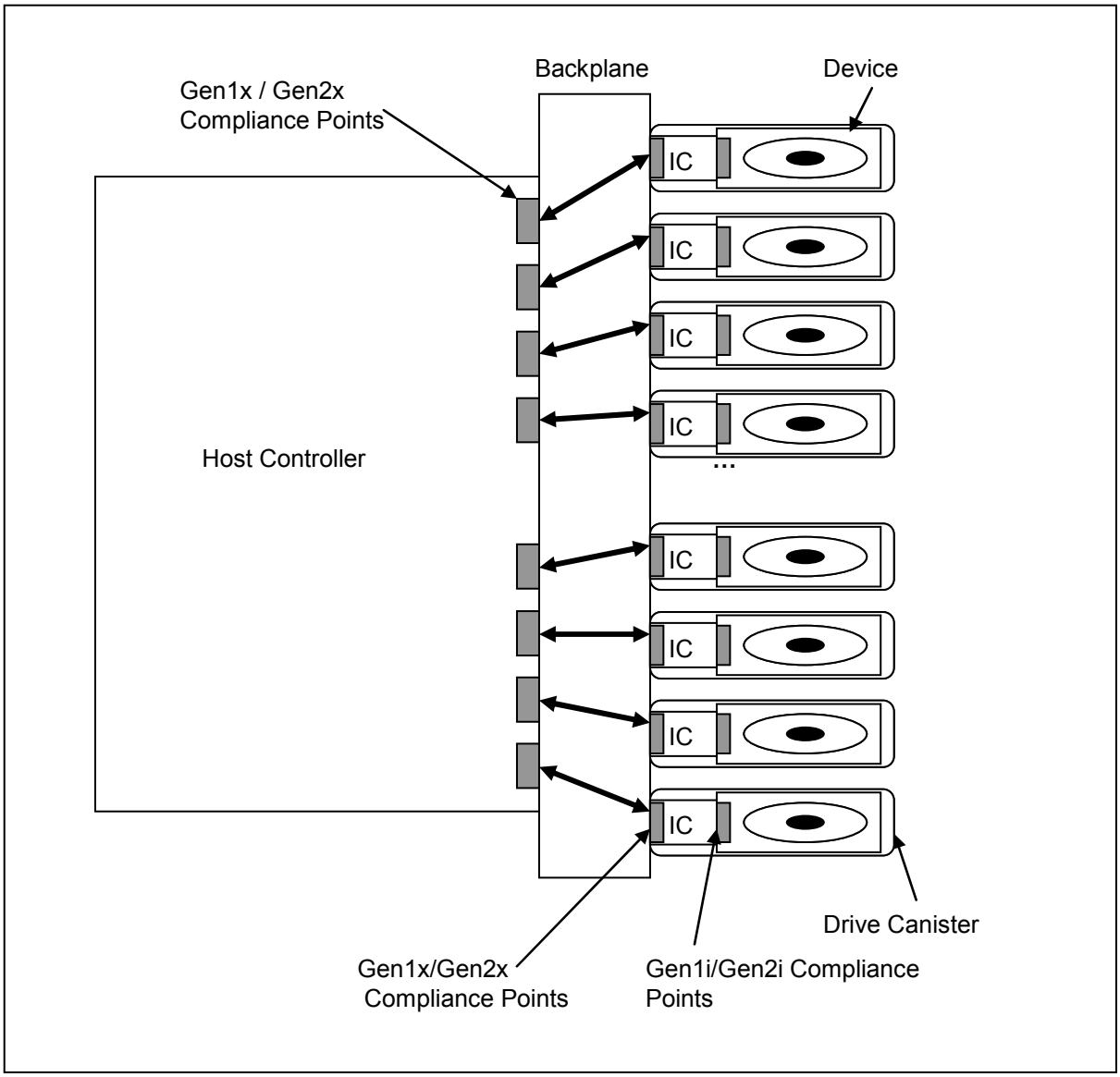


Figure 7 – Long Backplane to Device Application

5.2.4 Internal 4-lane Cabled Disk Arrays

In this application, Gen1i, Gen2i and Gen3i Serial ATA devices are connected to the host controller via the internal 4-lane cable solution. Gen1i, Gen2i and Gen3i specifications shall be met at each end of the 4-lane cable mated interface. The internal 4-lane cable shall connect to the device by one of the following two approaches.

- The host end of the internal 4-lane solution shall mate directly to the host controller board. The device end shall consist of four individual single lane Serial ATA cables for direct mate with up to four individual Serial ATA devices.
- The host end of the internal 4-lane solution shall mate directly to the host controller board. The device end of the internal 4-lane cable shall consist of a single internal 4-lane connector mated to a backplane which provides individual connection points to up to four Serial ATA devices. The backplane design is proprietary.

In the second solution, attenuation of the backplane reduces signal amplitude below specification limits at the compliance points. An IC, such as a Port Selector, shall be placed between the device and the internal 4-lane cable mated interface. Gen1i, Gen2i and Gen3i specifications shall be met at each end of the 4-lane cable mated interface and the device mated connector interface.

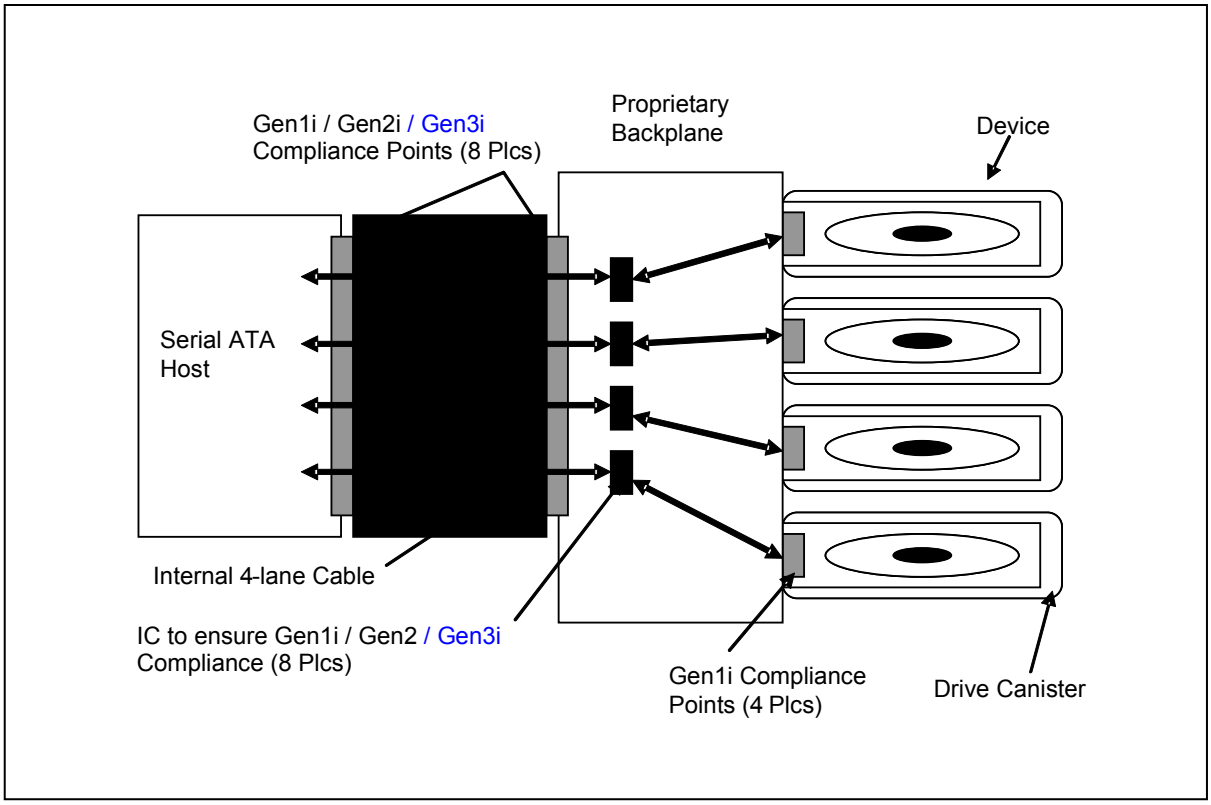


Figure 8 – Internal 4-lane Cabled Disk Array

NOTE: For Gen2i and Gen3i devices the link from the host to the drive (through the IC on the backplane) does not have an electrical specification for the “delivered signal” to the drive. Consider the compliance point for the backplane-drive connector (through the mated pair). If the signal at the compliance point, into a lab load meets all the requirements of Table 29 and

Table 31, and the backplane meets all the requirements of Table 30, then interoperability is confirmed.

However, it is anticipated that the additional trace length on the backplane between the IC and the backplane-device connector makes compliance to these specifications difficult. The burden for ensuring interoperability with all Gen1i/Gen2i devices falls upon the implementer of the system. Essentially the implementer would use simulations and empirical results to confirm that the compliance point at the backplane-device connector is equivalent to or better than a compliant host and cable combination.

5.2.5 System-to-System Interconnects – Data Center Applications (xSATA)

This application is defined as external storage applications that require more than one serial link between systems. This application uses the external Multilane cables defined in sections 6.5.2 and 6.5.3, and may be referred to as xSATA.

For system-to-system interconnects that require cables of approximately two meters or less (i.e. pedestal to pedestal, blade to blade, or intrarack connections) either Gen1m/Gen2m or Gen1x/Gen2x Electrical Specifications are used. For system-to-system interconnects that require distances greater than two meters (i.e. rack to rack,) Gen1x/Gen2x Electrical Specifications are used.

All external Serial ATA cables function at both 1.5 Gbps and 3.0 Gbps. Use of a cable that operates at 1.5 Gbps but not at 3.0 Gbps is allowed but this cable assembly shall not be interchangeable with standard Serial ATA cables. For example, if this 1.5 Gbps cable uses Serial ATA specified connectors, the cable shall be keyed to insure that it cannot plug into standard Serial ATA connections. Hot pluggability is a requirement in this application.

Compliance points are located at the mated bulkhead connectors of each system as shown in the example implementation in Figure 9. If a SATA endpoint device is included in the system, it shall not be connected directly to the external connectors. Instead, a bridge shall be used between the external connectors and the endpoint device. Some examples of bridges include repeaters/retimers, Port Multipliers, and RAID controllers.

NOTE: Interconnects between Gen1m/Gen2m and Gen1x/Gen2x systems are not allowed.

NOTE: Gen3 is not defined for xSATA.

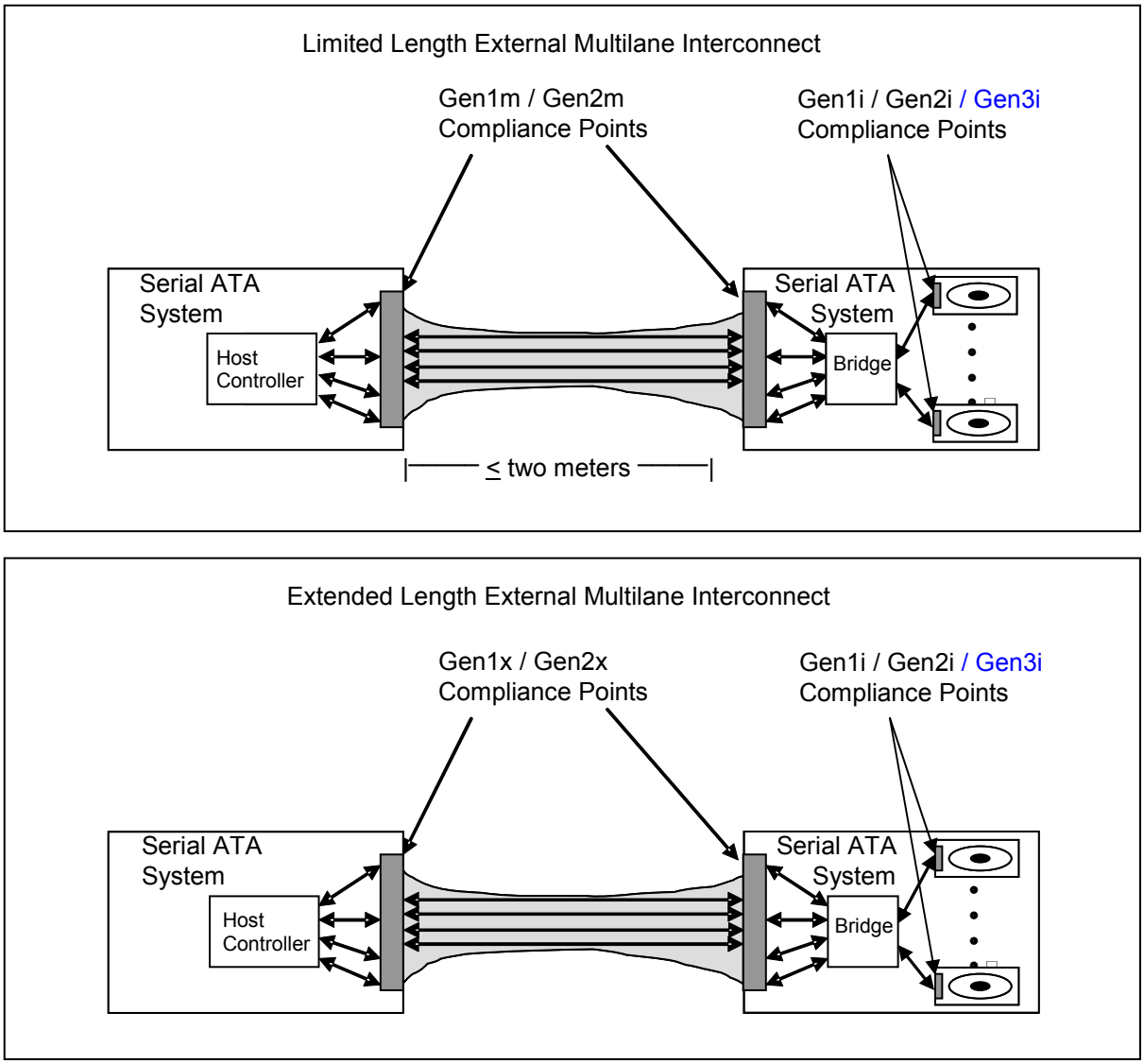


Figure 9 – System-to-System Data Center Interconnects

5.2.6 System-to-System Interconnects – External Desktop Applications (eSATA)

This application is aimed at external storage applications that require a single lane with approximately two meters or less of cable length. This application uses the external single lane connector system defined in section 6.5.1, and may be referred to as eSATA.

NOTE: Gen3i is not defined for eSATA.

In the example shown below, a device enclosure contains a Gen1i / Gen2i device and an interposer card which contains an integrated circuit unless the device is specifically designed for external connection. Regardless of the implementation, the outside of the device enclosure shall meet Gen1m specifications when operating at Gen1 speeds and Gen2m specifications when operating at Gen2 speeds. A cable/connector has been defined for this application which operates at both Gen1 and Gen2. The host system has an external connector which meets Gen1m specifications when operating at Gen1 speeds and Gen2m specifications when operating at Gen2 speeds.

The entire system shall meet the following requirements:

- The cable/connector shall operate at Gen1 and Gen2 speeds. Systems shall not deploy any cable which cannot operate at Gen2 speeds when the host system and device enclosure both comply with Gen2m electrical specifications.
- The host system and device enclosure shall comply with Gen1m specifications when operating at Gen1 speeds.
- The host system and device enclosure may operate at Gen2 speeds. However, if they operate at Gen2 speeds the host system and device enclosure shall comply with Gen2m specifications when operating at Gen2 speeds and shall also be able to operate at Gen1 speeds using Gen1m electrical specifications.
- The host system and the device enclosure shall comply with the Hot Plug Specifications in this document.
- NOTE: AC Coupling on the transmitters and receivers of the host system and device enclosure is strongly recommended for Gen1m and required for Gen2m.

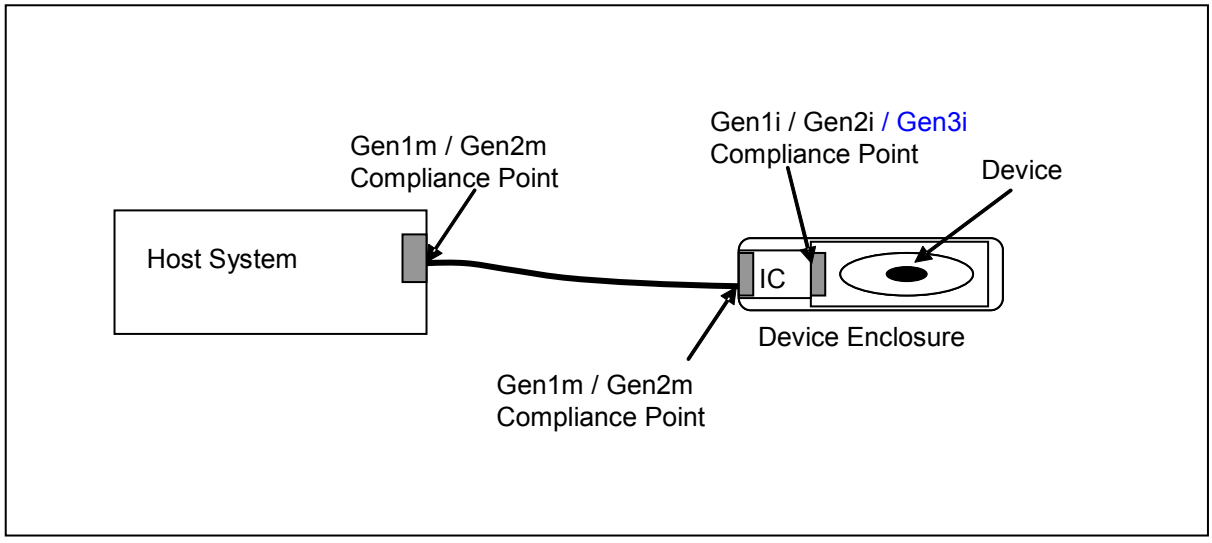


Figure 10 – External Desktop Application

5.2.7 Proprietary Serial ATA Disk Arrays

In this application, Serial ATA devices are connected to a backplane and the links are routed over a combination of internal backplanes or cables as well as an external cable to a Serial ATA system. There are not semiconductors between the devices and the system so the intermediate connectors are not compliance points. Although this application is allowed, the external connectors on the disk array shall not be standard Serial ATA connectors. This is to prevent users from connecting standard external cables between the system and the disk array since they may not function reliably.

NOTE: The designer faces significant challenges regarding signal integrity issues because of complexity of the interface, which may require additional margin. Validation/feasibility data has not been provided. Making this work is up to the system designer. Using a laboratory load to measured the host controller amplitude or jitter at the device connector and comparing with Table 33 is not appropriate. The receiver is tested to work at these amplitudes and jitter levels when the signal is applied from a lab-sourced signal.

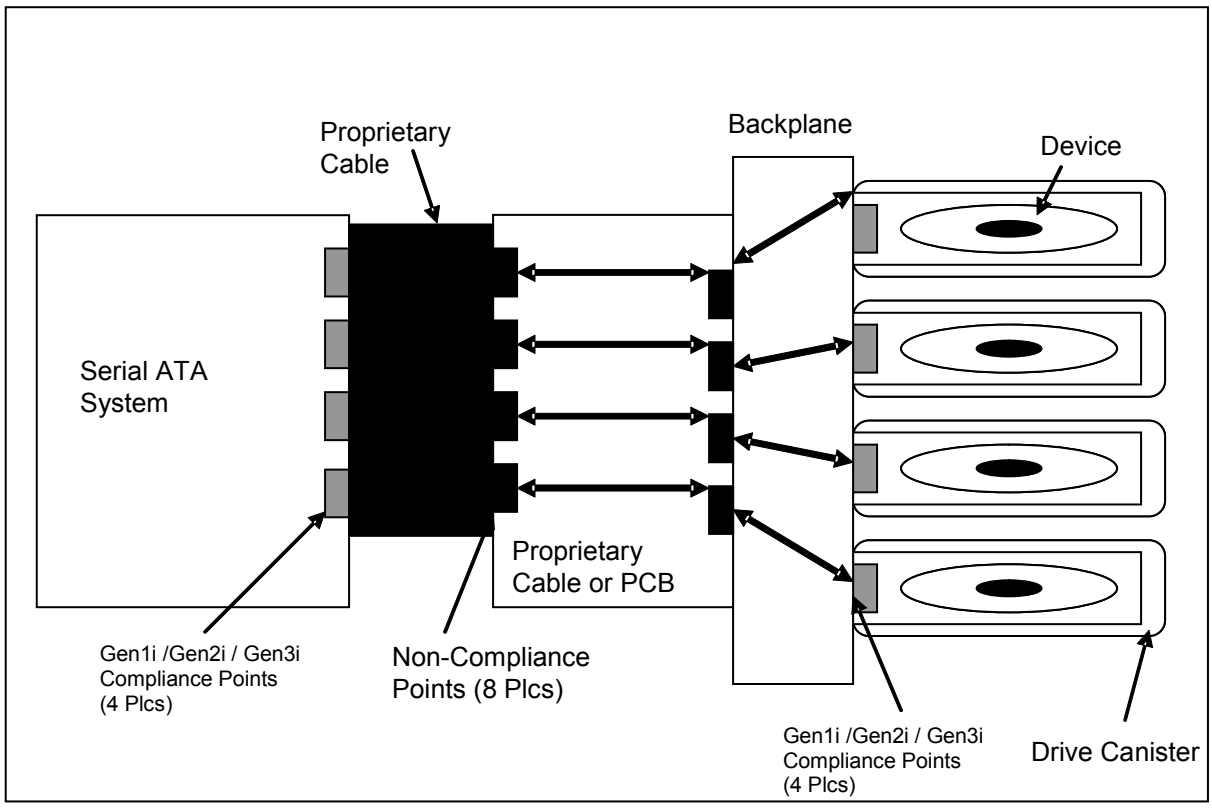


Figure 11 – SATA Disk Arrays

5.2.8 Serial ATA and SAS

The SAS reference is a standard that specifies a SCSI transport protocol over a serial link. The SAS standard borrows heavily from the SATA Phy, Link, and Transport layers. A SAS domain may support attachment to and control of unmodified SATA devices connected directly into the SAS domain using the Serial ATA Tunneled Protocol (STP).

5.2.9 Potential External SATA Incompatibility Issues

WARNING: The functionality of External Desktop and External Data Center cabled applications is not defined by this specification. Consequently, two systems could be connected which may not interoperate even though all the components comply with the electrical specifications defined in this document. External applications are not required to support Port Multipliers or Port Selectors. As a result, a host with an External Data Center connector could be connected to a disk array containing a Port Multiplier and the resulting system may not operate correctly.

5.2.10 Mobile Applications

Applications and compliance points for Serial ATA devices within or connected to mobile computers are not defined in this document, except for embedded applications. If any proprietary cables/connectors or electrical specifications are developed for this application, the system shall be designed so as to prevent connection with standard SATA components. If standard cables/connectors/electrical interfaces are used within the mobile computer, within the docking bay or to external storage components, these shall comply with the applicable requirements in this specification and interoperate properly with Serial ATA components.

Internal Applications:
It is expected that all internal interfaces comply with the Gen1i, Gen2i, and/or Gen3i specifications. Any mobile computer designer modifying electrical specifications of hosts and devices within the mobile computer is free to do so, however, all proprietary interfaces shall be designed so as to prevent connection with standard SATA components.

Docking Bay Applications:
Proprietary docking bay interfaces shall be designed so as to prevent connection with standard SATA components.

External Applications:
Applications for external Serial ATA interfaces on mobile computers may use either the External Desktop cable/connector (Gen1m/Gen2m) or the System-to-System Data Center cable/connector (Gen1m/Gen2m or Gen1x/Gen2x). Proprietary solutions shall be designed so as to prevent connection with standard SATA components.

Embedded Applications

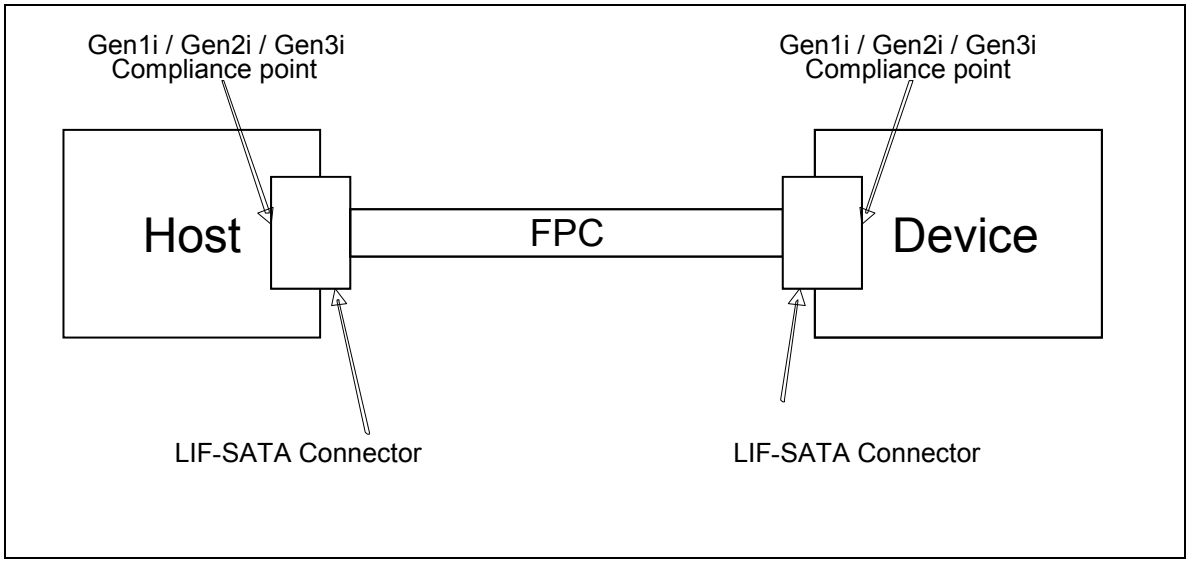


Figure 12 - Embedded LIF-SATA Application

In this application, the device is connected to a host controller via a flexible printed circuit cable (FPC). The FPC interfaces shall comply with Gen1i electrical requirements listed in Table 18. The compliance points are shown in Figure 12. Gen2i or Gen3i compliance is feature specific, but if implemented, shall also comply with the electrical requirements listed in Table 18.

5.2.11 Port Multiplier Example Applications

One possible application of the Port Multiplier is to increase the number of Serial ATA connections in an enclosure that does not have a sufficient number of Serial ATA connections for all of the devices in the enclosure. An example is shown in Figure 13. A Multilane cable with two Serial ATA connections is delivered to the enclosure. The enclosure contains eight Serial ATA devices. To create the appropriate number of Serial ATA connections, two 1-to-4 Port Multipliers are used to create eight Serial ATA connections.

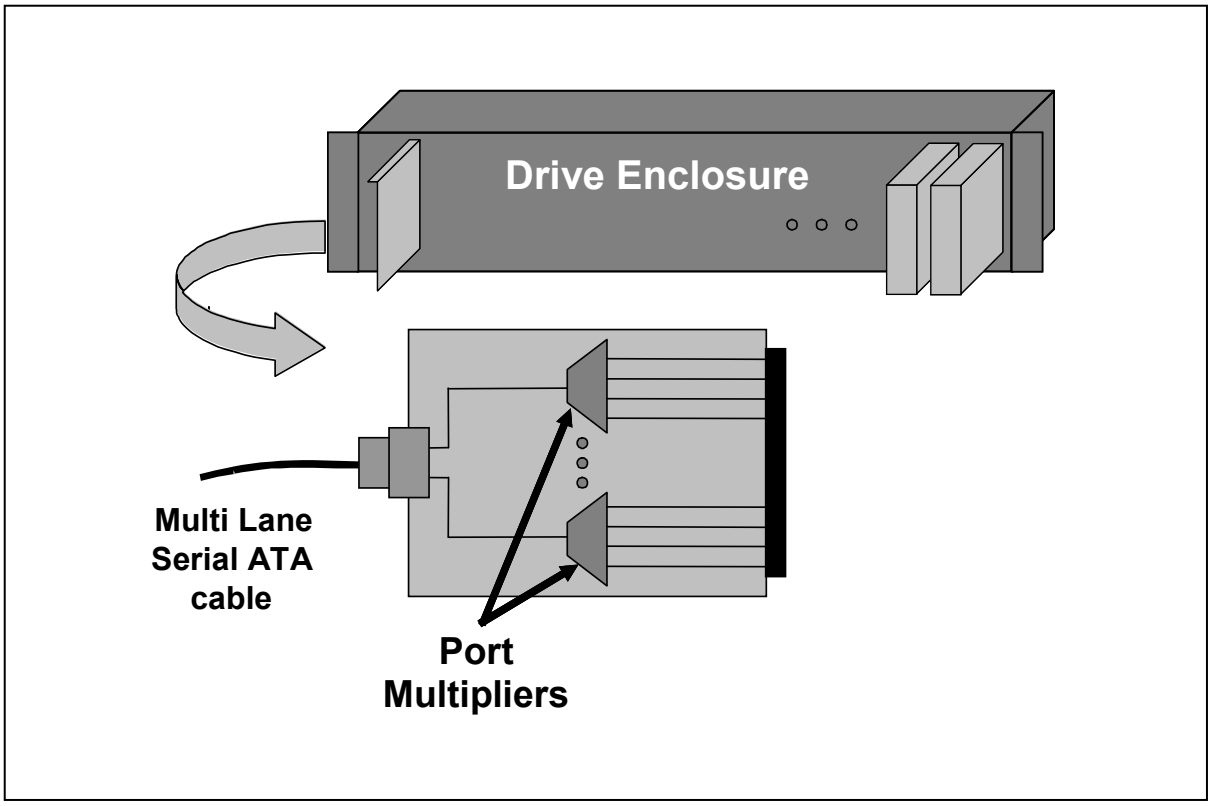


Figure 13 – Enclosure example using Port Multipliers with Serial ATA as the connection within the rack

Another example is shown in Figure 14. Fibre Channel, Infiniband, or Gigabit Ethernet is used as the connection within the rack to the enclosure. Inside the enclosure, a host controller creates two Serial ATA connections from the connection delivered. The enclosure contains eight Serial ATA devices. To create the appropriate number of Serial ATA connections, two 1-to-4 Port Multipliers are used to create eight Serial ATA connections.

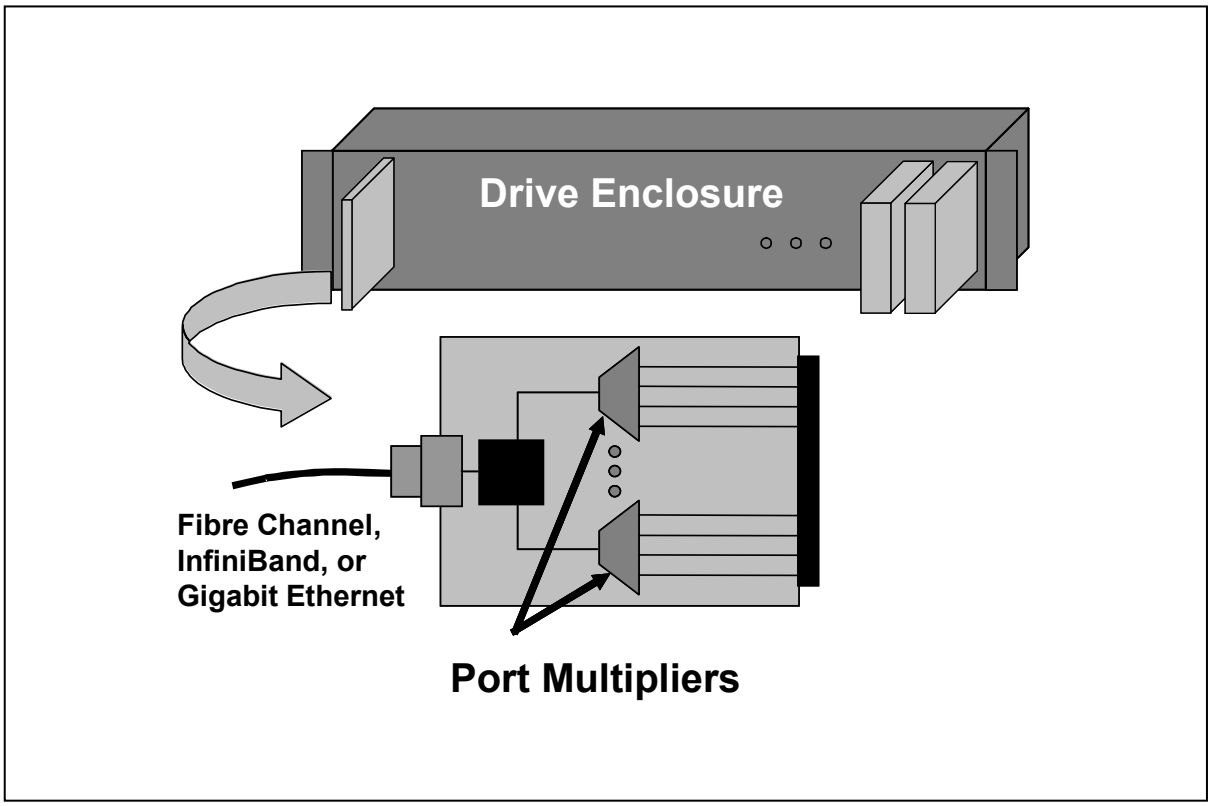


Figure 14 – Enclosure example using Port Multipliers with a different connection within the rack

The Port Multiplier allows host controllers with a modest number of connections to be used in these enclosures and then the connectivity is increased as product requirements dictate.

Another example application is using a Port Multiplier to increase the number of Serial ATA connections in a mobile docking station. The example shown in Figure 15 has a proprietary interface between the laptop and the docking station. The proprietary interface may route a Serial ATA connection from the laptop to the docking station or the docking station may create a Serial ATA connection itself. The docking station routes the Serial ATA connection to a Port Multiplier to create an appropriate number of Serial ATA connections for the number of devices to be attached.

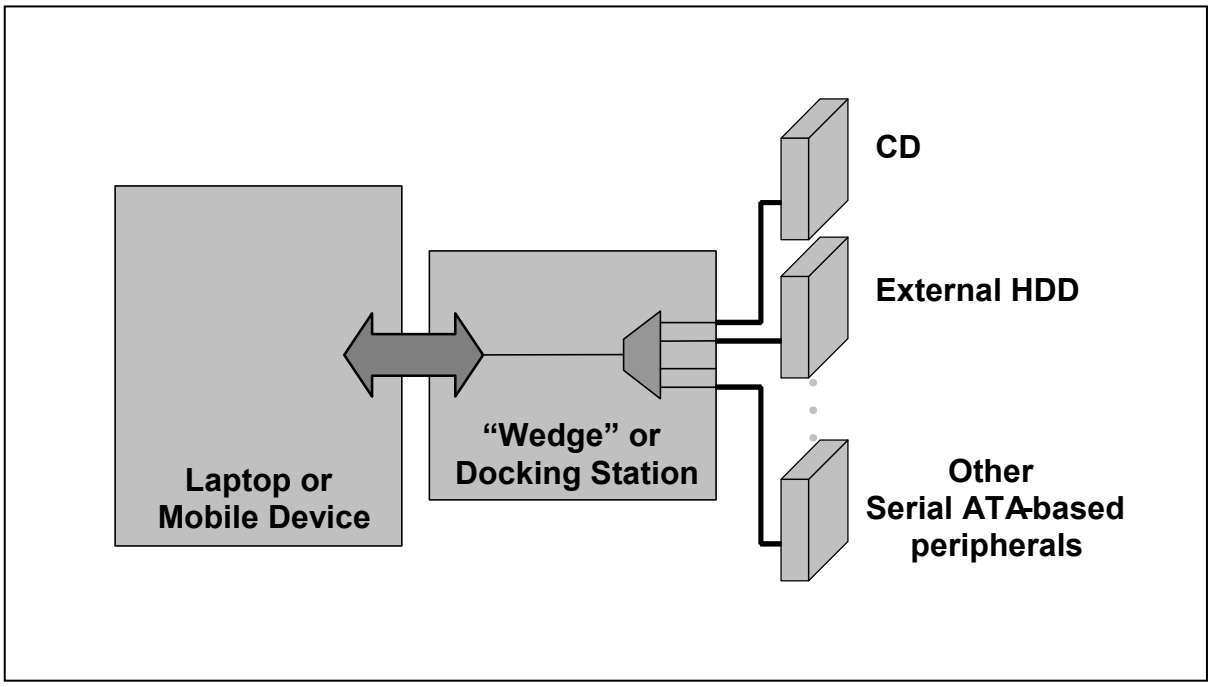


Figure 15 – Mobile docking station example using a Port Multiplier

These are a few examples of possible applications of the Port Multiplier and are not meant to be all encompassing.

6 Cables and Connectors

This section defines the connectors and cable assemblies for Serial ATA. It specifies:

- The mating interfaces between the connectors
- The connector location on the Serial ATA device
- The electrical, mechanical and reliability requirements of the connectors and cable assemblies

The mating interfaces of Serial ATA connectors are defined in terms of their front end (i.e. separable) characteristics only. All SATA internal and external connector contact mating areas shall have a gold or gold-compatible finish. Unless otherwise specified, connector back end characteristics including finish, PCB mounting features, and cable termination features are not defined.

6.1 Internal cables and connectors

6.1.1 Internal Single Lane Description

A Serial 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 16, is inserted directly into a backplane connector, illustrated as (g) in Figure 16. The device plug connector and the backplane 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 16, mates with the signal cable receptacle connector on one end of the cable, illustrated as (c) in Figure 16. 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 16. The signal cable wire consists of two twinax sections in a common outer sheath.

Besides the signal cable, 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 16, on one end and may be directly connected to the host power supply on the other end or may include a power cable receptacle on the other end. The power cable receptacle connector on one end of the power cable mates with the device power plug connector, shown as (b) in Figure 16. The other end of the power cable is attached to the host as necessary.

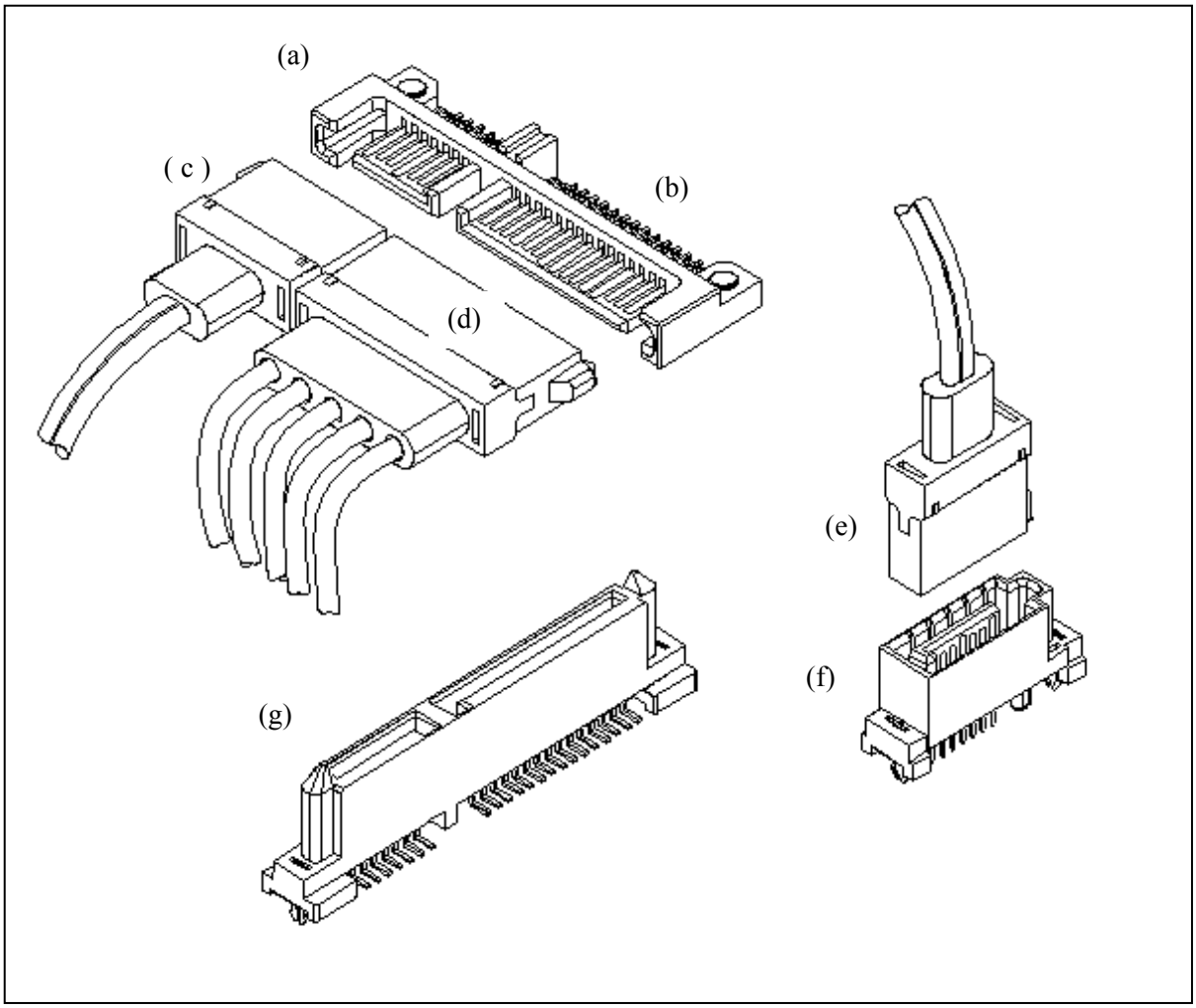


Figure 16 – Serial ATA connector examples

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) backplane connector mating directly with device plug connector (a) & (b).

6.1.1.1

Figure 17 shows a direct cable / connector connection and highlights the signal path of the differential TX and RX pairs.

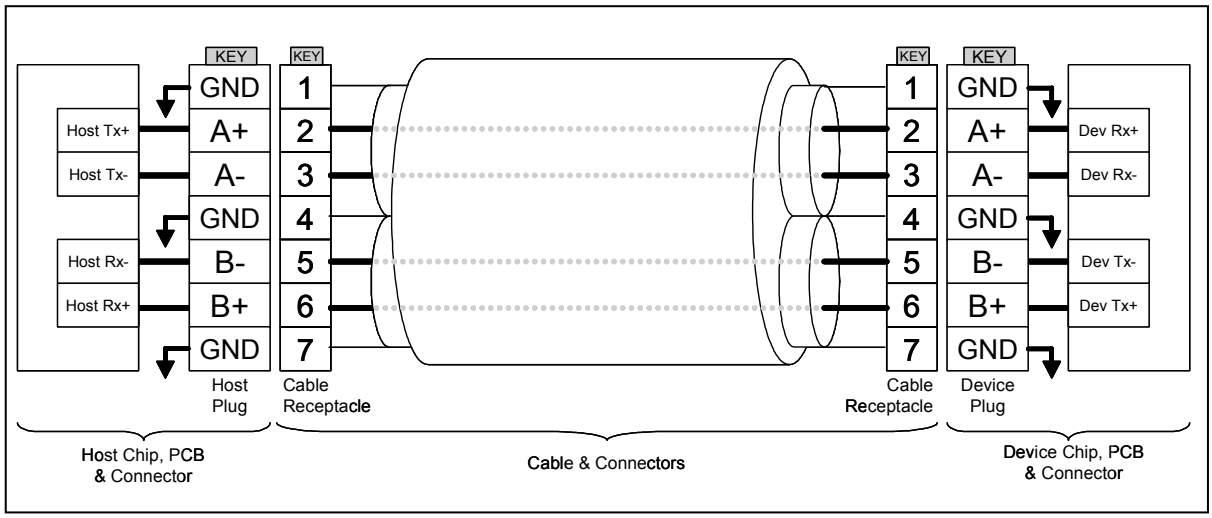


Figure 17 – SATA Cable / Connector Connection Diagram

The connector on the left represents the Host with TX/RX differential pairs connected to a cable. The connector on the right shows the Device with TX/RX differential pairs also connected to the cable. Notice also the ground path connecting the shielding of the cable to the Cable Receptacle.

Figure 18 shows the connection between host and device as a direct connection. It is similar to the cable/connector connection with the exception of the cable.

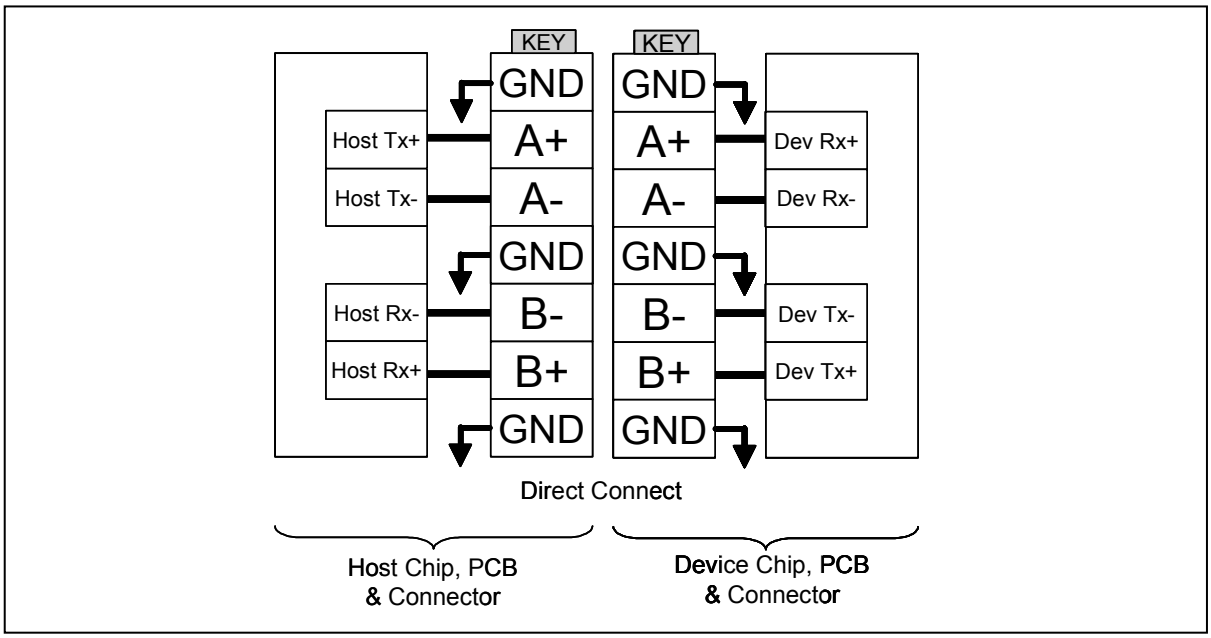


Figure 18 – SATA Host / Device Connection Diagram

In both cases the connection of the TX differential signal pair on the host side to the RX differential signal pair on the device side. A similar connection of the host RX pair to the device TX pair is also shown.

6.1.2 Connector locations

The device connector location is defined to facilitate blind mating.

Figure 19 and Figure 20 define the connector location on 5.25” devices. Optical devices shall locate the connector as indicated in the optical device connector location. Non-optical devices should locate the connector as indicated in the optical device connector location but may locate the connector as indicated in the non-optical device alternate connector location. The Serial ATA connector is nominally flush to the end of the device factor.

Figure 21 defines the connector location on side mounted 3.5” devices. Figure 22 defines the connector location on bottom mounted 3.5” devices. Refer to EIA-740 and SFF-8301 for 3.5” device form factor specifications. The Serial ATA connector is nominally flush to the end of the device factor.

Figure 23 defines the connector location on side mounted 2.5” devices. Figure 24 defines the connector location on bottom mounted 2.5” devices. Refer to EIA-720 and SFF-8201 for 2.5” device form factor specifications. The Serial ATA connector nominally protrudes 0.3 mm from the end of the device factor.

Figure 25 defines the Serial ATA connector location on side mounted 1.8” devices. Figure 26 defines the Serial ATA connector location on bottom mounted 1.8” devices. Refer to SFF-8111 for 1.8” device form factor specifications. The Serial ATA connector nominally protrudes 0.3 mm from the end of the device factor.

To ensure mating of devices to backplanes with proper mechanical and electrical interface and without physical conflict, Figure 38 illustrates the fully mated condition of the device to a nominally

flush backplane receptacle and Figure 27 defines the keep out zones for devices of all form factors. The application shall ensure that these areas do not contain any materials or construction that prevents the fully mated condition.

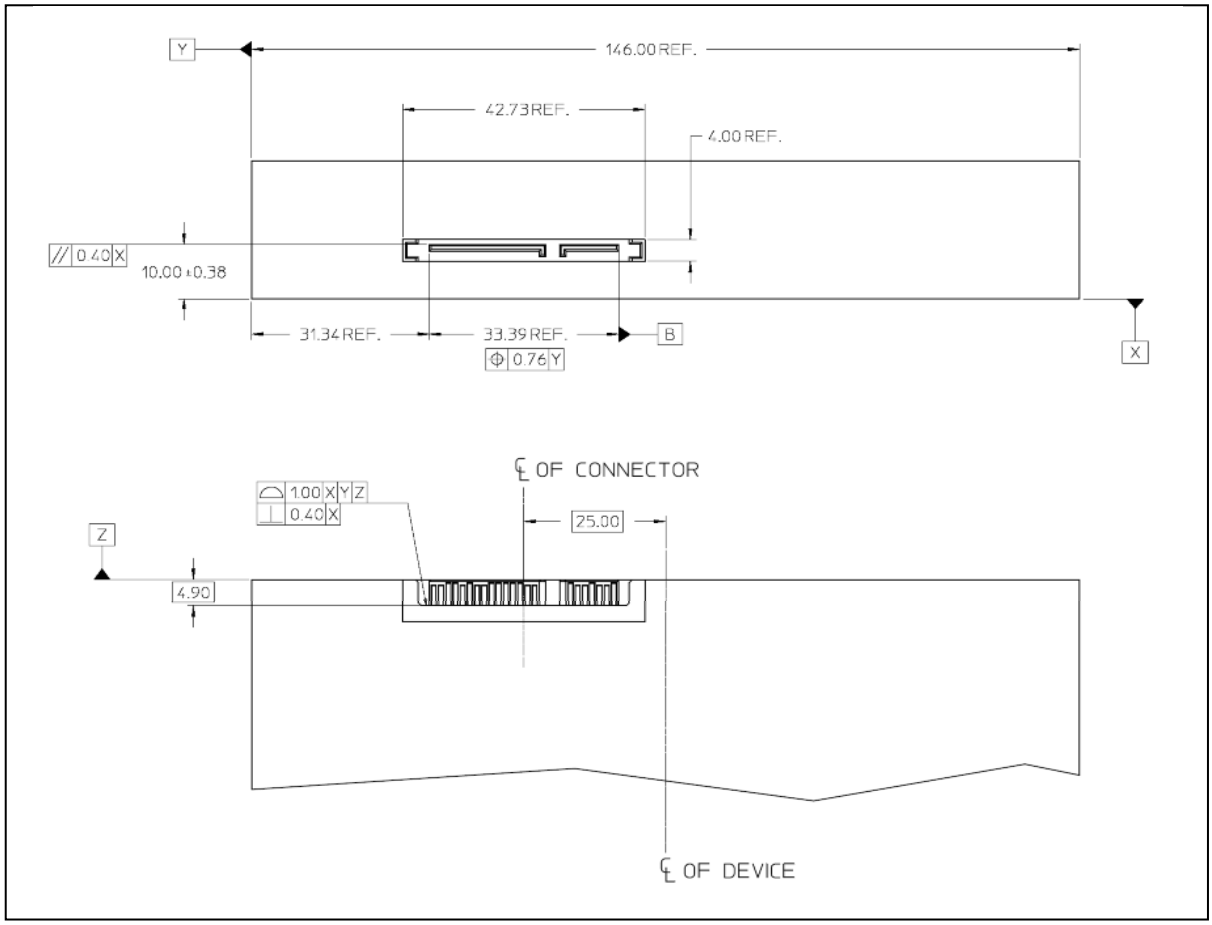


Figure 19 – Optical Device Plug Connector Location on 5.25" form factor

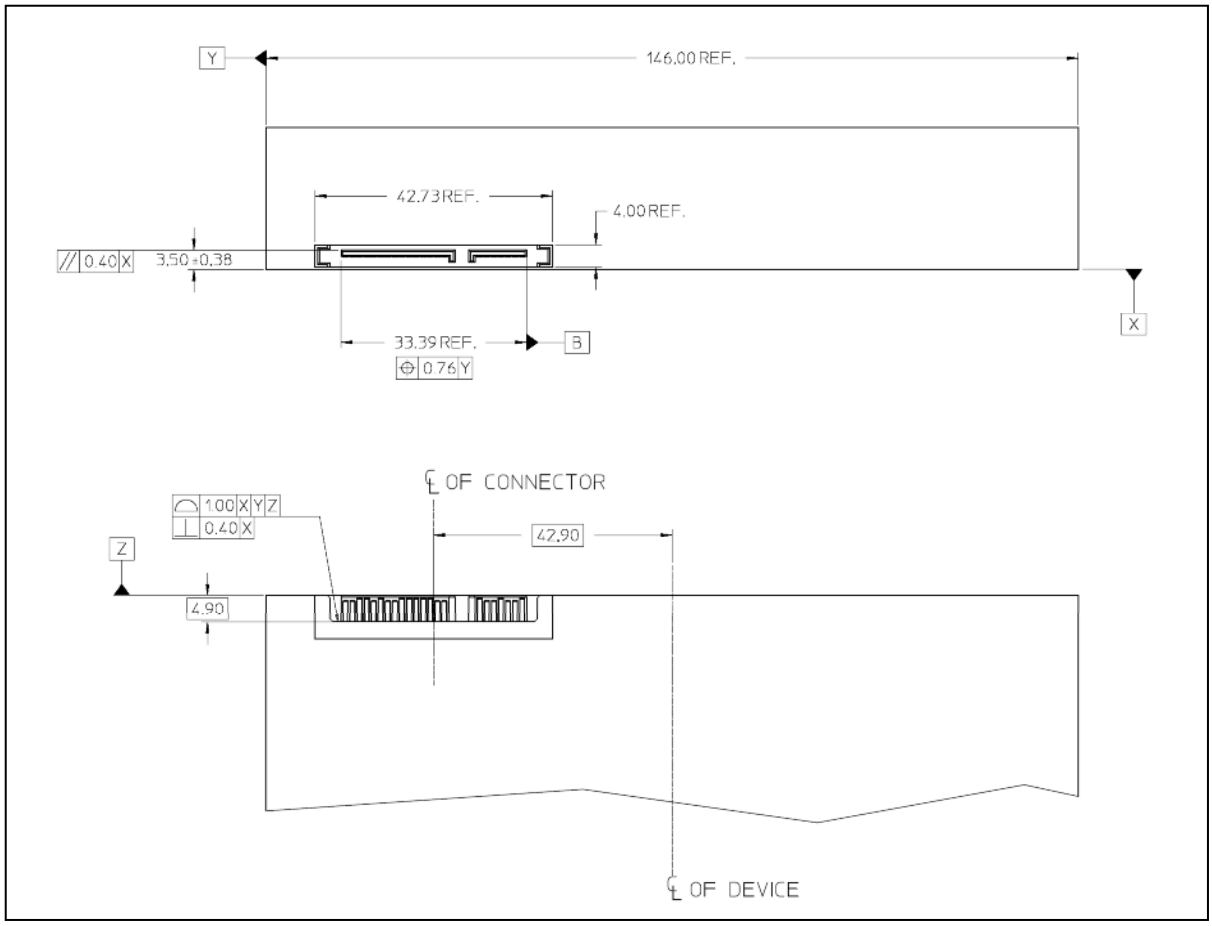


Figure 20 – Non-Optical Alternate Device Plug Connector Location on 5.25" form factor

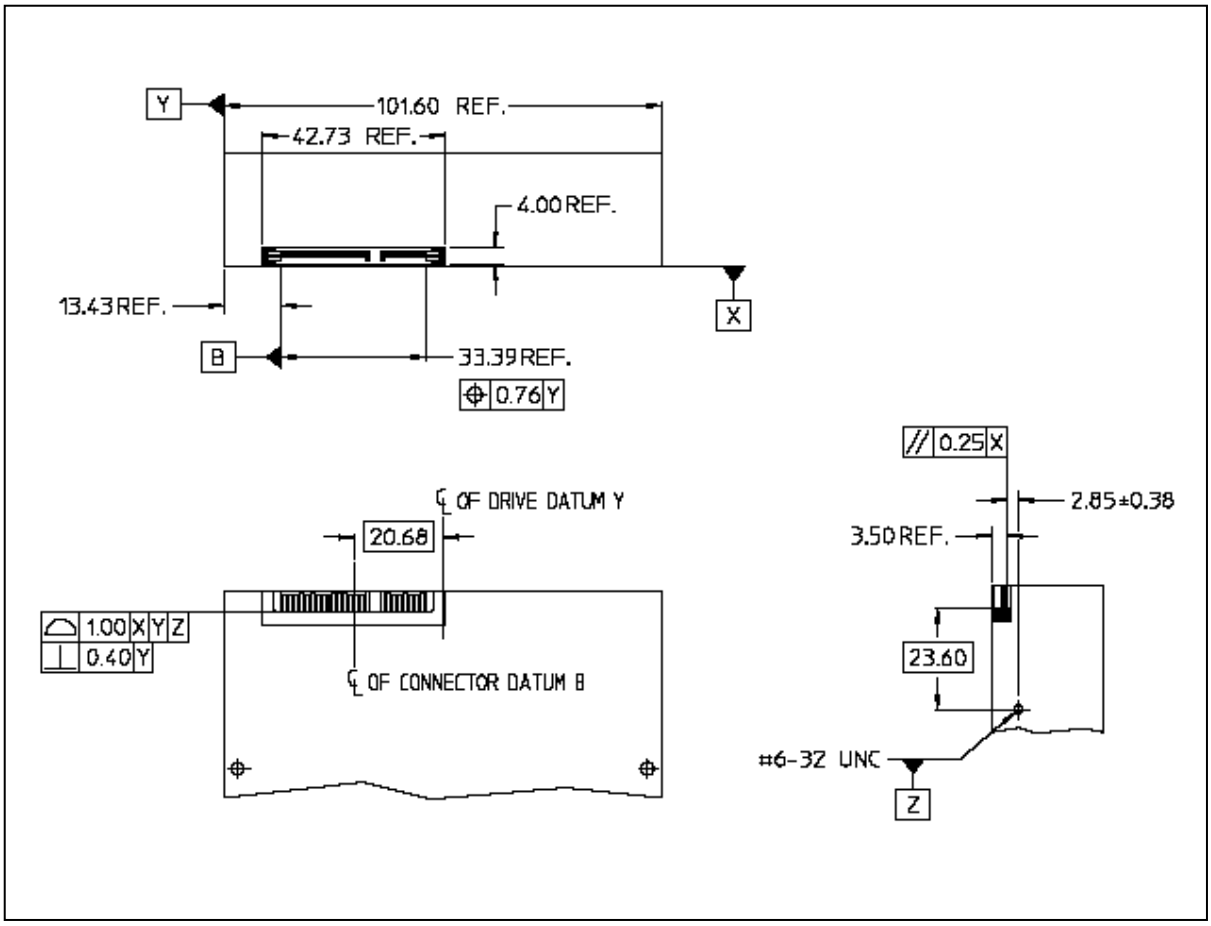


Figure 21 – Device Plug Connector Location on 3.5” Side Mounted Device

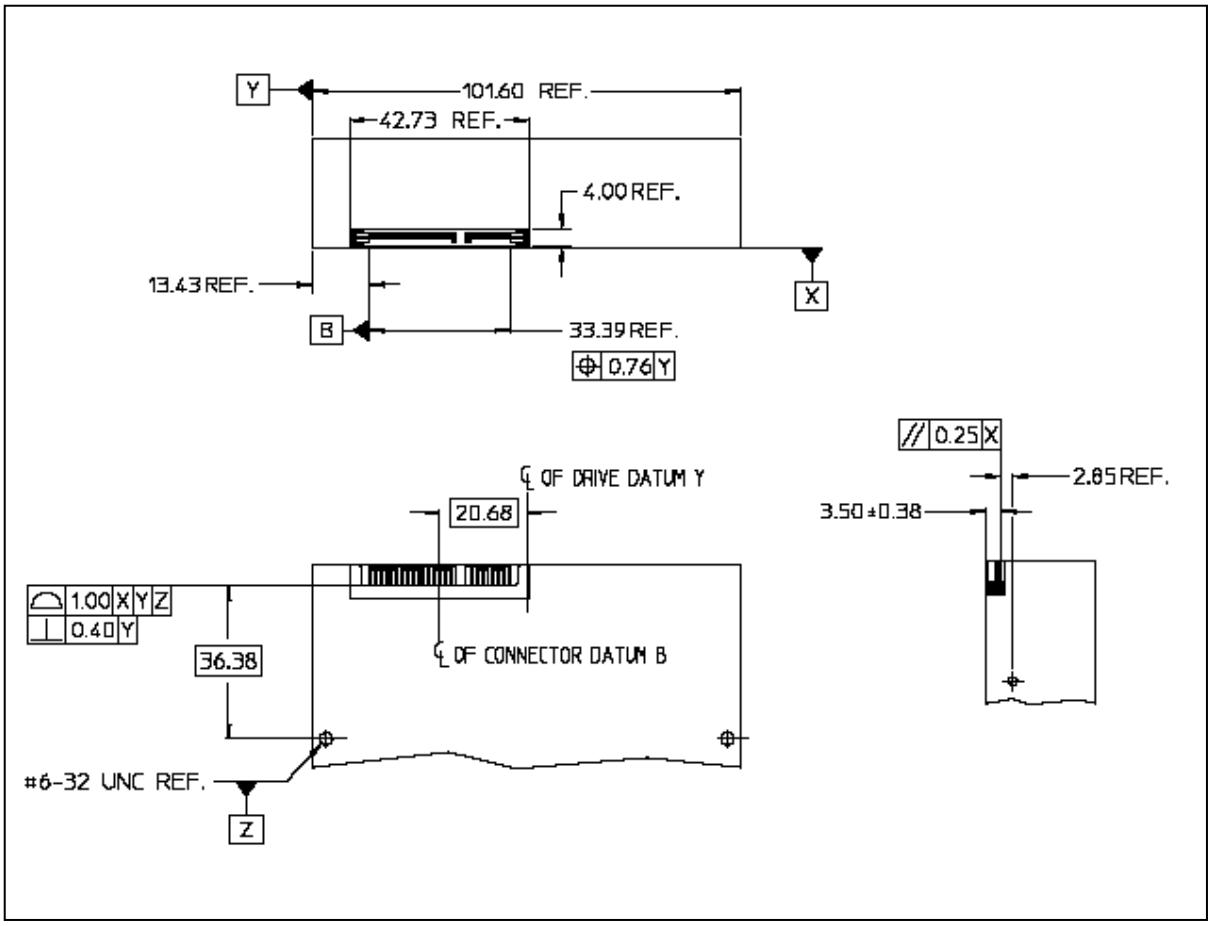


Figure 22 – Device Plug Connector Location on 3.5" Bottom Mounted Device

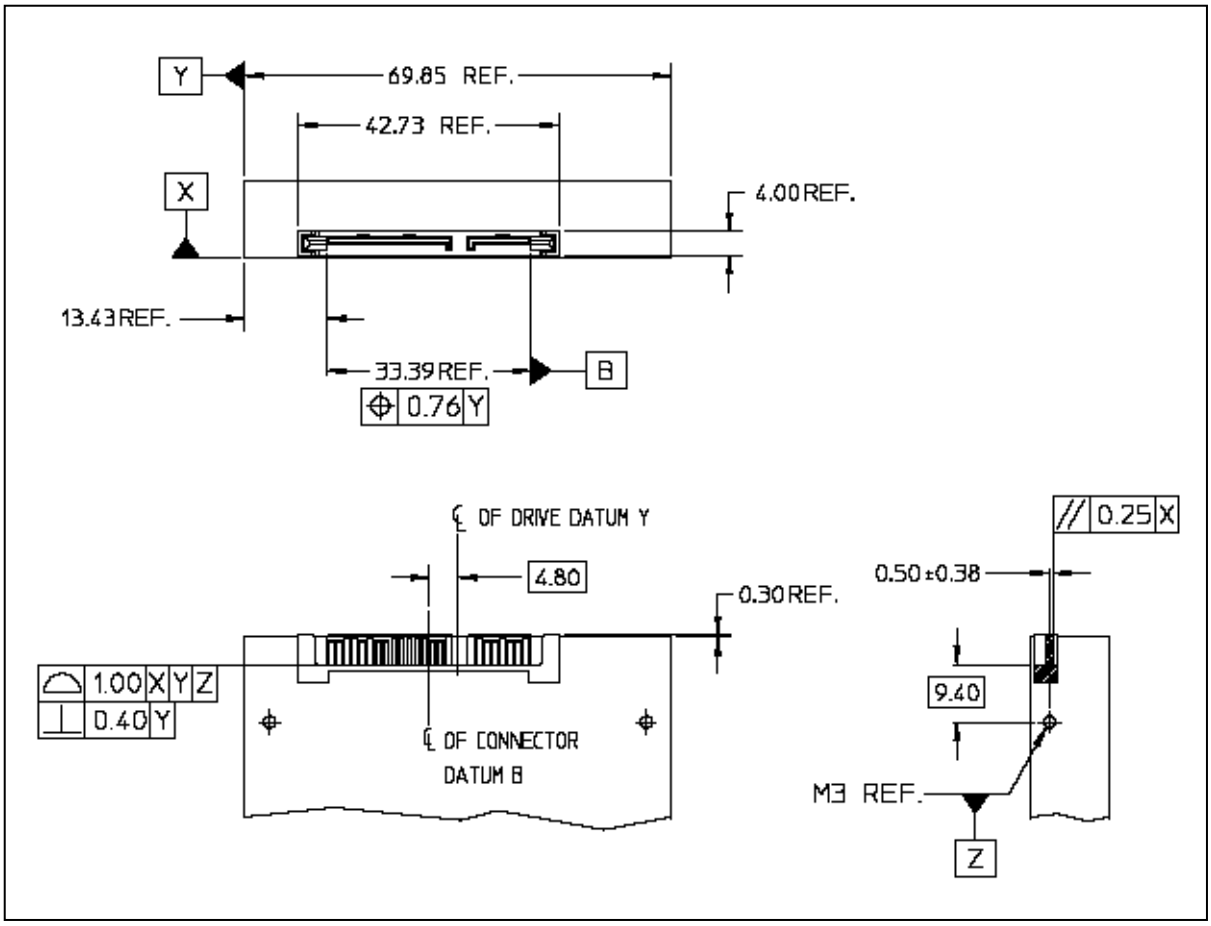


Figure 23 – Device Plug Connector Location on 2.5” Side Mounted Device

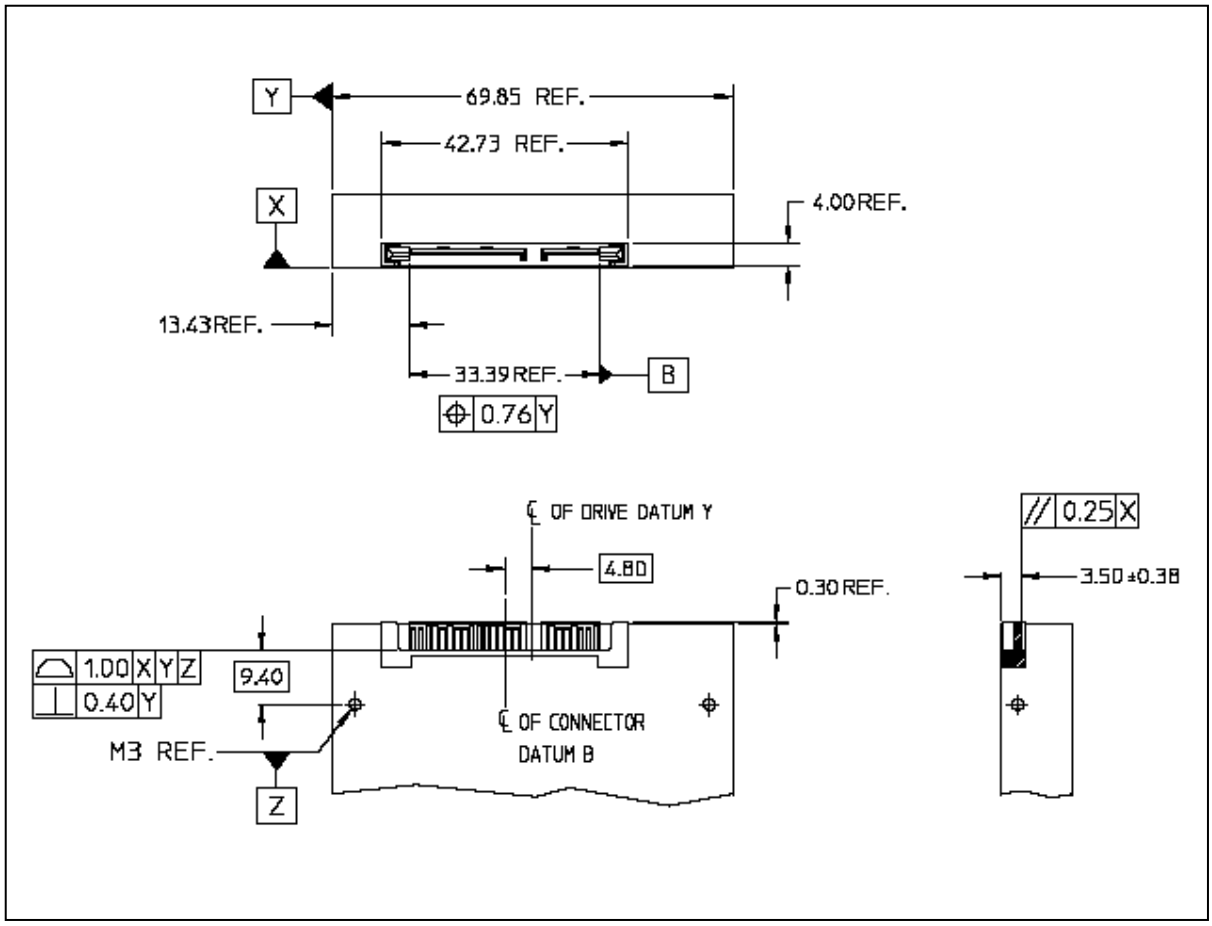


Figure 24 – Device Plug Connector Location on 2.5" Bottom Mounted Device

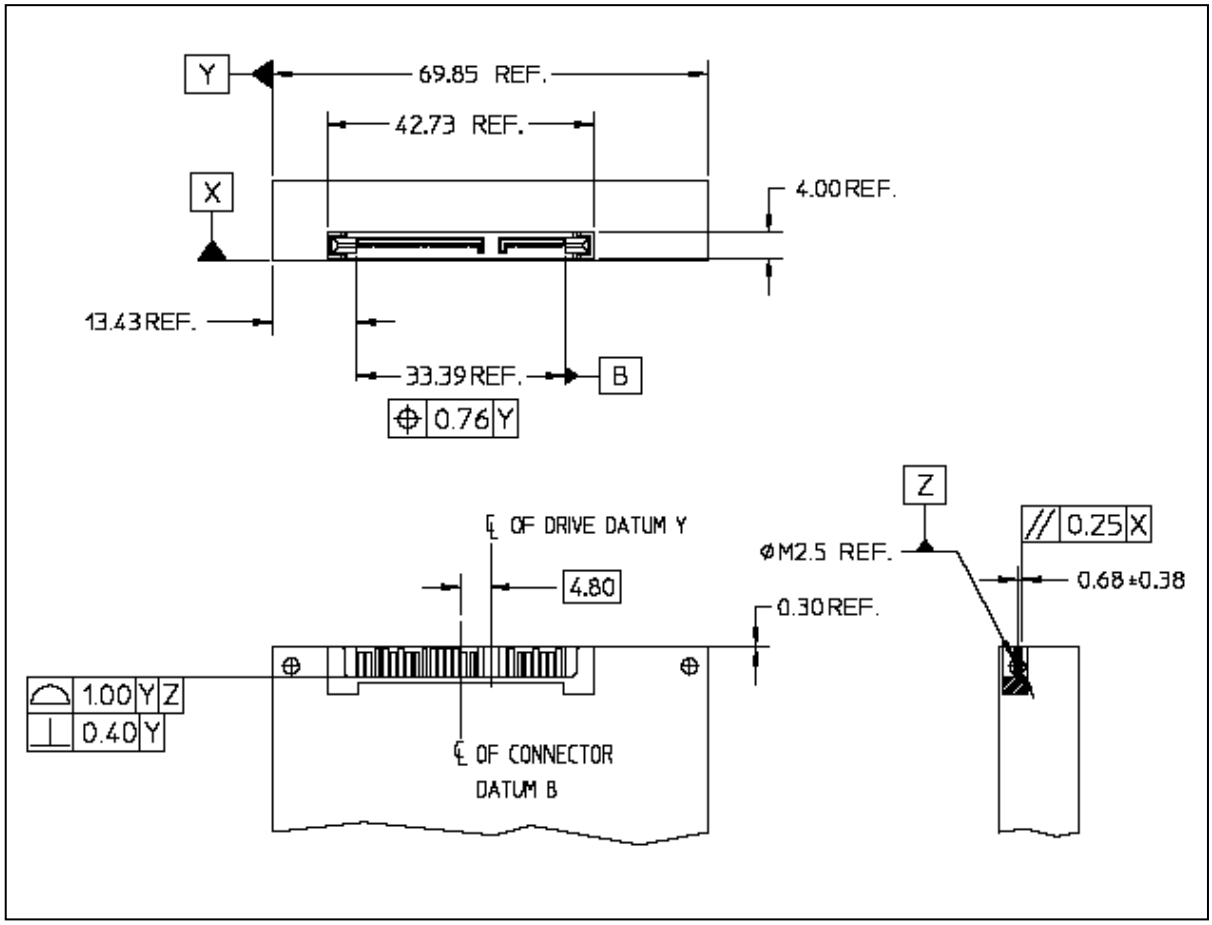


Figure 25 – Device Plug Connector Location on 1.8" Side Mounted Device

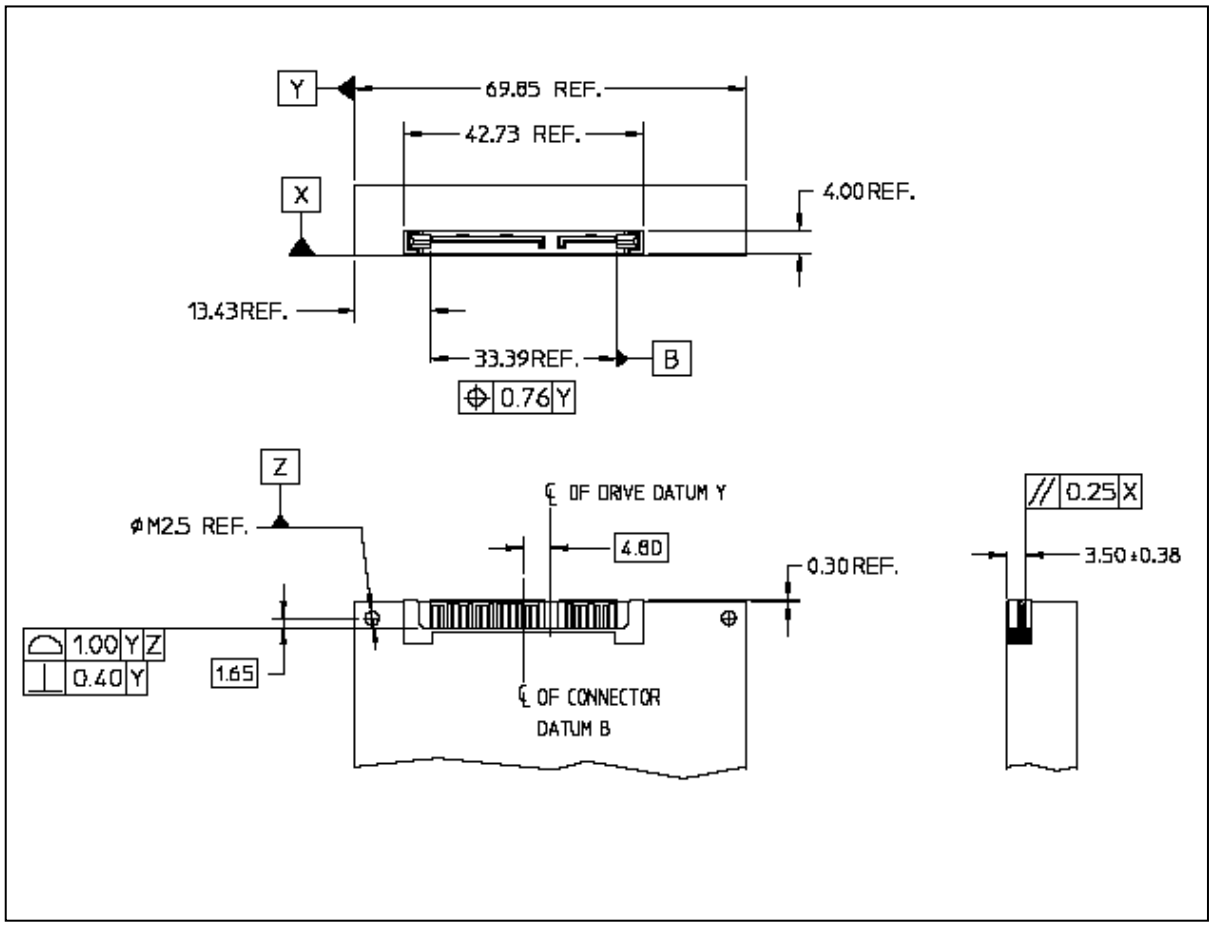


Figure 26 – Device Plug Connector Location on 1.8" Bottom Mounted Device

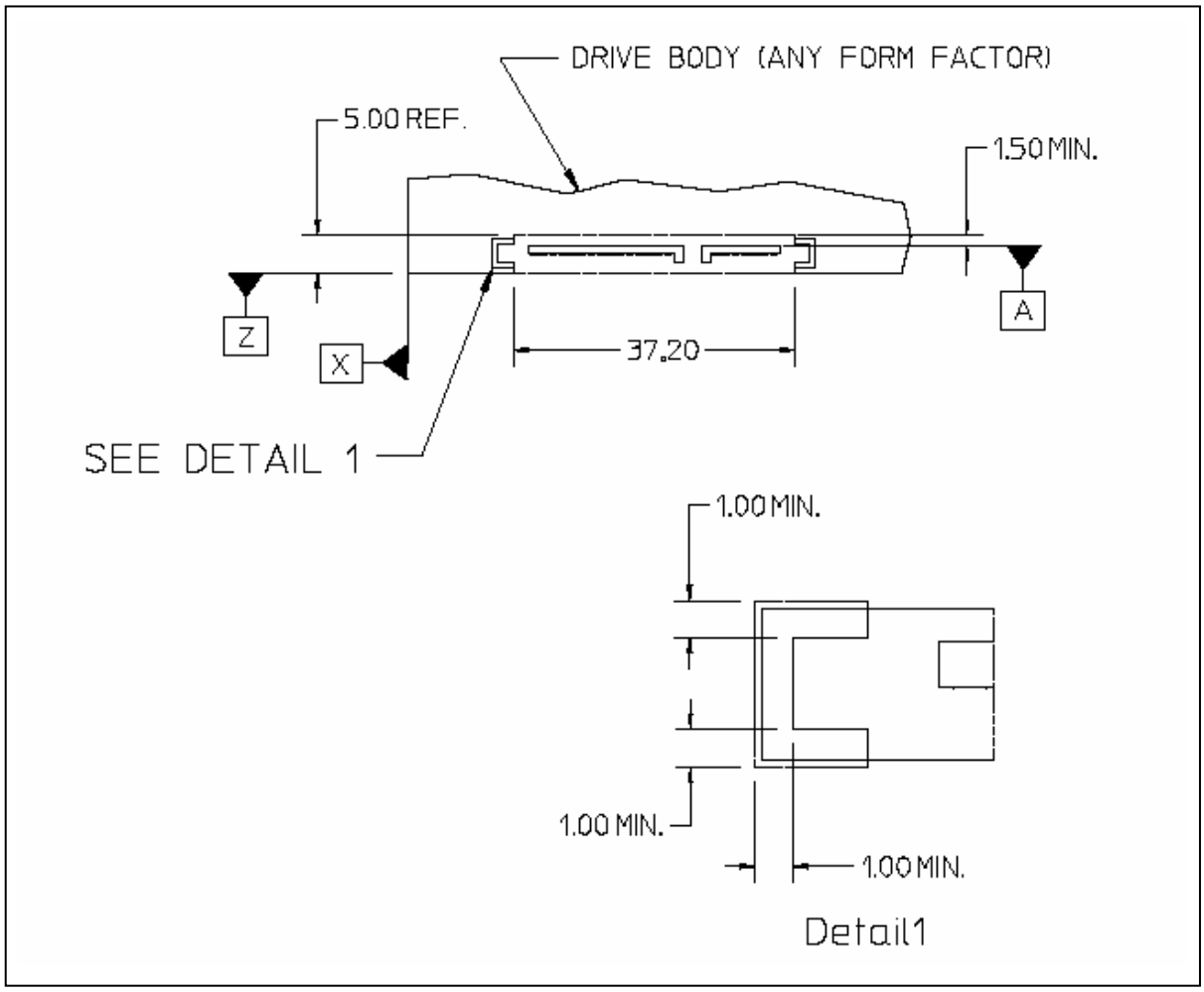


Figure 27 – Device Plug Connector Keep Out Zones

Notes:

1. The 1.50 keepout area above Datum A extends into the form factor to the connector Datum C in Figure 28.
2. The 1.00 keepout area shown in Detail 1 applies to both ends of the connector and extends from the connector housing outward to the outermost point of the form factor.

6.1.3 Mating interfaces

6.1.3.1 Device plug connector

Figure 28 and Figure 29 show the interface dimensions for the device plug connector with both signal and power segments. The device plug includes optional features to allow use of latching cables, fillets, and additional material to improve connector robustness. Table 3 defines the pin definitions and contact mating sequence for hot plug. These optional features should be included in all plug designs.

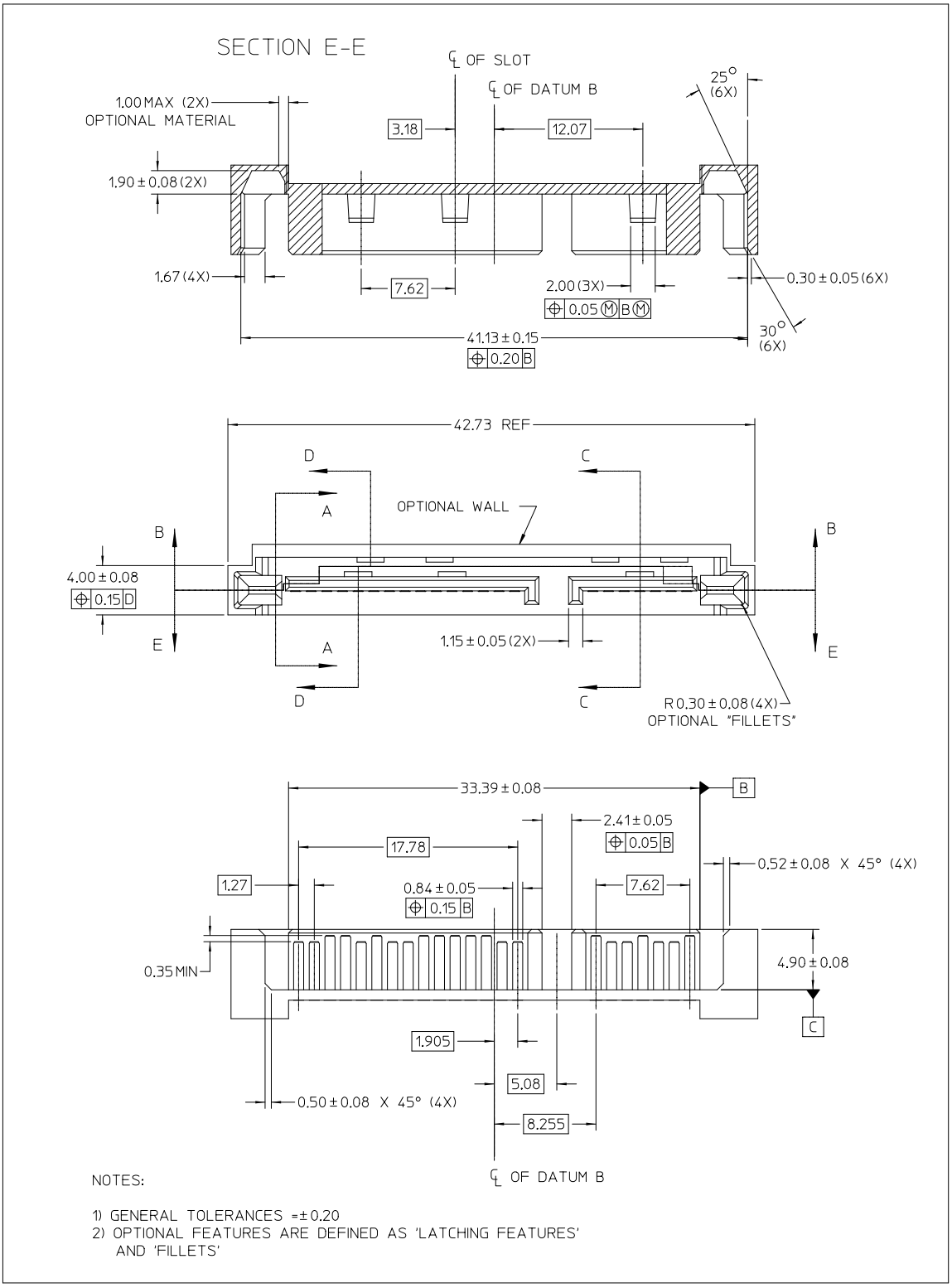


Figure 28 – Device Plug Connector

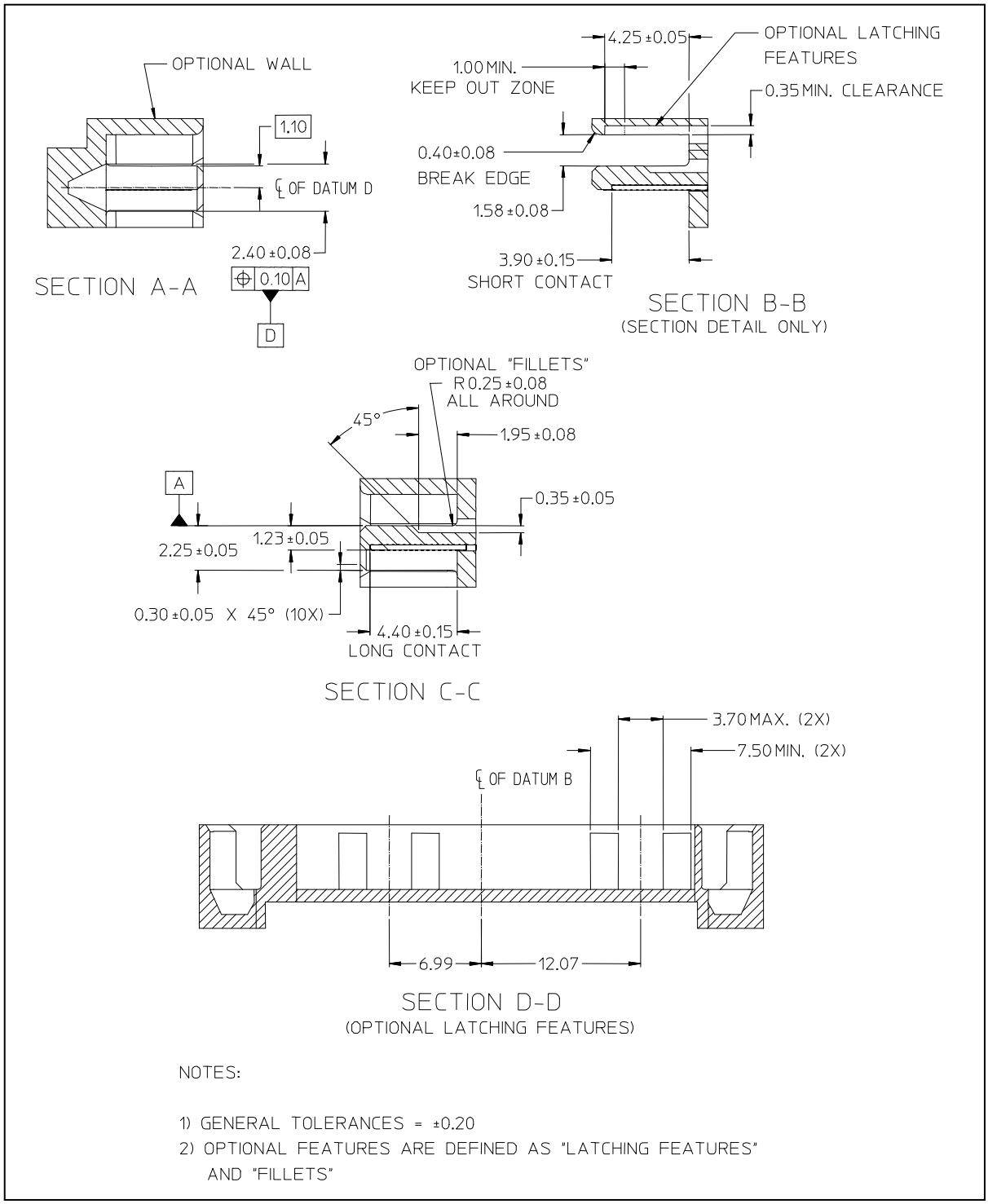


Figure 29 – Device Plug Connector (additional views)

6.1.3.2 Pin Signal Definition and Contact Mating Sequence

Table 3 details the pin names, types, and contact order of the two SATA plug options. A brief description is also included for signal, ground and power pins. There are total of 7 pins in the signal segment and 15 pins in the power segment.

Table 3 – Signal and Power SATA Plug and Nominal Mate Sequence

	Name	Type	Description	Cable Usage[2, 3]	Backplane Usage[3]
Signal Segment	S1	GND		1st Mate	2nd Mate
	S2	A+	Differential Signal Pair A	2nd Mate	3rd Mate
	S3	A-		2nd Mate	3rd Mate
	S4	GND		1st Mate	2nd Mate
	S5	B-	Differential Signal Pair B	2nd Mate	3rd Mate
	S6	B+		2nd Mate	3rd Mate
	S7	GND		1st Mate	2nd Mate
Key and Spacing separate signal and power segments[4]					
Power Segment	P1	V_{33}	3.3 V Power	2nd Mate	3rd Mate
	P2	V_{33}	3.3 V Power	2nd Mate	3rd Mate
	P3	V_{33}	3.3 V Power, Pre-charge	1st Mate	2nd Mate
	P4	GND		1st Mate	1st Mate
	P5	GND		1st Mate	2nd Mate
	P6	GND		1st Mate	2nd Mate
	P7	V_5	5 V Power, Pre-charge	1st Mate	2nd Mate
	P8	V_5	5 V Power	2nd Mate	3rd Mate
	P9	V_5	5 V Power	2nd Mate	3rd Mate
	P10	GND		1st Mate	2nd Mate
	P11	DAS/DSS	Device Activity Signal / Disable Staggered Spinup[1]	2nd Mate	3rd Mate
	P12	GND		1st Mate	1st Mate
	P13	V_{12}	12 V Power, Pre-charge	1st Mate	2nd Mate
	P14	V_{12}	12 V Power	2nd Mate	3rd Mate
	P15	V_{12}	12 V Power	2nd Mate	3rd Mate

NOTE:
1. The corresponding pin to be mated with P11 in the power cable receptacle connector shall always be grounded. For specific optional usage of pin P11 see section 6.7.
2. Although the mate order is shown, hot plugging is not supported when using the cable connector receptacle.
3. All mate sequences assume zero angular offset between connectors.
4. The signal segment and power segment may be separate.

Mating Configuration Notes

- All pins are in a single row with 1.27 mm (.050") pitch
- All ground pins in the Serial ATA device plug power segment (connector pins P4, P5, P6, P10, and P12) shall be bussed together on the Serial ATA device.
- The connection between the Serial ATA device signal ground and power ground is vendor specific.
- The following sets of voltage pins in the Serial ATA device plug power segment shall be bussed together on the Serial ATA device:
  - P1, P2, and P3 3.3 V power delivery and precharge
  - P7, P8, and P9 5 V power delivery and precharge
  - P13, P14, and P15 12 V power delivery and precharge
- The use of power delivery schemes that do not deliver all the specified voltages should only be used in scenarios where there is sufficient configuration control to ensure that the attached device does not require a supply voltage that is not provided.

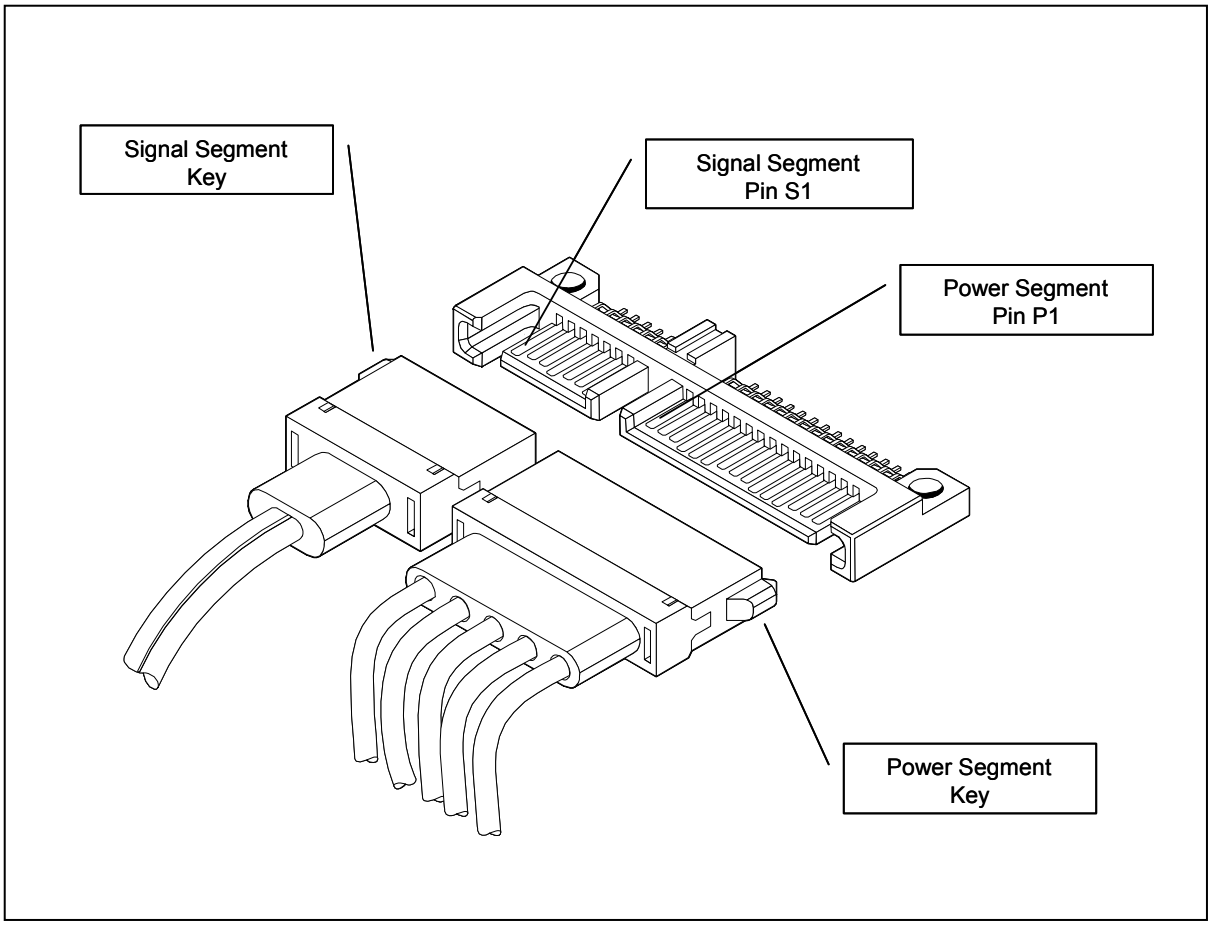


Figure 30 – Connector Pin and Feature Locations

6.1.4 Signal cable receptacle connector

Figure 31 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 32 defines an optional positive latch solution for internal cabled system applications. The latch requires the user to press and hold a release mechanism when disconnecting the cable. The latching feature option for device and host plug connectors are required in order to provide a latching surface. This latching feature option is called ClickConnect. Without a latching surface, there is no retention feature to hold a latching cable assembly in place.

It is optional to implement the latch on cable receptacles.

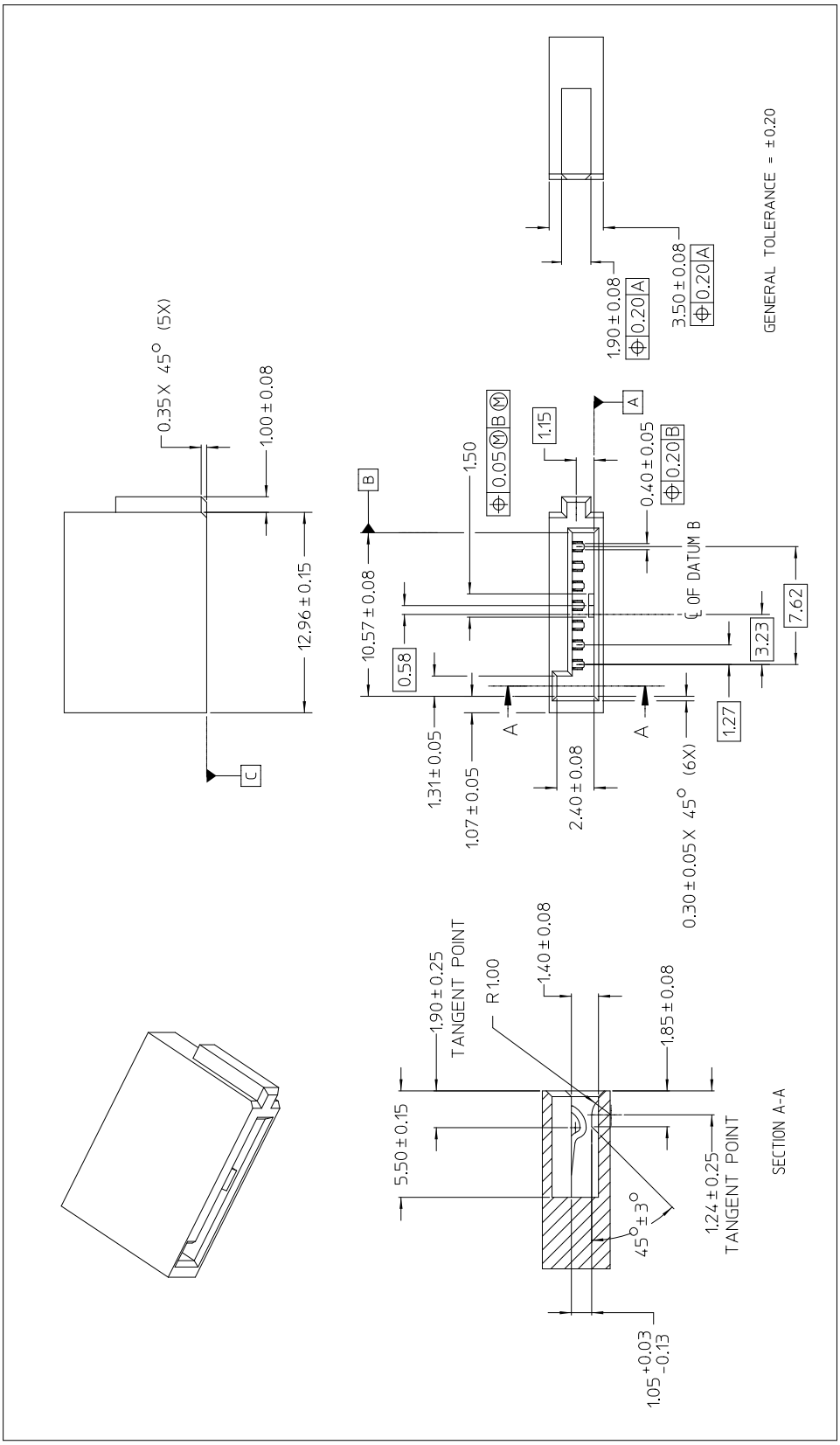


Figure 31 – Cable receptacle connector interface dimensions

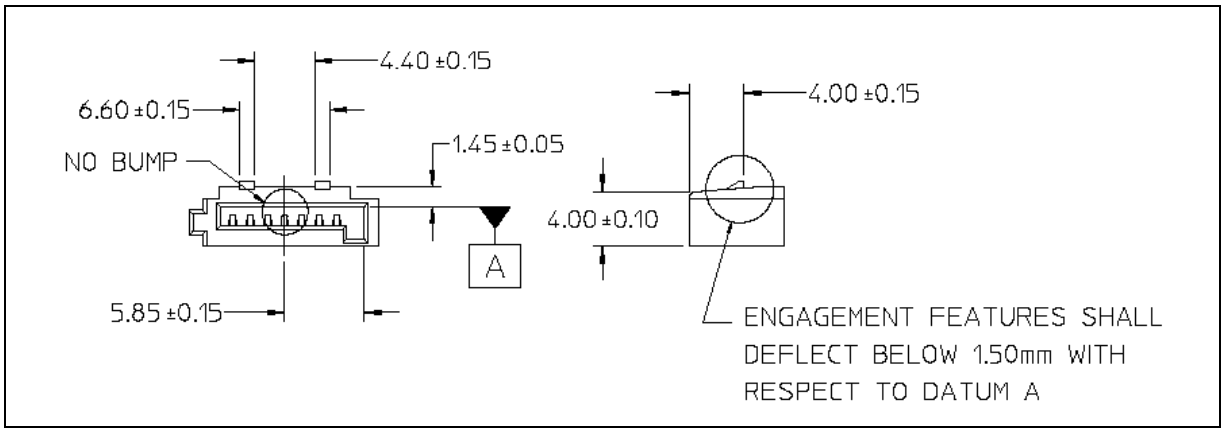


Figure 32 – Latching signal cable receptacle (ClickConnect)

The pin out of the cable receptacle connector is the mirror image of the signal segment of the device plug connector. Notice that:

- The two differential pin pairs are terminated with the corresponding differential cable pairs
- The ground pins are terminated with the cable drain wires, if it applies
- The choice of cable termination methods, such as crimping or soldering is up to each connector vendor

6.1.5 Signal host plug connector

The signal host plug connector is to be mated with one end of the Serial ATA cable assembly. The pinout of the host plug connector is the mirror image of the signal cable receptacle. Figure 33 shows the host plug connector interface definition. The host plug includes optional features to allow use of latching cables, fillets, and additional material to improve connector robustness.

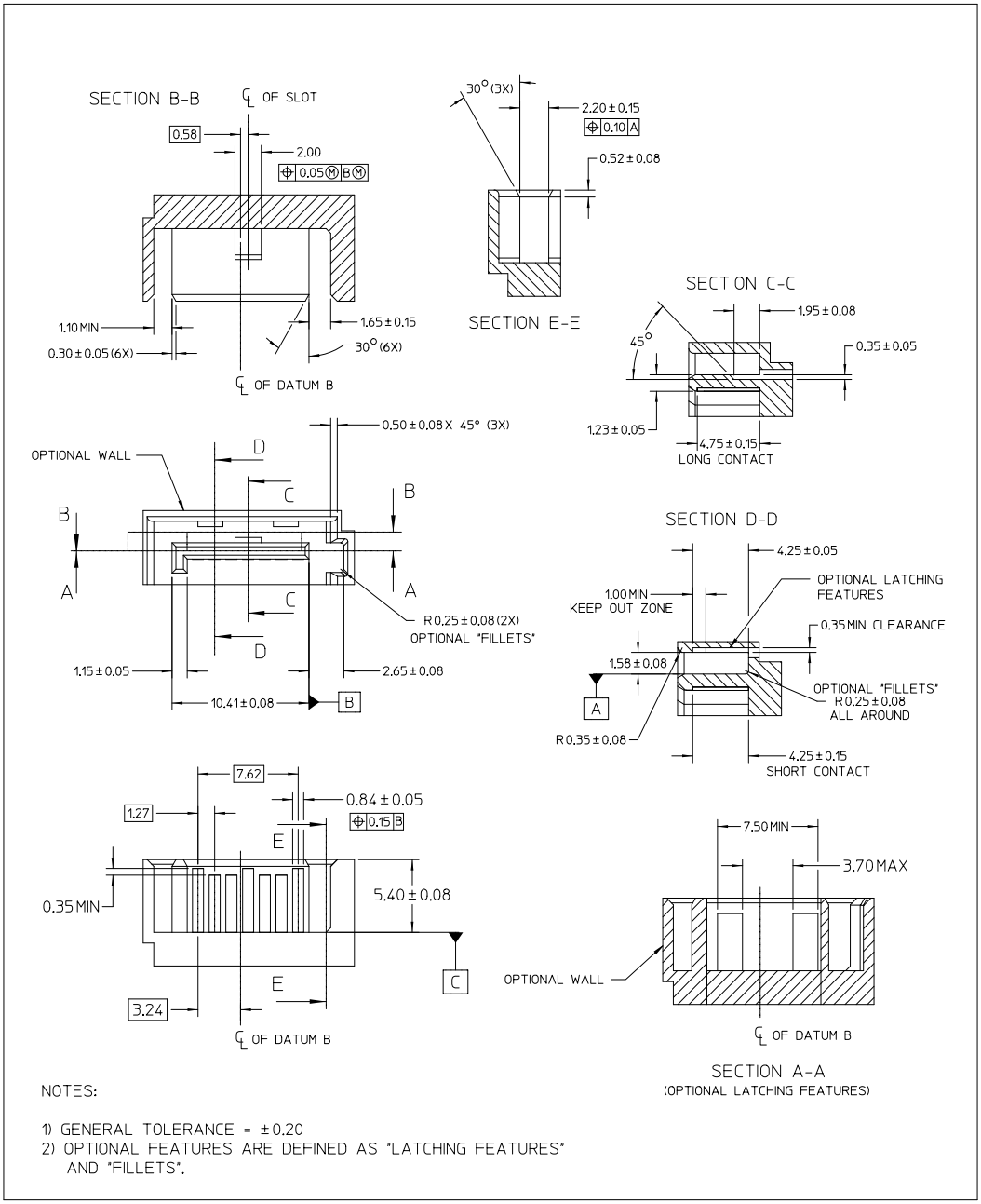


Figure 33 – Host signal plug connector interface dimensions

6.1.5.1 Internal plug stacking

The purpose of this recommended layout is to conserve motherboard, HBA and I/O controller printed circuit board space to support system density and size goals for such products.

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 34 shows the recommended clearance or spacing. Figure 35 shows the recommended clearance and orientation to allow access for latching cables.

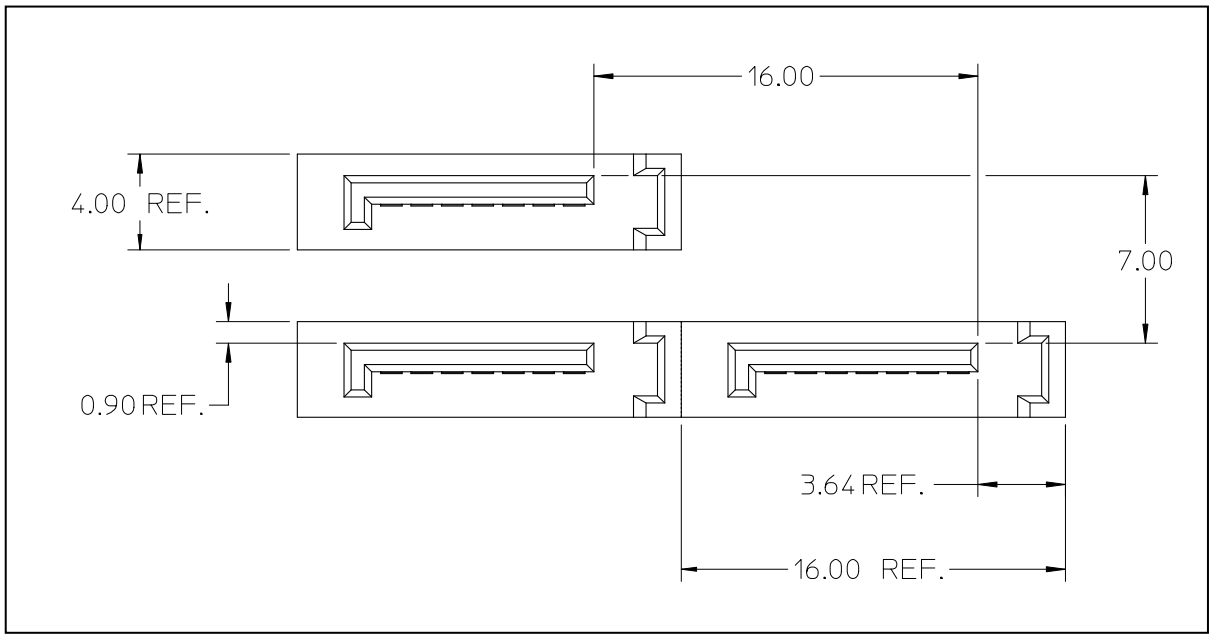


Figure 34 – Non-Latching Connector Stack Spacing and Orientation

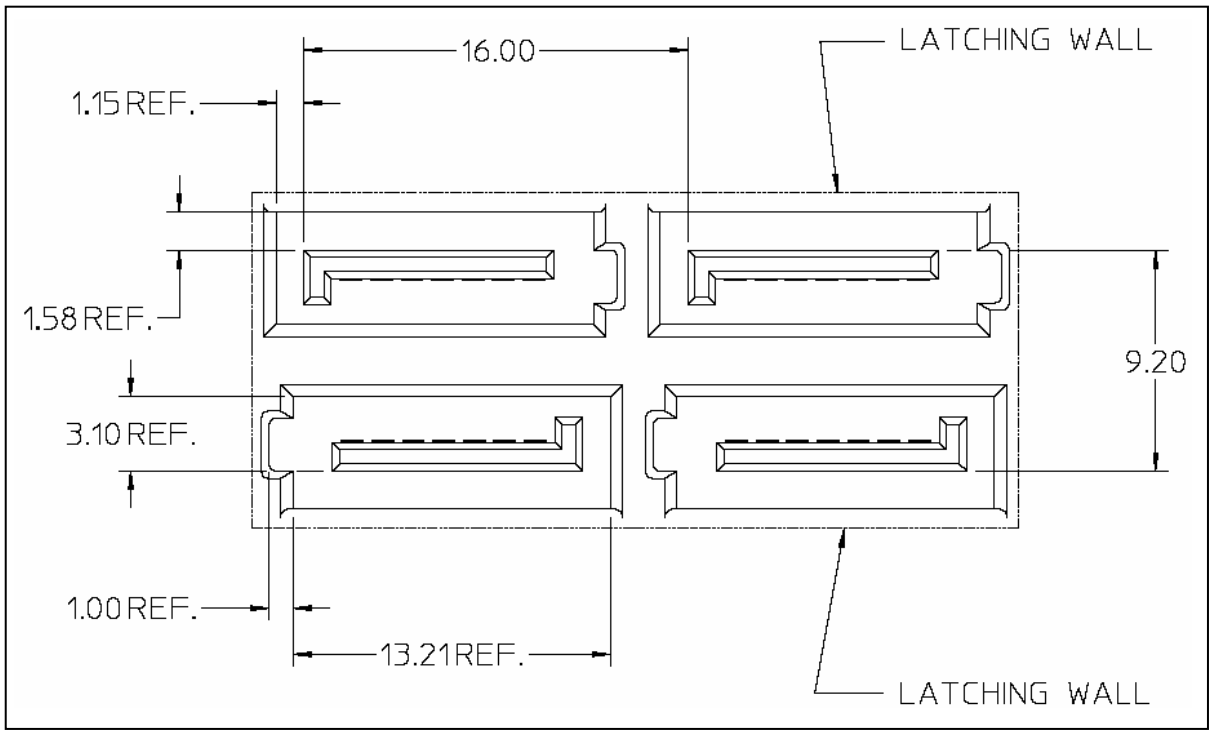


Figure 35 – Latching Connector Stack Spacing and Orientation

6.1.6 Backplane connector

The backplane connector is to be blind-mated directly with the device plug connector. The interface dimensions for the backplane connector are shown in Figure 36. Note that dimension B allows two values: 8.15 mm and 14.15 mm. There are two levels of contacts in the backplane 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. The backplane connector is not designed with any retention mechanism.

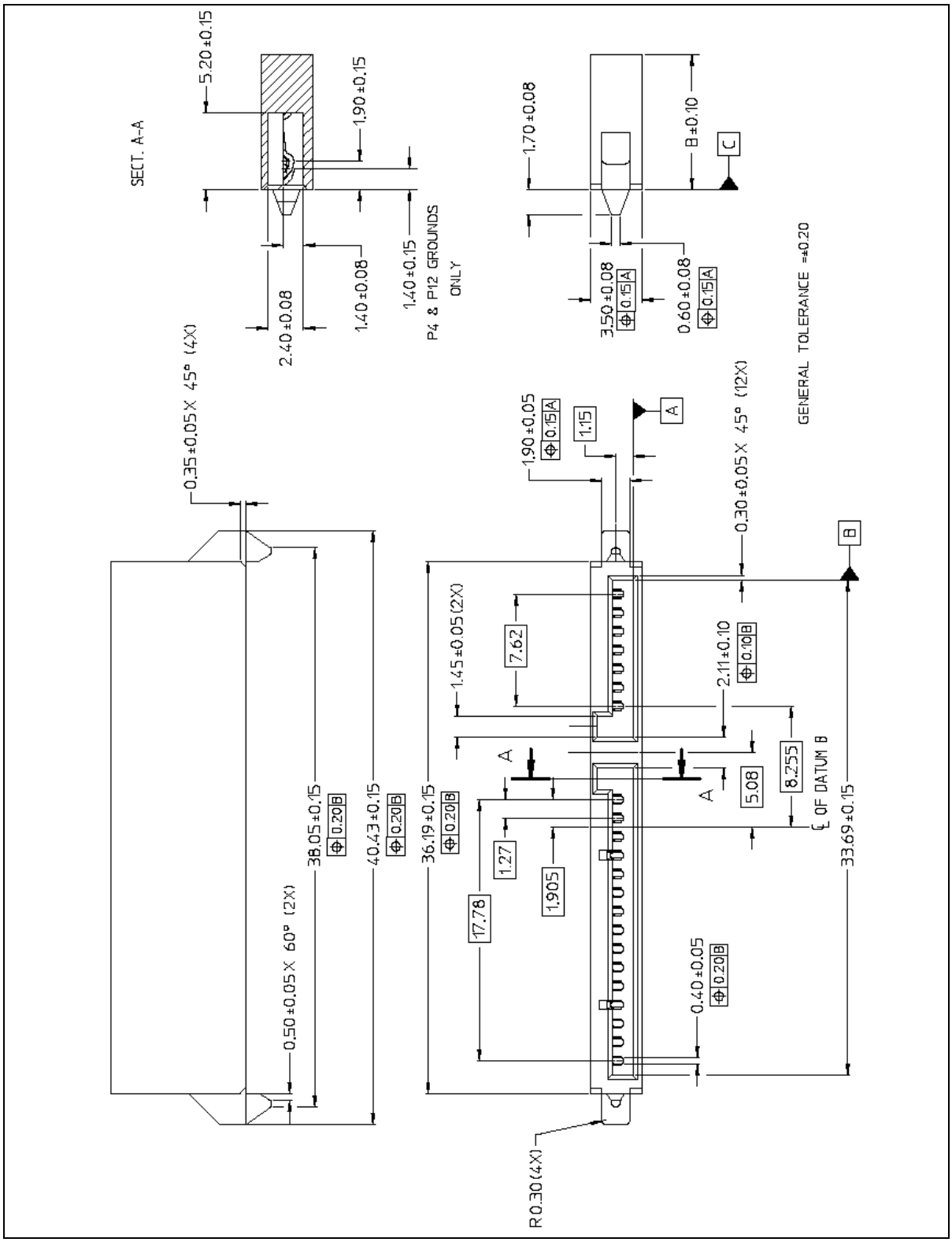


Figure 36 – Backplane connector interface dimensions

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

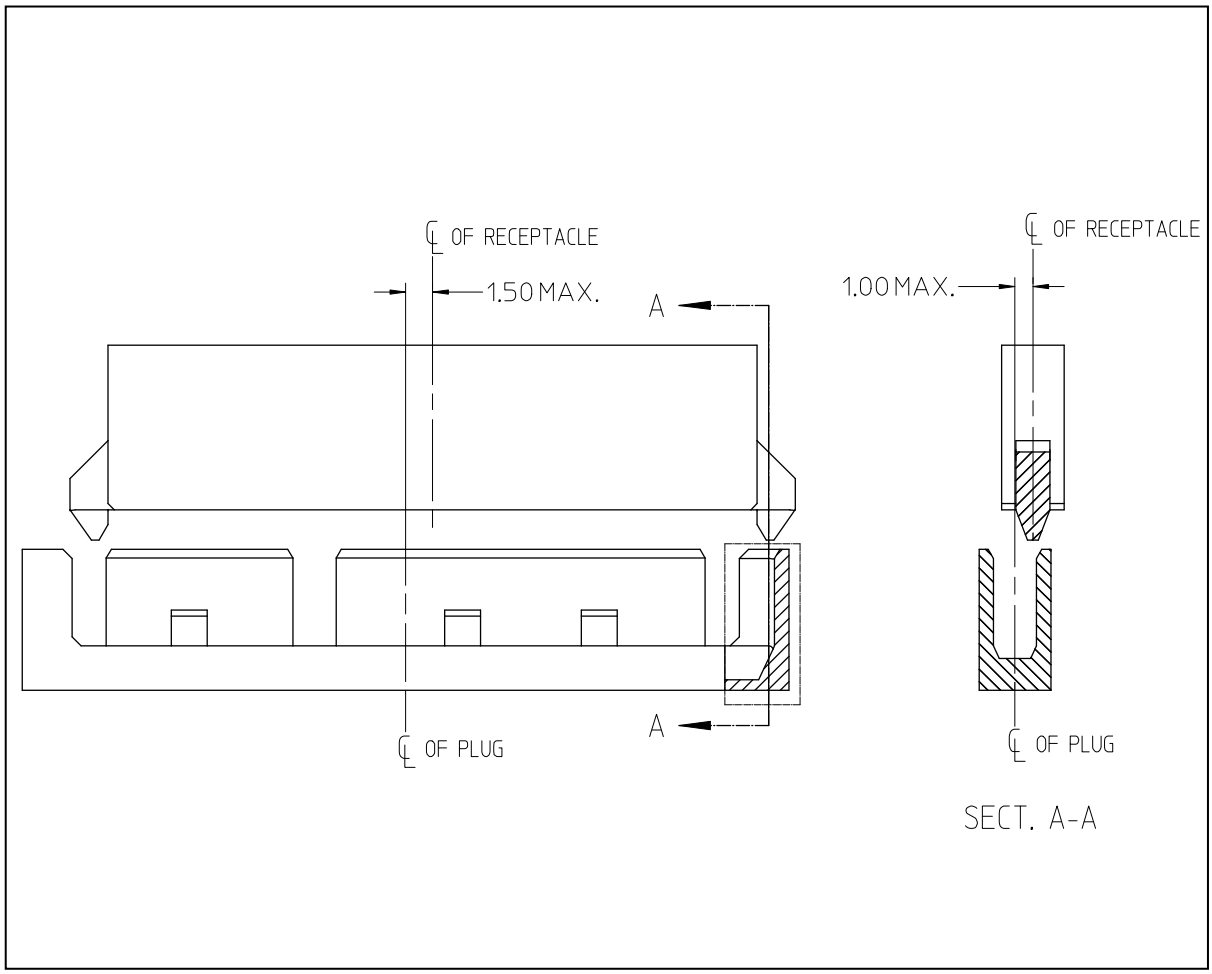


Figure 37 – Connector pair blind-mate misalignment tolerance

The device-to-backplane mating configuration is shown in Figure 38. The allowed values for dimension A and dimension B are shown in Table 4.

Table 4 – Allowed values for dimension A and B for device-to-backplane mating

Description	Standard	Extended
Device mated height (A)	8.45 mm	14.45 mm
Component clearance (B)	3.55 mm	9.55 mm

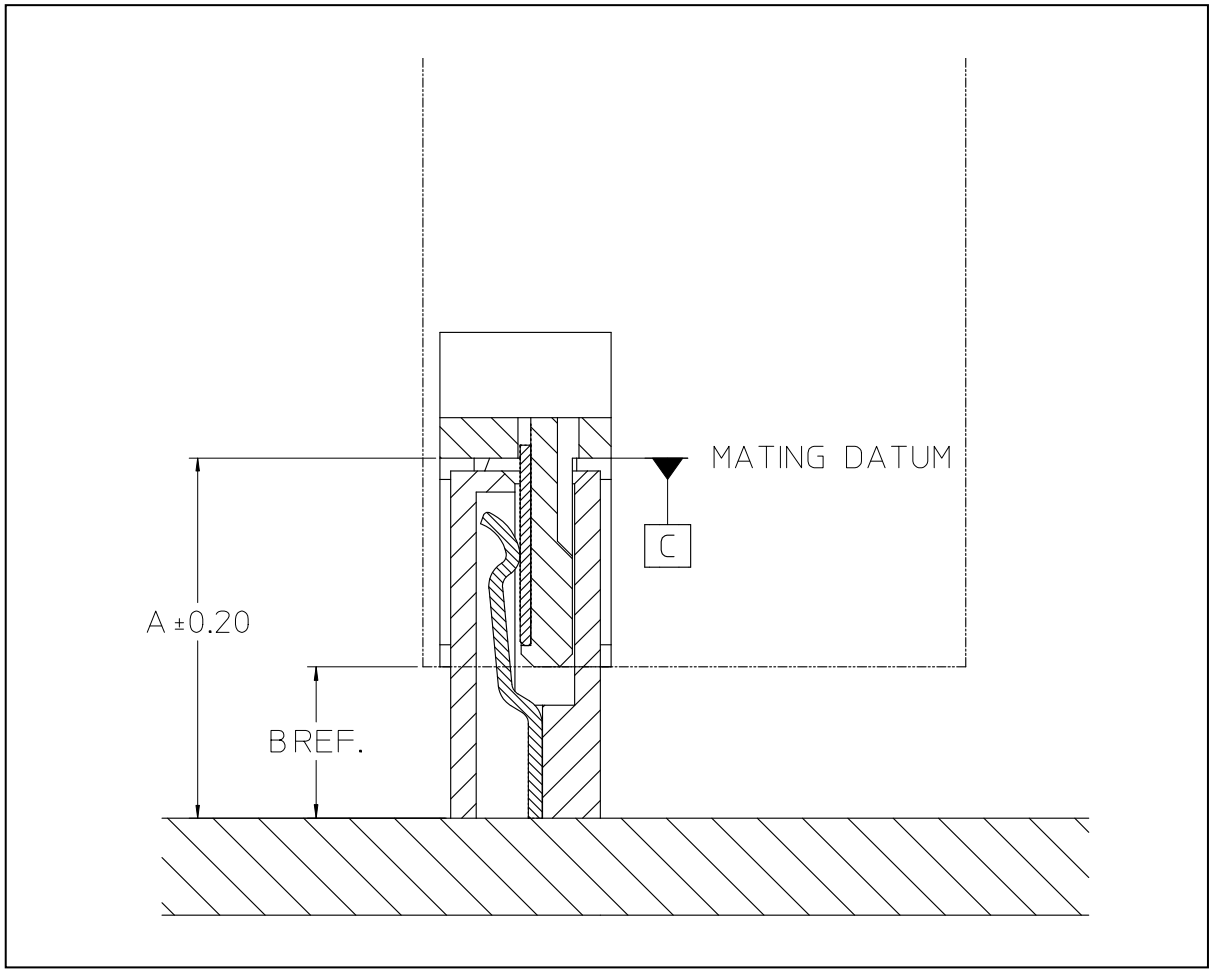


Figure 38 – Device-backplane mating configuration

6.1.7 Power cable receptacle connector

The power cable receptacle connector mates with the power segment of the device plug, bringing power to the device. Figure 39 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 3. Figure 40 defines an optional positive latch solution for internal cabled system applications. The latch requires the user to press and hold a release mechanism when disconnecting the cable. The latching feature option for device plug connectors is required in order to provide a latching surface. Without a latching surface, there is no retention feature to hold a latching cable assembly in place.

It is optional to implement the latch on cable receptacles.

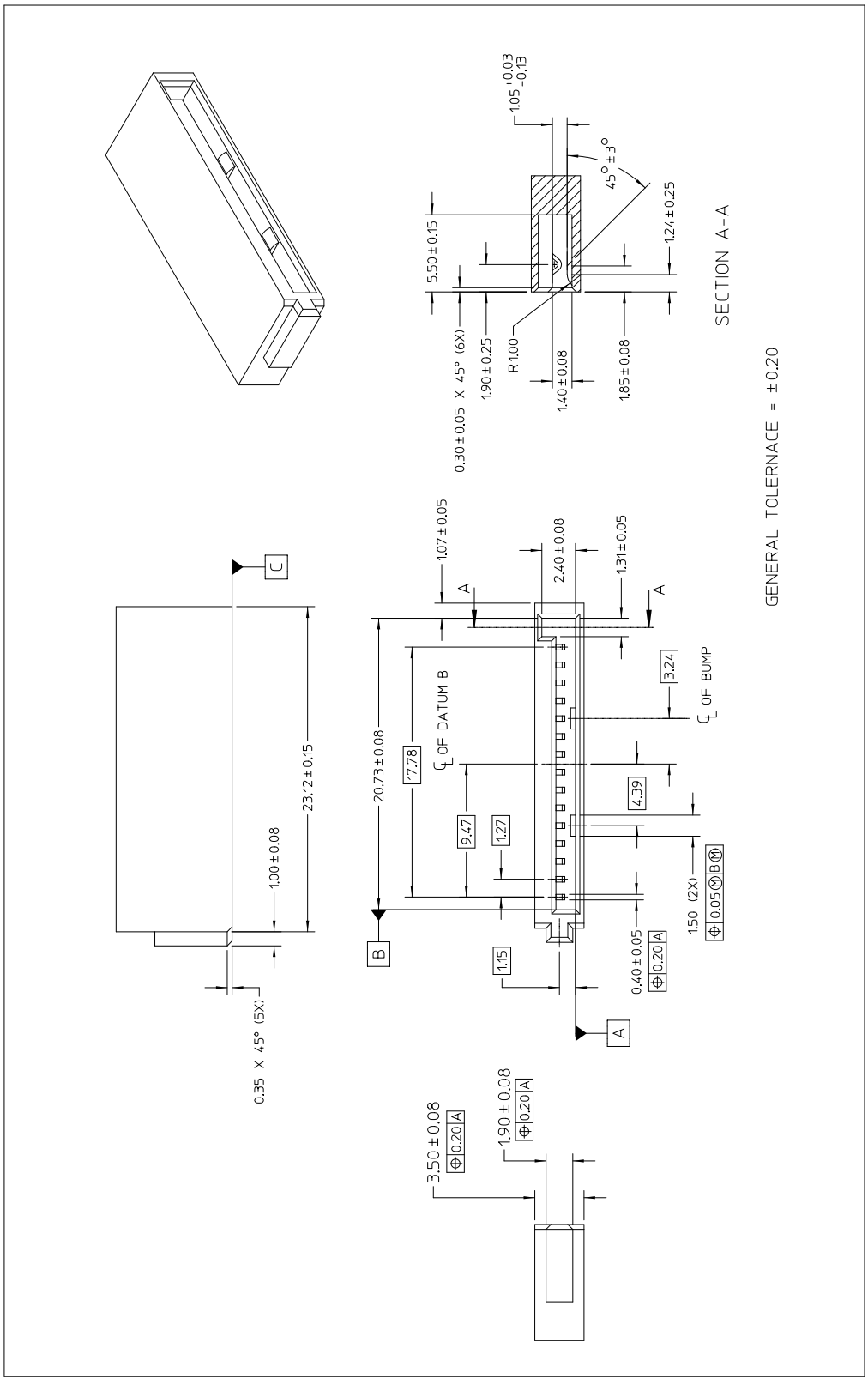


Figure 39 – Power receptacle connector interface dimensions

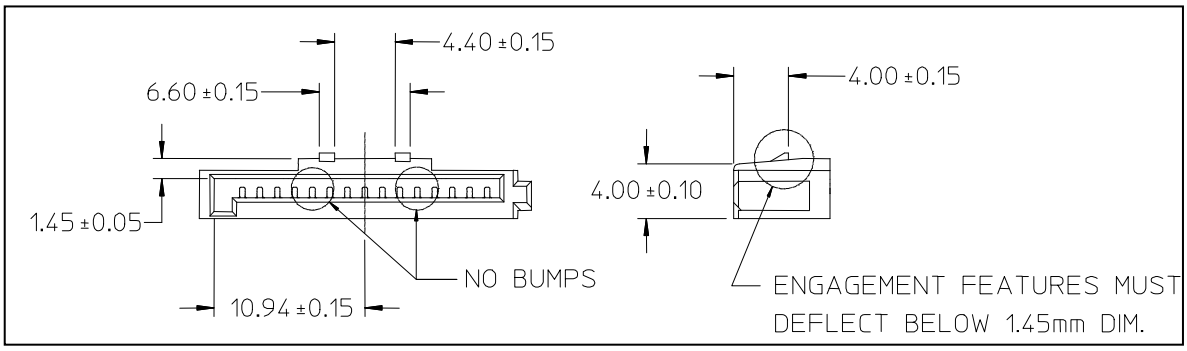


Figure 40 – Latching power cable receptacle

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.

6.1.8 Internal single lane cable

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

This specification does not specify a standard internal single lane cable. Any cable that meets the electrical requirements in section 6.3 is considered an acceptable internal single lane cable. 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 Figure 41 for an informational purpose only.

Although construction methodologies are not specified, there are a few essential elements of the Serial ATA cable that should be considered. Physical characteristics of the Serial ATA cable may include the following items. See Figure 41 for details.

- Shielded Pairs (2)
- Solid Tinned Copper (26 AWG)
- White Foam Polyolefin (43.5 mil Diameter)
- Parallel Drain Pairs (2 pr., 28 AWG – Solid Tinned Copper)
- Aluminized Polyester Foil (1 mil thick w/35mil overlap)
- Foil may be the blue longitudinal wrap that is sealed with heat
- Jacket (20 mil PVC wall)

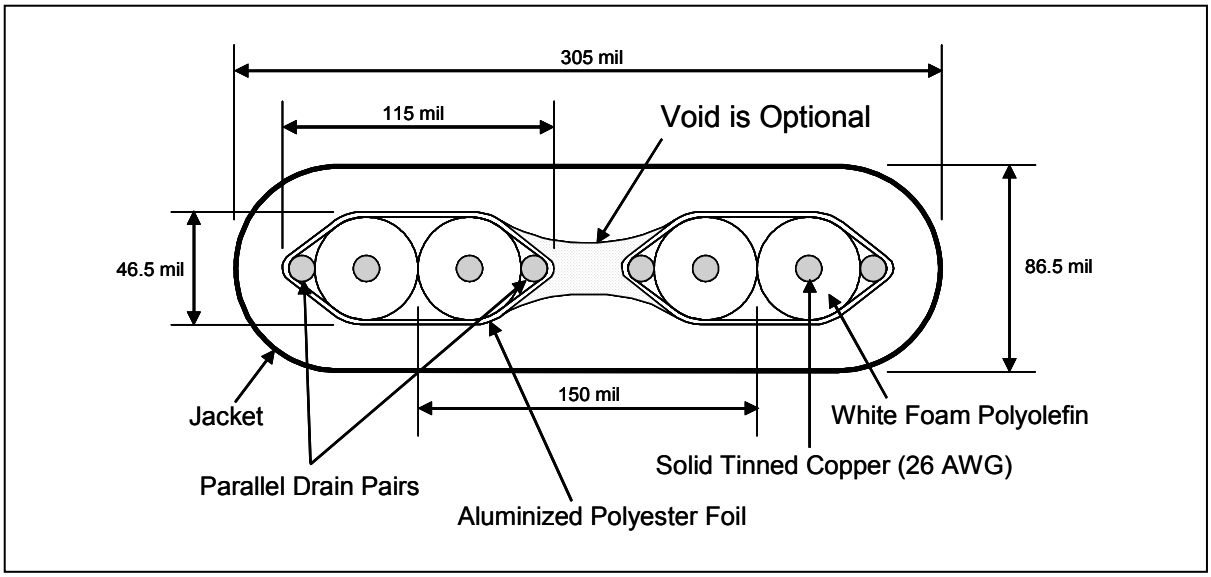


Figure 41 – Detailed cross-section of an example internal single lane cable

6.1.9 Connector labeling

Labeling on a connector with the connector manufacturer identifier or Serial ATA icon is optional.

6.1.10 Connector and cable assembly requirements and test procedures

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

- Mated
- Temperature: 15° to 35° C
- Relative Humidity: 20% to 80%
- Atmospheric Pressure: 650 mm to 800 mm of Hg

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

6.1.10.1 Housing and contact electrical requirements

Table 5 is the connector housing and contact electrical requirements.

Table 5 – 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 20mV maximum open circuit at 100mA maximum	• Initially 30 mΩ maximum. • Resistance increase 15 mΩ maximum after stress
Contact current rating (Power segment)	• Mount connector to a test PCB • Wire three adjacent pins in parallel for supply (or the minimum number required by the connector type) • Wire three adjacent pins in parallel for return (or the minimum number required by the connector type) • Apply a DC current of three times the current rating per contact to the supply pins, returning through the return pins. • 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.

6.1.10.2 Mechanical and environmental requirements

Table 6 lists the mechanical parameters and requirements, while Table 7 the environmental and reliability tests and requirements:

Table 6 – Mechanical test procedures and requirements

Visual and dimensional inspections	EIA-364-18 Viual, 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 shall meet all connector and cable mechanical requirements before and after the completion of the test.
Cable flexing	For round cable: EIA-364-41 Condition I Dimension x=3.7 x 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 us during flexing.
Insertion force Cabled signal connector	EIA-364-13 Measure the force necessary to mate the connector assemblies at a max. rate of 12.5 mm per minute.	45 N Max.
Removal force Cabled signal connector (Non-latching)	EIA-364-13 Measure the force necessary to unmate the connector assemblies at a max. rate of 12.5 mm per minute.	10 N Min. through 50 cycles
Insertion force Cabled power connector	EIA-364-13 Measure the force necessary to mate the connector assemblies at a max. rate of 12.5 mm per minute.	45 N Max.
Removal force Cabled power connector (Non-latching)	EIA-364-13 Measure the force necessary to unmate the connector assemblies at a max. rate of 12.5 mm per minute.	15 N Min. for cycles 1 through 5 10 N Min. through 50 cycles
Insertion force Backplane connector	EIA-364-13 Measure the force necessary to mate the connector assemblies at a max. rate of 12.5 mm per minute.	20 N Max.
Removal force Backplane connector	EIA-364-13 Measure the force necessary to unmate the connector assemblies at a max. rate of 12.5 mm per minute.	4 N Min. after 500 cycles
Removal force Cabled Latching connector Includes power and signal connectors	EIA-364-13 Apply a static 25 N unmating test load	No damage
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 Table 9.

Table 7 – 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 ms 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 us 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 us 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.
NOTE: 1. 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 Table 9. 2. Shock and vibration test fixture is to be determined by each user with connector vendors.		

An additional requirement is listed in Table 8.

Table 8 – Additional requirement

Parameter	Procedure	Requirement
Flammability	UL 94V-0	Material certification or certificate of compliance required with each lot to satisfy the Underwriters Laboratories follow-up service requirements.

It should be pointed out that this specification 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 shall be added to satisfy the application needs. For example, a user who requires a SMT connector may want to include additional requirements for SMT connector reliability.

6.1.10.3 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 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.

6.1.10.4 Test sequence

Table 9 shows the connector test sequences for five groups of tests.

Table 9 – Connector test sequences

Test or examination	Test group				
	A	B	C	D	E
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(1)			2(1)
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 –
1. 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)

6.1.11 Internal Multilane cables

This section defines standard cable assemblies and headers for connecting multiple Serial ATA links from a RAID host bus adapter to a backplane within the same enclosure or server, or for connecting multiple Serial ATA links from a host bus adapter to individual devices.

This cable/connector is based on the SFF-8484 specification. The SFF-8484 specification is also used by Serial Attached SCSI (SAS).

6.1.11.1 Conformance Criteria

- 2 or 4 lanes, Serial ATA signals.
- Either point to point with a high density connector on both ends of the cable or fanout with a high density connector on one end of the cable assembly and individual single lane connectors on the other end of the cable assembly.
- Additional pins/conductors for sideband signals.
- Specifications for PCB footprint for SMT, thru hole and press fit.
- RX, TX, RX, TX pin sequencing to minimize crosstalk.
- Ground reference between each pair.
- Flexible cable for routing and airflow.
- Performance supporting 1.5 Gbps and 3.0 Gbps.
- Compliance points are at the ends of a mated cable interface. If additional interconnect media between host and device exists, that portion of the design is proprietary and shall ensure the mated cable interface compliance points are met.

6.1.11.1.1 Electrical Requirements

The Internal Multilane cable assembly shall meet the electrical characteristics defined in Table 18. The Internal Multilane cable assembly shall operate at Gen1i and Gen2i levels and meet the electrical characteristics defined in Table 18. Since this cable is a Multilane and therefore has multi-aggressors, the additional requirement is to have crosstalk measured using the CXT method. The measured crosstalk shall meet the requirements listed in Table 18.

6.1.11.1.2 Component Descriptions

Three components are defined in this section for Multilane applications. Each component has a 2 Lane and a 4 Lane version. Pin assignments are provided at the end of this section.

- Cable Receptacles and Backshells
- Vertical Headers
- Right Angle Headers

6.1.11.1.3 Cable Receptacles and Backshells

Figure 42 and Figure 43 show isometric drawings of the internal Multilane cables and connectors. Refer to SFF-8484 for dimensions and mechanical details.

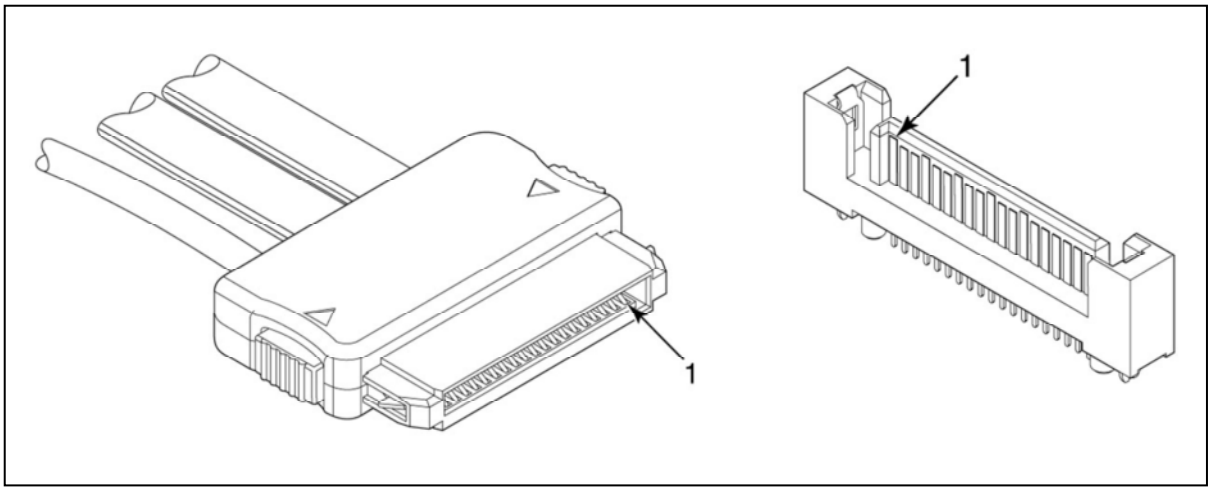


Figure 42 – Isometric drawings of the internal 2 Lane cable and connector

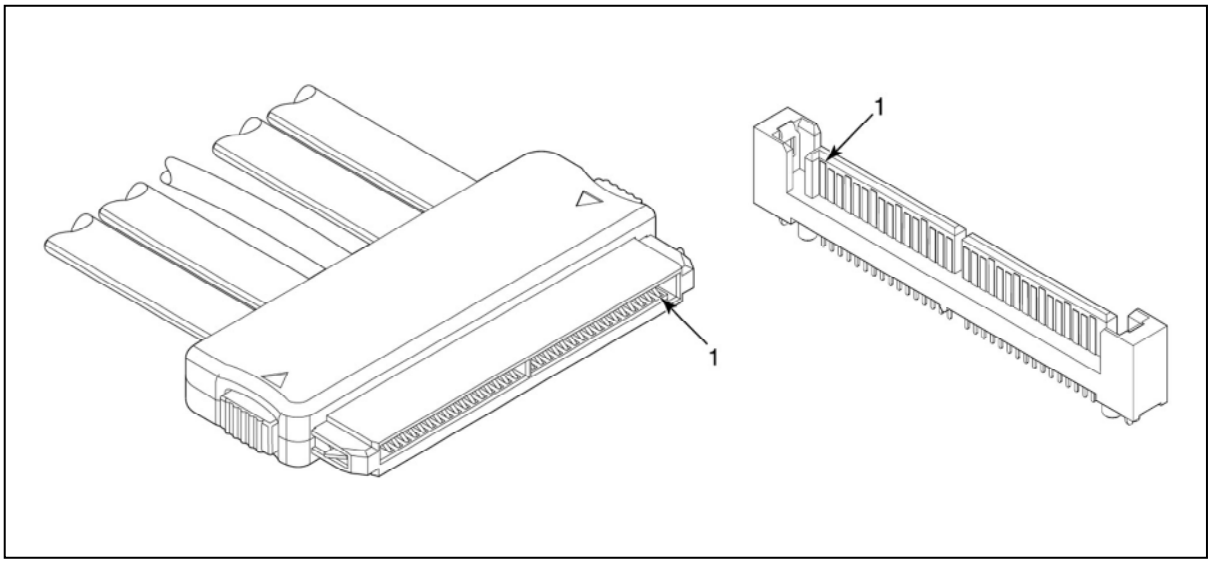
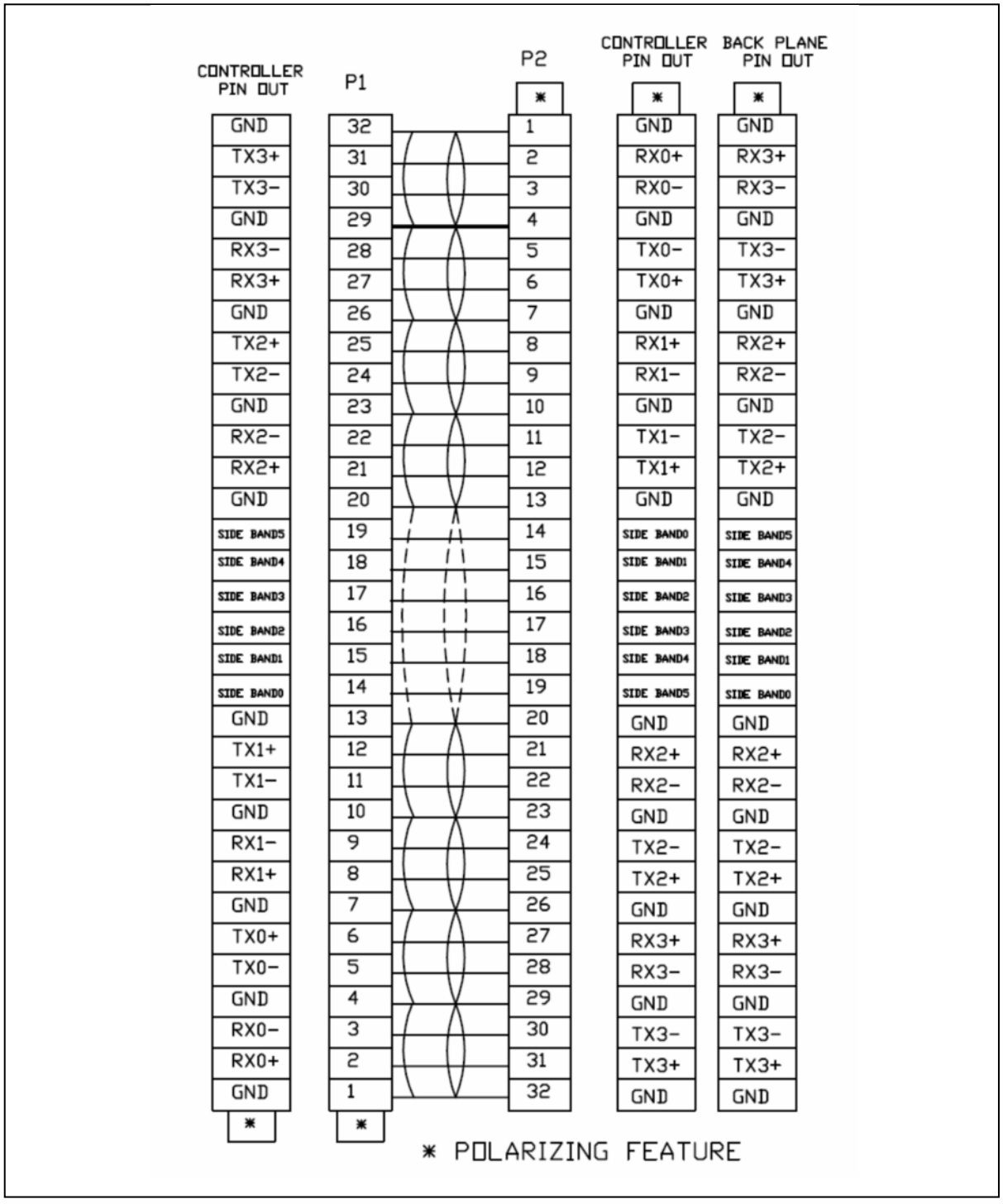


Figure 43 – Isometric drawings of the internal 4 Lane cable and connector

6.1.11.2 4 Lane Pin Assignments



Note: Dashed line around Side Band lines 14 – 19 indicates shielding optional based on application and may be connected to the adjacent signal grounds.

Figure 44 – 4 Lane Pin Assignments

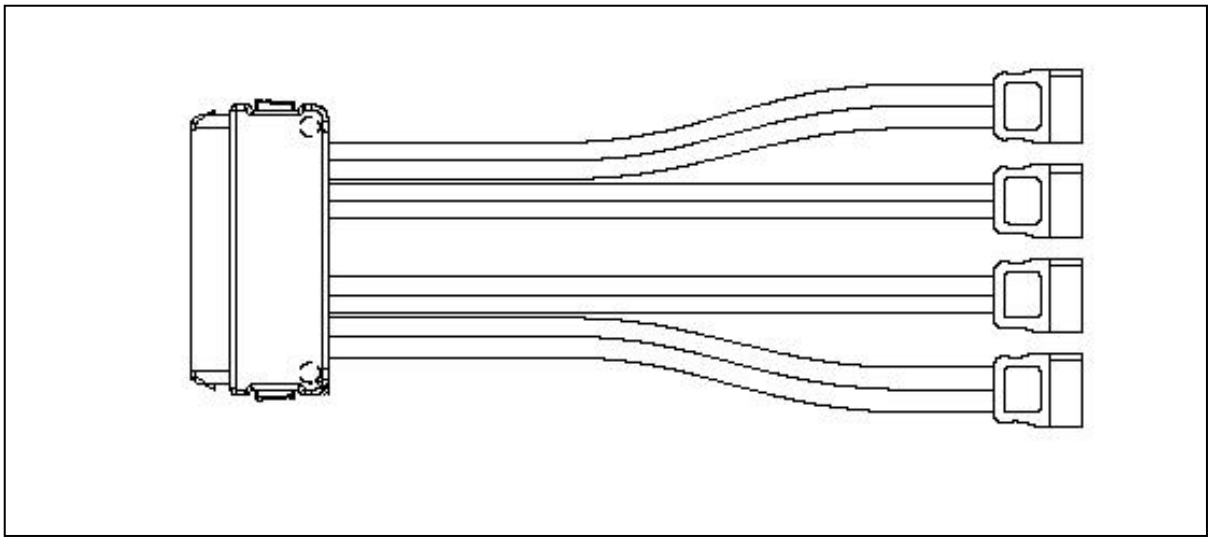
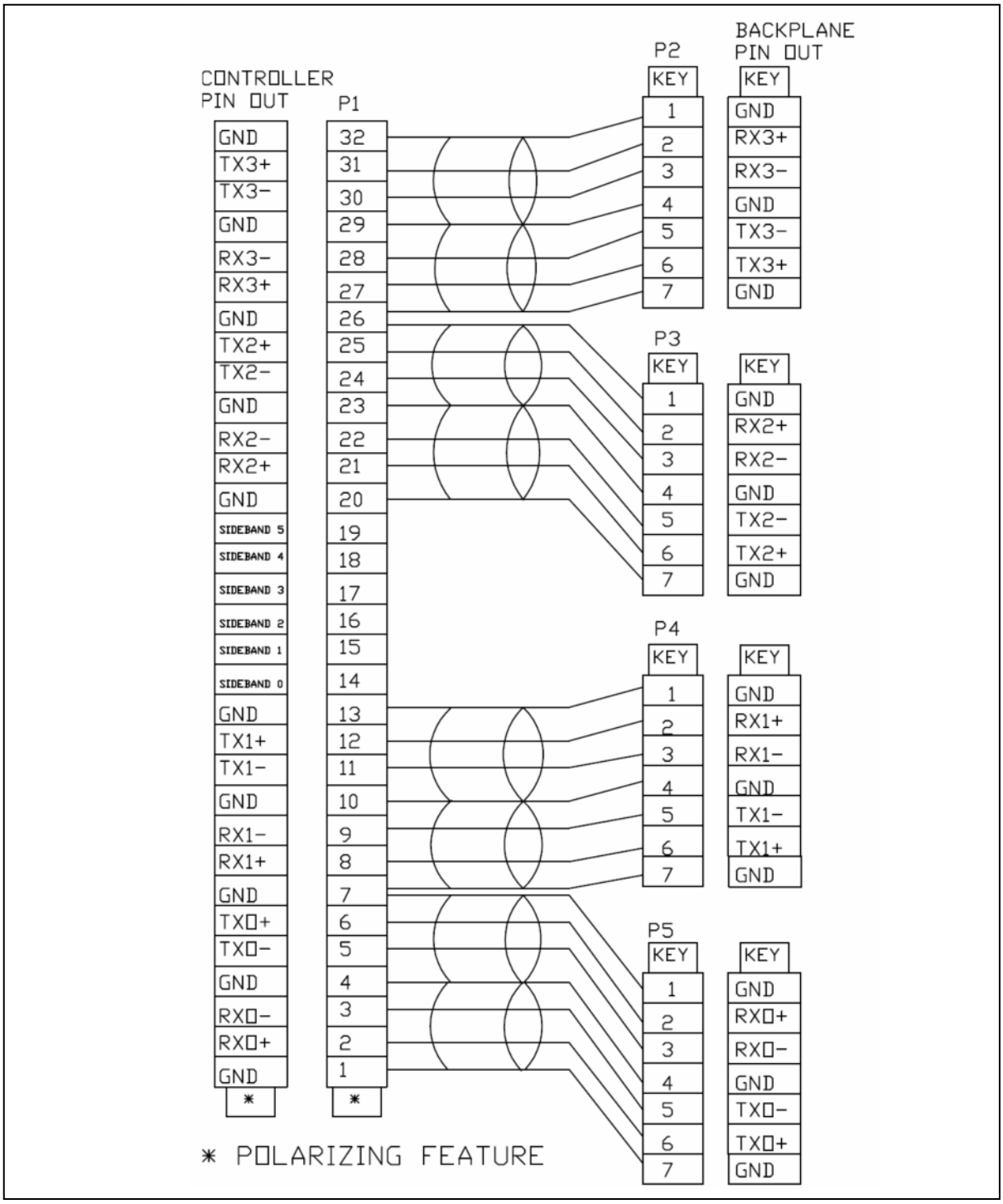


Figure 45 – 4 Lane to 4 x 1 Lanes, Fanout Implementation



Note: Sideband signals not shown

Figure 46 – 4 Lane Fanout Pin Assignments

6.1.11.3 2 Lane Pin Assignments

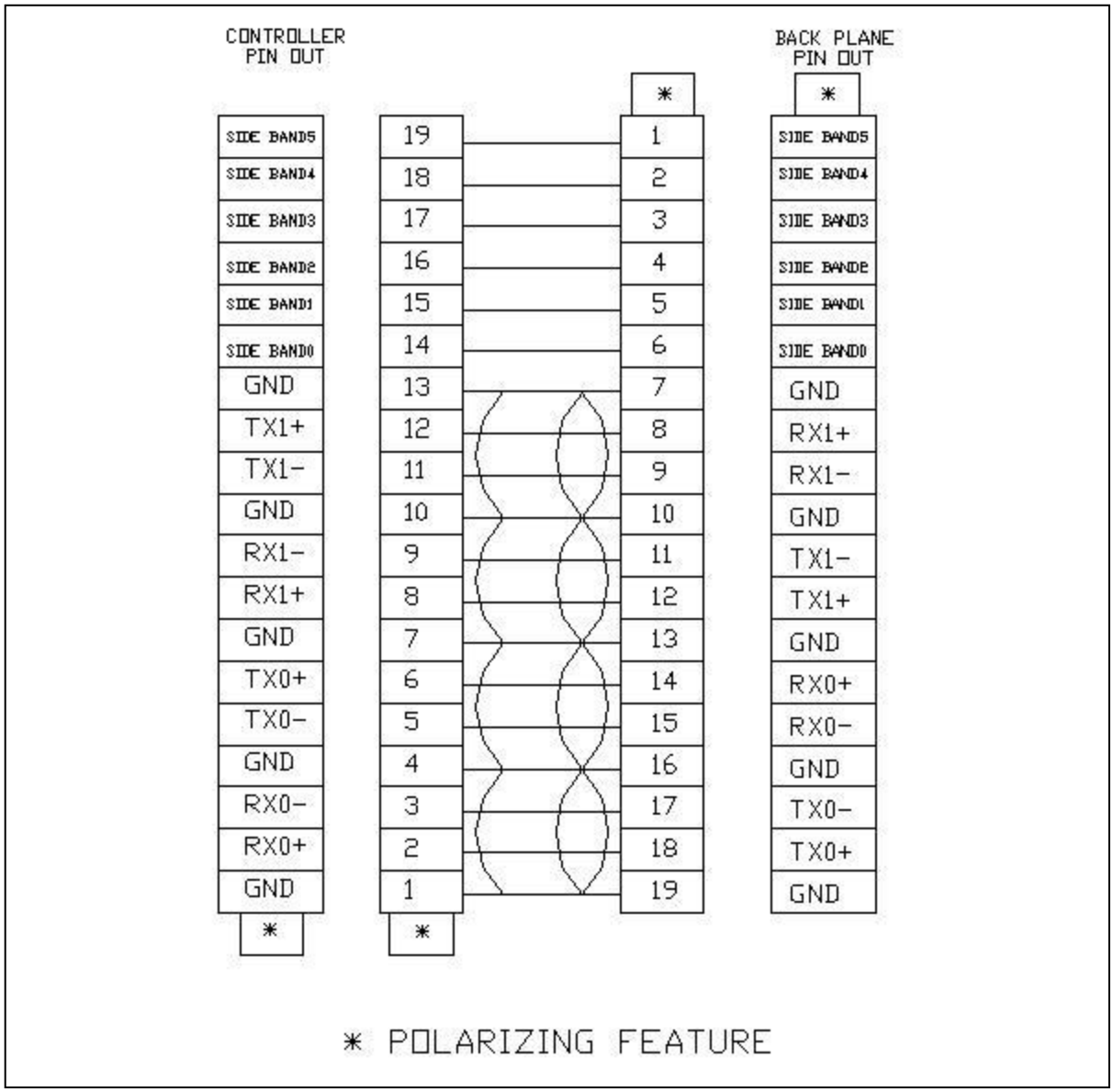


Figure 47 – 2 Lane Fanout Pin Assignments

6.1.12 Mini SATA Internal Multilane

This section defines standard cable assemblies and headers for connecting multiple Serial ATA links from a RAID host bus adapter to a backplane within the same enclosure or server, or for connecting multiple Serial ATA links from a host bus adapter to individual devices.

This cable/connector system is based on the SFF-8086 and SFF-8087 specifications. Both SFF-8086 and SFF-8087 specifications are also used by Serial Attached SCSI (SAS).

6.1.12.1 Conformance Criteria

- 4 lanes, Serial ATA signals.
- Cable length is 1 meter maximum.
- Either point to point with a high density connector on both ends of the cable or fanout with a high density connector on one end of the cable assembly and individual single lane connectors on the other end of the cable assembly.
- RX, TX, RX, TX pin sequencing to minimize crosstalk.
- Ground reference between each pair.
- Performance supporting 1.5 Gbps, 3.0 Gbps and 6.0 Gbps.
- Compliance points are at the ends of a mated cable interface. If additional interconnect media between host and device exists, that portion of the design is proprietary and shall ensure the mated cable interface compliance points are met.

6.1.12.1.1 Electrical Requirements

The Mini SATA Internal Multilane cable assembly shall operate at Gen1i, Gen2i and Gen3i levels and meet the electrical characteristics defined in Table 18. Since this cable is a Multilane and therefore has multi-aggressors, the additional requirement is to have crosstalk measured using the CXT method. The measured crosstalk shall meet the requirements listed in Table 18.

6.1.12.1.2 Component Descriptions

Detailed mechanical requirements are specified in SFF-8086 and SFF-8087.

6.1.12.1.3 Mechanical Requirements

Figure 48 shows the isometric drawings of the Mini SATA Internal Multilane cables and connectors. Refer to SFF-8086 and SFF-8087 for dimensions and mechanical details.
The Mini SATA Internal Multilane cables and connectors shall use the 36-circuit version plug and receptacle defined in SFF-8086 and SFF-8087.

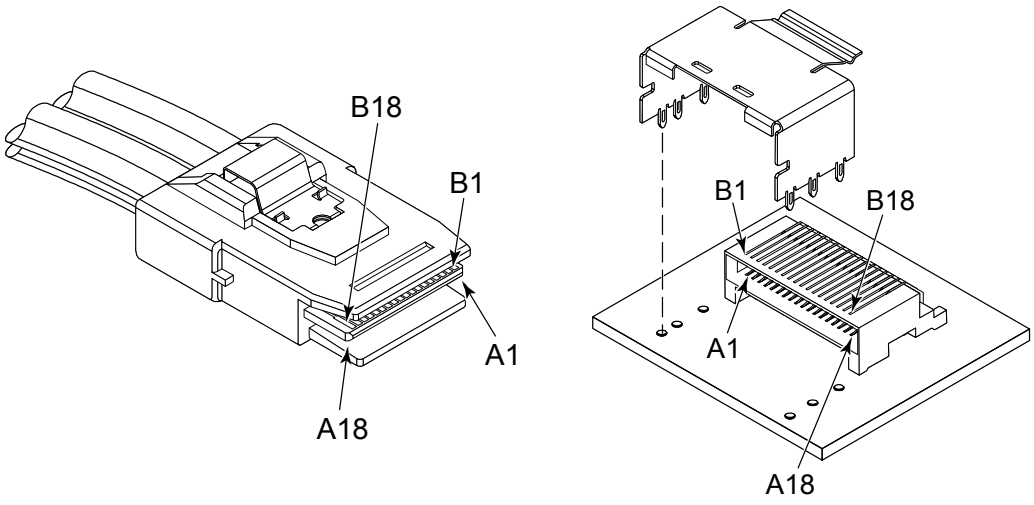


Figure 48 Isometric Drawings for Mini SATA Internal Multilane

6.1.12.2 Mini SATA Internal Multilane Pin Assignments

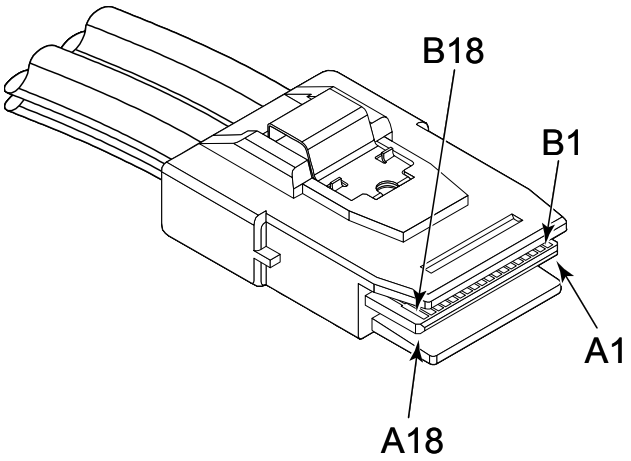
The Mini SATA Internal Multilane connector pin assignments are shown in Figure 49.

Pin assignments for sideband signals are based on the Internal Symmetrical Cable Assembly Implementation shown in Figure 50.

For host-to-backplane applications, sideband signals on the host are attached to the corresponding sideband signals on the backplane (e.g., SB0 of the host is attached to SB0 of the backplane). For host-to-host applications, sideband signals on one host are not attached to their corresponding sideband signals on the other host (e.g., SB0 of one host is attached to SB6 of the other host).

Figure 51 shows the Controller based fanout cable assembly.

Figure 52 shows the Backplane based fanout cable assembly



Signal	Signal pin
SIGNAL GND	A1
RX 0+	A2
RX 0-	A3
SIGNAL GND	A4
RX 1+	A5
RX 1-	A6
SIGNAL GND	A7
SB7 (host)/SB0 (backplane)	A8
SB3 (host)/SB1 (backplane)	A9
SB4 (host)/SB2 (backplane)	A10
SB5 (host)/SB6 (backplane)	A11
SIGNAL GND	A12
RX 2+	A13
RX 2-	A14
SIGNAL GND	A15
RX 3+	A16
RX 3-	A17
SIGNAL GND	A18
SIGNAL GND	B1
TX 0+	B2
TX 0-	B3
SIGNAL GND	B4
TX 1+	B5
TX 1-	B6
SIGNAL GND	B7
SB0 (host)/SB7 (backplane)	B8
SB1 (host)/SB3 (backplane)	B9
SB2 (host)/SB4 (backplane)	B10
SB6 (host)/SB5 (backplane)	B11
SIGNAL GND	B12
TX 2+	B13
TX 2-	B14
SIGNAL GND	B15
TX 3+	B16
TX 3-	B17
SIGNAL GND	B18

Figure 49 Mini SATA Internal Multilane Connector Pin Assignments

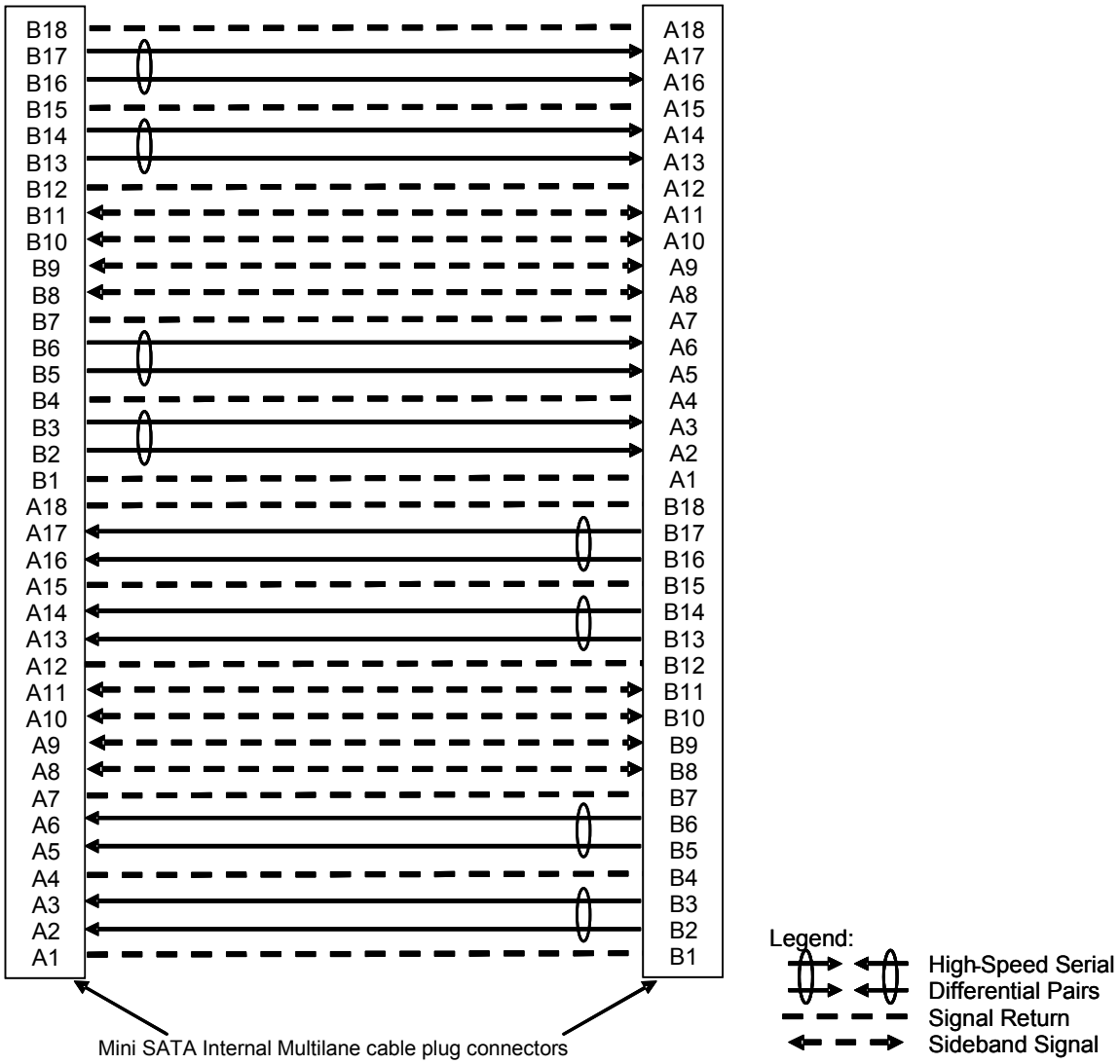
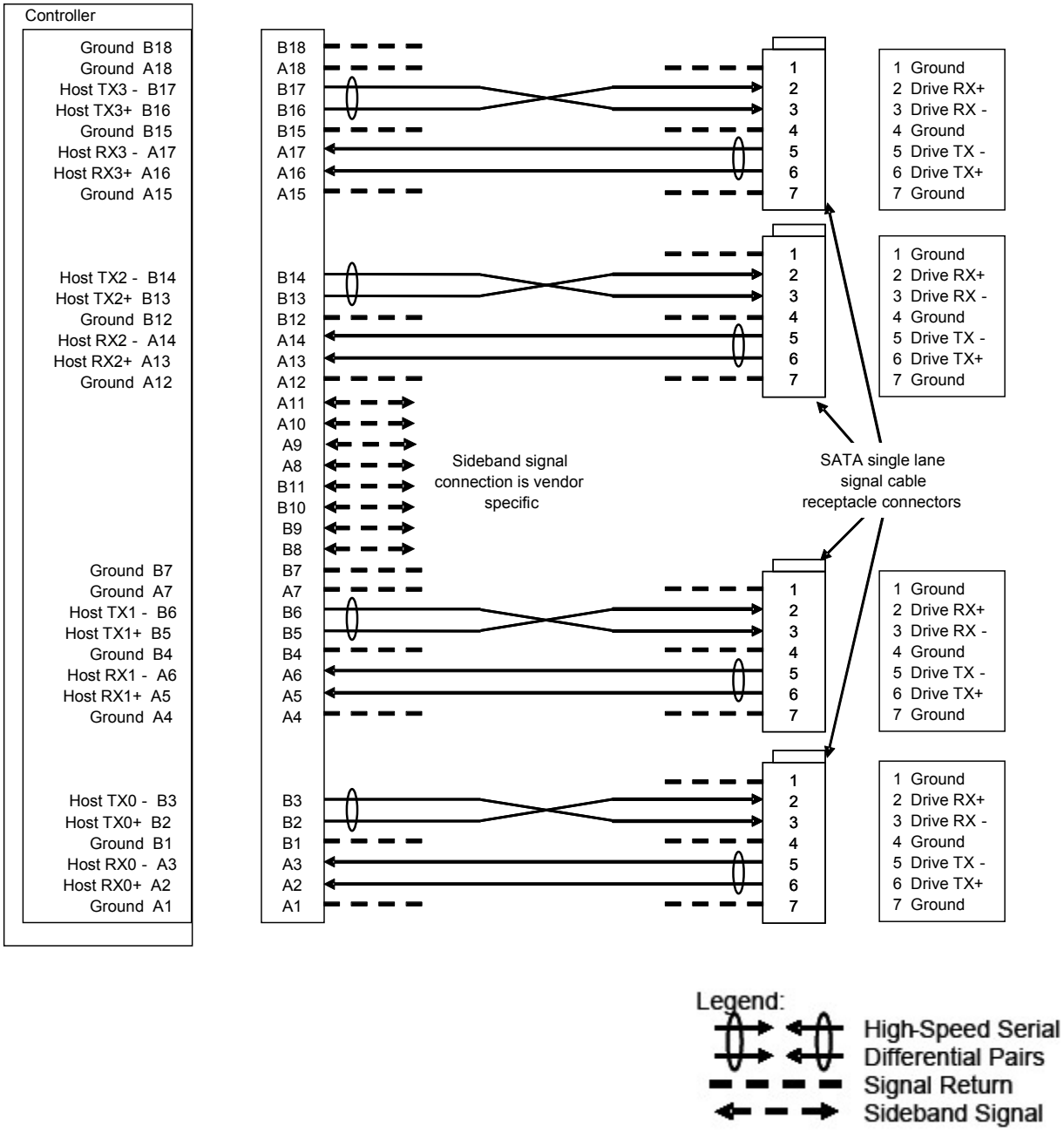
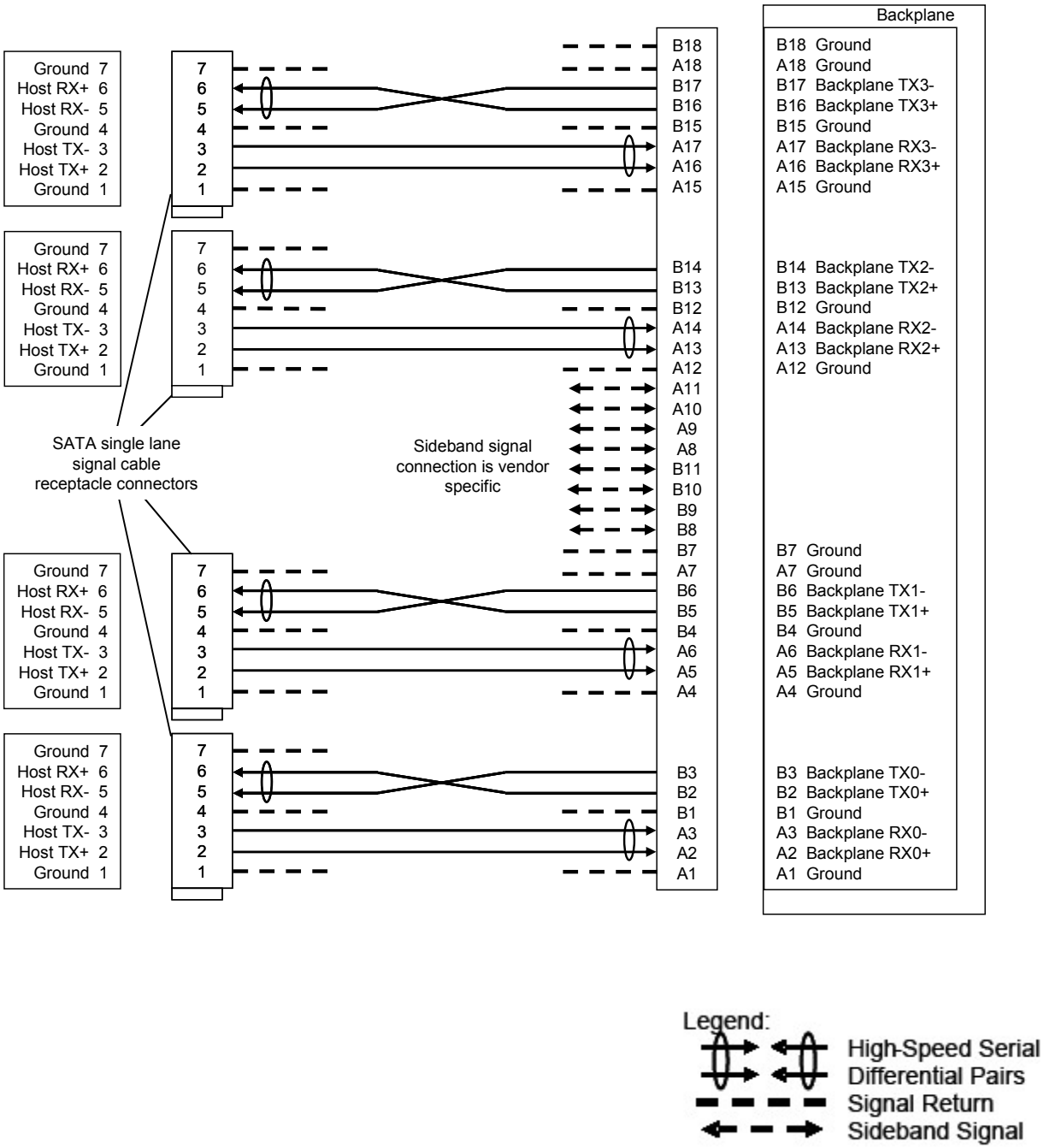


Figure 50 Mini SATA Internal Multilane System, Symmetric Cable Implementation



Note: The cable assembly shall connect each signal return on one end to at least one signal return on the other end. The cable assembly may connect one or more of the signal returns together.

Figure 51 Mini SATA Internal Multilane System, Controller based fanout cable implementation



Note: The cable assembly shall connect each signal return on one end to at least one signal return on the other end. The cable assembly may connect one or more of the signal returns together.

Figure 52 Mini SATA Internal Multilane System, Backplane based fanout cable implementation

6.2 Internal Micro SATA Connector for 1.8” HDD

This section provides capabilities required to enable a smaller Serial ATA 1.8” HDD (hard disk drive).

The definition supports the following capabilities:

- Supports Gen1 (1.5 Gbps) and Gen2 (3.0 Gbps) transfer rates
- Support for backplane (direct connect) and cable attachment usage models
- Support for hot plug in backplane (non-cabled) applications
- Support for 8.0 and 5.0 mm slim 1.8” form factor HDDs
- Support of 3.3 V with 5 V to meet future product requirements
- Support optional pins, P8 and P9 for vendor specific use

6.2.1 Usage model

The internal micro SATA connector may be used for the mobile usage model defined in section 5.2.10 The definition only supports internal 8.0 mm and 5.0 mm slim 1.8” form factor HDDs.

6.2.2 General description

The internal micro SATA connector is designed to enable connection of a slim 1.8” form factor HDD to the Serial ATA interface.

The internal micro Serial ATA connector uses the 1.27 mm pitch configuration for both the signal and power segments. The signal segment has the same configuration as the internal standard Serial ATA connector. The power segment provides the present voltage requirement support of 3.3 V, and includes a provision for a future voltage requirement of 5 V. In addition, there is a reserved pin, P7. Finally, there are two optional pins, P8 and P9, for vendor specific use.

The internal micro SATA connector is designed with staggered pins, for hot plug backplane (non-cabled) applications.

A special power segment key is located between pins P7 and P8. This feature prevents insertion of other Serial ATA cables.

Care should be taken in the application of this device so that excessive stress is not exerted on the device or connector. Backplane configurations should pay particular attention so that the device and connector are not damaged due to excessive misalignment.

6.2.3 Connector location

The internal micro SATA connector location on the HDD is shown in Figure 53 and Figure 54 for reference purposes. See SFF-8144 for form factor definition and connector location.

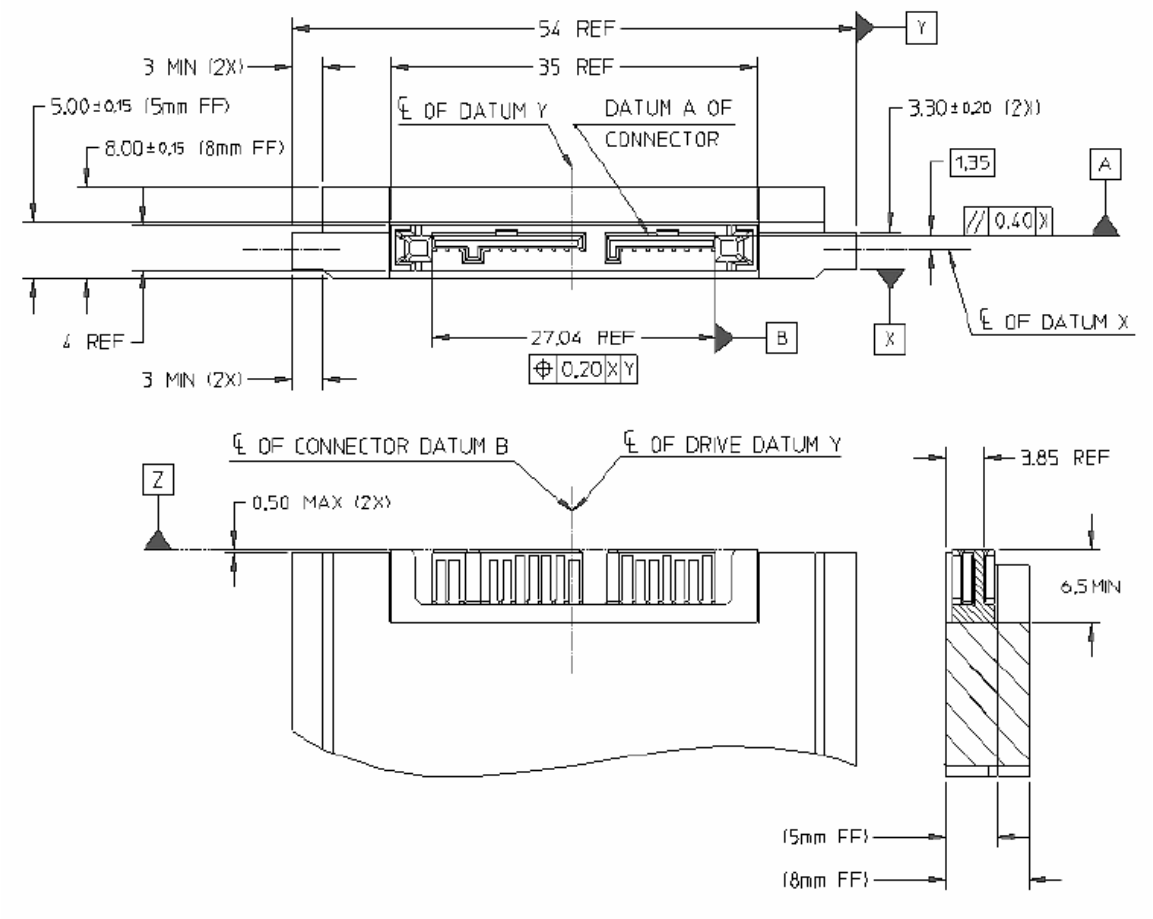


Figure 53 Device internal micro SATA connector location for 1.8” HDD

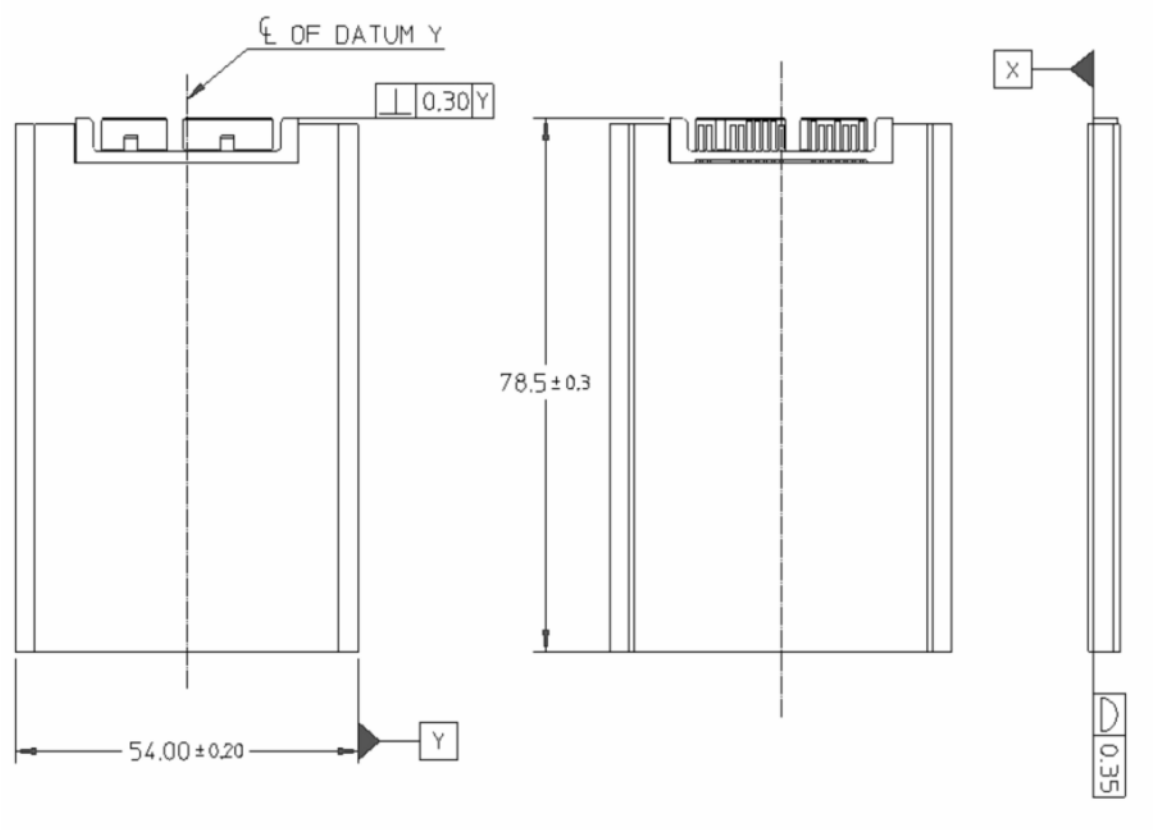
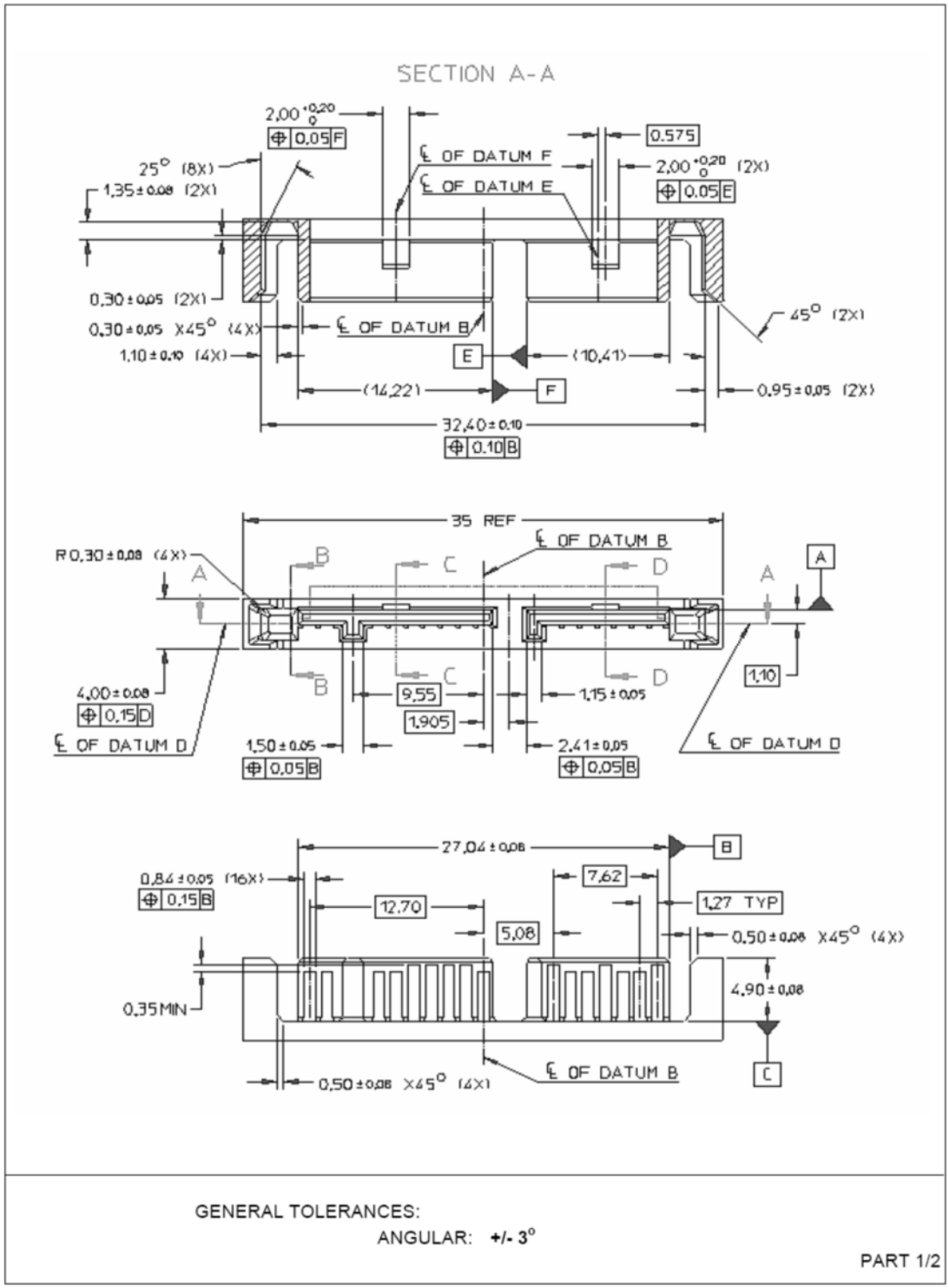


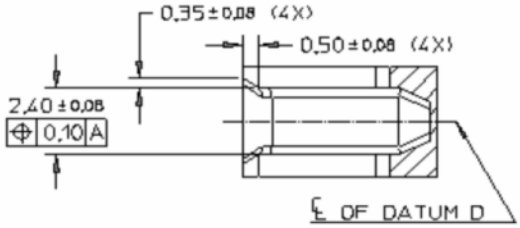
Figure 54 Device internal micro SATA connector location for 1.8” HDD

6.2.4 Mating interfaces

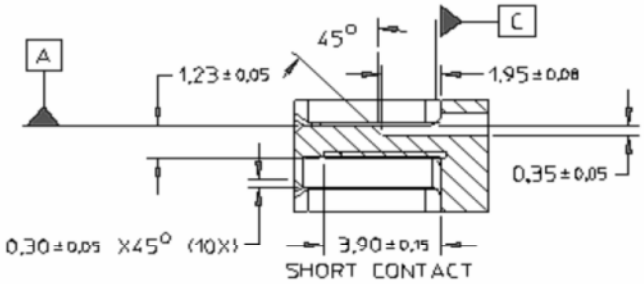
6.2.4.1 Device internal micro SATA plug connector

Figure 55 defines the interface dimensions for the internal micro SATA device plug connector with both signal and power segments.

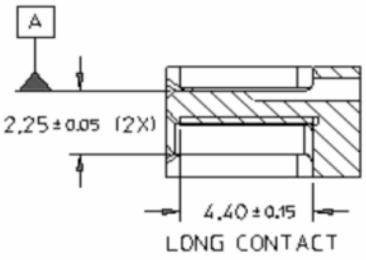




SECTION B-B



SECTION C-C



SECTION D-D

GENERAL TOLERANCES:

ANGULAR: +/- 3°

PART 2/2

Figure 55 Device internal micro SATA plug connector

6.2.4.2 Internal micro SATA backplane connector

Figure 56 defines the interface dimensions for the internal micro SATA backplane connector.

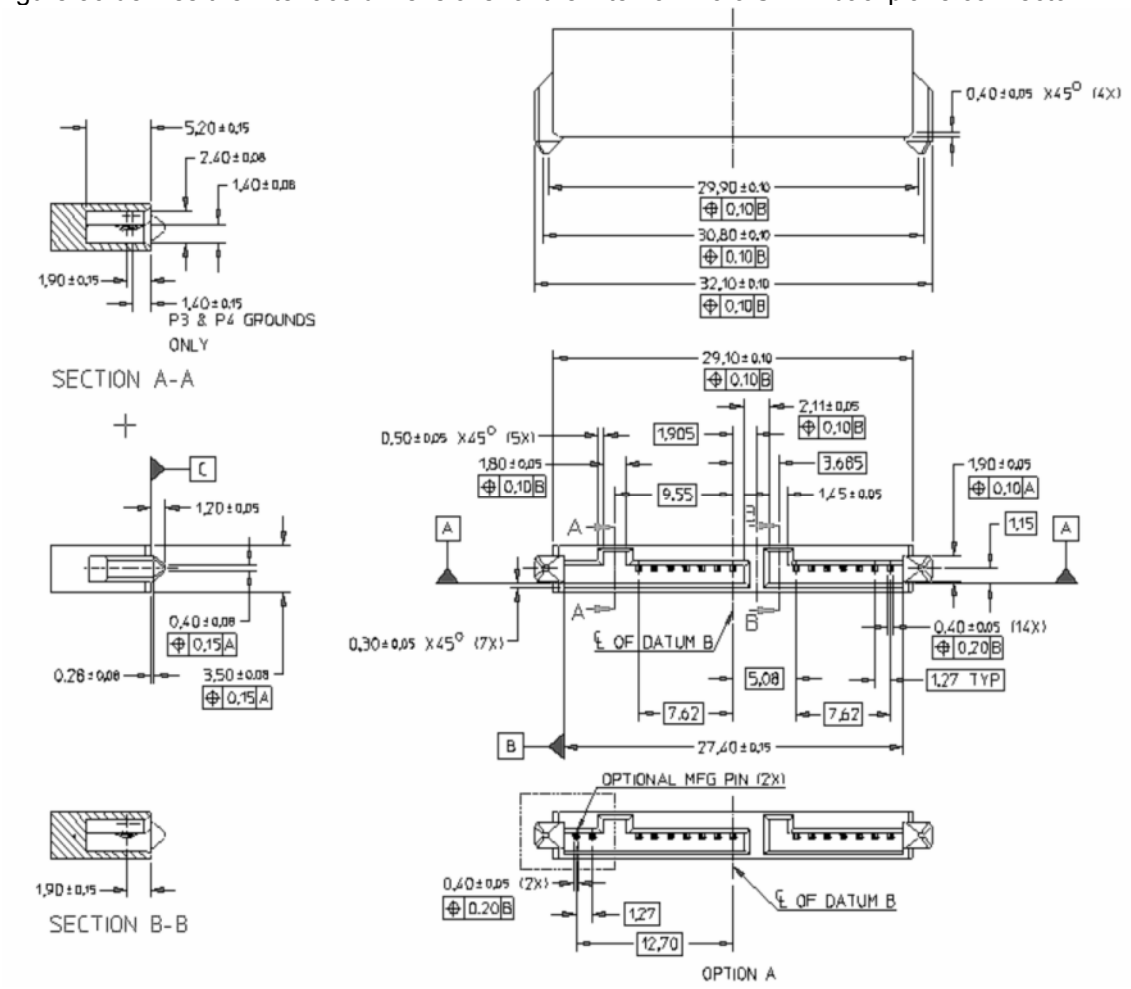


Figure 56 Internal micro SATA backplane connector

6.2.4.3 Internal micro SATA power receptacle connector

Figure 57 defines the interface dimensions for the internal micro SATA power receptacle connector.

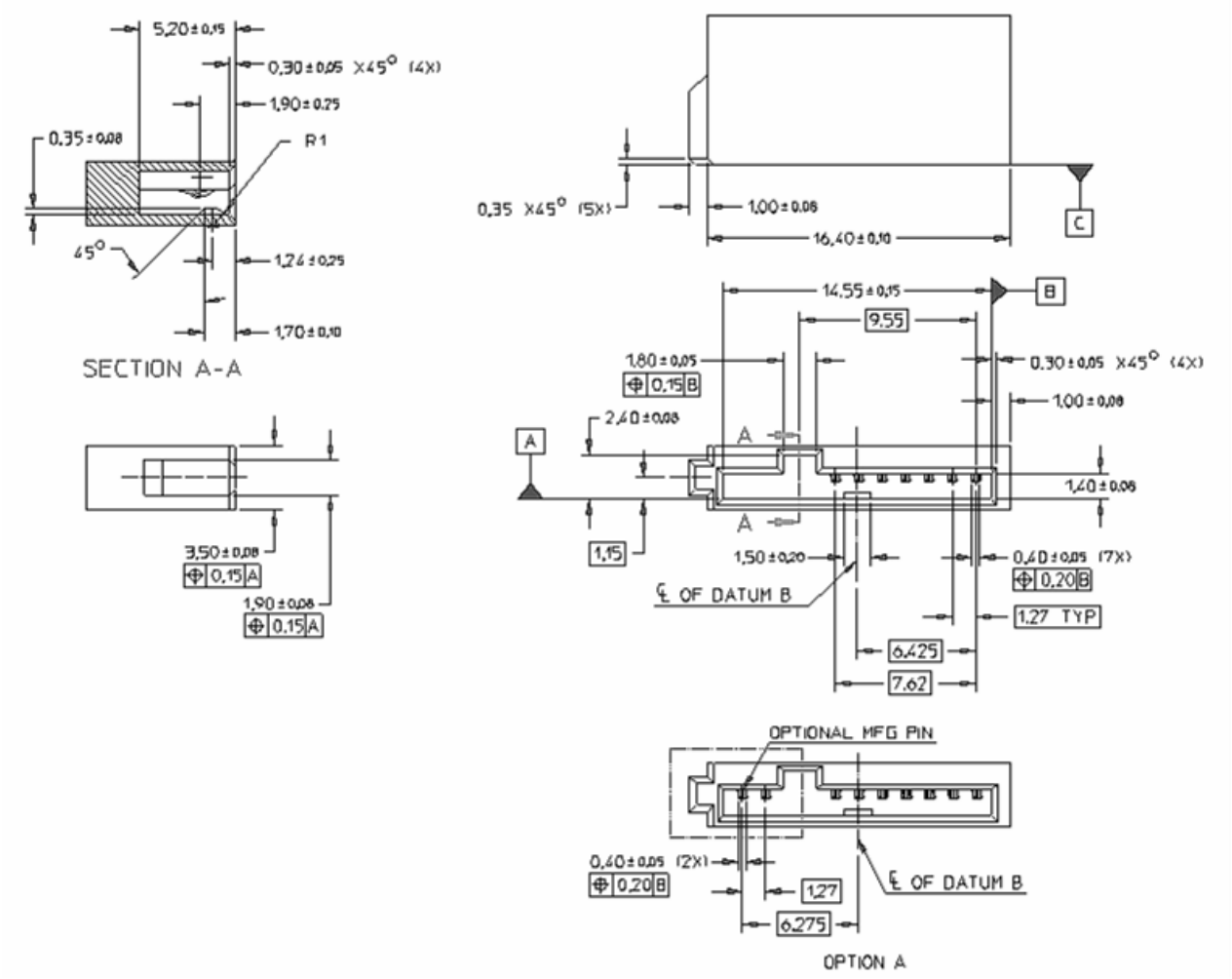


Figure 57 Internal micro SATA power receptable connector

6.2.4.4 Internal micro SATA connector pair blind-mate misalignment capability

The maximum blind-mate misalignment capabilities are ±1.50 mm and ±1.00 mm, respectively, for two perpendicular axes illustrated in Figure 58. Any skew angle of the plug, with respect to the receptacle, reduces the blind-mate capabilities.

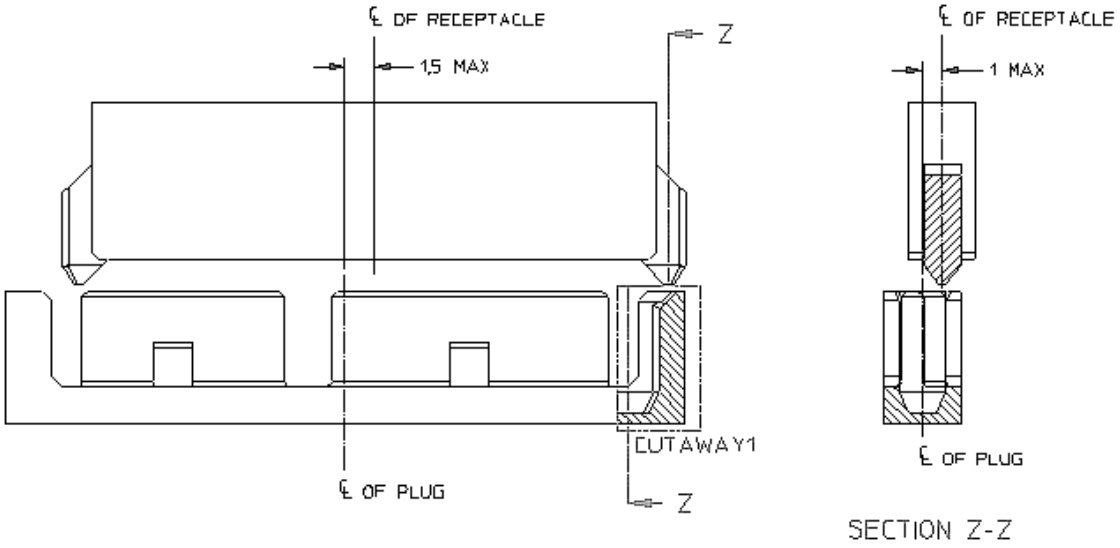


Figure 58 Internal micro SATA connector pair blind-mate misalignment capability

6.2.4.5 Internal micro SATA pin signal definition and contact mating sequence

Table 10 details the pin names, types and contact order of the two internal micro SATA Plug options. A brief description is also included for signal, ground and power pins. There are total of 7 pins in the signal segment and 9 pins in the power segment.

Table 10 – Signal and Power Internal Micro SATA Plug and Nominal Mate Sequence

	Name	Type	Description	Cable Usage[1,2]	Backplane Usage[2]
Signal Segment	Refer to Table 3.				
Spacing separate signal and power segments[3]					
Power Segment	P1	V_{33}	3.3 V Power	2nd Mate	3rd Mate
	P2	V_{33}	3.3 V Power, Pre-charge	1st Mate	2nd Mate
	P3	GND		1st Mate	1st Mate
	P4	GND		1st Mate	1st Mate
	P5	V_5	5 V Power, Pre-charge[4]	1st Mate	2nd Mate
	P6	V_5	5 V Power[4]	2nd Mate	3rd Mate
	P7	R	Reserved[5]	2nd Mate	3rd Mate
	Key	Key	Key	NC	NC
	P8	Optional	Vendor specific[6]	2nd Mate	3rd Mate
	P9	Optional	Vendor specific[6]	2nd Mate	3rd Mate

NOTE:
1. Although the mate order is shown, hot plugging is not supported when using the cable connector receptacle.
2. All mate sequences assume zero angular offset between connectors.
3. The signal segment and power segment may be separate.
4. The 5 V supply voltage pins are included to meet future product requirements and may optionally be provided on the power segment receptacle. Future revisions of this specification may require 5 V supply voltage be provided.
5. The corresponding pin to be mated with pin P7 in the power Internal Micro receptacle connector shall always be grounded.
6. No connect on the host side.

6.2.4.6 Internal micro SATA connector and cable assembly requirements and test procedures

The internal micro SATA connector and cable shall meet the requirements as defined for standard internal SATA cables and connectors in section 6.1.10, with the exceptions listed in Table 11.

Table 11 – Unique Connector Mechanical Testing Procedures and Requirements

Removal force	EIA-364-13	2.5 N Min. after 500 cycles

Backplane connector	Measure the force necessary to unmate the connector assemblies at a max. rate of 12.5 mm per minute.	
Insertion force Cabled power connector (non-latching)	EIA-364-13 Measure the force necessary to mate the connector assemblies at a max. rate of 12.5 mm per minute.	45 N Max.
Removal force Cabled power connector (non-latching)	EIA-364-13 Measure the force necessary to unmate the connector assemblies at a max. rate of 12.5 mm per minute.	10 N Min. for cycles 1 through 5 8 N Min. through 50 cycles

6.3 Internal Slimline cables and connectors

This section provides capabilities required to enable Serial ATA in “Slimline” optical disk drives.

The definition supports the following capabilities:

- Supports Gen1 (1.5 Gbps) and Gen2 (3.0 Gbps) transfer rates
- Support for backplane (direct connect) and cable attachment usage models
- Support for warm plug
- Support for 12.7 mm, 9.5 mm and 7.0 mm Slimline devices
- 5 V only power delivery

The definition has the following constraints:

- No device activity signal support
- Analog audio is not supported
- External cable and connector are not supported
- No hot plug support
- Warm plug support is usage model dependent

6.3.1 Usage Models

Support for three usage models. Internal fixed bay (direct attach), removable bay, and internal cable. The requirements for each usage model are specified in this section.

Internal Fixed Bay requirements:

- Support for 12.7, 9.5 and 7.0 mm slimline devices
- Direct attach support for 1.8” HDD not required
- Direct attach support for 2.5” HDD not required
- Minimum footprint
- Device presence detection not required

Removable Bay requirements:

- Support for 12.7 mm, 9.5 mm and 7.0 mm slimline devices
- Direct attach support for 1.8” HDD not required
- Direct attach support for 2.5” HDD not required. Attachment in carrier adapter shall conform to cabled usage model for Gen1i, Gen2i and Gen3i.
- Warm plug and blind mate required
- Un-powered device presence detection required
- No floppy support required

- No battery connector support required
- Support 500 insertions and removals. Design is scaleable to higher cycle counts.

Internal Cable requirements:

- Support for 12.7 mm, 9.5 mm and 7.0 mm slimline devices
- Direct attach support for 1.8" HDD not required
- Direct attach support for 2.5" HDD not required
- Blind mate not supported
- Device presence detection not supported
- Support passive and latching retention mechanisms
- Support internal single lane cables and connectors, defined in section 6.1
- Support 50 insertions and removals

6.3.2 General description

The Slimline connector is designed to enable connection of a "slimline" form factor drivedevice to the Serial ATA interface. The connector design accommodates a latching option for 9.5 mm and 12.7 mm slimline devices. A latching option may not be accommodated in 7 mm slimline devices. This internal Serial ATA connector uses the 5 mm (ref) connector height as low profile.

The connector is designed to fit into the presently used PATA (Parallel ATA) connector slimline designs with almost no changes to the slimline device case, media tray, or internal mechanics. It is anticipated that a simple replacement of the PATA PC board with a SATA controller and connector in the present slimline device designs may be possible.

The connector preserves the design of the signal portion of the present SATA connector and accommodates presently available SATA signal cables.

The power portion of the connector was reduced to six pins. Both +12 V and +3.3V were removed from the connector leaving +5 V as the sole supply voltage.

The standard +A, -A, +B, and –B signals are on the signal portion of the connector. In addition to +5 V and GND, the following signals were added to the power portion:

- DP – Device Present – Active low signal indicating device connect to the host. The device shall connect the DP pin to ground with a resistance of 1K ohms and a maximum tolerance of ± 10% ohms. This signal is not supported in a cabled environment. If a cabled connection is used and the cable is held in a fixed position, the cable shall follow backplane requirements. Host connection to the DP pin is optional. If un-used, no connection is allowed. If the host requires the use of the DP function, the maximum current it shall source is 4mA and the minimum is 0 mA.
- MD – Manufacturing Diagnostic – Signal pin used by device vendors during device testing. No host connection is allowed, device connection is optional.

6.3.3 Connector location and keep out zones

This section describes the location of the connector in the slimline device referenced from the standard locations. Keep out zones are also defined to allow blind mate capability.

6.3.3.1 Location

Table 12 shows the connector location references for the 7.0 mm, 9.5 mm and 12.7 mm slimline devices.

Table 12 – Slimline Connector Location References

Slimline Drive	Horizontal	Vertical	Depth
7.0 mm Slimline	Drive left edge to connector CL	Drive bottom edge to connector tongue top edge	Drive back edge to connector back wall
9.5 mm Slimline	Drive left edge to connector CL	Drive bottom edge to connector tongue top edge	Drive back edge to connector back wall
12.7 mm Slimline	Drive left edge to connector CL	Drive bottom edge to connector tongue top edge	Drive back edge to connector back wall

6.3.3.2 Keep out zones

The minimum panel opening is the maximum connector size plus the positional tolerance of the connector location within the slimline device.

The internal SATA connector location on the 7 mm slimline ODD is shown in Figure 59 for reference purposes only. Additional details may be found in SFF-8553.

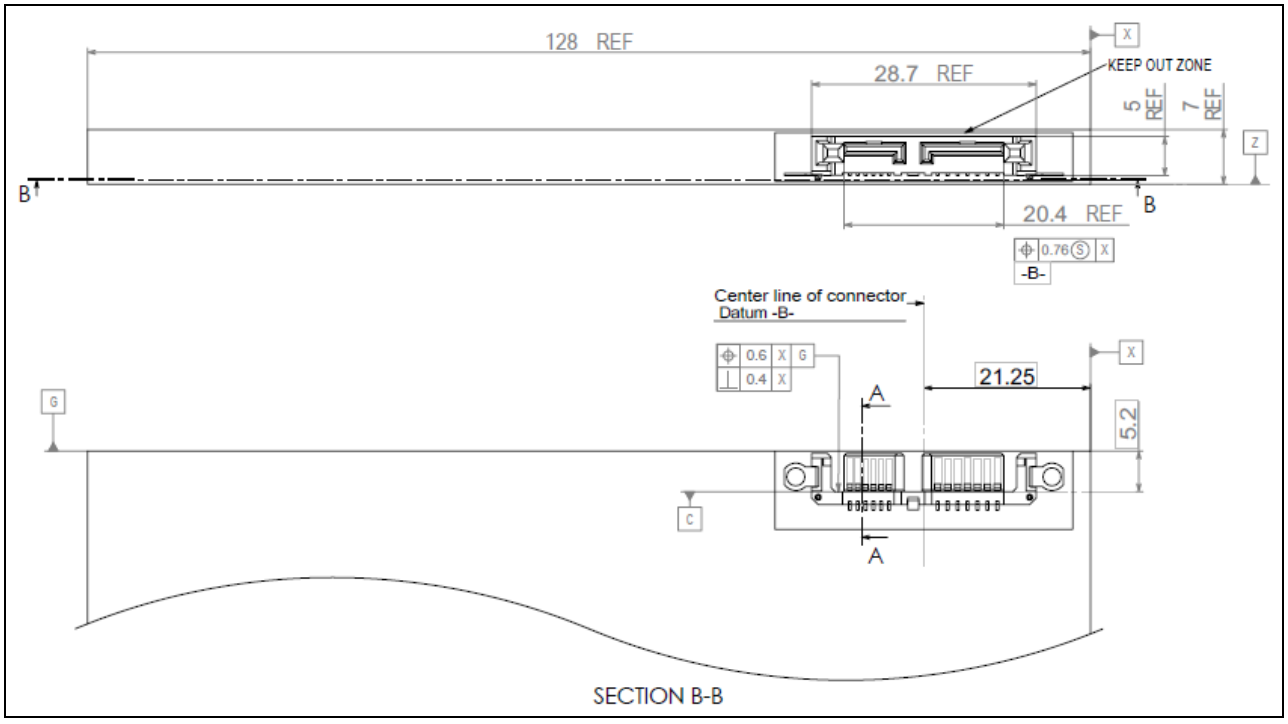


Figure 59 - 7.0 mm Slimline Drive Connector Locations

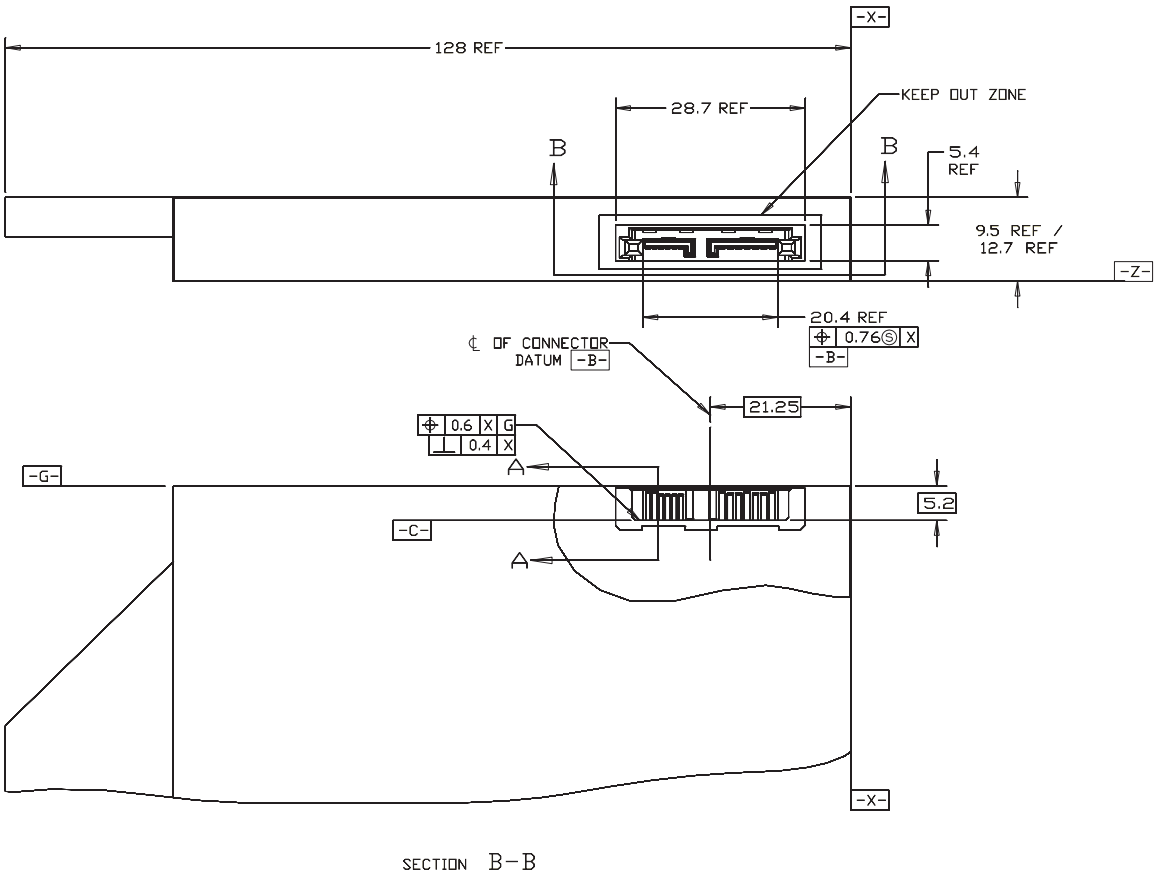


Figure 60 - 9.5 mm/12.7 mm Slimline Drive Connector Locations

7.0 mm Slimline Devices

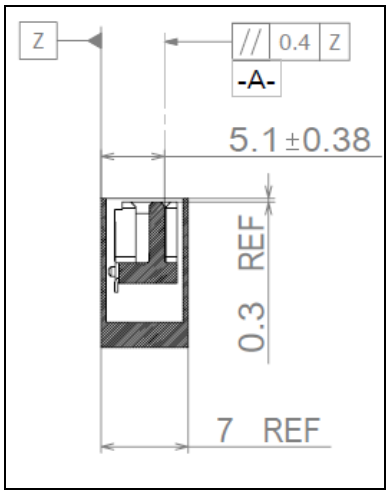


Figure 61 - 7.0 mm Slimline Drive Connector Location (Section A-A)

6.3.3.2.1 9.5 mm Slimline Drives

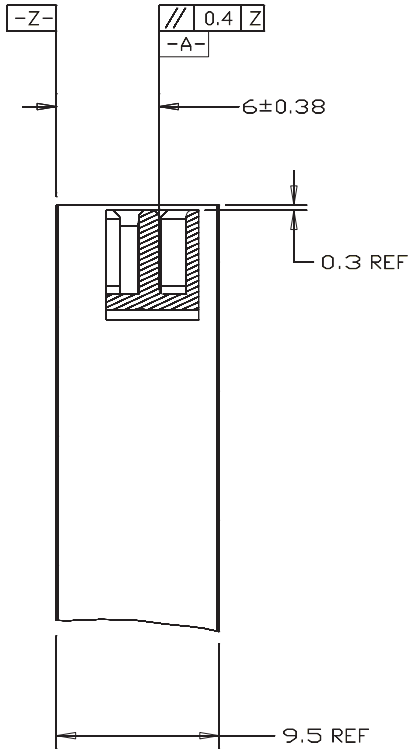


Figure 62 - 9.5 mm Slimline Drive Connector Location (Section A-A)

6.3.3.2.2 12.7 mm Slimline Drives

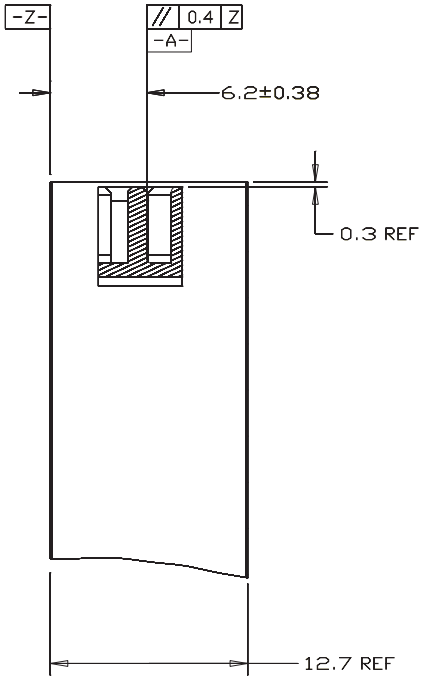


Figure 63 - 12.7 mm Slimline Drive Connector Location (Section A-A)

6.3.4 Mating interfaces

Serial ATA connectors are defined in terms of their mating interface or front end characteristics only. Connector back end characteristics including PCB mounting features and cable termination features are not defined.

Certain informative dimensions may be defined to indicate specific assumptions made in the design to meet certain requirements of form, fit, and function.

6.3.4.1 7 mm Slimline Device plug connector

This is the connector used in the 7 mm slimline device. Figure 2 and Table 1 show the interface dimensions for the device plug connector with both signal and power segments. General tolerances on these drawings are +/-0.20 mm.

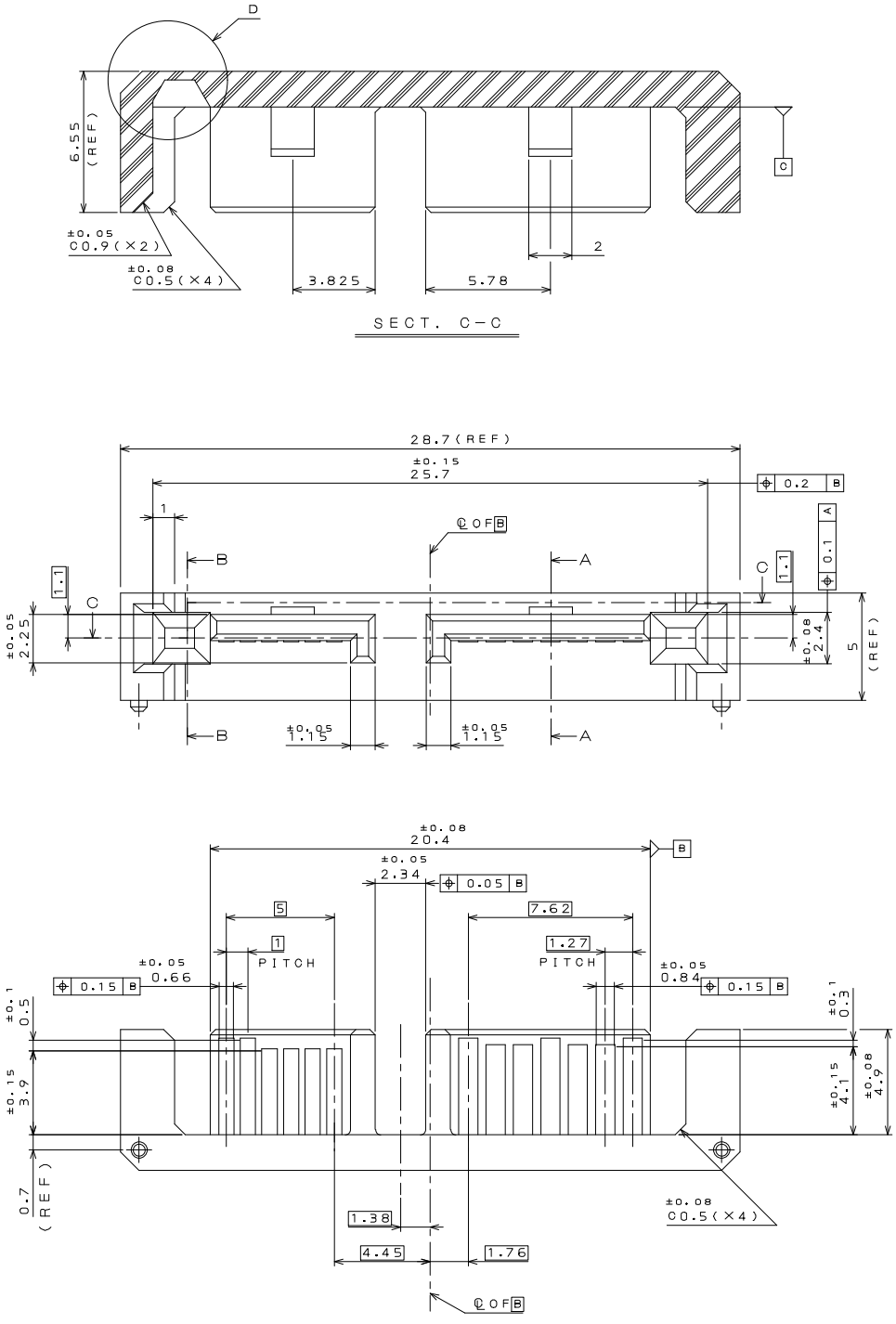


Figure 64 – 7 mm slimline device plug connector interface dimensions

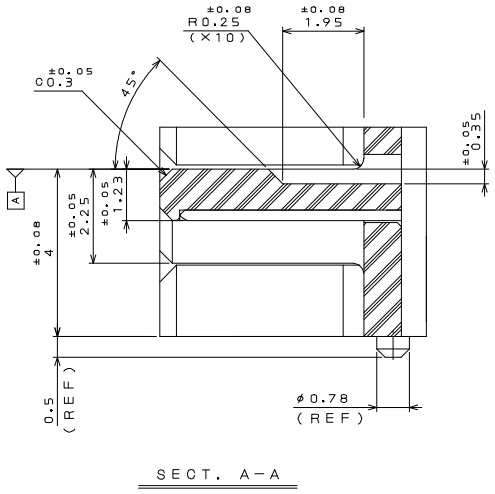


Figure 65 - 7.0 mm Slimline device plug connector interface dimensions Section A-A

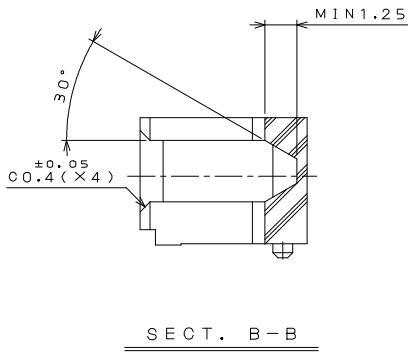


Figure 66 - 7.0 mm Slimline device plug connector interface dimensions Section B-B

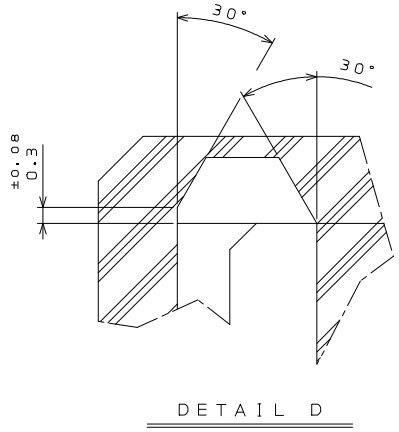


Figure 67 - 7.0 mm Slimline device plug connector interface dimensions detail D

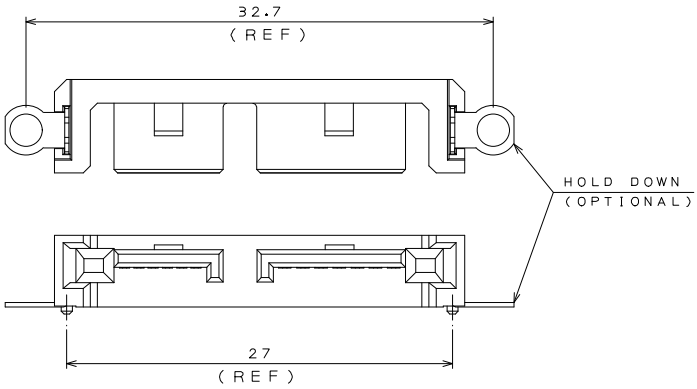


Figure 68 - 7.0 mm Slimline device plug connector interface dimensions optional hold down mounting

6.3.4.1.1 Connector pin signal definition

Refer to Table 13 for connector pin assignment and description.

6.3.4.1.2 Housing and contact electrical and mechanical requirement

The internal SATA connector and cable shall meet the requirements as defined for standard internal SATA cables and connectors in 6.1.10.1.

6.3.4.1.3 Connector and cable assembly requirements and test procedure

The connector and cable assembly requirements and test procedure shall confirm to those in 6.3.7.

6.3.4.2 9.5 mm and 12.7 mm Slimline Device plug connector

This is the connector used in the slimline device.

Figure 69 and Table 13 show the interface dimensions for the device plug connector with both signal and power segments. General tolerances on these drawings is +/-0.20 mm.

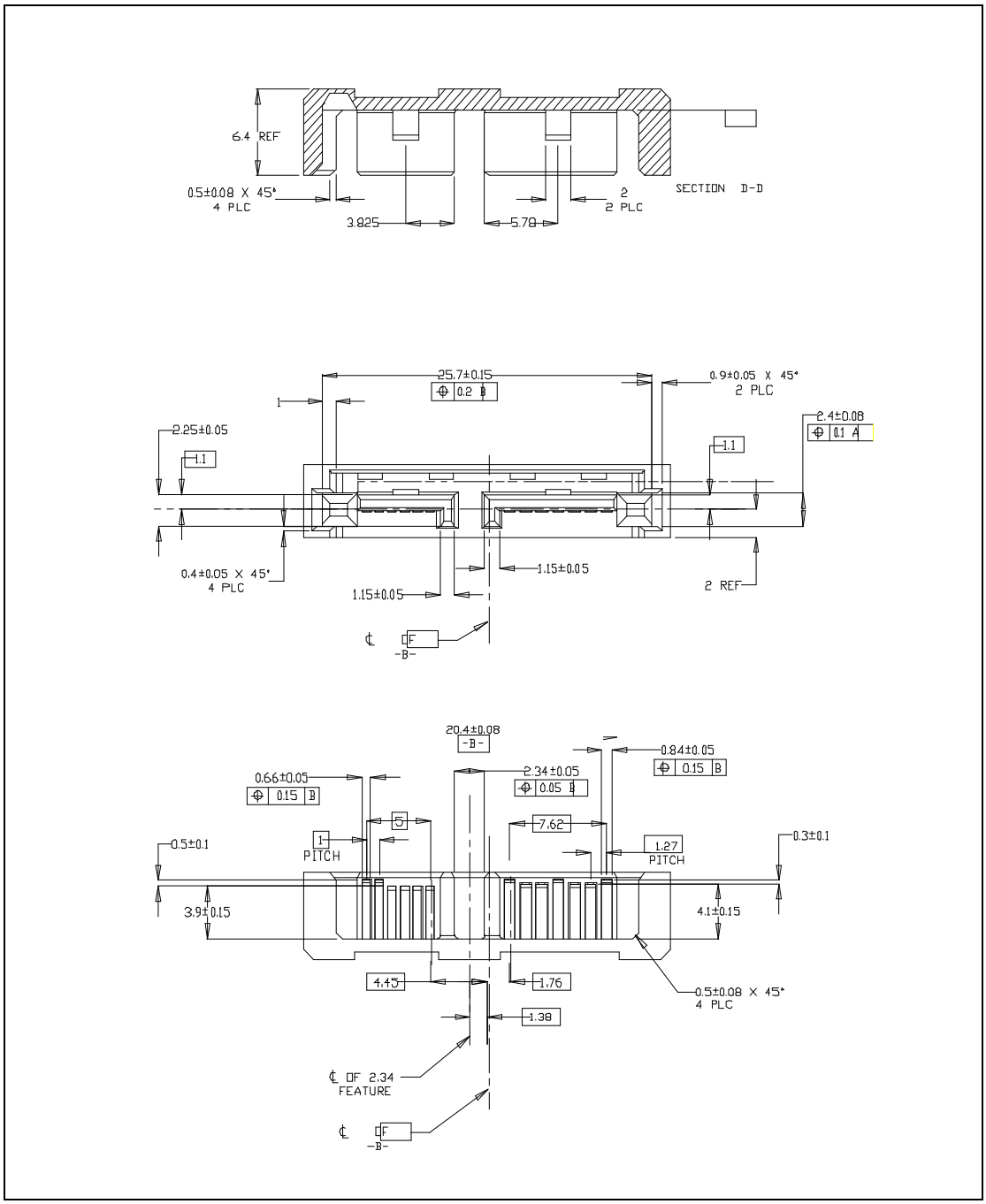


Figure 69 – Slimline Device plug connector interface dimensions

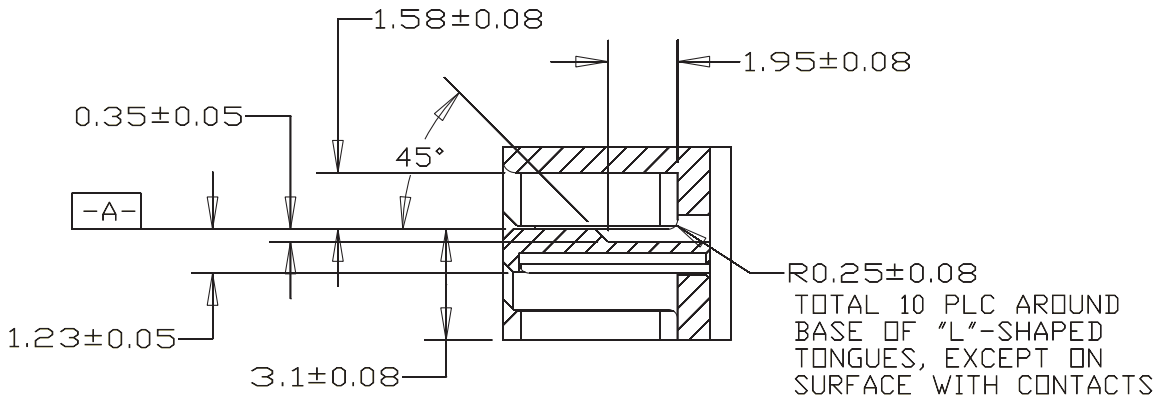


Figure 70 - Slimline Device plug connector interface dimensions (Section A-A)

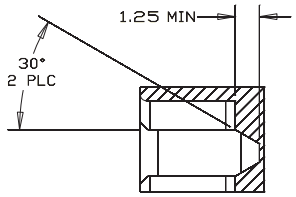


Figure 71 - Slimline Device plug connector interface dimensions (Section B-B)

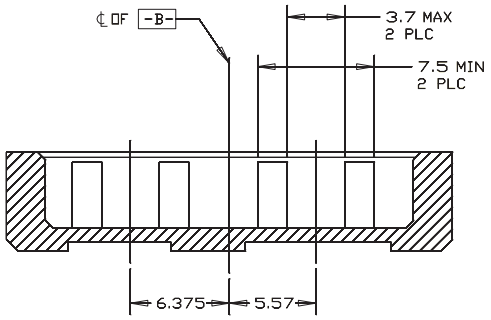


Figure 74 - Slimline Device plug connector interface dimensions (Section G-G)

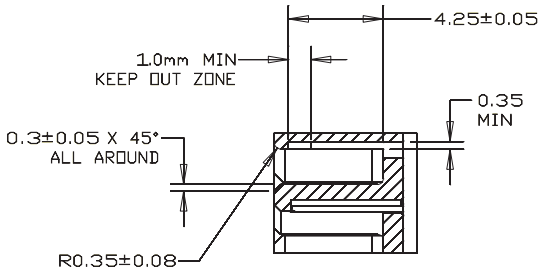


Figure 72 - Slimline Device plug connector interface dimensions (Section C-C)

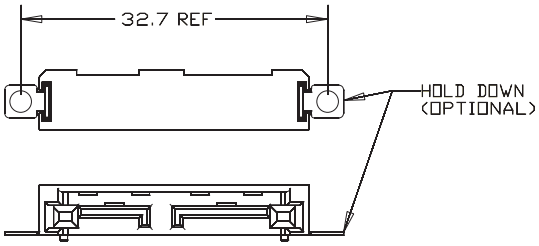


Figure 75 – Slimline Device Plug Connector Optional Hold Down Mounting

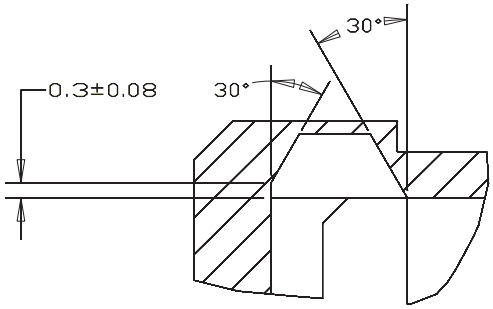


Figure 73 - Slimline Device plug connector interface dimensions (Detail F)

There are total of 7 pins in the signal segment and 6 pins in the power segment. The pin definitions are shown in Table 13. Note that the pin is numbered from the pin furthest from the power segment.

Table 13 – Slimline Device plug connector pin definition

	Name	Type	Description	Cable Usage	Backplane Usage
Signal Segment	Refer to Table 3.				
Signal Segment "L"					
Central Connector Gap					
Power Segment "L"					
Power Segment	P1	DP	Device Present	3rd mate	3rd mate
	P2	+5 V		2nd mate	2nd mate
	P3	+5 V		2nd mate	2nd mate
	P4	MD	Manufacturing Diagnostic	2nd mate	2nd mate
	P5	Gnd		1st mate	1st mate
	P6	Gnd		1st mate	1st mate
Power Segment Key					

NOTE:
1. All pins are in a single row with 1.00 mm (.039") pitch on the power segment portion.
2. Ground pins in the Serial ATA slimline device plug power segment (connector pins P5 and P6) shall be bussed together on the Serial ATA slimline device.
3. The connection between the Serial ATA slimline device signal ground and power ground is vendor specific.
4. The DP and MD signals shall be referenced to the power portion ground pins, P5 and P6.
5. The 5 V power delivery pins in the Serial ATA slimline device plug power segment (connector pins P2 and P3) shall be bussed together in the Serial ATA slimline device.

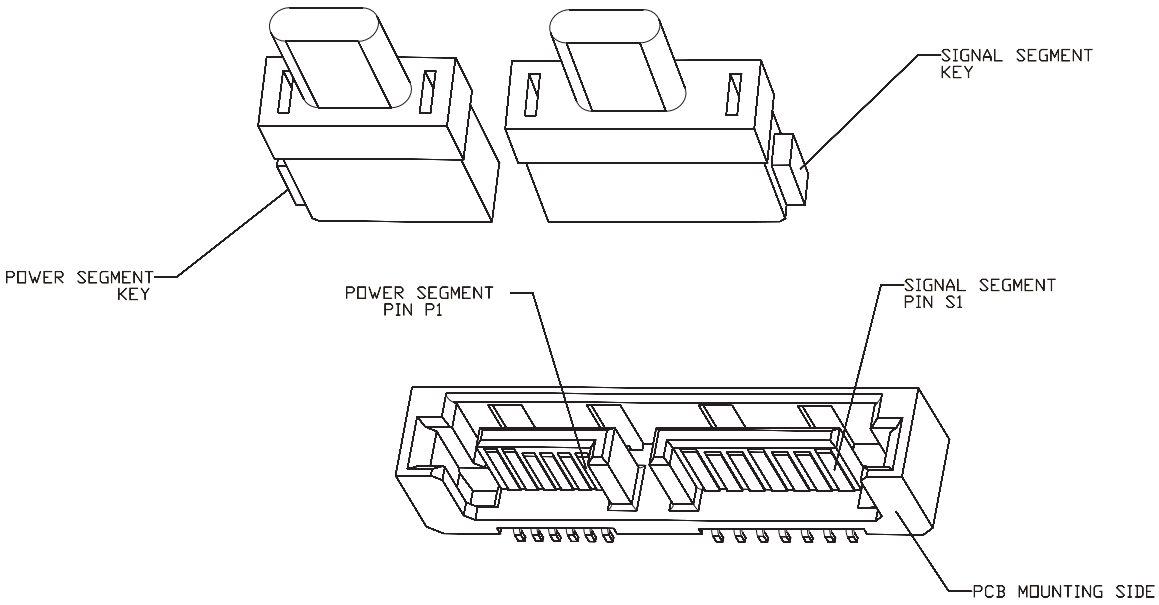


Figure 76 – Slimline Connector Pin and Feature Locations

6.3.4.2.1 Slimline Signal cable receptacle connector

The standard SATA signal cable receptacle is used. See section 6.1.4.

6.3.4.2.2 Slimline Power cable receptacle connector

The slimline power cable receptacle connector mates with the power segment of the device plug, bringing power to the device. Figure 77 shows the interface dimensions of the power receptacle connector. The pin out of the connector is the mirror image of the power segment of the device plug shown in Table 13. The MD and DP connector pins are optionally present in a cabled environment.

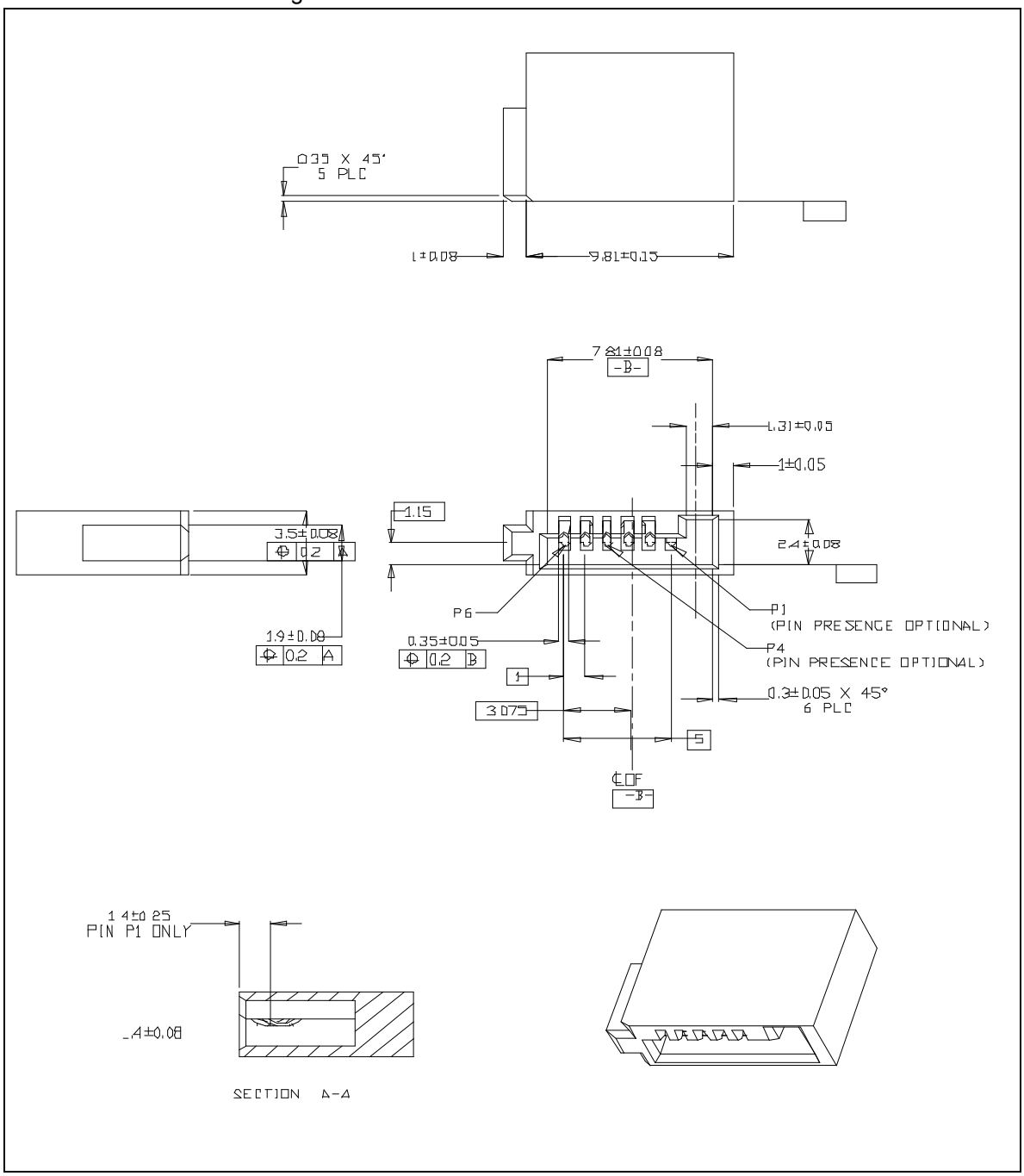


Figure 77 – Slimline Power receptacle connector interface dimensions

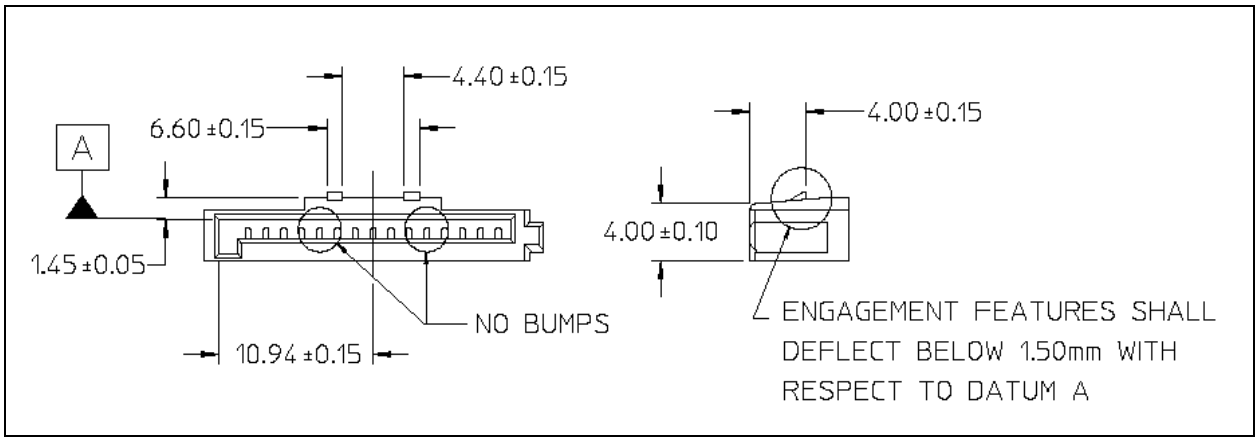


Figure 78 – Slimline Power receptacle connector Option with Latch

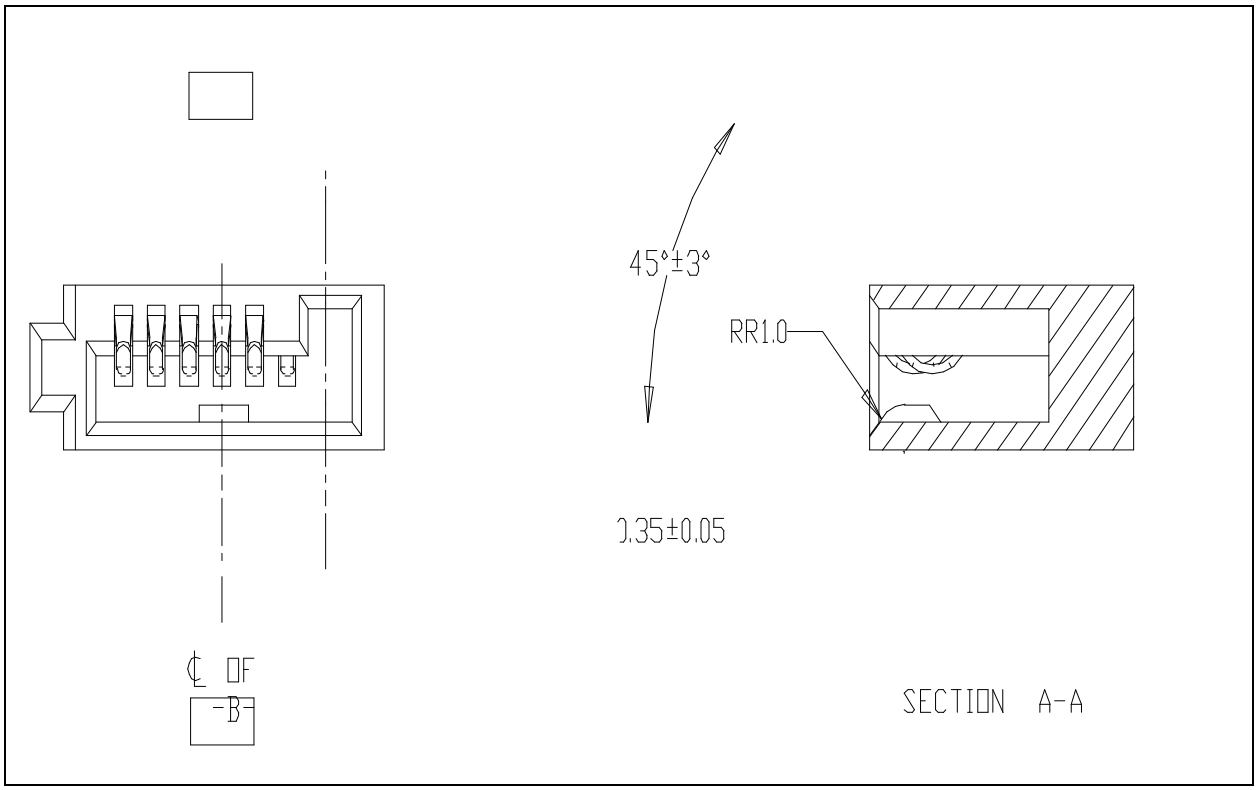


Figure 79 – Slimline Power receptacle connector Option with Bump

6.3.4.3 Slimline Host Receptacle Connector

The slimline 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 80. An appropriate external retention mechanism independent of the connector is required to keep the host PCB and the device in place. The slimline host receptacle connector is not designed with any retention mechanism.

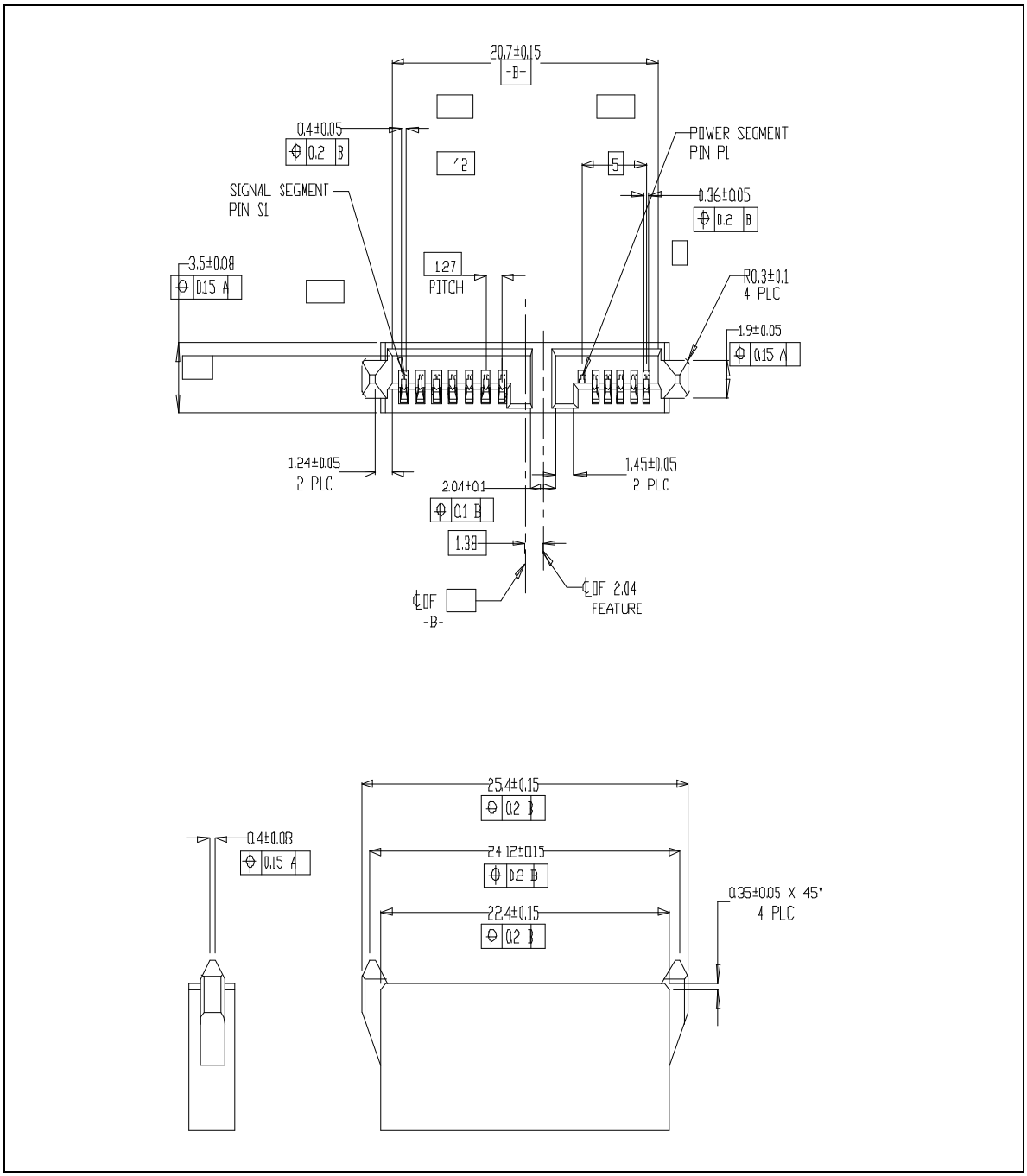


Figure 80 – Slimline Host receptacle connector interface dimensions

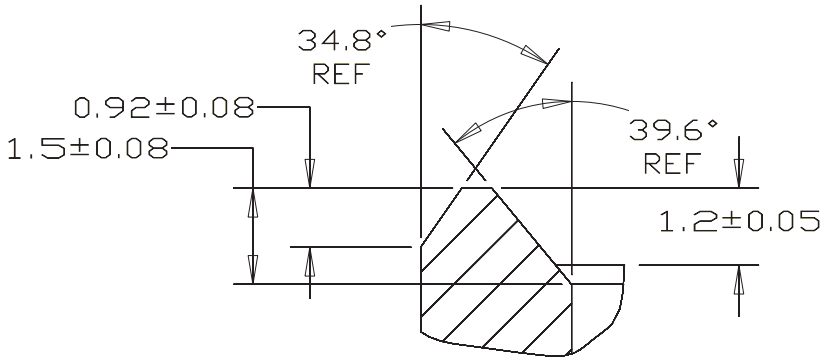


Figure 81 – Slimline Host receptacle connector interface dimensions Section C-C

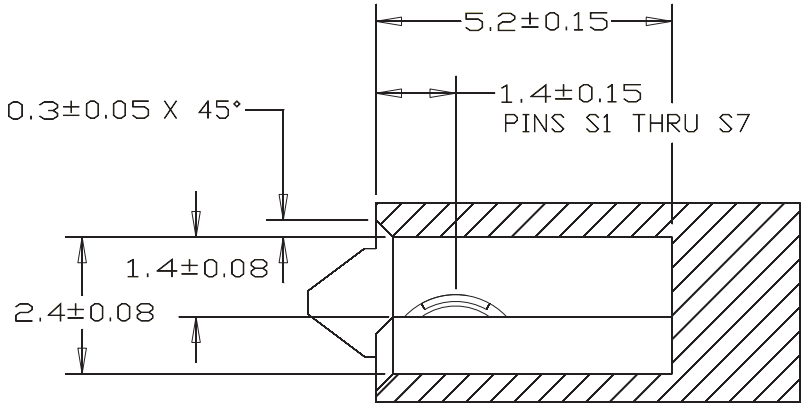


Figure 82 – Slimline Host receptacle connector interface dimensions Section X-X

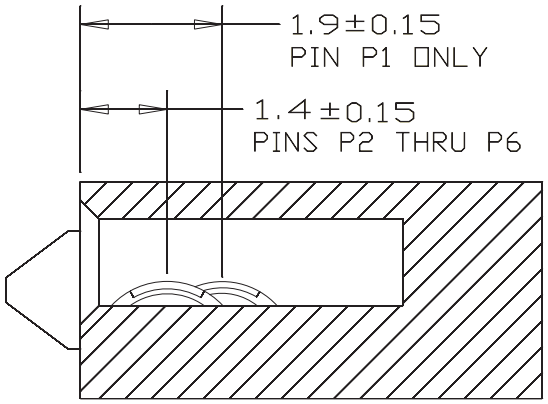


Figure 83 – Slimline Host receptacle connector interface dimensions Section Y-Y

6.3.5 Backplane connector configuration and blind-mating tolerance

The maximum blind-mate misalignment tolerances are ±1.50 mm and ±1.20 mm, respectively, for two perpendicular axes illustrated in Figure 84. Any skew angle of the plug, with respect to the receptacle, reduces the blind-mate tolerances.

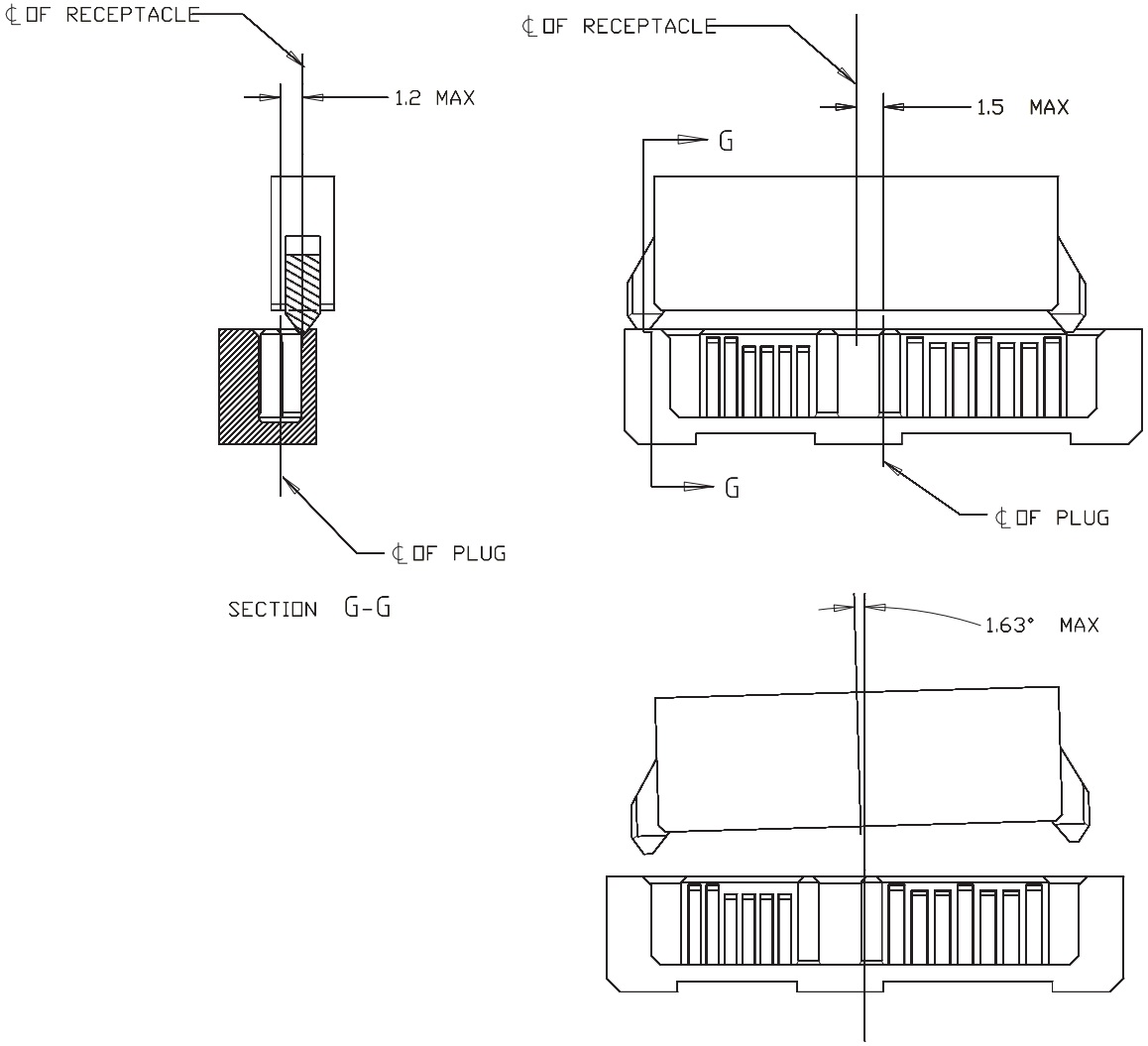


Figure 84 – Slimline Connector pair blind-mate misalignment tolerance

6.3.6 Connector labeling

Refer to section 6.1.9.

6.3.7 Connector and cable assembly requirements and test procedures

The connector and cable assembly requirements and test procedures shall conform to those in Table 6 with the following exceptions.

Table 14 – Slimline Connector Mechanical Test Procedures And Requirements

Removal force Backplane connector	EIA-364-13 Measure the force necessary to unmate the connector assemblies at a max. rate of 12.5 mm per minute.	2.5 N Min. after 500 cycles
Insertion force Cabled power connector Without latch	EIA-364-13 Measure the force necessary to mate the connector assemblies at a max. rate of 12.5 mm per minute.	45 N Max.
Removal force Cabled power connector Without latch	EIA-364-13 Measure the force necessary to unmate the connector assemblies at a max. rate of 12.5 mm per minute.	10 N Min. for cycles 1 through 5 8 N Min. through 50 cycles
Insertion force Cabled power connector With latch	EIA-364-13 Measure the force necessary to mate the connector assemblies at a max. rate of 12.5 mm per minute.	45 N Max.
Removal force Cabled power connector With latch	EIA-364-13 Apply a static 25 N unmating test load	No damage and no disconnect through 50 mating cycles

6.4 Internal LIF-SATA Connector for 1.8” HDD

This section provides capabilities required to enable a new smaller Serial ATA 1.8” HDD (hard disk drive).

The definition supports the following capabilities:

- Supports Gen1 (1.5 Gbps) and Gen2 (3 Gbps) transfer rates
- Support for FPC usage models
- Support for 8.0 and 5.0 mm slim 1.8” Form Factor (FF) HDD’s
- Support of 3.3 V with 5 V to meet future product requirements
- Support vendor pins, P18,19,20 and 21 reserved for HDD customer usage
- Support vendor pins, P22 and P23 for HDD manufacturing usage

This LIF-SATA is only for internal 8.0 mm and 5.0 mm slim 1.8” form factor devices.

NOTE: It is expected that the LIF-SATA interfaces comply with Gen1i and Gen2i specifications. The LIF-SATA connector can only be mated with FPC cable, the compliance point shall be after the mated assembly, where the necessary Tx and Rx measurements were measured.

6.4.1 General description

This internal LIF-SATA connector is designed to enable connection of a new family of slim 1.8” form factor HDD’s to the Serial ATA interface.

The internal LIF Serial ATA connector uses the 0.5 mm pitch configuration for both the signal and power segments. The signal segment has the same configuration as the internal standard Serial ATA connector but power segment provides the present voltage requirement support of 3.3 V, and provision for a future voltage requirement of 5 V. In addition, there is P8 that is defined as Device Activity Signal/Disable Staggered Spin up. Finally, there are 6 vendor pins, P18, 19, 20 and 21 for HDD customer usage, and P22 and P23 for HDD manufacturing usage.

Care must be taken in the application of this drive so that excessive stress is not exerted on the drive or connector.

6.4.2 Connector Locations

6.4.2.1 Connector location on Hard Disk Drive (HDD) Form Factor, - (Reference)

The internal LIF-SATA connector location on the HDD is shown in Figure 85 for reference purposes only. See SFF-8146 for form factor definition and connector location.

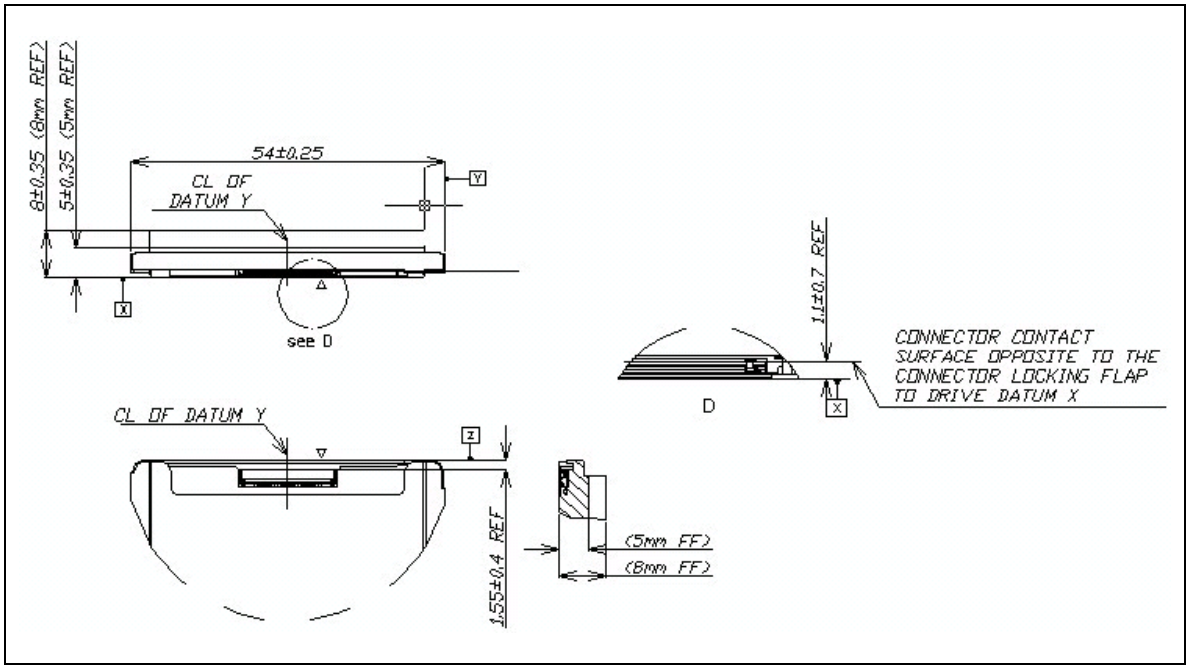


Figure 85 - Internal LIF-SATA connector location for 1.8" HDD

6.4.2.2 Connector location on Solid State Drive (SSD) Form Factor, - (Reference)

The internal LIF-SATA connector location on SSD Bulk Type is shown in Figure 86 for reference purposes only.

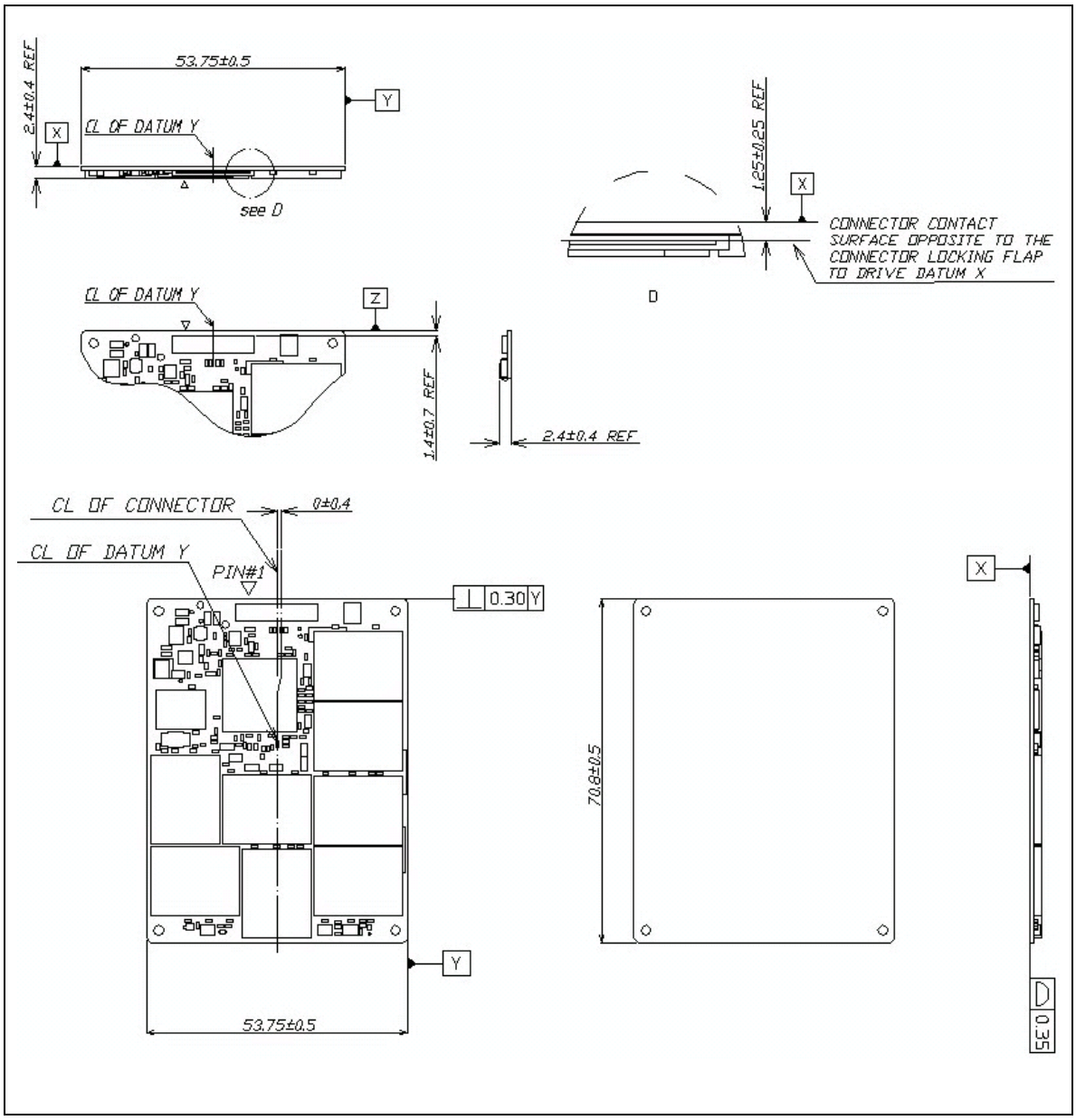
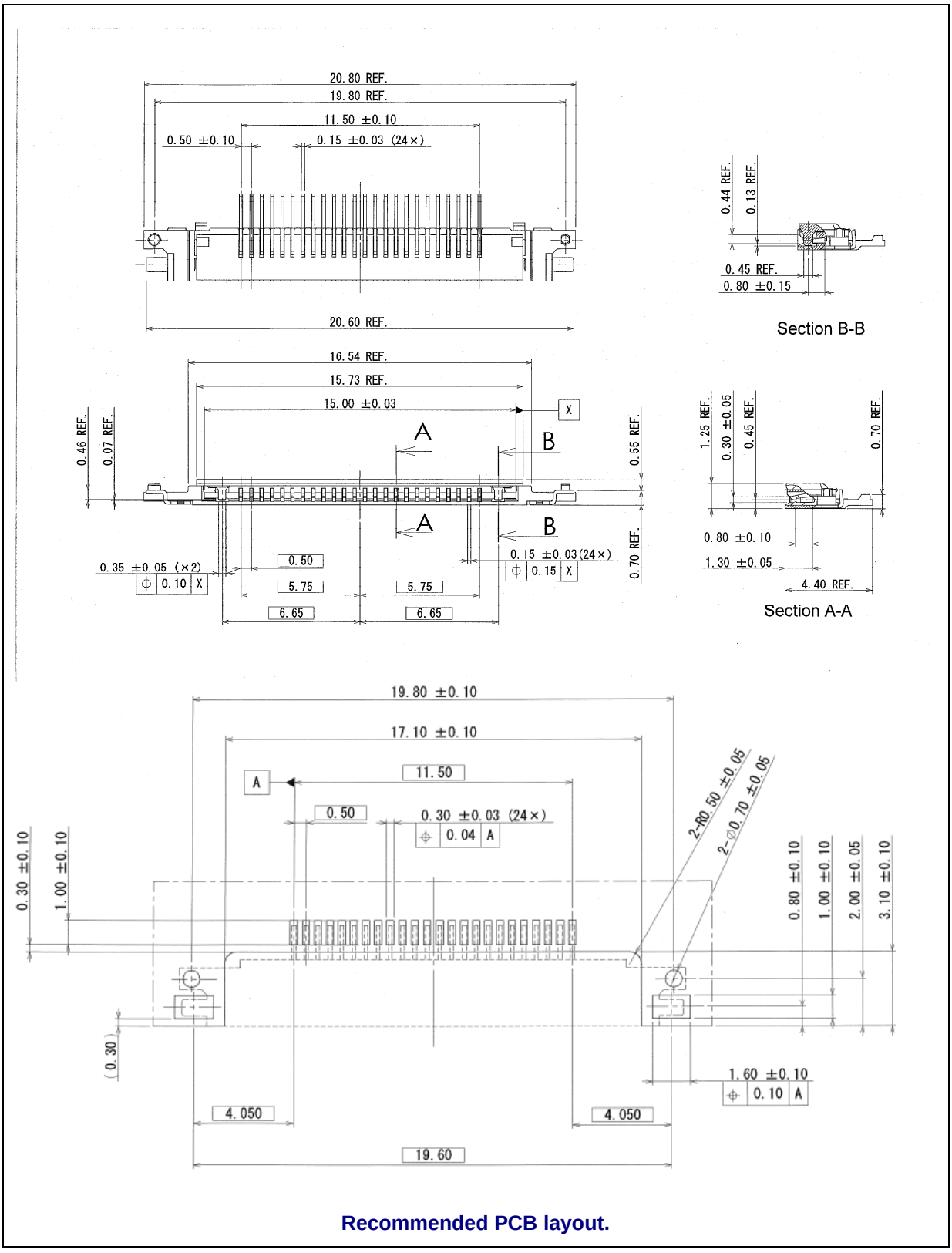


Figure 86 - Internal LIF-SATA connector location for 1.8” SSD bulk of single-sided mount type

6.4.3 Mating interfaces

6.4.3.1 Device internal LIF-SATA embedded type connector

Figure 87 defines the interface dimensions for the internal LIF-SATA embedded type connector with both signal and power segments.



Recommended PCB layout.

Figure 87 - Device internal LIF SATA embedded type connector

6.4.3.2 Device internal LIF-SATA surface mounting type connector

Figure 88 defines the interface dimensions for the internal LIF-SATA device surface mounting type connector.

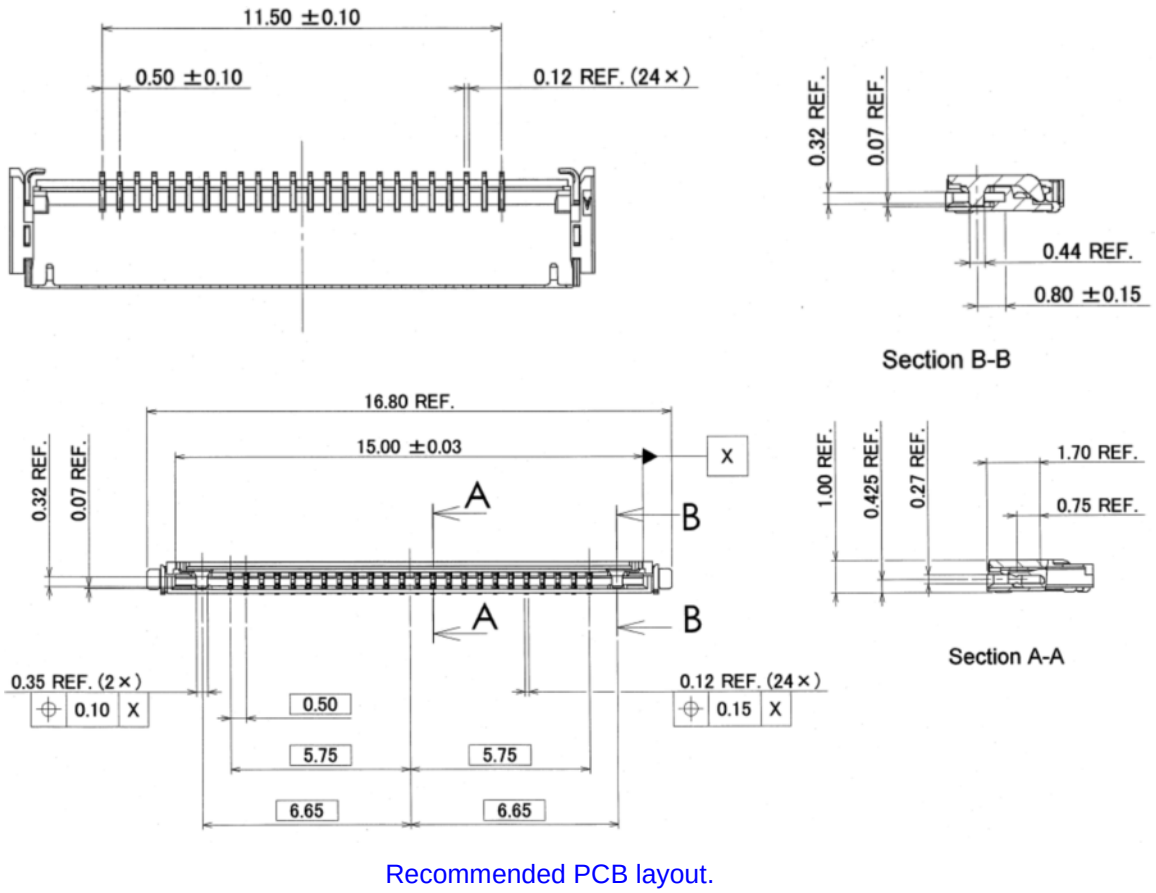


Figure 88 - Device internal LIF-SATA surface mounting type connector

6.4.3.3 FPC for Internal LIF-SATA

Figure 89 defines the interface dimensions for the FPC of LIF-SATA.

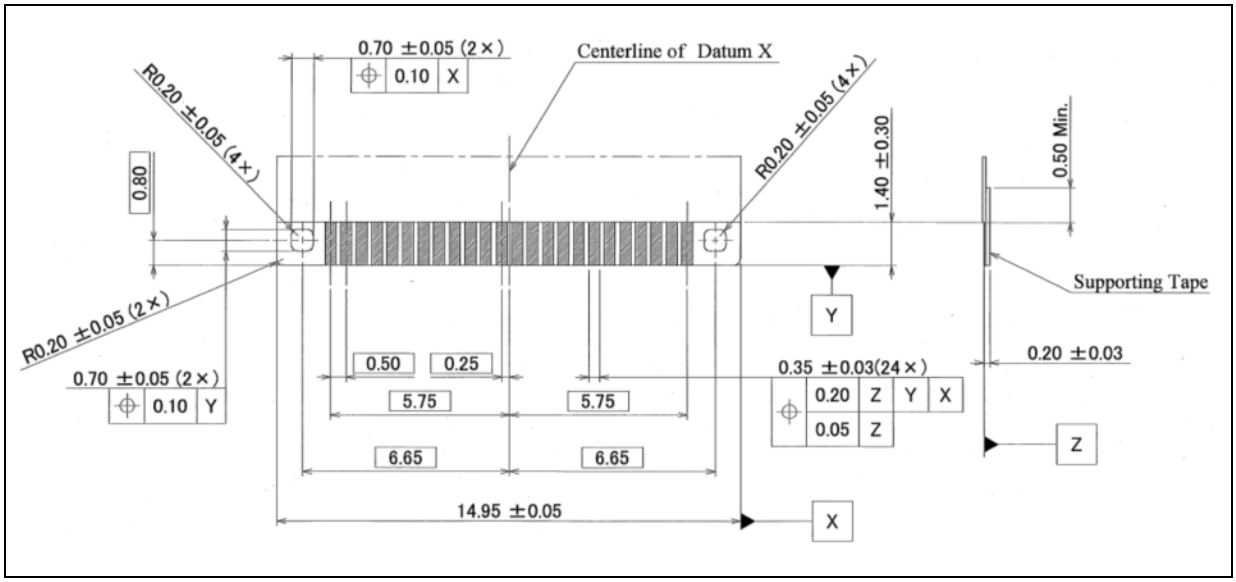


Figure 89 - FPC for Internal LIF SATA

6.4.4 Internal LIF-SATA pin signal definition and contact mating sequence

Table 15 details the pin names, types and contact order of the two internal LIF-SATA plug options. A brief description is also included for signal, ground and power pins. There are total 24 pins.

Table 15 - Signal and Power Internal LIF-SATA Plug

Name	Type	Description
P1	GND	
P2	V_{33}	3.3 V Power
P3	V_{33}	3.3 V Power
P4	GND	
P5	V_5	5 V Power
P6	V_5	5 V Power[1]
P7	GND	
P8	DAS/DSS	Device Activity Signal/Disable Staggered Spinup[2]
P9	GND	
P10	GND	
P11	A+	Differential Signal Pair A
P12	A-	
P13	GND	
P14	B-	Differential Signal Pair B
P15	B+	
P16	GND	
P17	GND	
P18	Vendor	Vendor Specific
P19	Vendor	Vendor Specific
P20	Vendor	Vendor Specific
P21	Vendor	Vendor Specific
P22	Vendor	Vendor Specific - Mfg pin[3]
P23	Vendor	Vendor Specific - Mfg pin[3]
P24	GND	

NOTE:
1. The 5 V supply voltage pins are included to meet future product requirements. Future revisions of this specification may require 5 V supply voltage be provided.
2. The corresponding pin to be mated with Pin8 shall always be grounded. For specific optional usage of Pin8 see section 6.4 Serial ATA Revision 3.0 spec.
3. No connect on the host side.

6.4.5 Housing and contact electrical requirement

The internal LIF-SATA connector and cable shall meet the requirements as defined for standard internal SATA cables and connectors in 6.1.10 of Serial ATA Revision 3.0, Connector and FPC assembly requirements and test procedures, with the exceptions listed below in Table 16.

Table 16 - Unique Connector Mechanical Testing Procedures and Requirements

Insertion force for FPC	Measure the force necessary to mate the connector assemblies at a max. rate of 12.5 mm per minute.	6 N Max.
Removal force for FPC	Measure the force necessary to unmate the connector assemblies at a max. rate of 12.5 mm per minute.	6 N Min. through 10 cycles
Durability	10 cycles for internal cabled application; Test done at a maximum rate of 200 cycles per hour.	No physical damage.
Minimum current	• Mount the connector to a test PCB • Wire power pins P2, P3, P5, P6 in parallel for power • Wire ground pins P1, P4, P7, P9, P10, P13, P16, P17, P24 in parallel for return • Supply 2A total DC current to the power pins in parallel, returning from the parallel ground pins • Record temperature rise when thermal equilibrium is reached	0.5 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.

6.5 External cables and connectors

6.5.1 External Single Lane

The External Single Lane system provides a single lane connection between a PC/Laptop and a commodity storage device using Gen1i/Gen2i Serial ATA devices. This interface is for the use of an external device that resides outside the PC chassis, similar to USB or 1394 hard drives and optical drives. While this does not exclude other usages, the requirements are derived based on this usage model.

Power is supplied to the external storage device via a separate means, which is outside the scope of this specification. This separate means is expected to be similar to power delivery for USB or 1394 external drives.

This section defines the external interconnect, compliance points, and associated electrical requirements/parameters for device interoperability.

The primary implementation is:

- An HBA connected to a shielded external connector. A buffer IC is required to interface to a Gen1i/Gen2i Serial ATA host unless the Serial ATA host is Gen1m/Gen2m compliant and designed for direct external connection.
- A shielded Serial ATA cable designed for external usage.
- A Serial ATA device enclosure with a corresponding external connector. A buffer IC is required to interface to a Gen1i/Gen2i Serial ATA device unless the Serial ATA device is Gen1m/Gen2m compliant and designed for direct external connection.

This implementation is shown in Figure 90.

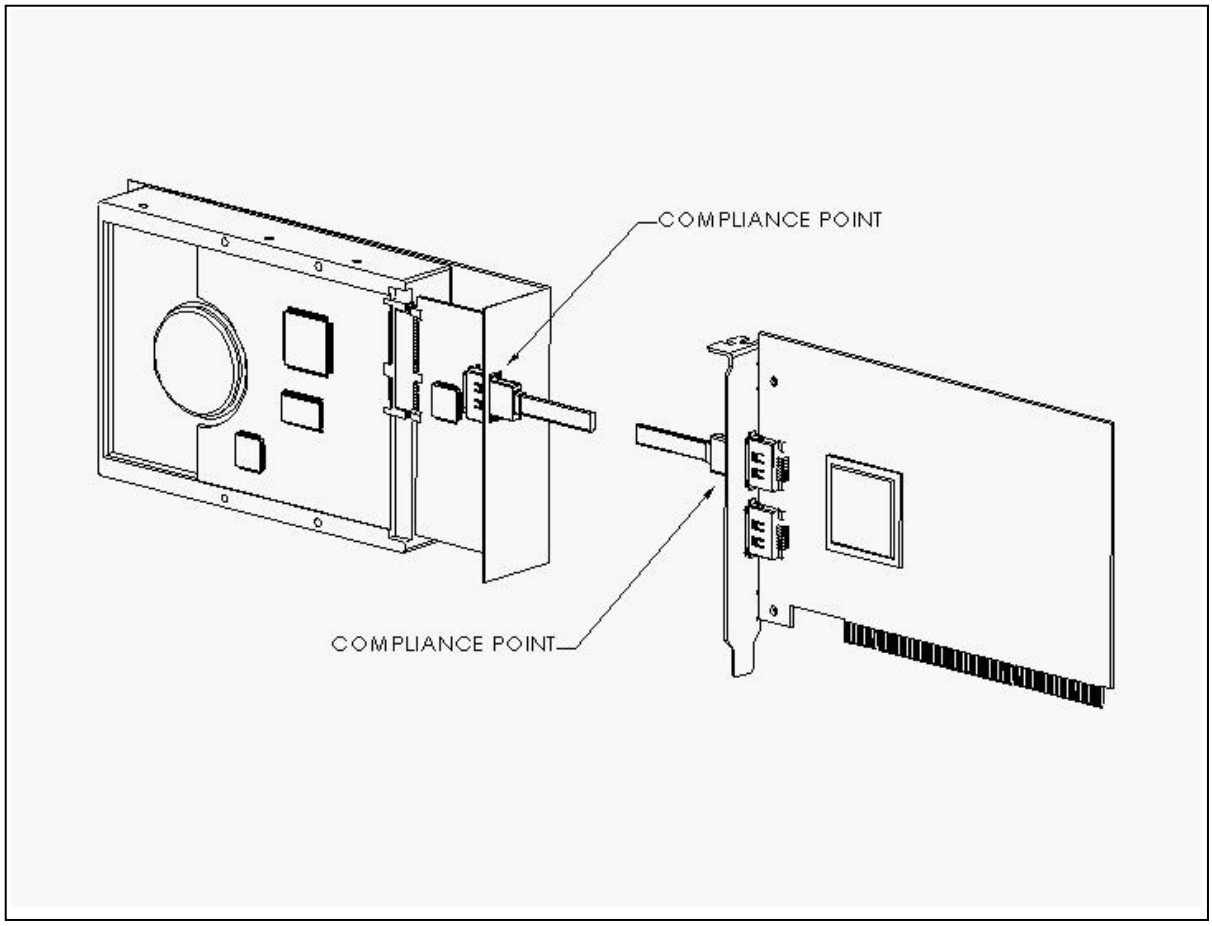


Figure 90 – Usage Model for HBA with external cable and single device enclosure

A second potential implementation is a Serial ATA host directly assembled on the motherboard connected to a shielded external connector via a pigtail to the motherboard connection. In this implementation, the external Serial ATA cable and the device assembly are similar to Figure 131, but another cable and connector pair between the motherboard and the external cable is introduced, placing an additional discontinuity point between host and device.

There are two compliance points for the External Single Lane system, one at each shielded external connector. As with other cable and connector system descriptions, interconnect between the IC/Phy and the connectors at the mating interface are outside the scope of the definition and are considered part of the delivered Phy solution. Implementations that have additional connections between the Phy/IC and the shielded external connector shall provide such interconnects as part of the engineered solution. For an implementation such as shown in Figure 91, the compliance points remain at the shielded external connectors.

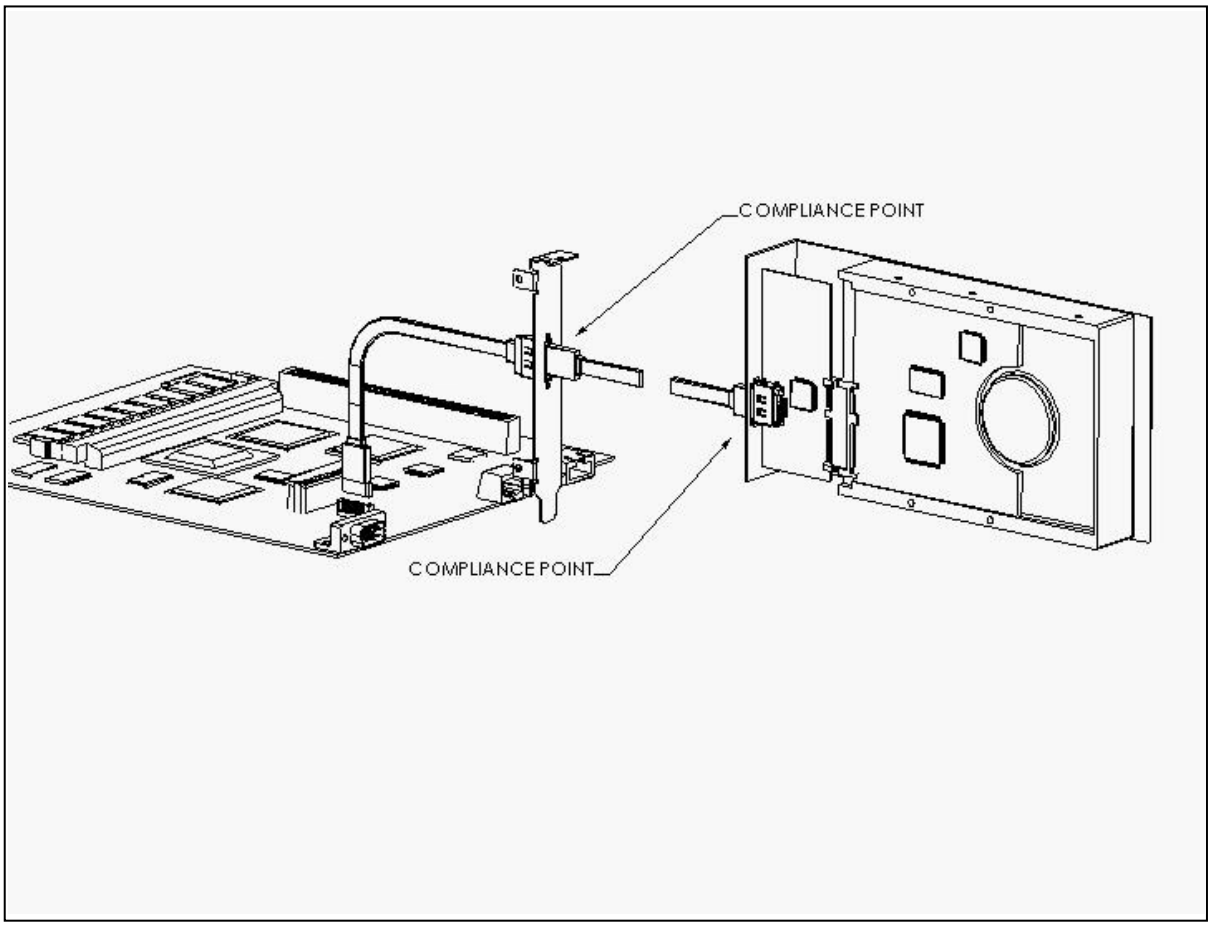


Figure 91 – Usage Model for on-Board Serial ATA Connector with extension cable to external cable to disk

The typical cable length is two meters (six feet); long enough to reach from a floor mounted PC to a device placed on the desktop. The compliance points for both ends of the external cable shall meet the Gen2m electrical specification.

The use of a standard internal Serial ATA host or device not specifically designed for direct external connection requires a buffer IC as part of the implementation. Note that since Single Lane External Serial ATA cables are a straight through design, (pin to pin), the host and device external connections, using the same external connector, have the same pin one locations but opposite signal definitions. Refer to Table 3 for both host and device connection signal assignments.

6.5.1.1 External Serial ATA Component General Descriptions

Five components are defined in this section to support external Serial ATA.

- A shielded external cable receptacle for use with shielded (external Serial ATA) cabling.
- A fully shielded RA PCB mounted SMT plug, (and reversed pin-out version).
- A fully shielded RA PCB mounted through-hole plug.
- A fully shielded vertical PCB mounted SMT plug.
- A fully shielded vertical PCB mounted through-hole plug.

Footprints and recommended panel cutouts are included to encourage greater interoperability from multiple vendors.

The external cable connector is a shielded version of the internal single lane connector defined in section 6, with these basic differences:

- The External connector has no “L” shaped key, and the guide features are vertically offset and reduced in size. This prevents the use of unshielded internal cables in external applications.
- To prevent ESD damage, the insertion depth is increased from 5 mm to 6.6 mm and the contacts are mounted further back in both the receptacle and plug.
- The retention features are springs built into the shield on both the top and bottom surfaces.

External Serial ATA Connector renderings:

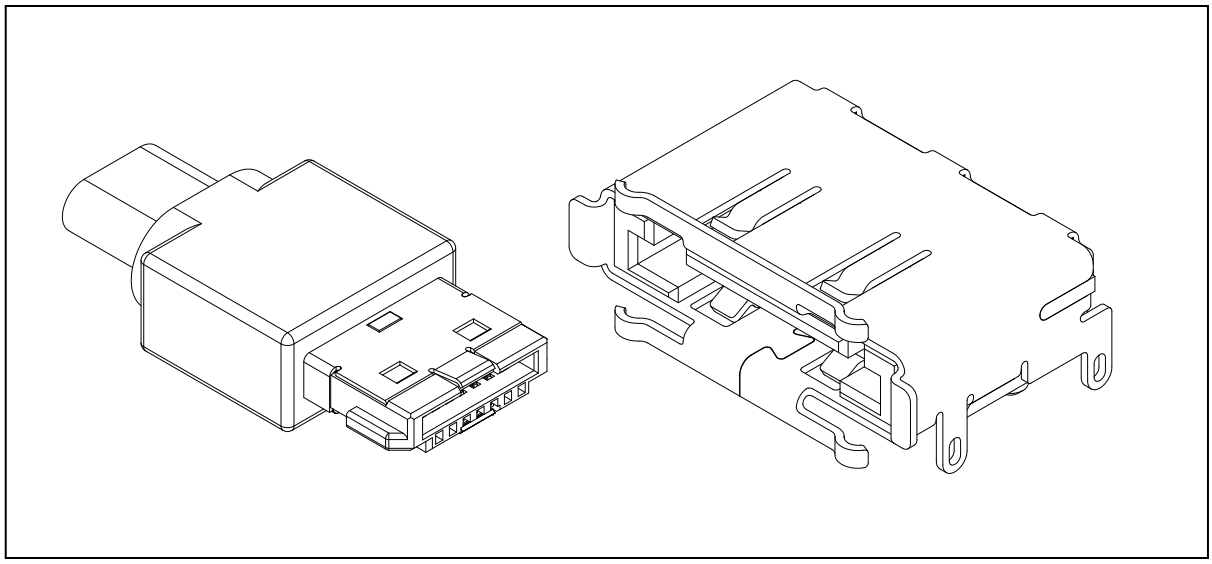


Figure 92 – Renderings of External Serial ATA cable receptacle and right angle plug

6.5.1.1.1 External Serial ATA Connector Mechanical drawings

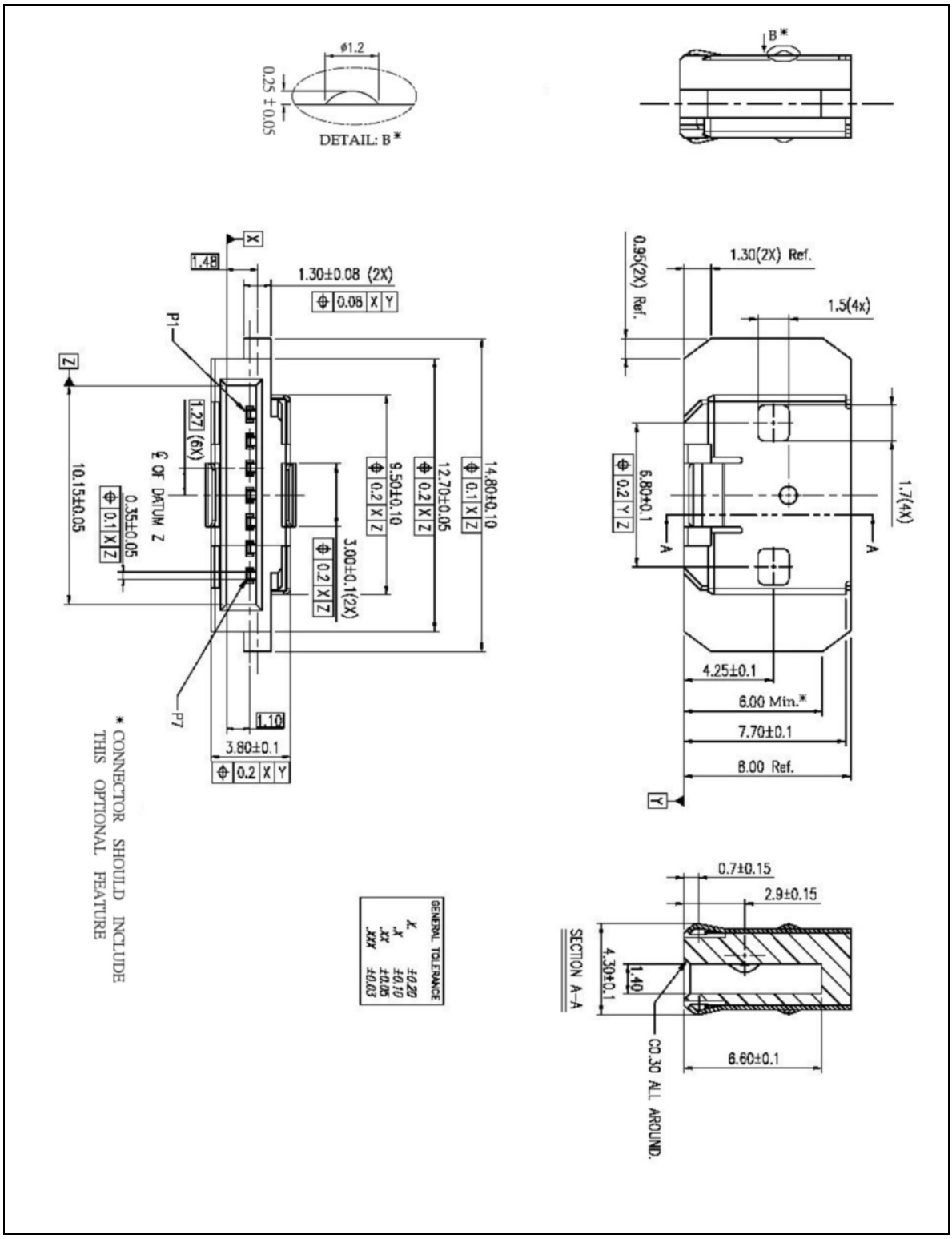


Figure 93 – Mechanical dimensions of External Serial ATA cable receptacle assembly

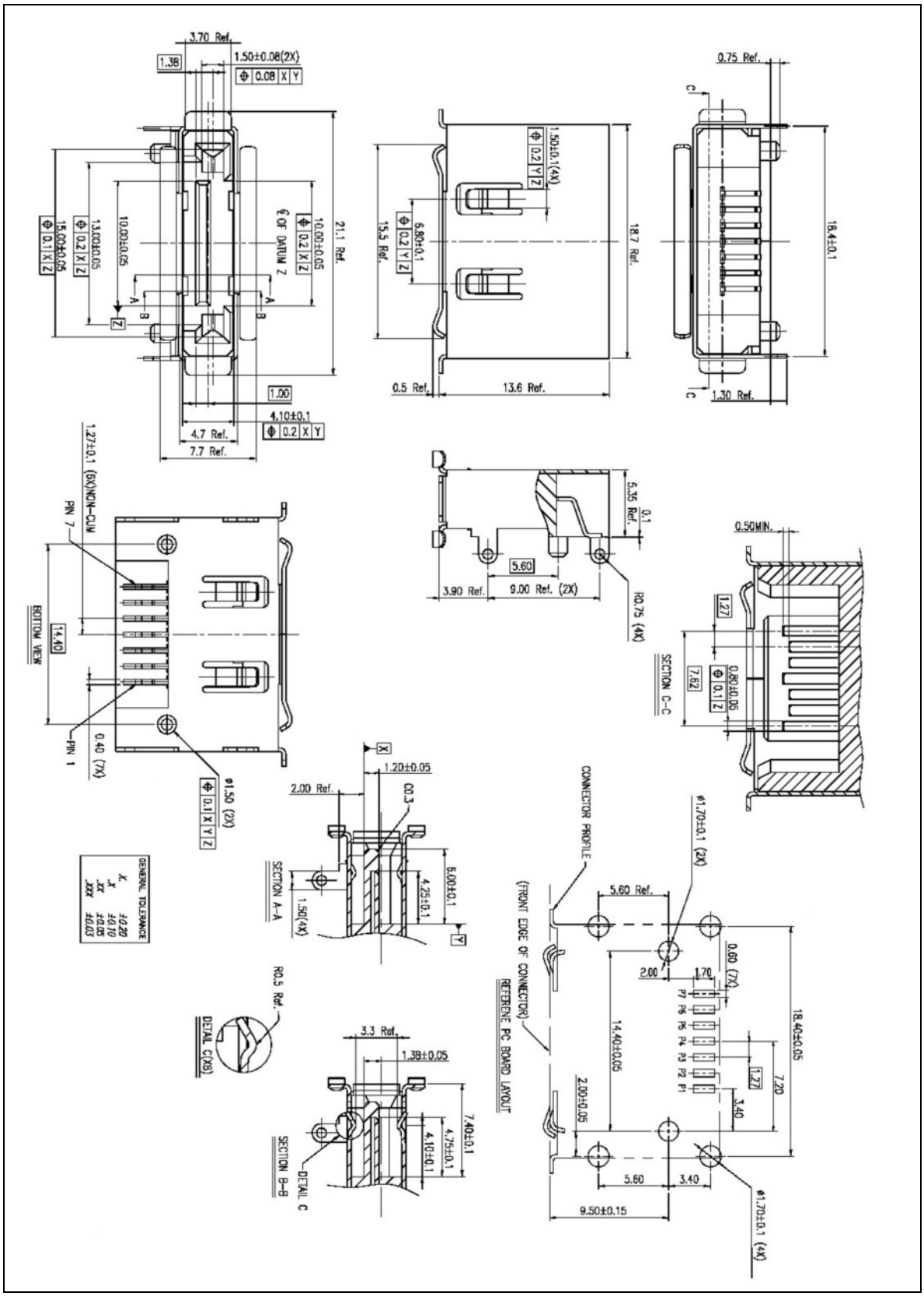


Figure 94 – Mechanical dimensions of External Serial ATA RA SMT plug

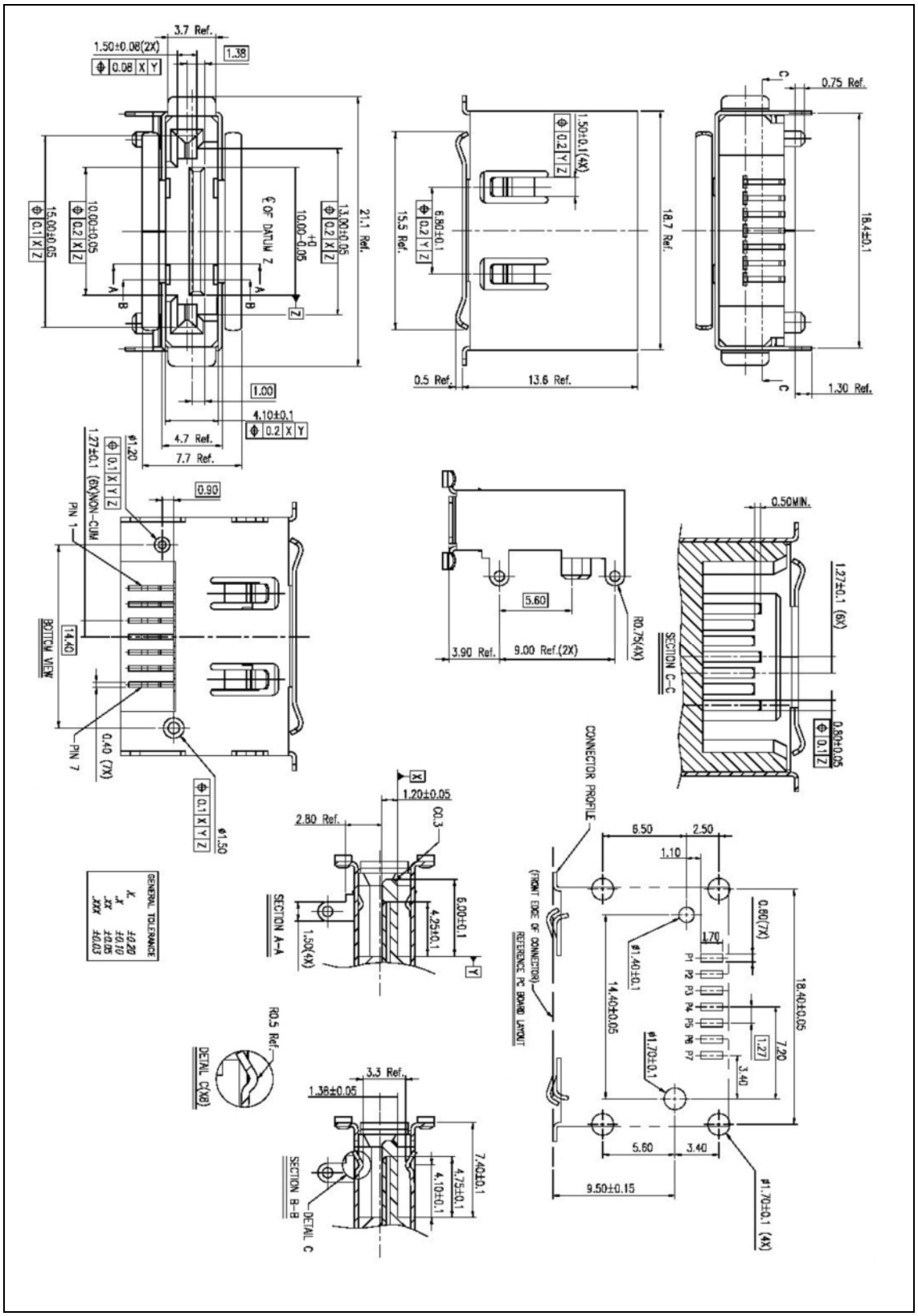


Figure 95 – Mechanical dimensions of External SATA RA SMT plug – Reversed Pin Out

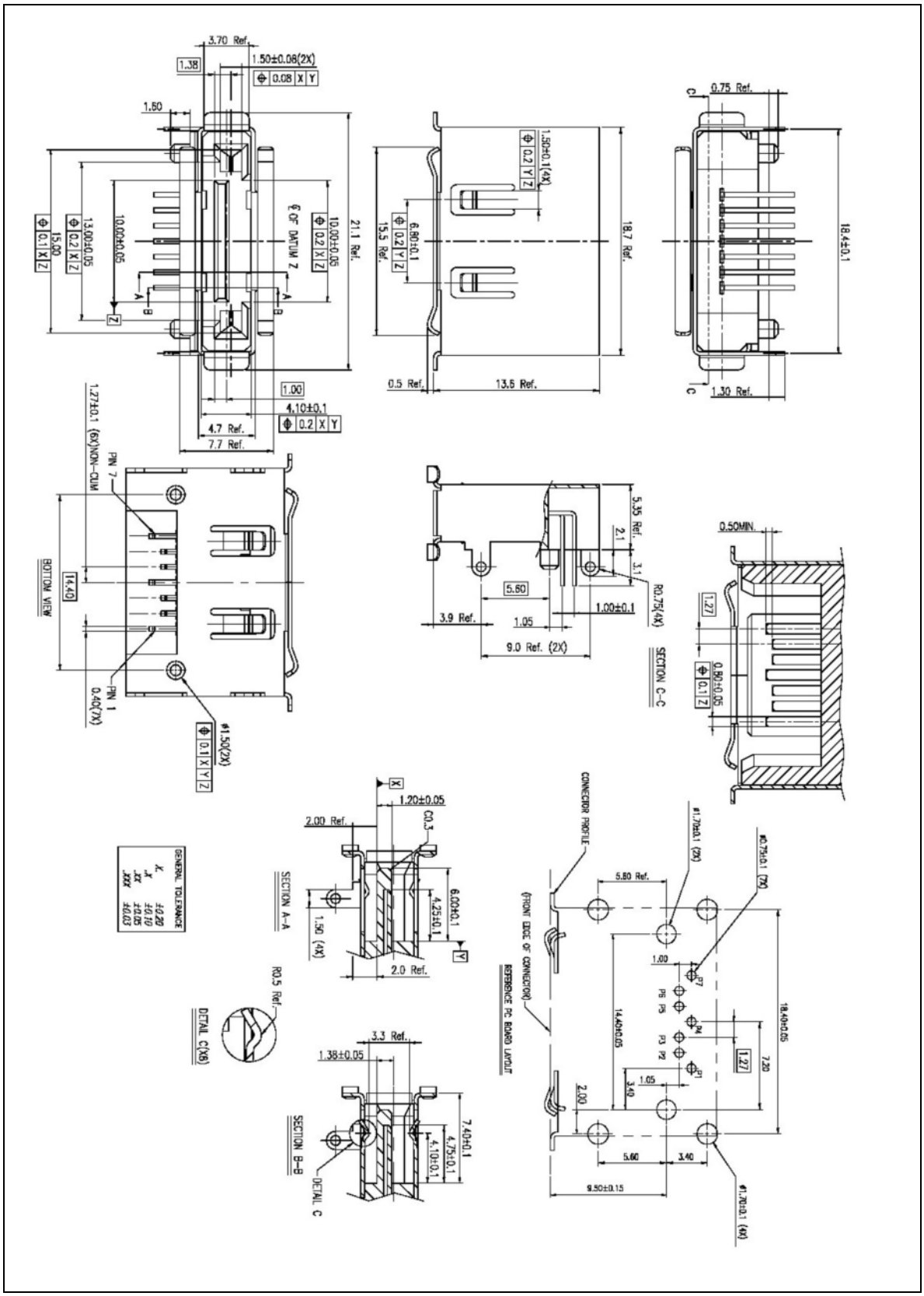


Figure 96 – Mechanical dimensions of External Serial ATA RA Through-hole

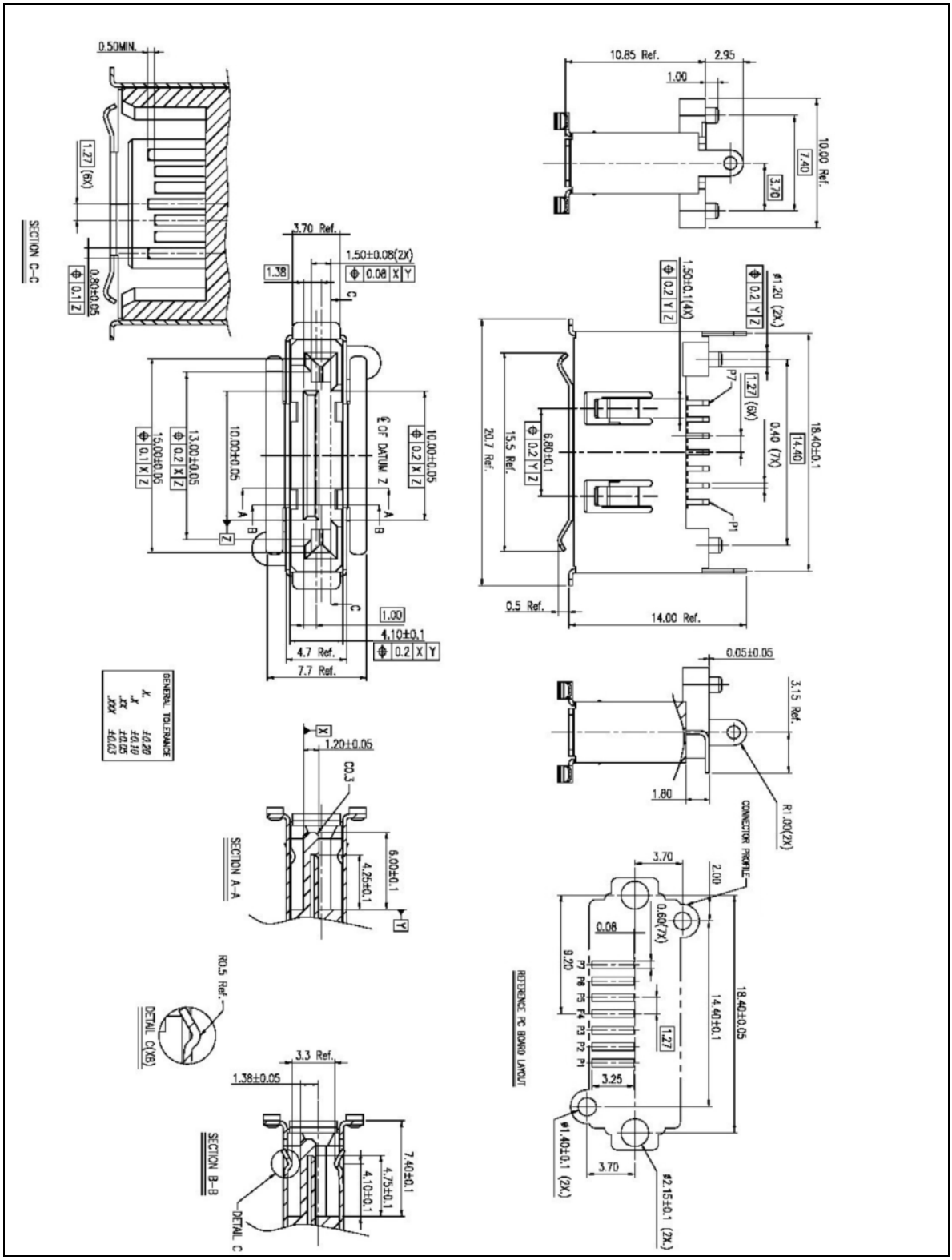


Figure 97 – Mechanical dimensions of External Serial ATA Vertical SMT plug

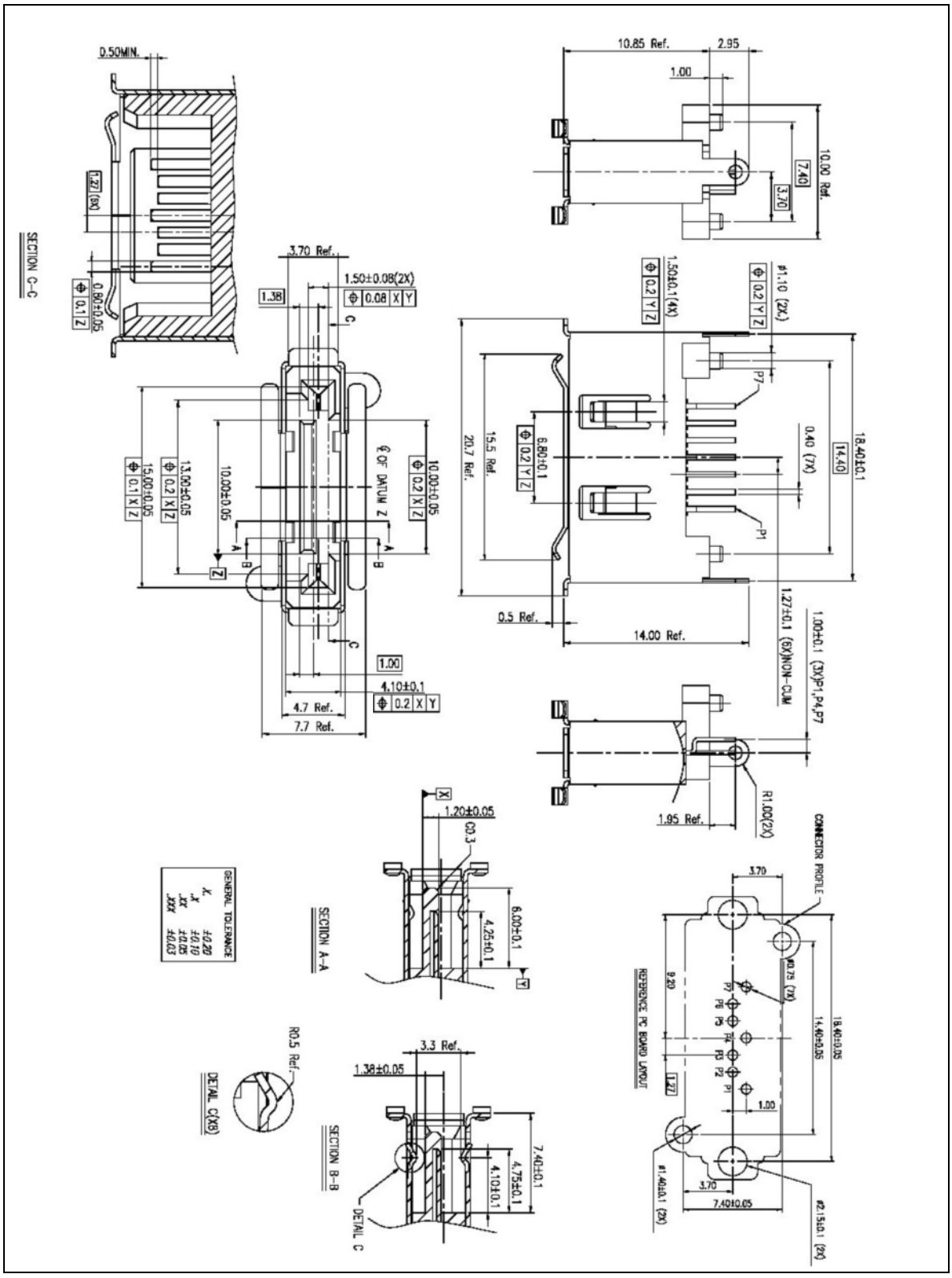


Figure 98 – Mechanical dimensions of External Serial ATA Vertical Through-hole plug

6.5.1.2 External Serial ATA Electrical Requirements

The external cable assembly shall meet the electrical characteristics defined in Table 19.

The Single Lane External Serial ATA Data Interface Phy electrical performance shall comply with the following:

- Electrical characteristics defined in Table 19, Gen1m and/or Gen2m at the shielded external connector compliance points.
- Support hot plugging and non-powered device attachment.
- AC coupling is required at the device interface and recommended at the host interface.

6.5.1.3 External Serial ATA Mechanical Requirements

The external connector mechanical performance specifications shall be consistent with the internal single lane connector specifications in section 6.5.1, with the following exceptions:

- Durability shall be 2500 cycles with no exposure of the base metal of the signal contacts.
- Insertion force shall be a maximum of 40 N.
- Removal force shall be a minimum of 10 N at the conclusion of the durability test.

6.5.1.4 External Serial ATA Device Direct Connection Requirements

Serial ATA devices may have data interfaces specifically designed for direct connection in the Single Lane External Serial ATA environment without requiring a buffer IC. The data interface of the Single Lane External Serial ATA device shall comply with all external Serial ATA mechanical requirements of section 6.5.1.3. The Phy electrical performance shall comply with the following:

- Electrical characteristics defined in section 6.6, Gen1m and/or Gen2m at the shielded external connector compliance points.
- Support hot plugging and non-powered device attachment.
- AC coupling is required at the device interface and recommended at the host interface.

6.5.2 External Multilane

This section defines standard cable assemblies and headers for connecting multiple Serial ATA channels from a RAID host bus adapter to an intelligent backplane in an adjacent JBOD (just a bunch of disks) unit. The RAID HBA and the JBOD units are envisioned as using different power supplies.

This cable/connector set is based on the SFF-8470 specification. The SFF-8470 specification also describes the cable/connector set used by Serial Attached SCSI (SAS).

6.5.2.1 Multilane Cable Conformance Criteria

The External Multilane cable/connector shall be used with Gen1m/Gen2m and Gen1x/Gen2x signal levels only. If Gen1m/Gen2m signal levels are used, the cable length is limited to two meters. If Gen1x/Gen2x signal levels are used, cable lengths up to and greater than two meters are supported.

6.5.2.1.1 Electrical Parameters

The External Multilane cable assembly operating at Gen1m and Gen2m levels shall meet the electrical characteristics defined in Table 20.

The External Multilane cable assembly operating at Gen1x and Gen2x levels shall meet the electrical characteristics defined in Table 21.

6.5.2.1.2 Mechanical Parameters

Detailed mechanical requirements are specified in SFF-8470, reference type 4X with thumbscrews. The PCI add-in card form factor supports two 4X interfaces.

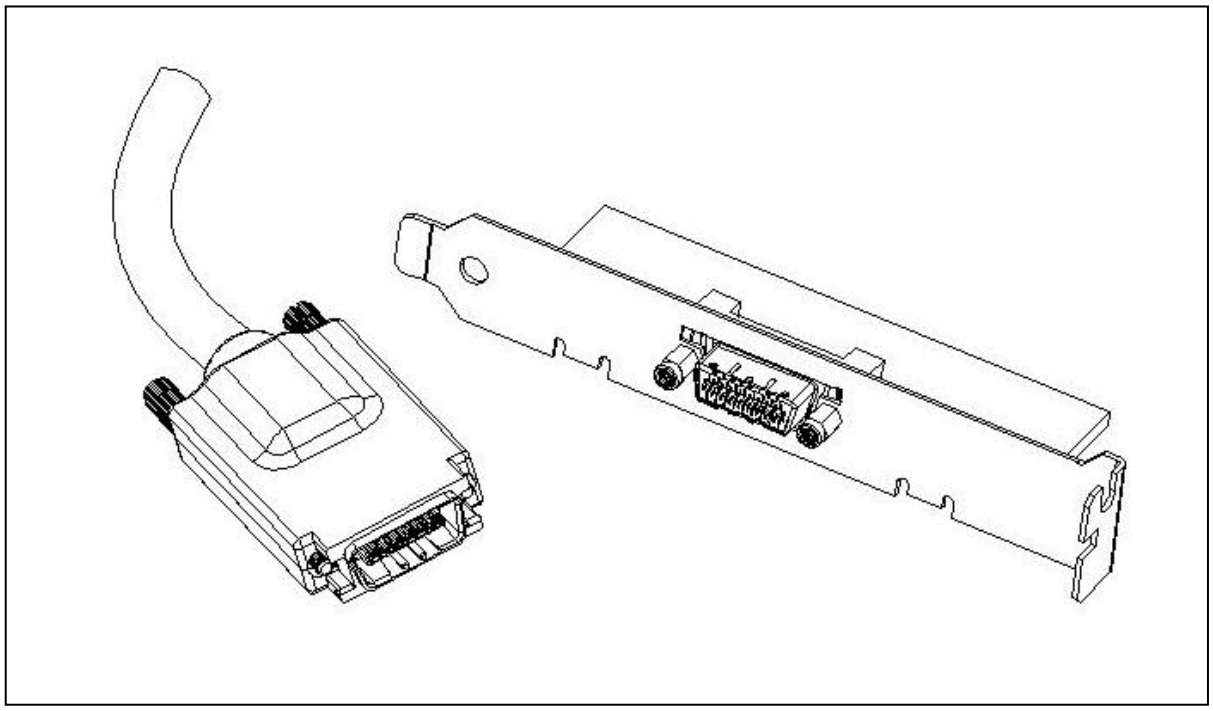


Figure 99 –External Multilane cable and connector

6.5.2.2 Keying Requirements

The Serial ATA External Multilane cable/connector may include keying features, a variant from the SFF-8470. The Serial ATA key locations are shown in Figure 100.

Optional keying allows connection between Serial ATA HBAs and JBODs but disallows connection to SAS HBAs or JBODs if the SAS units do not have connectors with key slots. If present, the External Multilane cable connector blocking key locations shall be 3, 4 and 5 and the corresponding mating connector blocking key locations shall be 1, 2, and 6.

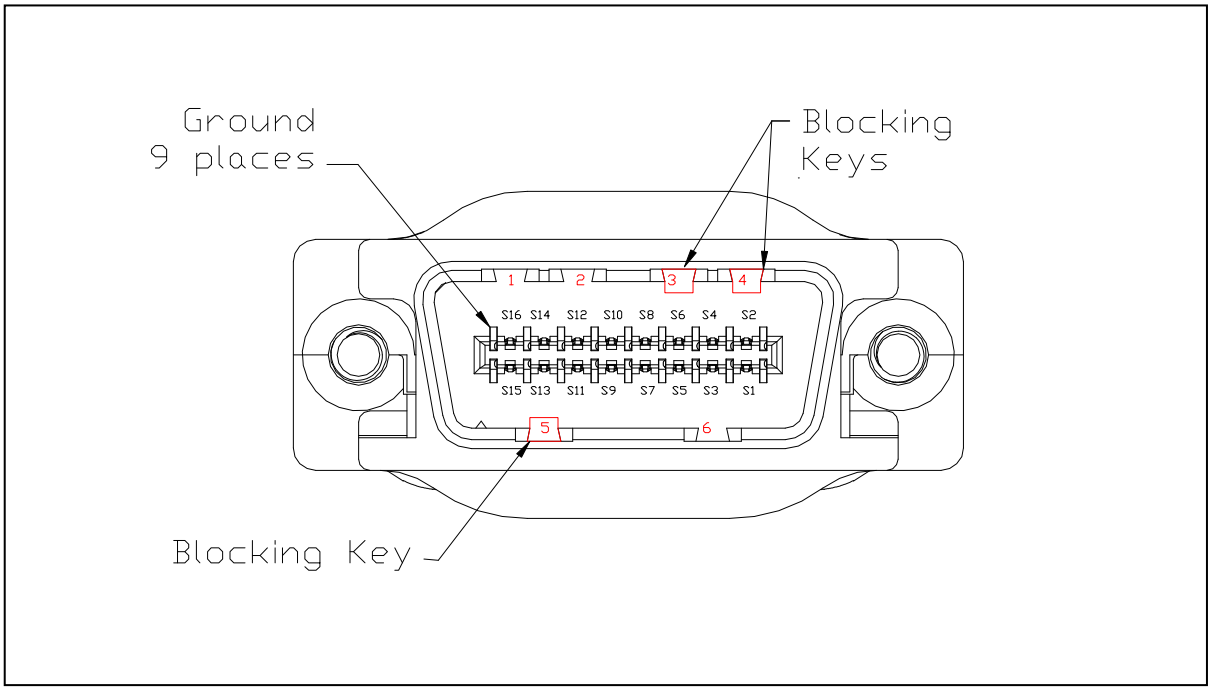


Figure 100 – Multilane Cable Connector Blocking Key Locations

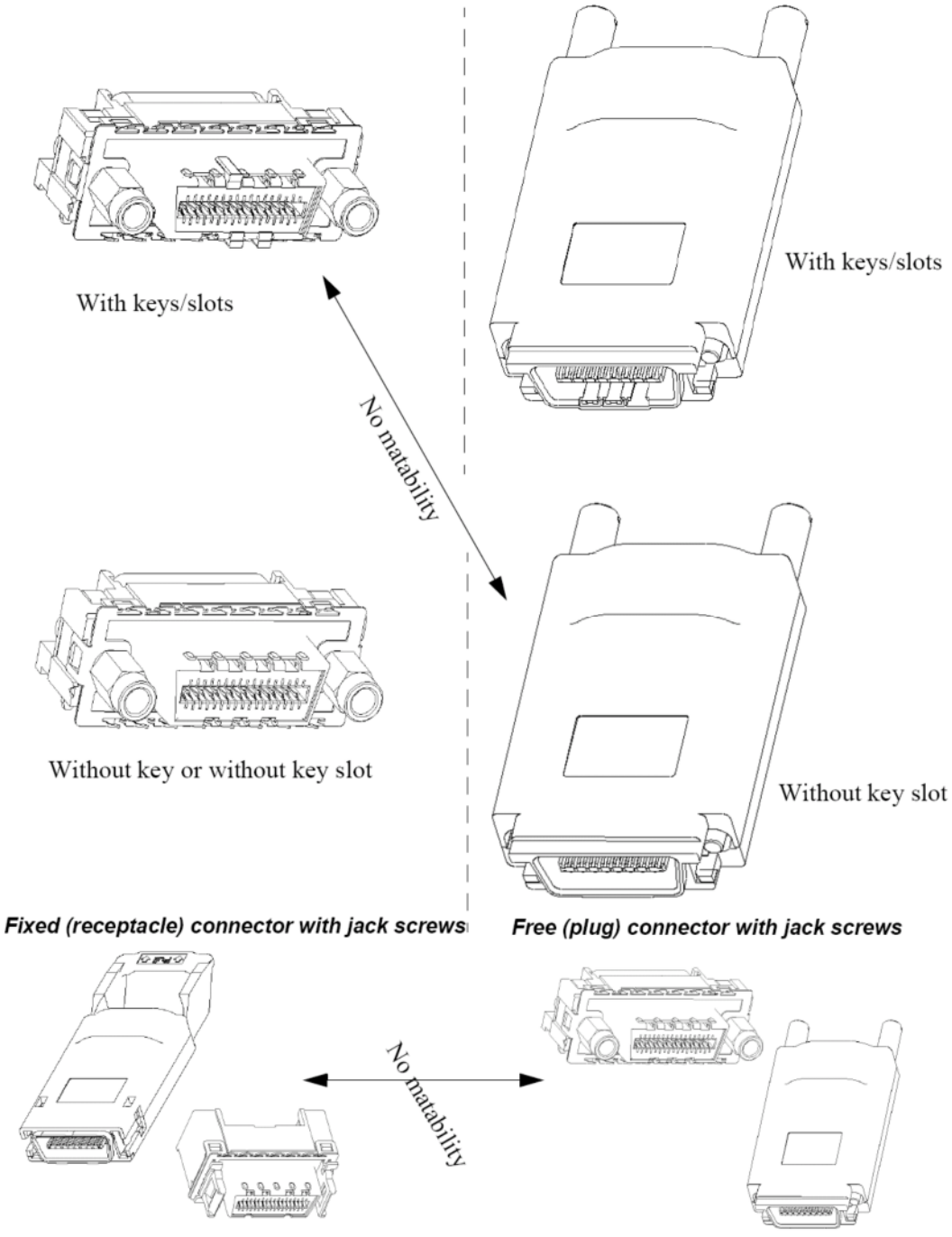


Figure 101 – Plug/Receptacle Keying

6.5.2.3 4 Lane Pin Assignments

Table 17 – Multilane Pin Assignments

Signal	Signal pin to use based on number of physical links supported by the cable			
	One	Two	Three	Four
Rx 0+	S1	S1	S1	S1
Rx 0-	S2	S2	S2	S2
Rx 1+	N/C	S3	S3	S3
Rx 1-	N/C	S4	S4	S4
Rx 2+	N/C	N/C	S5	S5
Rx 2-	N/C	N/C	S6	S6
Rx 3+	N/C	N/C	N/C	S7
Rx 3-	N/C	N/C	N/C	S8
Tx 3-	N/C	N/C	N/C	S9
Tx 3+	N/C	N/C	N/C	S10
Tx 2-	N/C	N/C	S11	S11
Tx 2+	N/C	N/C	S12	S12
Tx 1-	N/C	S13	S13	S13
Tx 1+	N/C	S14	S14	S14
Tx 0-	S15	S15	S15	S15
Tx 0+	S16	S16	S16	S16
SIGNAL GROUND	G1-G9			
CHASSIS GROUND	Housing			
Note: N/C: Not connected				

6.5.3 Mini SATA External Multilane

This section defines standard cable assemblies and headers for connecting multiple Serial ATA channels from a RAID host bus adapter to an intelligent backplane in an adjacent JBOD (just a bunch of disks) unit. The RAID HBA and the JBOD units are envisioned as using different power supplies.

This cable/connector system is based on the SFF-8086 and SFF-8088 specifications. Both SFF-8086 and SFF-8088 specifications are also used by Serial Attached SCSI (SAS).

6.5.3.1 Conformance Criteria

The External Multilane cable/connector shall be used with either Gen1m/Gen2m or Gen1x/Gen2x signal levels.

- 4 lanes, SAS/Serial ATA signals.
- Cable length is two meters maximum for Gen1m and Gen2m applications.
- Cable length is up to and greater than two meters for Gen1x and Gen2x applications.
- RX, TX, RX, TX pin sequencing to minimize crosstalk.
- Ground reference between each pair.
- Performance for 3.0 Gbps.
- Keying features for “m” and “x” cables.

6.5.3.1.1 Electrical Parameters

The External Multilane cable assembly operating at Gen1m and Gen2m levels shall meet the electrical characteristics defined in Table 20.

The External Multilane cable assembly operating at Gen1x and Gen2x levels shall meet the electrical characteristics defined in Table 21.

6.5.3.1.2 Mechanical Parameters

Detailed mechanical requirements are specified in SFF-8086 and SFF-8088.

The Mini SATA External Multilane cables and connectors shall use the 26-circuit version plug and receptacle defined in SFF-8086 and SFF-8088.

The pull-tab for the Mini SATA External Multilane connector, if present, shall be red (Pantone #207).

6.5.3.2 Mini SATA External Multilane Keying Requirements

The Mini SATA External Multilane cable/connector shall include keying features from SFF-8088. The Serial ATA defined key locations are shown in Figure 102.

Unique keying requirements for x-level signal levels, is shown in Figure 103. Unique keying requirements for m-level signal levels is shown in Figure 104.

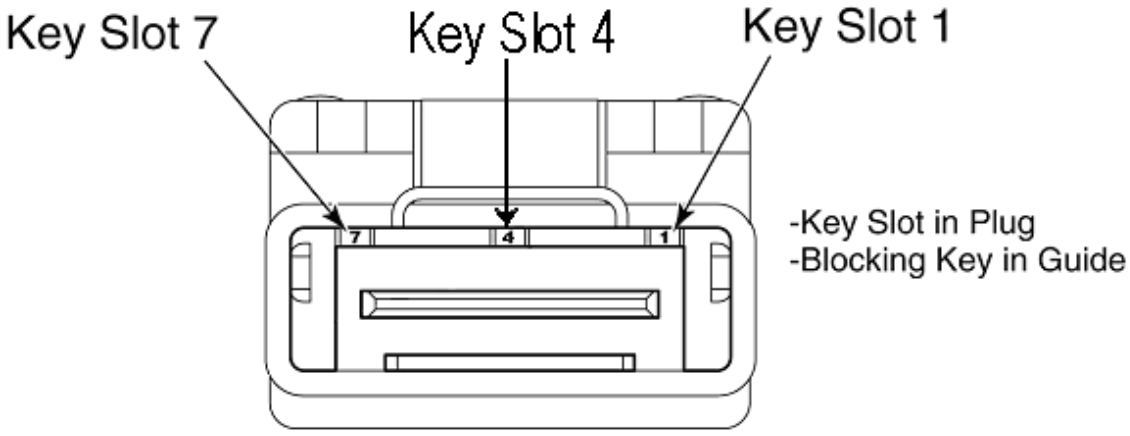


Figure 102 Mini SATA External Multilane System, Key Features

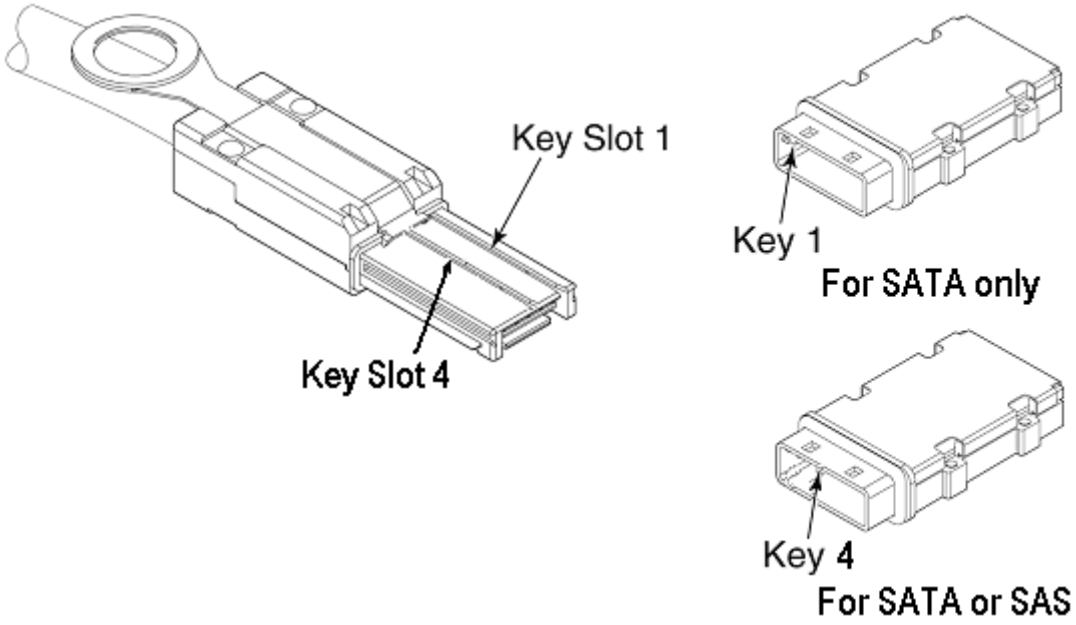


Figure 103 Mini SATA External Multilane System, Key Slots 1 and 4 for "x level" Signals

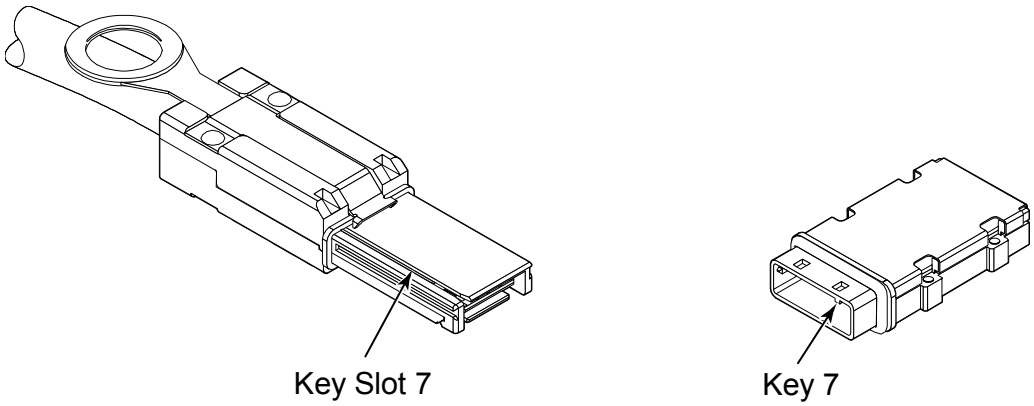
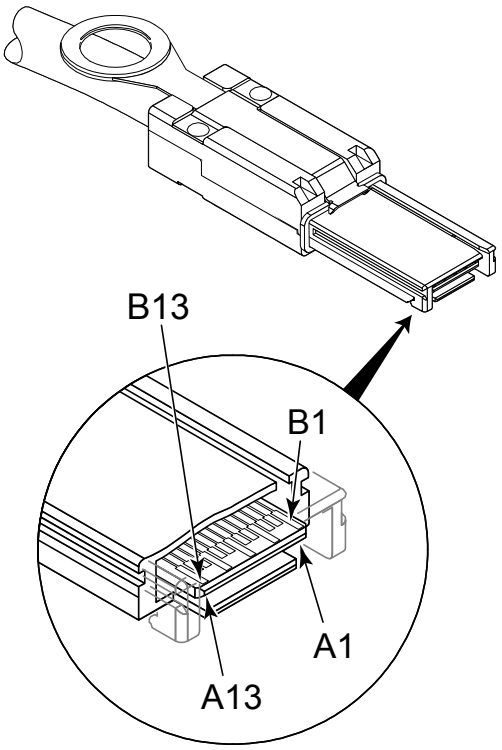


Figure 104 Mini SATA External Multilane System, Key Slots 7 for "m level" Signals

6.5.3.3 Mini SATA External Multilane Pin Assignments

The Mini SATA External Multilane connector pin assignments are shown in Figure 105.



Signal	Signal pin
SIGNAL GND	A1
RX 0+	A2
RX 0-	A3
SIGNAL GND	A4
RX 1+	A5
RX 1-	A6
SIGNAL GND	A7
RX 2+	A8
RX 2-	A9
SIGNAL GND	A10
RX 3+	A11
RX 3-	A12
SIGNAL GND	A13
SIGNAL GND	B1
TX 0+	B2
TX 0-	B3
SIGNAL GND	B4
TX 1+	B5
TX 1-	B6
SIGNAL GND	B7
TX 2+	B8
TX 2-	B9
SIGNAL GND	B10
TX 3+	B11
TX 3-	B12
SIGNAL GND	B13
CHASSIS GROUND	HOUSING

Figure 105 Mini SATA External Multilane Connector Pin Assignments

6.6 Cable and Connector Electrical Specifications

The purpose of this section is to specify the electrical characteristics of the cable and connector for all cabled usage models. The electrical characteristics defined herein describe relevant electrical characteristics required for high-speed signal transmission. An example test methodology is presented in Table 22 and Table 23 that may be used as a tool for characterizing cables, connectors, and PCB signal paths, i.e. microstrip and stripline traces. Different test methodologies and/or equipment may be used as long as they provide equivalent results.

The cable and connector shall meet the electrical requirements listed below before and after all the tests listed in Table 6 and Table 9 are performed.

6.6.1 Serial ATA Cable

6.6.1.1 Electrical Requirements

The electrical requirements for the internal single lane and Multilane Serial ATA cables and connectors for systems operating at Gen1i, Gen2i or Gen3i levels are listed in Table 18.

Table 18 – Internal Cable / Connector Measurement Parameter and Requirements

Parameter	Requirement	Procedure
Mated Connector Differential Impedance	100 Ohms ±15%	P1
Cable Absolute Differential Impedance	100 Ohms ±10%	P2
Cable Pair Matching Impedance	±5 Ohms	P3
Common Mode Impedance	25 - 40 Ohms	P4
Maximum Insertion Loss of Cable (10-4500 MHz)	6 dB	P5
Maximum Crosstalk, single lane: NEXT (10-4500 MHz)	26 dB loss	P6
Maximum Crosstalk, Multilane: CXT (10 – 4500 MHz)	30 dB loss	P7
Maximum Rise Time	85 ps (20-80%)	P8
Maximum Inter-Symbol Interference	50 ps	P9
Maximum Intra-Pair Skew	10 ps	P10

Note: The Internal Multilane and Mini SATA Internal Multilane maximum crosstalk is different than single lane. Since these cables are Multilane and have multi-aggressors, the additional requirement is to have crosstalk measured using the CXT method.

The electrical requirements for the External Single Lane cable and connector for systems operating at Gen1m or Gen2m levels are defined in Table 19.

Table 19 – External Single Lane Cable / Connector Measurement Parameter and Requirements

Parameter	Requirement	Procedure
Mated Connector Differential Impedance	100 Ohms ±15%	P1
Cable Absolute Differential Impedance	100 Ohms ±10%	P2
Cable Pair Matching Impedance	±5 Ohms	P3
Common Mode Impedance	25 - 40 Ohms	P4
Maximum Insertion Loss of Cable (10-4500 MHz)	8 dB	P5
Maximum Crosstalk: NEXT (10-4500 MHz)	26 dB loss	P6
Maximum Rise Time	150 ps (20-80%)	P8
Maximum Inter-Symbol Interference	50 ps	P9
Maximum Intra-Pair Skew	20 ps	P10

The electrical requirements for the External Multilane cable and connector for systems operating at Gen1m or Gen2m levels are defined in Table 20.

Table 20 – Limited External Multilane Cable / Connector Measurement Parameter and Requirements

Parameter	Requirement	Procedure
Mated Connector Differential Impedance	100 Ohms ±10%	P1
Cable Absolute Differential Impedance	100 Ohms ±5%	P2
Cable Pair Matching Impedance	±5 Ohms	P3
Common Mode Impedance	25 Ohms - 40 Ohms	P4
Maximum Insertion Loss of Cable (10-4500 MHz)	8 dB	P5
Maximum Crosstalk: CXT (10-4500 MHz)	30 dB CXT	P7
Maximum Rise Time	150 ps (20-80%)	P8
Maximum Inter-Symbol Interference	50 ps	P9
Maximum Intra-Pair Skew	20 ps	P10

Note: External Multilane cables for Gen1m or Gen2m signaling are limited to 2 meters in length.

The electrical requirements for the External Multilane cable and connector for systems operating at Gen1x or Gen2x levels are defined in Table 21.

Table 21 – Extended External Multilane Cable / Connector Measurement Parameter and Requirements

Parameter	Requirement	Procedure
Mated Connector Differential Impedance	100 Ohms ±10%	P1
Cable Absolute Differential Impedance	100 Ohms ±5%	P2
Cable Pair Matching Impedance	±5 Ohms	P3
Common Mode Impedance	25 Ohms - 40 Ohms	P4
Maximum Insertion Loss of Cable (10-4500 MHz)	16 dB	P5
Maximum Crosstalk: CXT (10-4500 MHz)	30 dB CXT	P7
Maximum Rise Time	150 ps (20-80%)	P8
Maximum Inter-Symbol Interference	60 ps	P9
Maximum Intra-Pair Skew	20 ps	P10

Note: All External Multilane cables longer than 2 meters use Gen1x or Gen2x levels.

6.6.2 Cable/Connector Test Methodology

6.6.2.1 Test Equipment

The following list identifies the type and performance of suggested equipment to perform the characterization procedures outlined in Table 22 and Table 23.

- High Bandwidth Sampling Oscilloscope
- TDR Module – < 35 ps (20% - 80%) Edge Rate Step Response
- Vector Network Analyzer – 4 port, 13.5 GHz BW
    - Suggested 20 GHz BW
- High Performance Coaxial Cables – = 20 GHz BW
- Low Jitter 3.0 Gb/s Pattern Source – 20%-80% rise time of 136 ps minimum. (The rise time should be as close to 136 ps as is practical.)

6.6.2.2 Test and Measurement Conditions

Unless otherwise specified, all tests and measurements shall be performed under the following conditions:

- Cable/Connector Mated
- Temperature: 15° to 35° C
- Relative Humidity: 20% to 80%
- Atmospheric Pressure: 650 mm to 800 mm of Hg

6.6.2.3 Test Fixture Considerations

Characterization of the cable/connector configuration requires an interface between the unit under test (UUT) and the test equipment and is commonly referred to as the test fixture. A primary objective in using a test fixture is to eliminate, as much as possible, the adverse signal integrity effects of the PCB. The following guidelines should be followed when defining the test fixture. Consider the following:

- The test fixture should use differential microstrip traces (100 ±5 Ohms) over a ground plane (single ended 50 ±2.5 Ohms).
- Open or shorted traces with the same length as the input signal traces shall be provided to enable the following:
    - Establish system input rise time
    - Synchronize pulses
    - Establish reference plane
- Traces for crosstalk measurements should diverge from each other.
- Provisions for attenuation reference measurement should also be provided.

6.6.2.4 Test Definition / Methodology

There are a number of steps within the test procedures used in preparation of making a measurement and are referred to as common procedures. They consist of calibration, de-skewing, establishing a reference plane, and establishing a rise time reference trace.

Prior to performing any procedures or gathering data, ensure that the test equipment has been properly calibrated.

The methodology to complete each of the common procedures is outlined in Table 22.

Table 22 – Common Interconnect Measurement Procedure Methodologies

C1	**Minimizing Skew between V+ and V-, Diff Signals**
	1. Define Differential Channel Stimulus a. Channel 1/3 positive edge step response (V^+) b. Channel 2/4 negative edge step response (V^-) 2. Differential Response – Identify the differential signal (V_{diff}) as a scope response, e.g. Use math function to obtain $V_{diff} = V^+ - V^-$ (CH1-CH2 or CH3-CH4) 3. Minimize skew between the V+ (positive) and V- (negative) edges by adjusting either the V+ (CH1/3) or V- (CH2/4) edge forward or backward in time until both edges align to within 1ps.
C2	**Establishing a reference plane at the connector**
	1. Follow calibration procedures outlined in the firmware of the oscilloscope. 2. Select the define reference plane option within the scope firmware to establish a reference plane at the input of the test fixture. 3. When establishing a new reference plane, use precision 50 Ohm loads or precision air lines that are terminated with 50 Ohm loads for the test fixture.
C3	**Establishing the rise time reference trace**
	1. Configure the TDR modules to generate a differential step impulse response and identify the differential rising edge of the trace. 2. Identify the high and low voltage values of the impulse response. 3. Identify the 20% and 80% voltage levels and verify that the rise time of the step impulse is between 25 ps and 35 ps. There are two methods for adjusting the step impulse response to be within the desired range: a. The system rise time is to be set via equipment filtering techniques. The filter programmed equals $\sqrt{t^2_{r(observed)} - t^2_{r(stimulus)}}$ b. Capture the measurement data and perform a post processing step to filter the captured data to the desired rise time within a waveform viewer or TDR SW application. 4. Once the correct rise time has been established, verify the rise time using the reference traces on the PCB fixture.

The test methodologies and procedures outlined in Table 23 refer to the common procedures described in Table 22. The actual specification requirement values for each of these methodologies are listed in section 6.6.1.1.

Table 23 – Interconnect Test Methodologies / Procedures

P1	**Mated Connector Differential Impedance**
	1. Calibrate the instrument and system using the measurement traces, then follow common procedures C1, C2, and C3. 2. The instrument rise time shall be set or the results filtered for a minimum of 55 ps to a maximum of 70 ps (20-80%) system risetime. The system risetime shall be set as close to 70 ps (20-80%) as practical. 3. Measure and record the maximum and minimum values of the near end connector differential impedance.
P2	**Cable Absolute Differential Impedance**
	1. Calibrate the instrument and system using the measurement traces, then follow common procedures C1, C2, and C3. 2. The instrument rise time shall be set or the results filtered for a minimum of 55 ps to a maximum of 70 ps (20-80%) system risetime. The system risetime shall be set as close to 70 ps (20-80%) as practical. 3. Measure and record maximum and minimum cable differential impedance values from the TDR trace in the first 500 ps of cable response following any vestige of the connector response.
P3	**Cable Pair Matching**
	1. Calibrate the instrument and system using the measurement traces, then follow common procedures C1, C2, and C3. 2. The instrument rise time shall be set or the results filtered for a minimum of 55 ps to a maximum of 70 ps (20-80%) system risetime. The system risetime shall be set as close to 70 ps (20-80%) as practical. 3. Measure and record the single-ended cable impedance of each cable within a pair, e.g. Z_{L1}, Z_{L2}. 4. Measure and record maximum and minimum cable impedance values from the TDR trace in the first 500 ps of cable response following any vestige of the connector response, e.g. $Z_{L1\text{-}max}$, $Z_{L1\text{-}min}$ and $Z_{L2\text{-}max}$, $Z_{L2\text{-}min}$. 5. The desired parameter equals $Z_{max} = Z_{L1\text{-}max} - Z_{L2\text{-}max}$ and $Z_{min} = Z_{L1\text{-}min} - Z_{L2\text{-}min}$.

P4	Common Mode Impedance
	1. Calibrate the instrument and system using the measurement traces, then follow common procedures C1, C2, and C3. 2. The instrument rise time shall be set or the results filtered for a minimum of 55 ps to a maximum of 70 ps (20-80%) system risetime. The system risetime shall be set as close to 70 ps (20-80%) as practical. 3. Select the negative edge step response channel to be a positive edge step response such that both channels generate a positive edge step response. 4. Measure the even mode impedance from the TDR trace of the first step generator in the first 500 ps of cable response following any vestige of the connector response. 5. Perform a math function on the waveform to divide the even mode impedance response by 2. The result is the Common Mode Impedance. 6. Make the same measurement and math calculation of the second step generator. 7. The Common Mode Impedance for each step generator shall meet the requirement.
P5	**Insertion Loss**
	1. Calibrate the instrument and system using the measurement traces, then follow common procedures C1 and C2. 2. 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, e.g. $IL_{fixture}$. 3. Measure and record the IL of the sample, which includes fixturing IL, over a frequency range of 10 to 4500 MHz, e.g. IL_{system}. 4. The insertion loss of the sample is calculated by $IL_{sample} = IL_{system} – IL_{fixture}$.

P6	Differential to Differential Crosstalk: NEXT
	1. Calibrate the instrument and system using the measurement traces, then follow common procedures C1, C2, and C3. 2. Terminate the far ends of the reference trace with characteristic impedance loads of 50 Ohms. 3. Measure and record the system and fixturing crosstalk. It is defined as the noise floor, e.g. V_{noise}. 4. Terminate the far ends of the device and listen lines with characteristic impedance loads of 50 Ohms. 5. Connect the source to the device pair and the receiver to the near-end of the listen pair. 6. Measure the NEXT over a frequency range of 10 to 4500 MHz, e.g. V_{NEXT}. 7. Verify that the sample crosstalk is out of the noise floor, e.g. $V_{NEXT} > V_{noise}$.

P7	Multilane (Multi Disturber) Differential Crosstalk: ML-CXT

1. Calibrate the instrument and system using the measurement traces, then follow common procedures C1, C2, and C3.

2. Terminate the far ends of the reference trace with characteristic impedance loads of 50 Ohms.

3. Measure and record the system and fixturing crosstalk. It is defined as the noise floor, e.g. V_{noise}.

4. Terminate the far ends of the device and listen lines with characteristic impedance loads of 50 Ohms.

5. Connect the source to the device pair and the receiver to the near-end of the listen pair.

6. Measure the CXT over a frequency range of 10 to 4500 MHz, e.g. V_{CXT}.

7. Verify that the sample crosstalk is out of the noise floor, e.g. $V_{CXT} > V_{noise}$.

8. $$MLCXT(f) = -20 \times \log\left(\sum_{A=1}^{7^1} 10^{-V_{CXT}(f)_l/20}\right)$$

where:

MLCXT(f) is the Multilane Cable assembly Crosstalk at frequency f observed on any given receive lane.

V_{CXT} (f)A is the relative crosstalk at frequency f between the receiver/victim and any combination of aggressor A[1], which exhibits less than 40 dB of isolation.

f is the frequency ranging from 10 MHz to 4.5 GHz

A is the 1,2 –7 (receiver/victim to aggressor[1] pair combinations)

1. Aggressor: Any lane identified as a Tx (input) shall be considered as a potential aggressor. This includes near end and far end Tx lane

Note:

MLCXT summation calculations accounts for any aggressor[1] relative to a receiver/victim pair which exhibit less than 40 dB of isolation.

P8	Differential Rise Time
	1. Calibrate the instrument and system using the measurement traces, then follow common procedures C1, C2, and C3. 2. Connect the TDR step impulse response generators to the near end of the signal path under test. 3. Record the output rise time at the far end of the signal path under test.
P9	**Inter-Symbol Interference**
	Note: As incident (test system induced) DJ may not be de-convolved from the end results, it's critical one use a high quality (low jitter) fixture and stimulus system when performing this measurement. 1. Connect a differential pattern source at the input of the test fixture. The 20%-80% rise time and fall time of the pattern source shall be 136 ps minimum. The rise and fall times should be as close to 136 ps as is practical, to minimize the resulting DJ and produce the most accurate results. Generate a LBP at 3.0 Gbps through the fixture. The Lone Bit Pattern emphasizes ISI. 2. Using a JMD, evaluate the Deterministic Jitter (DJ) introduced at the end of the cable.
P10	**Intra-Pair Skew**
	1. Calibrate the instrument and system using the measurement traces, then follow common procedures C1, C2, and C3. 2. Measure the propagation delay of each single ended signal within a pair at the mid point of the voltage swing, e.g. t_{delay} = V_{mid+} - V_{mid-} where $V_{mid} = \frac{V_{high} - V_{low}}{2}$.

6.7 Power Segment Pin P11 Definition (Optional)

Pin P11 of the power segment of the device connector may be used by the device to provide the host with an activity indication and it may be used by the host to indicate whether staggered spin-up should be used. To accomplish both of these goals, pin P11 acts as an input from the host to the device prior to PHYRDY for staggered spin-up control and then acts as an output from the device to the host after PHYRDY for activity indication. The activity indication provided by pin P11 is primarily for use in backplane applications. Reference section 13.14 for information on activity LED generation for desktop applications.

A device may optionally support activity indication via pin P11, staggered spin-up control via pin P11, or both features. If neither feature is supported, then pin P11 is a no-connect at the device as specified in Table 3.

A host may only support one pin P11 feature, either receiving activity indication or staggered spin-up disable control. If a host supports receiving activity indication via pin P11, then the host

shall not use pin P11 to disable staggered spin-up. If a host does not support receiving activity indication via pin P11, then the host may use pin P11 to disable staggered spin-up.

6.7.1 Device Activity Signal

6.7.1.1 Electrical Definition

The signal the device provides for activity indication is a low-voltage low-current driver intended for efficient integration into current and future IC manufacturing processes. The signal is not suitable for directly driving an LED and shall first be buffered using a circuit external to the device before driving an LED.

The activity signal is based on an open-collector or open-drain active-low driver. The device shall tolerate the activity signal being shorted to ground. The device shall tolerate a no-connect floating activity signal.

Table 24 and Table 25 define the electrical parameters and requirements for the activity signal for both the device and the host. Figure 106 is an example of an activity signal implementation for illustrative purposes. Note that the host should not rely on a particular resistor pull-up value on the device side, nor should the device rely on particular host resistor values. No direct support for wired-OR signals from multiple devices is accommodated. Host implementations that produce a single activity signal by combining multiple device inputs should buffer the signals prior to combining them.

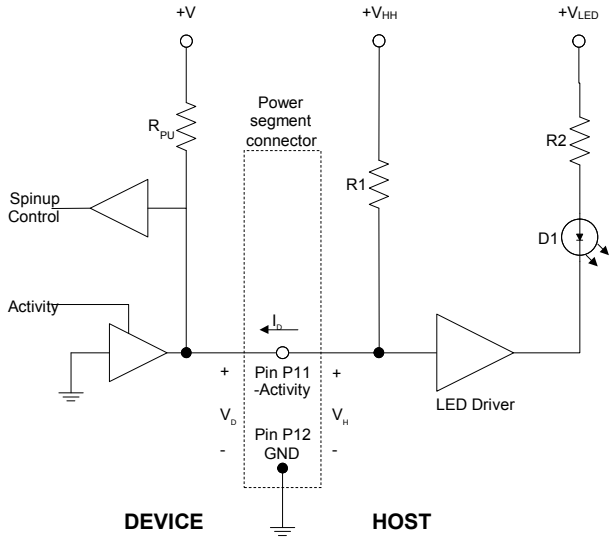


Figure 106 – Example activity signal electrical block diagram

All voltage references in Table 24 and Table 25 are to ground pin P12 on the device connector. All voltages and currents in Table 24 and Table 25 are measured at pin P11 on the device connector.

Table 24 – Power segment pin P11 activity signal electrical parameters

Parameter	Min Value	Max Value	Description & Conditions
V_{DIn}	-0.5 V	2.1 V	Tolerated input voltage
V_{DAct}	0 mV	225 mV	Device output voltage when driving low under the condition I_D less than or equal to 300 uA
V_{DInact}	-0.1 V	3.3 V	Device output voltage when not driving low
I_{DInact}	-10 uA	100 uA	Device leakage current when not driven

Table 25 – Host activity signal electrical parameters

Parameter	Min Value	Max Value	Description & Conditions
V_{HIn}	-0.5 V	3.3 V	Tolerated input voltage
V_{HH}		2.1 V	Host voltage presented to device when device not driving signal low. (Also see section 6.7.2 for staggered spin-up control).
V_{HL}	-0.1 V		Minimum allowable host voltage that may be presented to the device.
I_{HAct}		300 uA	Host current delivered to device when device driving signal low. Value specified at V_{DAct} voltage of 0 V.

6.7.1.2 LED Driver Circuit (Informative)

The LED driver circuit provided by the host to drive an activity LED is vendor specific. Figure 107 illustrates two conceptual driver circuits that satisfy the electrical requirements and provide a signal suitable for driving an activity LED. Variations in the driver circuits may be employed to drive the LED when active or to drive the LED when the device is inactive through the use of an inverting or non-inverting buffering arrangement.

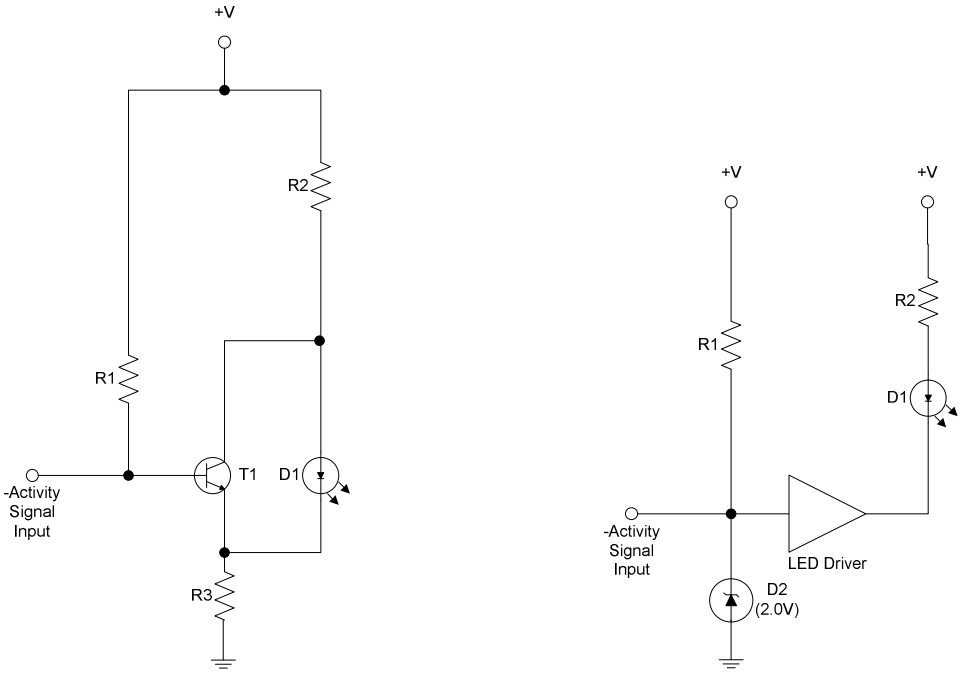


Figure 107 – Example host LED driver circuits

6.7.1.3 Functional Definition

Table 26 defines the two activity signal states and the corresponding conditions.

Table 26 – Activity signal functional states

State	Condition
Signal asserted (driven low)	Command(s) outstanding[1]
Signal negated (high impedance)	All other conditions

Notes:
1. Devices may omit asserting the activity signal for commands that do not access the media and have an expected service time too short to allow visual perception of the signal. Command(s) outstanding does not include the software reset, power-on reset, or COMRESET command protocols. As a consequence, pin P11 shall not be driven low by the device prior to return of the reset signature for the reset command protocols. This is behaviorally different than the parallel ATA DASP- signal.

6.7.2 Staggered Spin-up Disable Control

6.7.2.1 Electrical and Functional Definition

The staggered spin-up feature is defined in section 13.10. Devices may optionally provide support to disable staggered spin-up through pin P11 of the power segment connector. The staggered spin-up disable control is an active asserted low host signal.

Before the device spins up its media, devices that support staggered spin-up disable control shall detect whether pin P11 is asserted low by the host. If pin P11 is asserted low the device shall disable staggered spin-up and immediately initiate media spin-up. If pin P11 is not connected in the host (floating), devices that support staggered spin-up disable through pin P11 shall enable

staggered spin-up. Table 27 defines the electrical signal requirements for the device detection of staggered spin-up disable.

Table 27 – Host staggered spin-up control electrical requirements

Parameter	Min Value	Max Value	Description & Conditions
V_{HEnb}	1.8 V	V_{HHmax}	Host voltage presented to device to not disable staggered spin-up in devices that support staggered spin-up control. Value specified for all allowable I_{DInact} leakage currents.
V_{HDis}	-0.1 V	225 mV	Host voltage presented to device to disable staggered spin-up in devices that support staggered spin-up control. Value specified for all allowable I_{DInact} leakage currents.

The staggered spin-up control indication provided by a host or storage subsystem shall be static and shall not be changed while power is applied. If the signal is pulled low by the host during the staggered spin-up disable detection period, the signal shall remain low. Devices shall disable the activity signal if the host signals staggered spin-up disable.

If supported, the device shall sample the staggered spin-up disable condition after the time DC power is applied and before PHYRDY is asserted.

6.7.2.2 Staggered Spin-up Disable Circuit (Informative)

The host circuit for signaling staggered spin-up disable by pulling pin P11 low is vendor specific. Figure 108 illustrates a conceptual host circuit that would satisfy the electrical requirements for signaling staggered spin-up disable. It is permissible for the host to statically short pin P11 to ground or for the host to actively drive the signal low.

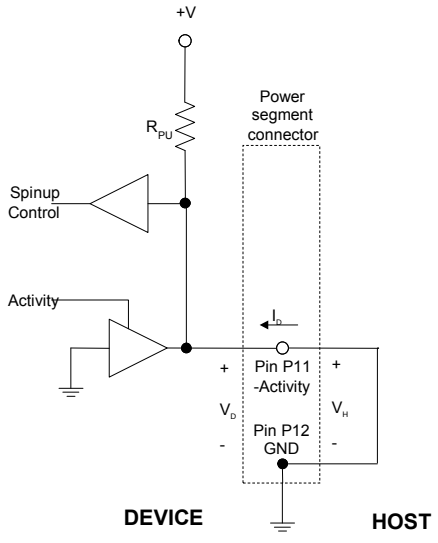


Figure 108 – Example host circuit for signaling staggered spin-up disable

6.8 Precharge and Device Presence Detection

For a storage subsystem, hot-plug capability is required as well as the ability to seamlessly handle presence detection in those cases where the storage subsystem may remove power to individual receptacles.

6.8.1 Device Requirements

In order to accommodate hot-insertion with the use of the precharge feature as well as a means for presence detection, Serial ATA devices shall bus together all power delivery pins for each supply voltage.

6.8.2 Receptacle Precharge (Informative)

The Serial ATA device connector has been specifically designed to accommodate a robust hot-plug capability. One feature of the device connector is the ability for receptacles to limit the instantaneous inrush current through the use of a precharge scheme. This scheme relies on one power delivery contact for each voltage being longer than the remaining contacts in order to allow power to be delivered through this longer contact through a current limiting device. Figure 109 illustrates one hot-plug power delivery scheme that utilizes the precharge connector feature.

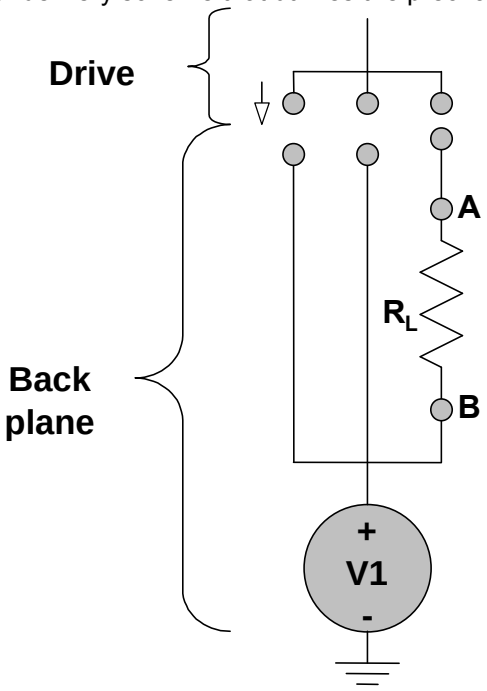


Figure 109 – Typical precharge configuration

All burden for limiting the inrush current for a newly inserted device is borne by the receptacle/backplane. The exact current limiting resistor size appropriate for a particular backplane solution depends on the details of the implementation. A few of the variables to be considered in sizing the current limiting resistor include:

- Device insertion velocity
- Effective capacitance of the inserted device
- Contact current carrying capacity

A survey of these variables by the group indicated that for one particular application, the maximum insertion velocity yielded a contact precharge time of approximately 3 ms. A poll of several disk drive vendors indicated a typical effective capacitance for disk drive devices of approximately 20 uF. For illustrative purposes, these values are presumed in an example scenario for estimating the precharge resistor value. The amount of time required to charge the effective capacitance to 90% of full charge is roughly 2.2 • R • C. Thus:

$$T = 2.2 \cdot R \cdot C_{EQ}$$

$$R = \frac{T}{2.2 \cdot C_{EQ}}$$

For the example charging time of 3 ms and an effective capacitance of 20 uF, the resultant precharge resistor value is approximately:

$$R = \frac{3ms}{2.2 \cdot 20uF} = 68\Omega$$

Because the Serial ATA power conductors support currents up to 1.5 A, the computed resistor size may be substantially reduced without adverse consequence in order to reduce the sensitivity to the device's actual effective capacitance. For the 12 V supply rail, the resistor may be as small as 8 Ohms and still not exceed the current carrying capacity of the precharge contact. Depending on the details of the actual enclosure subsystem design, typical precharge resistor values for the illustrative example scenario may therefore be in the range of 10-20 Ohms.

6.8.3 Presence Detection (Informative)

Presence detection relies on the device signaling the host using the OOB sequence to indicate its presence after a hot insertion. This approach presumes the device is inserted into a hot receptacle and also presumes the device inserted is not malfunctioning. In a storage subsystem, these assumptions may not be appropriate since such storage solutions may have the ability to unpower individual device receptacles in order to make device insertion/removal safer. Thus, a means for determining device presence in a receptacle that does not have power applied and without the device having to function is desired.

One possible device presence detection mechanism utilizes the precharge circuit outlined in section 6.8.2. The basic approach is to determine presence of a device by measuring the impedance between points A and B in the diagram. Because devices bus together their respective power delivery contacts, the impedance between points A and B in the diagram is R_L with no device present and is effectively zero with a device inserted in the receptacle.

Figure 110 illustrates one possible circuit for handling device presence detect with the receptacle either powered or unpowered. The example circuit is subject to tolerance buildup of the selected components and the supply voltages, and are only presented as conceptual examples.

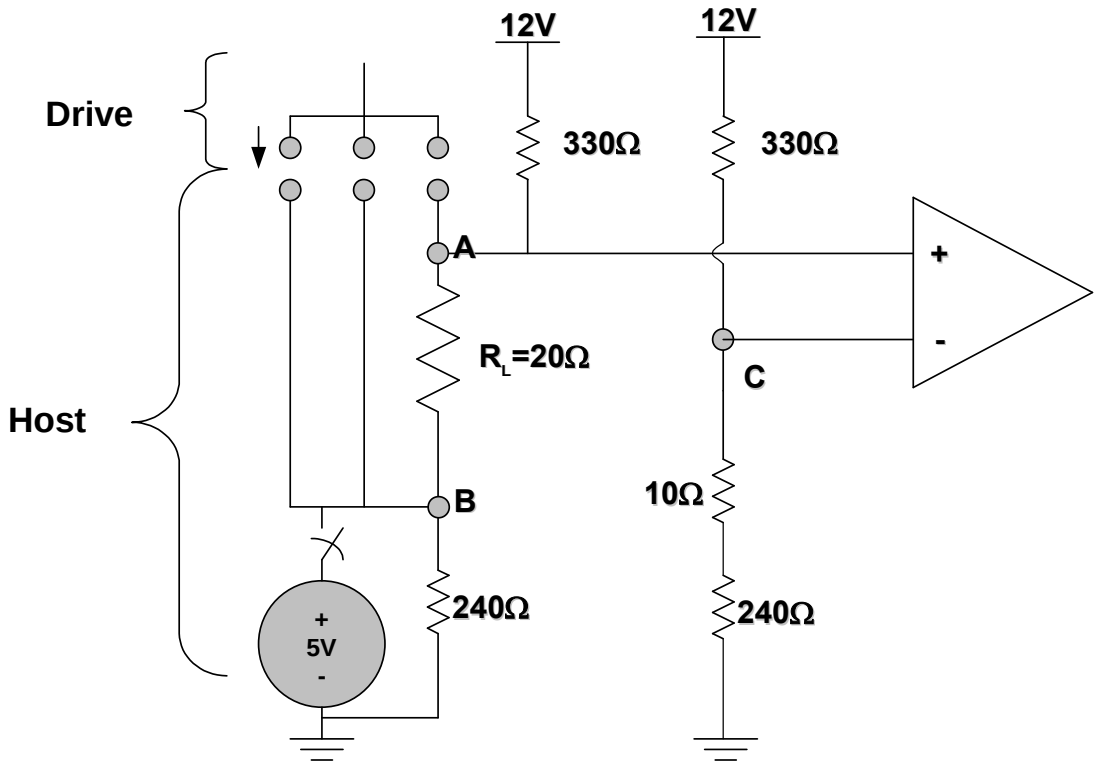


Figure 110 – Example presence detection implementation

Table 28 – Comparator voltages for alternate example presence detection circuit

	Receptacle Powered		Receptacle unpowered	
Device not present	V_A = 5.40 V	V_C = 5.17 V	V_A = 5.29 V	V_C = 5.17 V
Device present	V_A = 5.0 V	V_C = 5.17 V	*V_A = 5.05 V	V_C = 5.17 V

*If the inserted device provides a finite impedance to ground, then V_A should be lower than this value increasing the voltage differential further and increasing the margins.

7 Phy Layer

This section describes the physical layer of Serial ATA. The information that is provided is comprised of two types – informative and normative. Unless otherwise described, the information should be considered normative in nature and is included in this document as a necessary requirement in order to properly allow a piece of equipment to attach to another piece of equipment. The normative information is deliberately structured to constrain and define areas only to the degree that is required for compatibility. The information that is provided and marked informative is provided only to help the reader better understand the normative sections and should be taken as examples only. Exact implementations may vary.

7.1 Descriptions of Phy Electrical Specifications

The following terms have been developed for the various Electrical Specifications:

- **Gen1i**: Generation 1 Electrical Specifications: These are the 1.5 Gbps electrical specifications for internal host to device applications.
- **Gen1m**: Generation 1 Electrical Specifications for Short Backplane and external cabling applications: These are the 1.5 Gbps electrical specifications aimed at short 1.5 Gbps internal backplane applications, External Desktop Applications using the external single lane cable, and System-to-System Data Center Applications using external Multilane cables up to two meters in length. These include only modified receiver Differential input specifications. All other electrical specifications relating to Gen1m compliance points are identical to Gen1i specifications. This specification is for these limited applications only and is not intended for any other system topology.
- **Gen1x**: Extended Length 1.5 Gbps Electrical Specifications: These are electrical specifications aimed at 1.5 Gbps links in Long Backplane and System-to-System Data Center Applications supporting external Multilane cables up to and greater than two meters. These specifications are based upon the SAS references.
- **Gen2i**: Generation 2 Electrical Specifications: These are 3.0 Gbps electrical specifications for internal host to device applications.
- **Gen2m**: Generation 2 Electrical Specifications for Short Backplane and External Desktop Applications: These are 3.0 Gbps electrical specifications aimed at short internal backplane applications, External Desktop Applications using the external single lane cable, and System-to-System Data Center Applications using external Multilane cables up to two meters in length. These include only modified receiver differential input specifications. All other electrical specifications relating to Gen2m compliance points are identical to Gen2i specifications. This specification is for this limited application only and is not intended for any other system topology.
- **Gen2x**: Extended Length 3.0 Gbps Electrical Specifications: These are electrical specifications aimed at 3.0 Gbps links in Long Backplane and System-to-System Data Center Applications supporting external Multilane cables up to and greater than two meters. These specifications are based upon the SAS references.
- **Gen3i**: Generation 3 Electrical Specifications: These are 6.0 Gbps electrical specifications for internal host to device applications.

7.1.1 List of Services

- Transmit a 1.5 Gbps,3.0 Gbps, or 6.0 Gbps differential NRZ serial stream at specified voltage levels.
- Provide a 100 Ohm matched termination (differential) at the transmitter.
- Serialize a 10, 20, 40, or other width parallel input from the Link for transmission.
- Receive a 1.5 Gbps,3.0 Gbps, or 6.0 Gbps differential NRZ serial stream with data rates of ±350 ppm with +0/-5000 ppm (due to SSC profile) from the nominal data rate.
- Provide a 100 Ohms matched termination (differential) at the receiver.

- Extract data (and, optionally, clock) from the serial stream.
- De-serialize the serial stream.
- Detect the K28.5 comma character and provide a bit and word aligned 10, 20, 40, or other width parallel output.
- Provide specified OOB signaling detection and transmission.
- Use OOB signaling protocol for initializing the Serial ATA interface, and use this OOB sequence to execute a pre-defined speed negotiation function.
- Perform proper power-on sequencing and speed negotiation.
- Provide device status to Link layer.
  - Device present.
  - Device absent.
  - Device present but failed to negotiate communications.
- Optionally support power management modes.
- Optionally perform transmitter and receiver impedance calibration.
- Handle the input data rate frequency variation due to a spread spectrum transmitter clock.
- Accommodate request to go into Far-End retimed loopback, and other BIST Activate FIS test modes of operation when commanded.

7.1.2 Low Level Electronics Block Diagrams (Informative)

The following block diagrams are provided as a reference for the following sections of this document. Although informative in nature, the functions of the blocks described herein provide the basis upon which the normative specifications apply. The individual blocks provided are provided as an example of one possible implementation.

7.1.2.1 Physical Plant Block Diagram

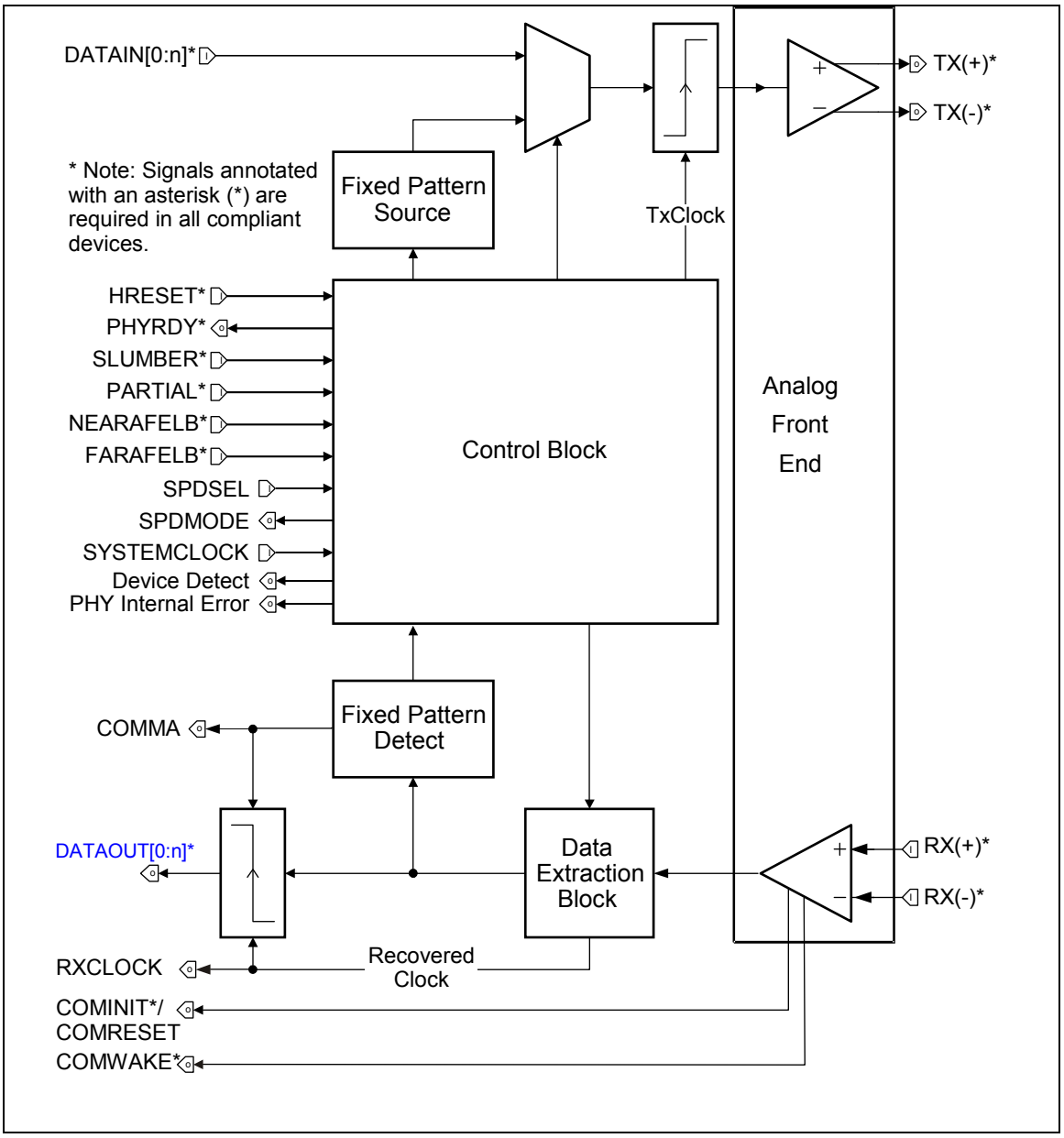


Figure 111 – Physical Plant Overall Block Diagram (Informative)

Physical Plant Overall Block Diagram Description

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 OOB 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 $ALIGN_P$ activity.
Fixed pattern detect	This block provides the support circuitry to allow proper processing of the $ALIGN_P$ 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 Phy layer for serialization and transmission.
PHYRESET	This input signal causes the Phy to initialize to a known state and start generating the COMRESET OOB signal across the interface.
PHYRDY	Signal indicating Phy has successfully established communications. The Phy is maintaining synchronization with the incoming signal to its receiver and is transmitting a valid signal on its transmitter.
SLUMBER	Causes the Phy layer to transition to the Slumber power management state.
PARTIAL	Causes the Phy layer to transition to the Partial power management state
NEARAFELB	Causes the Phy to loop back the serial data stream from its transmitter to its receiver
FARAFELB	Causes the Phy to loop back the serial data stream from its receiver to its transmitter
SPDSEL	Causes the control logic to automatically negotiate for a usable interface speed or sets a particular interface speed. The actual functionality of this input is vendor specific and varies from manufacturer to manufacturer.

SPDMODE — Output signal that reflects the current interface speed setting. The actual functionality of this signal is vendor specific and varies from manufacturer to manufacturer.

SYSTEMCLOCK — This input is the 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 de-serialized 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.

COMRESET / COMINIT

Host: Signal from the OOB detector that indicates the COMINIT OOB signal is being detected.
Device: Signal from the OOB detector that indicates the COMRESET OOB signal is being detected.

COMWAKE — Signal from the OOB detector that indicates the COMWAKE OOB signal is being detected.

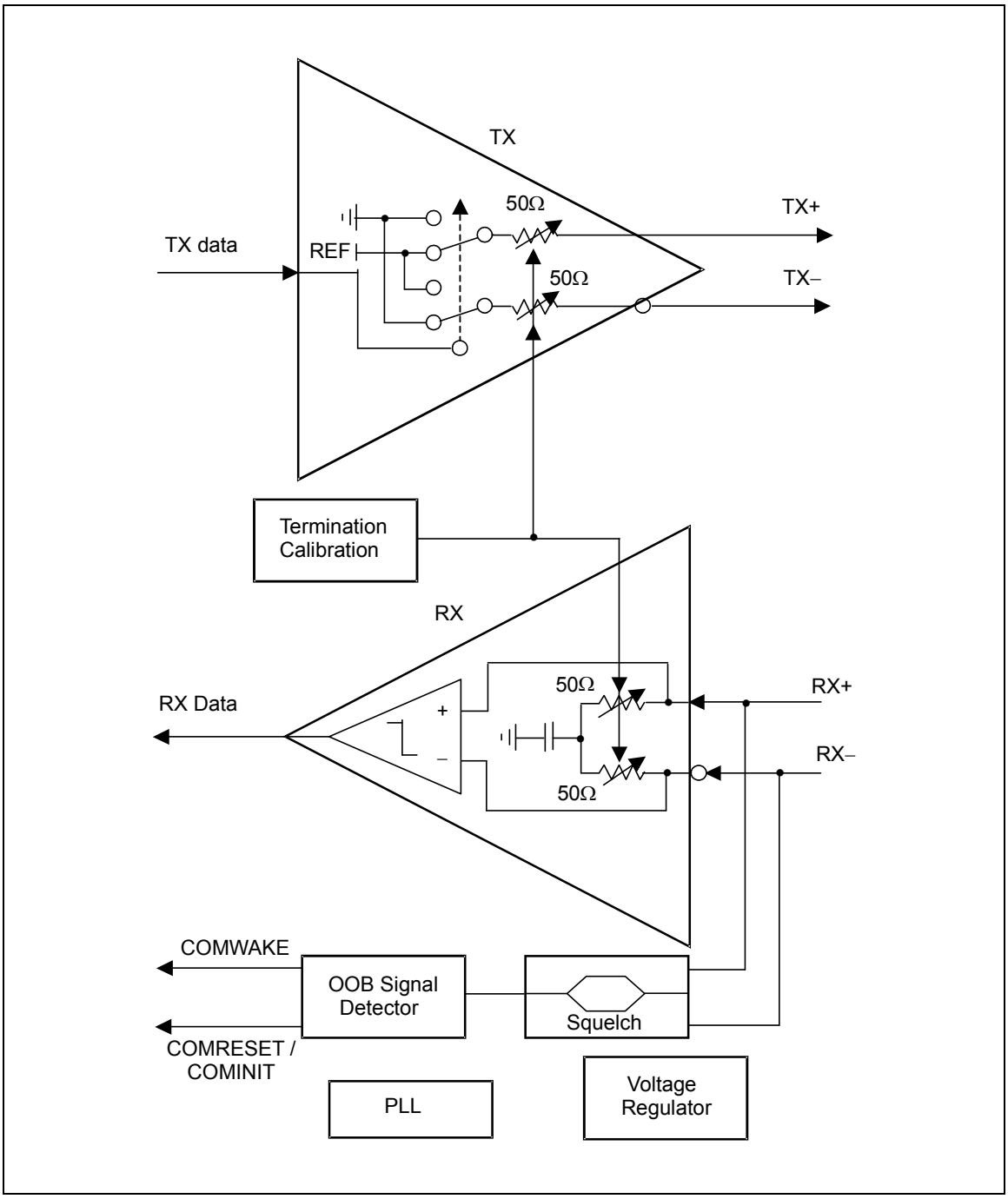


Figure 112 – Analog Front End (AFE) Block Diagram

[External AC caps not shown]

7.1.2.2 Analog Front End (AFE) Block Diagram Description

TX	This block contains the basic high-speed driver electronics.
RX	This block contains the basic high-speed receiver electronics.
Termination calibration	This block is used to establish the impedance of the RX block in order to properly terminate the high-speed serial cable.
Squelch	This block establishes a limit so that detection of a common mode signal may be properly accomplished.
OOB signal detector	This block decodes OOB signal from the high-speed input signal path.
PLL	This block is used to synchronize an internal clocking reference so that the input high-speed data stream may be properly decoded.
Voltage Regulator	This block stabilizes the internal voltages used in the other blocks so that reliable operation may be achieved. This block may or may not be required for proper operation of the balance of the circuitry. The need for this block is implementation specific.
TX+ / TX-	This is the same signal as described in the previous section: Physical plant overall block diagram description.
RX+ / RX-	This is the same signal as described in the previous section: Physical plant overall block diagram description.
TxData	Serially encoded 10b data attached to the high-speed serial differential line driver.
RxData	Serially encoded 10b data attached to the high-speed serial differential line receiver.
COMWAKE	This is the same signal as described in the previous section: Physical plant overall block diagram description.
COMRESET / COMINIT	This is the same signal as described in the previous section: Physical plant overall block diagram description.

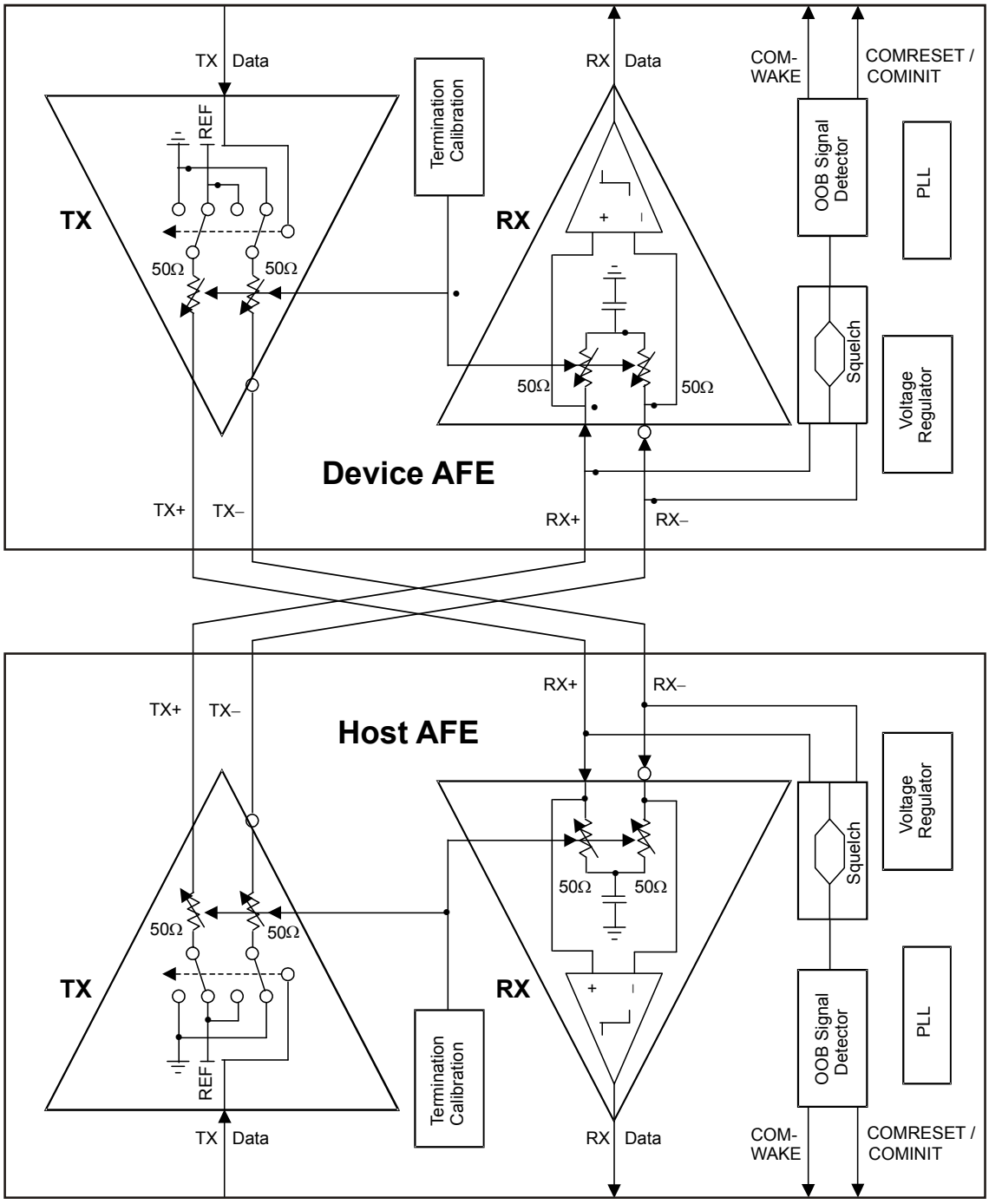


Figure 113 – Analog Front End (AFE) Cabling

[Editors note: External AC Coupling Capacitors not shown]

7.1.3 Compliance Testing

This document provides electrical specifications that when met by hosts, devices, and interconnects, satisfy the link performance specifications when combined into a system. This section of the document provides an overview of how to determine whether a Host, Device, or Interconnect is compliant to the specifications of this document.

Each electrical specification requires a specific measurement, test setup and data patterns. This section ties all of these requirements together to aid the reader in understanding what is needed for compliance testing.

Table 29, Table 30, Table 31, Table 32, Table 33, and Table 34 detail the electrical requirements for SATA compliance. Each requirement is defined in section 7.2.2. Jitter is described in section 7.3. Measurement methods for each specification are detailed in section 7.4. Section 7.5 discusses Interface States relating to OOB and power management. Section 6.6 describes the Interconnect requirements.

The Phy layer is divided into a transmitter, interconnect, and a receiver.

The SATA link is a full duplex point to point link as continuous data activity exists on each direction. For purposes of compliance testing of Hosts and Devices, the full duplex link is broken into two simplex links, one for the Host transmitting to the Device and the other for the Device transmitting to the Host. Each link is tested for compliance separately. Each transmitter to receiver Link contains the following elements:

Transmitter (IC/PCB/SATA Connector) – to –
Interconnect (Connector/Cable or PCB/Connector) – to –
Receiver (SATA Connector / PCB / IC)

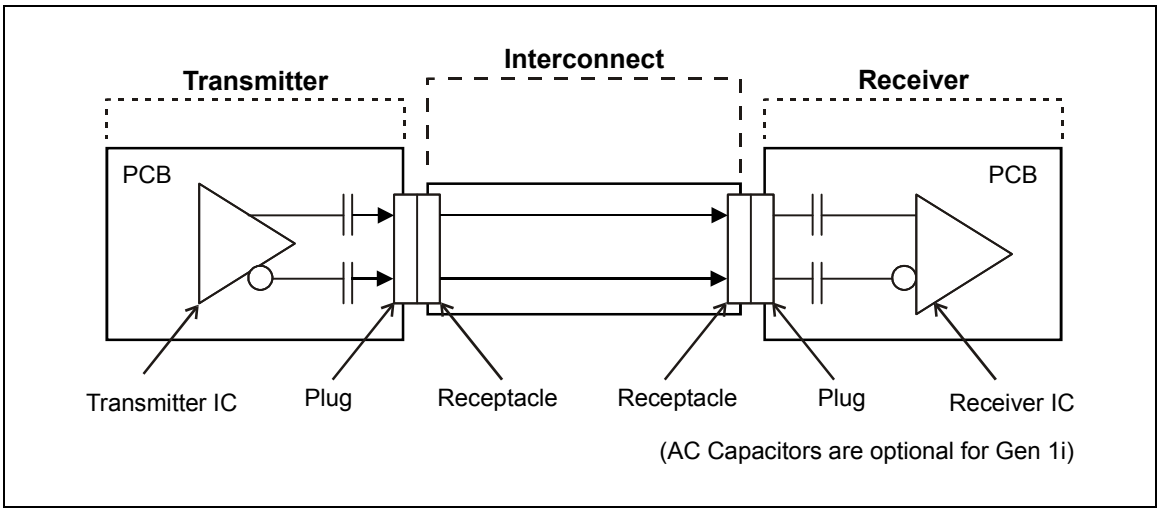


Figure 114 – The Simplex Link

In testing the compliance of SATA components that make up a system there are five Compliance Areas to be measured:

1. “Transmitted Signal”– Examine the transmitted signal quality at the compliance point for the Host/Device into a Laboratory Load. Electrical specifications include amplitude, rise/fall time, frequency, jitter, etc. The Electrical specifications apply to the signal output from the Transmitter-Under-Test at the mated connector when driving a Laboratory Load. No attempt has been made to specify the signal when attached to a cable, backplane or directly into another Device. Actual signals “In-System” may vary.

2. “Transmitter” – Examine all specified characteristics of the Transmitter from the compliance point. This includes specifications such as differential and common-mode impedance. The “Transmitter” includes the IC, which incorporates the transmitter, the PCB, the SATA connector as well as any additional components between the IC and the SATA connector.
3. “Receiver” – Examine all specified characteristics of the Receiver from the compliance point. This includes specifications such as differential and common-mode impedance. The “Receiver” includes the SATA connector, the PCB and the IC, which incorporates the receiver as well as any additional components between the IC and the SATA connector.
4. “Receiver Tolerance” – When the Receiver is presented with a worst-case “Lab-Sourced Signal”, and operating with its active transmitter, shall meet the specified Frame Error Rates. This requires carefully controlled signal sources in order to generate a worst-case signal.
5. "Interconnect" -- Examine all specified characteristics of the interconnect, using test equipment. The interconnect includes SATA connector pairs at each end. The testing requirements and procedures are described in section 6.6.

In order to determine compliance to the specifications of this document, measurements shall be performed separately with Host, Device, or Interconnect being tested when connected to test equipment. Compliance tests are not done with a Host, Device, or Interconnect connected together. Unless otherwise specified, all compliance measurements shall be taken through the mated connector pair.

NOTE: The electrical specifications in the Receiver Tolerance Table do not describe the characteristics of the received signal; these describe the Lab-Sourced Signal calibrated into a Laboratory Load and subsequently applied to the Receiver. The Receiver Tolerance Table does not describe the characteristics of a signal from a Transmitter through an Interconnect into a Laboratory Load. Received signals in a system are potentially worse due to the non-ideal impedance match of the transmitter and the receiver.

7.1.4 Link Performance

The performance of a SATA system with Host and Device linked together with an Interconnect is measured by the frame error rate, using a set of reference frames, defined by a specific set of ordered test patterns within the frame. An operating Host-Device duplex link that meets the Frame Error Rate (FER) specifications of Table 29 for both of its simplex links is deemed to fulfill Serial ATA performance levels. A Host or Device is commanded to generate the various test patterns through the use of the BIST Activate FIS or other vendor unique commands to the device under test.

7.2 Electrical Specifications

The goal of this specification is to provide a description of characteristics to ensure interoperability of SATA components; devices, hosts, and interconnects. Any combination of compliant components should provide the stated link performance. Secondly a means of validation to the requirements is described in section 7.4. Validation consists of performing tests on individual SATA components.

Serial ATA devices and hosts shall comply with the electrical specifications shown in Table 29, Table 30, Table 31, Table 32**,** Table 33**,** and Table 34**.** The transmitter consists of the driver integrated circuit (IC), printed circuit board, and mated connector pair. The receiver consists of the receiver IC, printed circuit board, and mated connector pair.

Unless otherwise stated, all specifications include the mated connector pair.

7.2.1 Physical Layer Requirements Tables

Table 29 – General Specifications

Parameters	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
Channel Speed	Gbps	Nom	1.5			3.0			6.0	7.2.2.1.1	-
Fbaud	GHz	Nom	1.5			3.0			6.0	-	-
FER, Frame Error Rate		Max	8.2e-8 at 95% confidence level			8.2e-8 at 95% confidence level			8.2e-8 at 95% confidence level	7.2.2.1.2	7.4.1
$T_{UI,}$ Unit Interval	ps	Min	666.4333			333.2167			166.6083	7.2.2.1.3	7.4.14
		Nom	666.6667			333.3333			166.6667		
		Max	670.2333			335.1167			167.5583		
f_{tol}, TX Frequency Long Term Accuracy	ppm of Fbaud	Min	-350			-350			-350	7.2.2.1.4	7.4.7
		Max	+350			+350			+350		
f_{SSC}, Spread-Spectrum Modulation Frequency	kHz	Min	30			30			30	7.2.2.1.5 7.3.3	7.4.14
		Max	33			33			33		
SSC_{tol}, Spread-Spectrum Modulation Deviation	ppm of Fbaud	Min	-5350			-5350			-5350	7.2.2.1.6 7.3.3	7.4.14
		Max	+350			+350			+350		

Parameters	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
$V_{cm,dc}$, DC Coupled Common Mode Voltage	mV	Min	200		-	(AC only)			(AC only)	7.2.2.1.7	7.4.5
		Nom	250		-	(AC only)			(AC only)		
		Max	450		-	(AC only)			(AC only)		
$V_{cm,ac\ coupled}$, AC Coupled Common Mode Voltage	mV	Min	0		-	-			-	7.2.2.1.8	7.4.29
		Max	2000		-	-			-		
Z_{diff}, Nominal Differential Impedance	Ohm	Nom	100		-	-			100	7.2.2.1.9	7.4.26
$C_{ac\ coupling}$ AC Coupling Capacitance	nF	Max	12			12			12	7.2.2.1.10	7.4.17
$t_{settle,cm}$, Common Mode Transient Settle Time	ns	Max	10		-	-			-	7.2.2.2.13	7.2.5.3
V_{trans}, Sequencing Transient Voltage	V	Min	-2.0			-2.0			-2.0	7.2.2.1.11	7.4.16
		Max	2.0			2.0			2.0		

Table 30 – Transmitter Specifications

Parameters	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
V_{trans}, Sequencing Transient Voltage LL	V	Min	-			-			-1.2	7.2.2.1.12	7.4.30
		Max	-			-			1.2		
Z_{diffTX}, TX Pair Differential Impedance	Ohm	Min	85		85	-		85	-	7.2.2.2.1	7.4.26
		Max	115		115	-		115	-		
$Z_{s\text{-}eTX}$, TX Single-Ended Impedance	Ohm	Min	40		-	-			-	7.2.2.2.2	7.4.27

Parameters	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
$RL_{DD11,TX}$, TX Differential Mode Return Loss (All Values Min)	dB	75 MHz-150MHz	14	14	-	-	-	-	-	7.2.2.2.3	7.4.13
		150 MHz-300 MHz	8	8	-	14	14	-	-		
		300 MHz-600 MHz	6	6	-	8	8	-	-		
		600 MHz-1.2 GHz	6	6	-	6	6	-	-		
		1.2 GHz-2.4 GHz	3	3	-	6	6	-	-		
		2.4 GHz-3.0 GHz	1	-	-	3	3	-	-		
		3.0 GHz-5.0 GHz	-	-	-	1	-	-	-		
$RL_{DD11,TX}$, TX Differential Mode Return Loss Start for slope	dB	Min at 300MHz	-			-	-	-	14	7.2.2.2.6	7.4.13
Slope of TX Differential Mode Return Loss	dB/dec	Nom	-			-	-	-	-13		
TX Differential Mode Return Loss Max Frequency	GHz	Max	-			-	-	-	3		

Parameters	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
$RL_{CC11,TX}$, TX Common Mode Return Loss (all Values Min)	dB	150 MHz-300 MHz	-			8	5	-	-	7.2.2.2.4	7.4.13
		300 MHz-600 MHz	-			5	5	-	-		
		600 MHz-1.2 GHz	-			2	2	-	-		
		1.2 GHz-2.4 GHz	-			1	1	-	-		
		2.4 GHz-3.0 GHz	-			1	1	-	-		
		3.0 GHz-5.0 GHz	-			1	-	-	-		
$RL_{DC11,TX}$, TX Impedance Balance (all values Min)	dB	150 MHz-300 MHz	-			30	30	-	30	7.2.2.2.5	7.4.13
		300 MHz-600 MHz	-			20	20	-	30		
		600 MHz-1.2 GHz	-			10	10	-	20		
		1.2 GHz-2.4 GHz	-			10	10	-	10		
		2.4 GHz-3.0 GHz	-			4	4	-	10		
		3.0 GHz-5.0 GHz	-			4	-	-	4		
		5.0 GHz-6.5 GHz	-			-	-	-	4		

Table 31 – Transmitted Signal Requirements

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
V_{diffTX}, TX Differential Output Voltage	mVppd	Min	400	400	400	400	400	400	-	7.2.2.2.7	7.4.5
		Min	-	-	-	-	-	-	240		7.4.3
		Nom	500		-	-			-		7.4.5
		Max	600		1600	700		1600	-		
		Max	-		-	-		-	900		7.4.3
UI_{VminTX}, TX Minimum Voltage Measurement Interval	UI		0.45-0.55		0.5	0.45-0.55		0.5	-	7.2.2.2.8	7.4.5
			-		-	-		-	0.50		7.4.3.2
$t_{20-80TX}$, TX Rise/Fall Time	ps (UI)	Min 20-80%	100 (.15)		67 (.10)	67 (.20)			33 (0.20)	7.2.2.2.9	7.4.4
		Max 20-80%	273 (.41)		273 (.41)	136 (.41)			68 (0.41)		
t_{skewTX}, TX Differential Skew	ps	Max	20			20		15	20	7.2.2.2.10	7.4.15
$V_{cm,acTX}$, TX AC Common Mode Voltage	mVp-p	Max	-			50		-	-	7.2.2.2.11	7.4.20

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
$V_{cm,acTX}$, TX AC Common Mode Voltage	dBmV(pk) (rms)	3 GHz Max	-		-	-		-	26	7.2.2.2.12	7.4.21
		6 GHz Max	-		-	-		-	30		
$D_{VdiffOOB}$, OOB Differential Delta	mV	Max	-		25	25			25	7.2.2.2.14	7.4.23
D_{VcmOOB}, OOB Common Mode Delta	mV	Max	-		50	50			50	7.2.2.2.15	7.4.22
R/F_{bal}, TX Rise/Fall Imbalance	%	Max	-		-	20		-	-	7.2.2.2.16	7.4.19
Amp_{bal}, TX Amplitude Imbalance	%	Max	-		10	10			-	7.2.2.2.17	7.4.18
TJ at Connector, Clk-Data, $f_{BAUD}/500$ JTF Defined	UI	Max	0.37		-	0.37		-	-	7.2.2.2.18 7.3	7.4.8 7.4.9
DJ at Connector, Clk-Data, $f_{BAUD}/500$ JTF Defined	UI	Max	0.19		-	0.19		-	-		
Jitter Transfer Function Bandwidth (D24.3, high pass -3dB)	MHz	Min	1.1		-	1.1		-	-	7.3.2	7.4.8
		Nom	2.1		-	2.1		-	-	7.3.2	7.4.8
		Max	3.1		-	3.1		-	-	7.3.2	7.4.8

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
Jitter Transfer Function Peaking	dB	Min	0		-	0		-	-	7.3.2	7.4.8
		Nom	0		-	0		-	-	7.3.2	7.4.8
		Max	3.5		-	3.5		-	-	7.3.2	7.4.8
Jitter Transfer Function Low Frequency Attenuation	dB	Min	69		-	69		-	-	7.3.2	7.4.8
		Nom	72		-	72		-	-	7.3.2	7.4.8
		Max	75		-	75		-	-	7.3.2	7.4.8
Jitter Transfer Function Low Frequency Attenuation Measurement Frequency	kHz		30±1%		-	30±1%		-	-	7.3.2	7.4.8
Jitter Transfer Function Bandwidth (D24.3, high pass -3dB) (Gen3)	MHz	Min	-		-	-		-	2.2	7.3.2.4	7.4.8
		Nom	-		-	-		-	4.2	7.3.2.4	7.4.8
		Max	-		-	-		-	6.2	7.3.2.4	7.4.8
Jitter Transfer Function Peaking (Gen3)	dB	Min	-		-	-		-	0	7.3.2.4	7.4.8
		Nom	-		-	-		-	0	7.3.2.4	7.4.8
		Max	-		-	-		-	3.5	7.3.2.4	7.4.8

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
Jitter Transfer Function Low Frequency Attenuation (Gen3)	dB	Min	-		-	-		-	35.2	7.3.2.4	7.4.8
		Nom	-		-	-		-	38.2	7.3.2.4	7.4.8
		Max	-		-	-		-	41.2	7.3.2.4	7.4.8
Jitter Transfer Function Low Frequency Attenuation Measurement Frequency (Gen3)	kHz		-		-	-		-	420 +/- 1%	7.3.2.4	7.4.8
TJ after CIC, Clk-Data, $f_{BAUD}/1667$	UI	Max	-		0.55	-		0.55	-	7.2.2.2.18 7.3	7.4.8 7.4.9
DJ after CIC, Clk-Data, $f_{BAUD}/1667$	UI	Max	-		0.35	-		0.35	-		
TJ before and after CIC, Clk-Data JTF Defined	UI	Max	-			-			RJ p-p meas. + 0.34	7.2.2.2.18 7.3	7.4.8 7.4.10
RJ before CIC, MFTP Clk-Data JTF Defined	UI	Max	-			-			0.18 p-p (2.14 ps 1 sigma)		

Table 32 – Receiver Specifications

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
Z_{diffRX}, RX Pair Differential Impedance	Ohm	Min	85			-		85	-	7.2.2.3.1	7.4.26
		Max	115			-		115	-		
$Z_{s\text{-}eRX}$, RX Single-Ended Impedance	Ohm	Min	40		-	-			-	7.2.2.3.2	7.4.27

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
$RL_{DD11,RX}$, RX Differential Mode Return Loss (all values Min)	dB	75MHz-150MHz	18	18	-	-	-	-	-	7.2.2.3.3	7.4.13
		150 MHz-300 MHz	14	14	-	18	18	-	-		
		300 MHz-600 MHz	10	10	-	14	14	-	-		
		600 MHz-1.2 GHz	8	8	-	10	10	-	-		
		1.2 GHz-2.4 GHz	3	3	-	8	8	-	-		
		2.4 GHz-3.0 GHz	1	-	-	3	3	-	-		
		3.0 GHz-5.0 GHz	-	-	-	1	-	-	-		
$RL_{DD11,RX}$, RX Differential Mode Return Loss	dB	Min at 300MHz	-			-			18	7.2.2.2.6	7.4.13
Slope of RX Differential Mode Return Loss	dB/dec	Nom	-			-			-13		
RX Differential Mode Return Loss Max Frequency	GHz	Max	-			-			6.0		

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
$RL_{CC11,RX}$, RX Common Mode Return Loss (all values Min)	dB	150 MHz-300 MHz	-			5	5	-	-	7.2.2.3.4	7.4.13
		300 MHz-600 MHz	-			5	5	-	-		
		600 MHz-1.2 GHz	-			2	2	-	-		
		1.2 GHz-2.4 GHz	-			1	1	-	-		
		2.4 GHz-3.0 GHz	-			1	1	-	-		
		3.0 GHz-5.0 GHz	-			1	-	-	-		

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
$RL_{DC11,RX}$, RX Impedance Balance (all values Min)	dB	150 MHz-300 MHz	-			30	30	-	30	7.2.2.3.5	7.4.13
		300 MHz-600 MHz	-			30	30	-	30		
		600 MHz-1.2 GHz	-			20	20	-	20		
		1.2 GHz-2.4 GHz	-			10	10	-	10		
		2.4 GHz-3.0 GHz	-			4	4	-	10		
		3.0 GHz-5.0 GHz	-			4	-	-	4		
		5.0 GHz-6.5 GHz	-			4	-	-	4		

Table 33 – Lab-Sourced Signal (for Receiver Tolerance Testing)

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
V_{diffRX}, RX Differential Input Voltage		Min	325	240	275	275	240	275	-	7.2.2.5.1	7.4.6
	mVppd	Min	-	-	-	-	-	-	240		7.4.3 7.4.12
		Nom	400		-	-	-	-	-		7.4.6
		Max	600		1600	750	750	1600	-		
		Max	-		-	-	-	-	1000		7.4.3 7.4.12
$t_{20\text{-}80RX}$, RX Rise/Fall Time	ps (UI)	Min 20-80%	100 (.15)		67 (.10)	67 (.20)			-	7.2.2.5.2	7.4.4
			-		-	-			62 (0.37)		7.4.4 7.4.12
		Max 20-80%	273 (.41)			136 (.41)			-		7.4.4
			-			-			75 (0.45)		7.4.4 7.4.12
UI_{VminRX}, RX Minimum Voltage Measurement Interval	UI		-		0.5	0.5			-	7.2.2.5.3	7.4.6
			-		-	-			0.5		7.4.3.2
t_{skewRX}, RX Differential Skew	ps	Max	-		80	50		75	30	7.2.2.5.4	7.4.15
$V_{cm,acRX}$, RX AC Common Mode Voltage	mVp-p	Max	100		150	100		150	100	7.2.2.5.5	7.4.11

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
$f_{cm,acRX}$, AC Common Mode Frequency	MHz	Min	2			2			2	7.2.2.5.6	7.4.11
		Max	200			200			200		

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
TJ at Connector, Clk-Data, $f_{BAUD}/500$ JTF Defined	UI	Max	0.60			0.60		-	-		
DJ at Connector, Clk-Data, $f_{BAUD}/500$ JTF Defined	UI	Max	0.42			0.42		-	-		
TJ at Connector, Clk-Data, $f_{BAUD}/1667$	UI	Max	-		0.65	-		0.65	-	7.2.2.5.7 7.3	7.4.8 7.4.11
DJ at Connector, Clk-Data, $f_{BAUD}/1667$	UI	Max	-		0.35	-		0.35	-		
TJ after CIC, Clk-Data JTF Defined	UI	Max	-		-	-		-	0.60	7.2.2.5.7 7.3	7.4.8 7.4.12
RJ before CIC, MFTP Clk-Data JTF Defined	UI	Max	-		-	-		-	0.18 p-p (2.14 ps 1 sigma)		

Table 34 – OOB Specifications

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
V_{thresh}, OOB Signal Detection Threshold	mVppd	Min	50		120	75		120	75	7.2.2.6.2	7.4.24
		Nom	100		-	125		-	125		
		Max	200		240	200		240	200		
UI_{OOB}, UI During OOB Signaling	ps	Min	646.67			646.67			646.67	7.2.2.6.3	-
		Nom	666.67			666.67			666.67		
		Max	686.67			686.67			686.67		
COMINIT/ COMRESET and COMWAKE Transmit Burst Length	ns	Min	103.5							7.2.2.6.4 7.2.2.6.1	7.4.25
		Nom	106.7								
		Max	109.9								
COMINIT/ COMRESET Transmit Gap Length	ns	Min	310.4							7.2.2.6.5	7.4.25
		Nom	320.0								
		Max	329.6								
COMWAKE Transmit Gap Length	ns	Min	103.5							7.2.2.6.6	7.4.25
		Nom	106.7								
		Max	109.9								

Parameter	Units	Limit	Electrical Specification							Detail Cross-Ref Section	Measurement Cross-Ref Section
			Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Gen3i		
COMWAKE Gap Detection Windows	ns	May detect	35 ≤ T < 175							7.2.2.6.7	7.4.25
		Shall detect	101.3 ≤ T ≤ 112								
		Shall not detect	T < 35 or T ≥175								
COMINIT/ COMRESET Gap Detection Windows	ns	May detect	175 ≤ T < 525							7.2.2.6.8	7.4.25
		Shall detect	304 ≤ T ≤ 336								
		Shall not detect	T < 175 or T ≥ 525								

7.2.2 Phy Layer Requirements Details

7.2.2.1 General Specifications Details

This section contains the details on Table 29 entries.

7.2.2.1.1 Channel Speed

A reference value showing the nominal rate of data through the channel.

7.2.2.1.2 Frame Error Rate

Frame error rate is the measure of link performance using all the intermediate circuit blocks in the chain from low-level Phy layer, Link layer, through Transport layer. Frame error rate is a system level test, not a compliance test. Error detection is at the frame level using the CRC (Cyclic Redundancy Check) error detection mechanism, and respective reporting to the higher layer levels.

7.2.2.1.3 Unit Interval

This is the operating data period (nominal value, architecture specific), excluding jitter. This value includes the long-term frequency accuracy and the Spread Spectrum Clock FM frequency deviation (rounded to 4 places).

This is the time interval value of each cycle of the Reference Clock.

7.2.2.1.4 TX Frequency Long Term Stability

This specifies the allowed frequency variation from nominal; this does not include frequency variation due to jitter, Spread Spectrum Clocking, or phase noise of the clock source.

7.2.2.1.5 Spread Spectrum Modulation Frequency

The modulation frequency of the Spread Spectrum frequency modulation. See further details of Spread Spectrum in section 7.3.3.

7.2.2.1.6 Spread Spectrum Modulation Deviation

This is the allowed frequency variation from the nominal Fbaud value in Table 27 when Spread Spectrum Clocking (SSC) is used. This deviation includes the long-term frequency variation of the transmitter clock source, and the SSC frequency modulation on the transmitter output. The frequency variation limits are measured using the SSC profile measurement described in section 7.4.11. See further details of Spread Spectrum in section 7.3.3.

7.2.2.1.7 DC Coupled Common Mode Voltage (Gen1i)

The Common mode DC level is defined as [(TX+) + (TX-)]/2 and [(RX+) + (RX-)]/2 measured at the mated connector.

This requirement only applies to Gen1i DC-coupled designs (no blocking capacitors) that hold the common-mode DC level at the connector. The four possible common mode biasing configurations shown in Figure 115 below demonstrate that only DC-coupled designs need sustain the specified common-mode level to ensure interoperability. AC coupled designs may allow the DC level at the connector to float. The SATA interfaces defined as Gen1x, Gen2i, Gen2x, and Gen3i shall be AC-coupled and this requirement does not apply to these.

A DC-coupled receiver shall weakly hold the common-mode level of its inputs to the $V_{cm,dc}$ value specified in Table 29. A DC-coupled transmitter shall transmit with the $V_{cm,dc}$ value specified in Table 29 while driving into a 100 Ohm differential impedance.

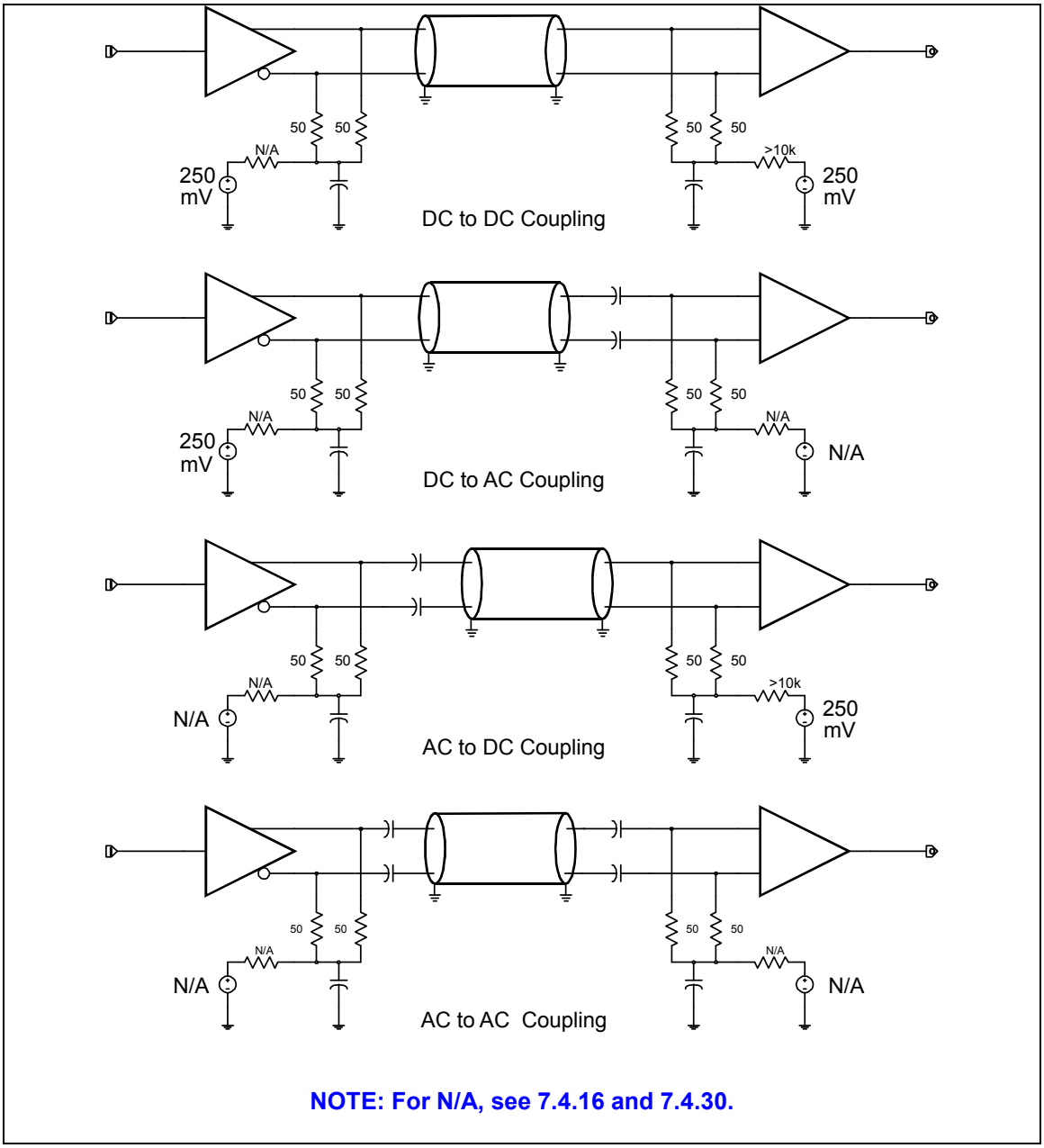


Figure 115 – Common Mode Biasing Examples for Gen1i (Informative)

7.2.2.1.8 AC Coupled Common Mode Voltage

The SATA interface, defined as Gen1i, may be AC or DC coupled as shown in Figure 115. The SATA interfaces defined as Gen1x, Gen2i, Gen2x, and Gen3i shall be AC-coupled.
Figure 116 shows an example of a fully AC-coupled system.

Compliance points for SATA are defined at the connector. The AC coupled common mode voltage in Table 29 defines the open circuit DC voltage level of each single-ended signal at the IC side of the coupling capacitor in an AC coupled Phy and it shall be met during all possible power and electrical conditions of the Phy including power off and power ramping. Since the Gen1x, Gen2i, Gen2x, and Gen3i specification defines only the signal characteristics as observable at the connector, this value is not applicable to those specifications. The common mode transient requirements defined in Table 29 were determined sufficient to limit stresses on the attached components under transient conditions, which was the sole intent of the AC, coupled common mode voltage requirement. Due to this, the following is true even for Gen1i where $V_{cm,ac\ coupled}$ applies: AC coupled common mode voltage levels outside the specified range may be used provided that the transient voltage requirements of Table 29 are met.

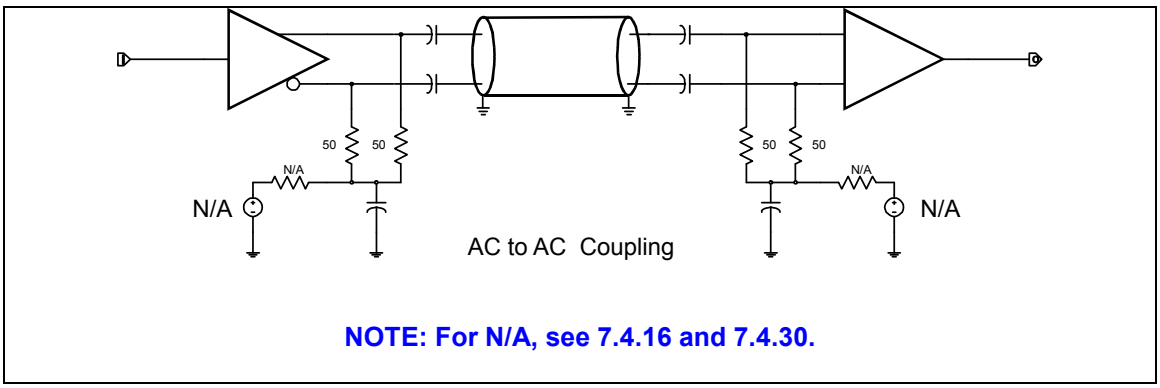


Figure 116 – Common Mode Biasing for Gen1x, Gen2i, Gen2x, and Gen3i

7.2.2.1.9 Nominal Differential Impedance (Gen1i)

The nominal impedance of all components in a SATA system.

7.2.2.1.10 AC Coupling Capacitance

The value of the coupling capacitor used in AC coupled implementations. AC coupling is optional for Gen1i and mandatory of Gen1x, Gen2i, Gen2x, and Gen3i.

Coupling Capacitor Characteristics (Informative): The physical size of the capacitor should be as small as practical to reduce the capacitance to ground. Body sizes larger than 0603 (or values less than 300 pF) should be avoided since they are likely to result in a failure of the return loss requirements in 7.2.2.2.3, and 7.2.2.3.3.

The physical size of the capacitor should be as small as practical to reduce the capacitance to ground. Body sizes larger than 0603 should be avoided, as they are likely to result in a failure of the return loss requirements in Table 30 and Table 31.

7.2.2.1.11 Sequencing Transient Voltage

This parameter addresses the transient voltages on the serial data bus during power sequencing and power mode changes. Since either the receiver or the transmitter may be affected by power

sequencing transients, the term "aggressor" is used to indicate the sequencing interface circuit and the term "victim" is used for the interface circuit receiving the transient.

In order to limit the voltage and energy seen by the victim receiver or transmitter circuitry during power sequencing, several parameters of the aggressor and victim are involved. Although parameters of the victim, such as common mode voltage and single ended impedance, affect the observed transient, this measurement addresses limiting the aggressor contribution.

The aggressor common mode voltage, single ended impedance, and AC coupling capacitor value determine the level of the sequencing transient. This measurement addressed the common mode voltage of the aggressor. The rate of change of the power on or power off ramp also affects this level. The limits provided allow for power up or power down ramps at rates faster than the time constants of the signal lines, although practical systems may not achieve this rate. This measurement shall include the test conditions of power on and power off ramping at the fastest possible rate expected in systems using the Phy, as well as any power mode transitions.

7.2.2.1.12 Sequencing Transient Voltage Lab Load (Gen3i)

This parameter addresses the transient voltages on the serial data bus during power sequencing under the test condition of a lab load (see 7.2.2.4 for lab load details). Measuring the transient voltage with a lab load combines the effects of the bias voltage and series termination. Separately measuring the impedance of the circuit is not required. Since some circuits calibrate the impedance after power ramping is compete, measuring impedance during a transient condition becomes challenging.

An open circuit voltage measurement, as outlined in 7.4.16, Sequencing Transient Voltage, shall be required to prevent overstressing victim circuits with high impedance common mode voltage.

7.2.2.2 Transmitter Specification Details

This section contains the details on Table 30 entries.

7.2.2.2.1 TX Pair Differential Impedance (Gen1i)

As seen by a differential TDR with 100 ps (max) edge looking into connector (20%-80%). Measured with TDR in differential mode.

7.2.2.2.2 TX Single-Ended Impedance (Gen1i)

As seen by TDR with 100 ps (max) edge looking into connector (20%-80%). The TDR is 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.

This requirement shall be met during all possible power and electrical conditions of the Phy including power off and power ramping.

7.2.2.2.3 TX and RX Differential Mode Return Loss (Gen2i, Gen2m) (Gen1i, Gen1m alternate)

This specification describes transmitter output impedance and receiver input impedance in terms of both the peak value of a reflection given an incident step of known risetime and also in terms of return loss. The return loss measurement shall be sufficient to verify compliance with Gen1 and Gen2 requirements. In order to ensure Gen1 designs passing the TDR differential impedance method as previously required are not invalidated due to this change, either method shall be sufficient to verify compliance with Gen 1 requirements. Verification of compliance by both methods shall not be required.

The differential mode return loss is defined as the ratio (expressed in dB) of differential mode incident power to differential mode reflected power both at a 100 Ohm impedance level. In the system environment the purpose of controlling the return loss of devices and hosts is to limit signal reflections that cause data dependent jitter. These signal reflections in question are over and above those that exist in compliance testing when connected to a matched source or load.

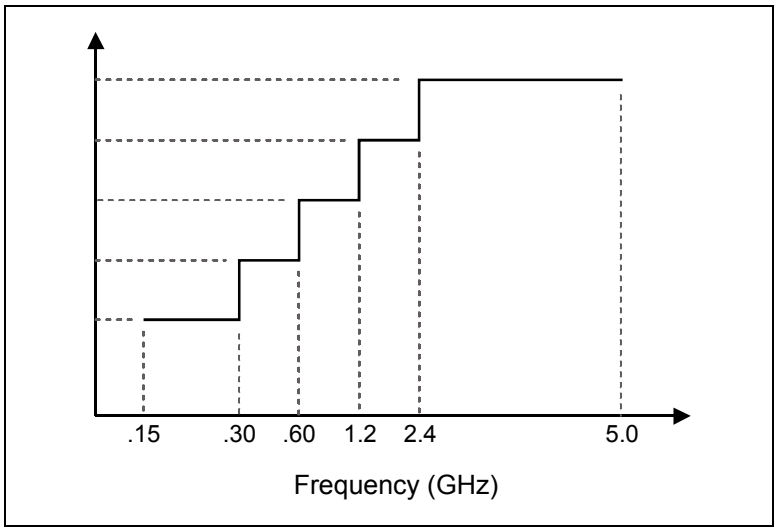


Figure 117 – Differential Return Loss Limits

7.2.2.2.4 TX Common Mode Return Loss (Gen2i, Gen2m)

The common mode return loss is defined as the ratio (expressed in dB) of common mode incident power to common mode reflected power both at a 25 Ohm impedance level. The intended signal propagation mode in SATA is the differential mode. However, imperfections in the system create some coupling between the common and differential modes. This has three consequences: radiated emissions, noise susceptibility, and signal degradation. Common mode reflections exacerbate these impairments. The common mode return loss is a bound on the magnitude of common mode reflections in the system.

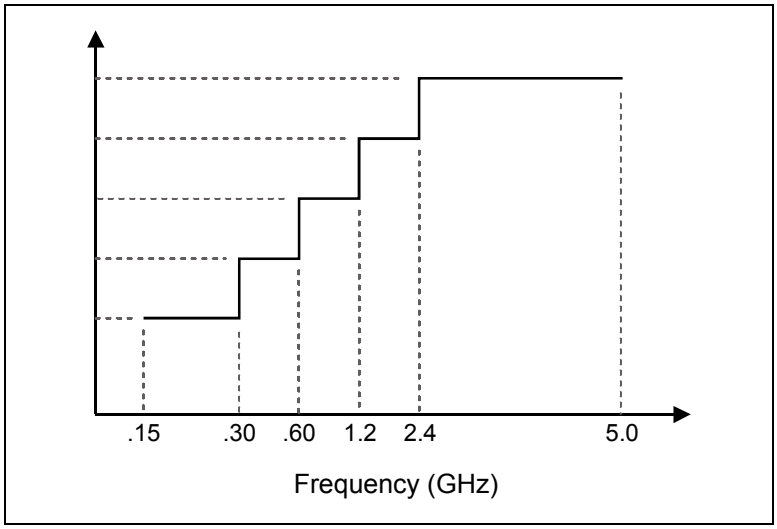


Figure 118 – Common Mode Return Loss Limits

7.2.2.2.5 TX Impedance Balance (Gen2i, Gen2m, Gen3i)

Impedance balance is defined as the ratio (expressed in dB) of common mode incident power at a 100 Ohm impedance level to differential mode reflected power at a 25 Ohm impedance level. The impedance balance is a bound on the coupling between common and differential modes.

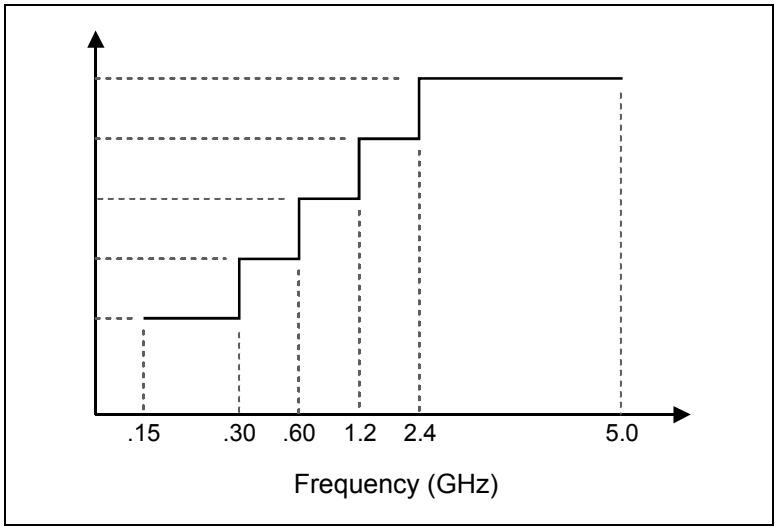


Figure 119 – Impedance Balance Limits

7.2.2.2.6 TX and RX Differential Mode Return Loss (Gen3i)

This specification describes transmitter output impedance and receiver input impedance in terms of return loss. Return loss is specified as starting and ending points, with a defined slope between. The return loss shall remain below the line shown in Figure 120, from the starting frequency to the ending frequency.

The differential mode return loss is defined as the ratio (expressed in dB) of differential mode incident power to differential mode reflected power both at a 100 Ohm impedance level. In the system environment the purpose of controlling the return loss of devices and hosts is to limit signal reflections that cause data dependent jitter. These signal reflections in question are over and above those that exist in compliance testing when connected to a matched source or load.

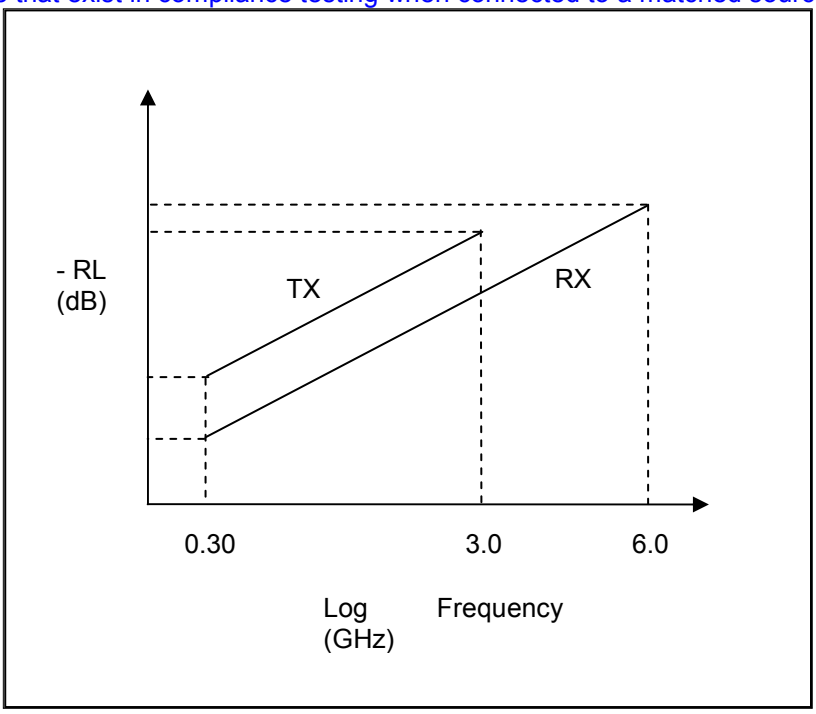


Figure 120 – Differential Return Loss Limits,Gen3i, TX and RX

Transmitted Signal Requirements Details

This section contains the details on Table 31 entries.

7.2.2.2.7 TX Differential Output Voltage

The differential voltage [(TX+) – (TX-)] measured at the Transmitter shall comply with the respective electrical specifications of section 7.2.

This is measured at mated Serial ATA connector on transmit side including any pre-emphasis. For Gen3i the maximum differential output voltage is likewise measured at the TX compliance point, but the minimum differential output voltage is measured after the Gen3i CIC. (see section 7.4.3)

7.2.2.2.8 TX Minimum Voltage Measurement Interval

The point within a UI where the signal shall meet minimum levels.

7.2.2.2.9 TX Rise/Fall Time

Rise times and fall times are measured between 20% and 80% of the signal, see Figure 121. The rise and fall time requirement $t_{r/f}$ applies to differential transitions (TX+ – TX-), for both normal and OOB signaling.

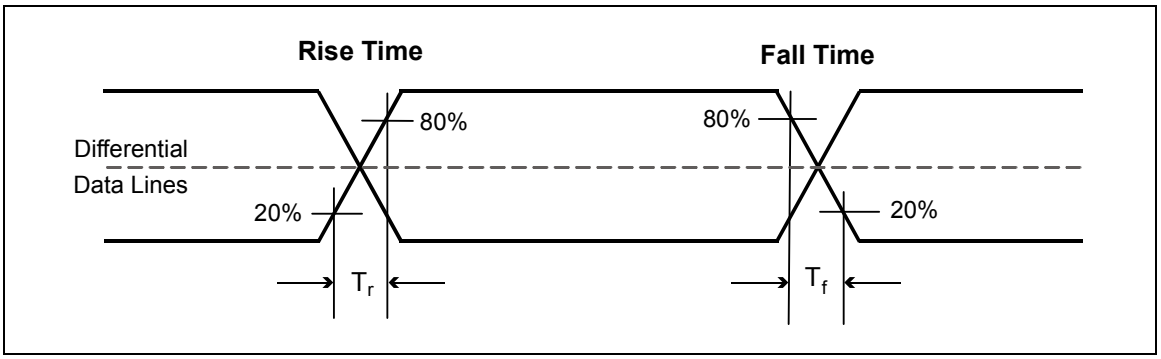


Figure 121 – Signal Rise and Fall Times

7.2.2.2.10 TX Differential Skew

TX Differential Skew is the time difference between the single-ended mid-point of the TX+ signal rising/falling edge, and the single-ended mid-point of the TX- signal falling/rising edge. It is an important parameter to control as excessive skew may result in increased high frequency jitter and common mode noise levels seen at the far end of the interconnect. The effects on the receiver are addressed in more detail in section 7.2.2.5.4. Excessive TX Differential Skew also increases EMI emissions.

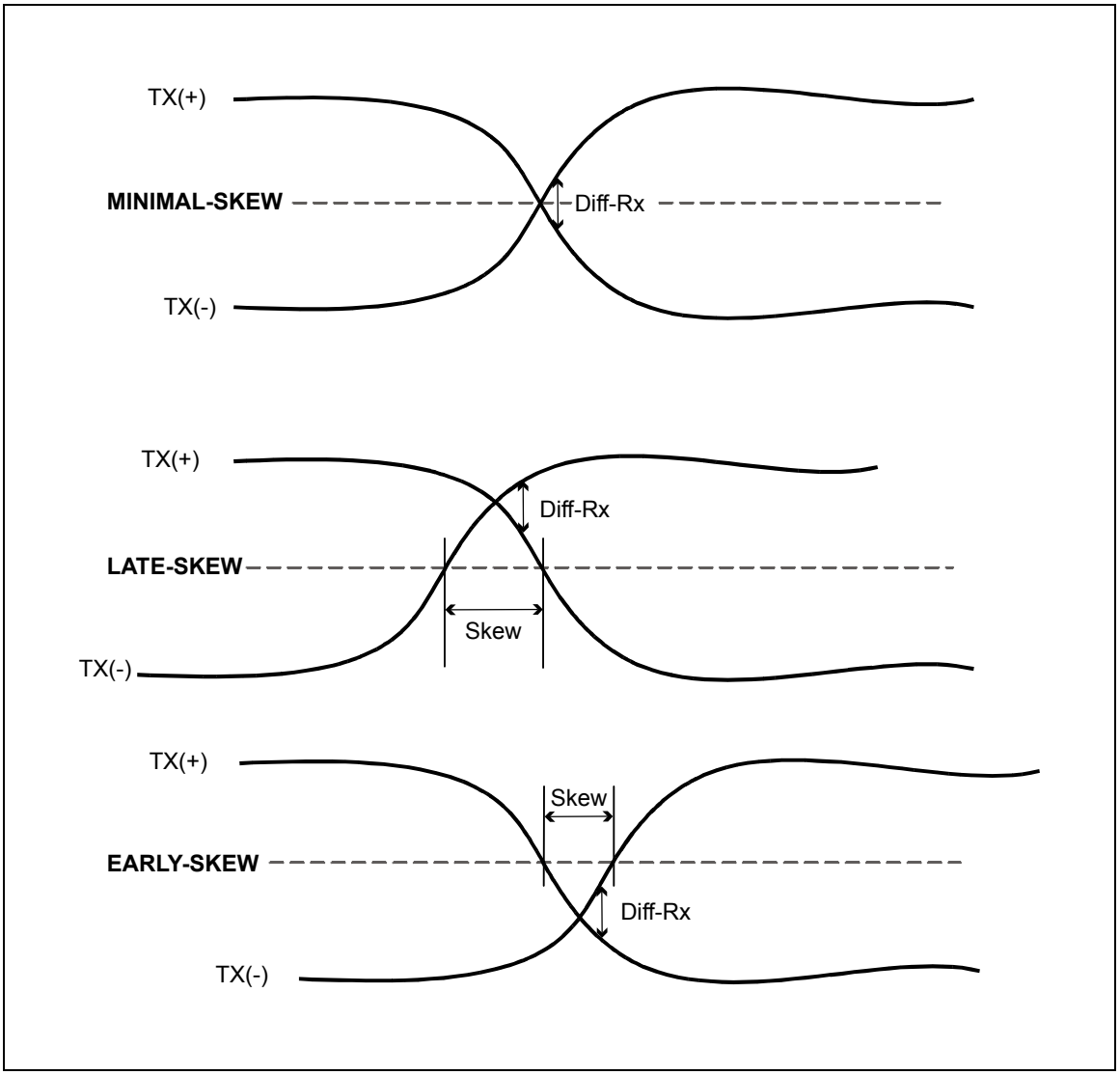


Figure 122 – TX Intra-Pair Skew

7.2.2.2.11 TX AC Common Mode Voltage (Gen2i, Gen1x, Gen2x)

Maximum sinusoidal amplitude of common mode signal measured at the transmitter connector.

The Transmitter shall comply to the electrical specifications of section 7.2, 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 us.

7.2.2.2.12 TX AC Common Mode Voltage (Gen3i)

Maximum sinusoidal amplitude of common mode signal measured at the transmitter connector.

The Transmitter shall not deliver more output voltage than that specified in Table 30 using the common mode voltage measuring technique defined in section 7.4.21.

7.2.2.2.13 Common Mode Transient Settle Time (Gen1i)

In Gen1i transmitters, this is the maximum time for common-mode transients to settle to within 25 mV of their previous state common mode voltage during transitions to and from the idle bus condition.

7.2.2.2.14 OOB Differential Delta (Gen2i,Gen3i, Gen2m,Gen1x, Gen2x)

The difference between the average differential value during the idle bus condition and the average differential value during burst on transitions to and from the idle bus condition.

During OOB transmission, imperfections and asymmetries in transmitters may generate error signals that impair proper detection by a receiver. The OOB Differential Delta describes an error from the difference in transmitter DC offset during the idle and active conditions. Since the transmitter is alternating between idle and active conditions each with different DC offsets, an AC error voltage is generated which is a square wave at about 1 / (2*106 ns) = 4.7 MHz. The AC error voltage propagates through the interconnect and causes an offset in the receiver OOB detector.

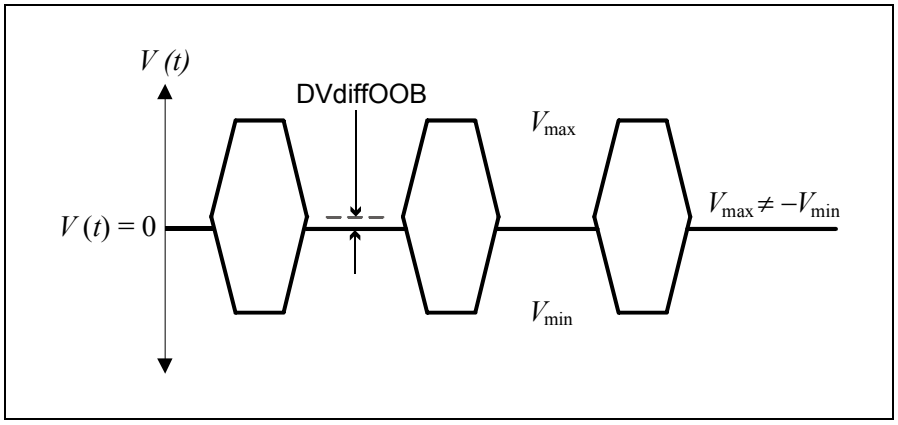


Figure 123 – OOB Differential Delta (at Compliance Point with AC Coupling)

7.2.2.2.15 OOB Common Mode Delta (Gen2i, Gen3i,Gen1x, Gen2x)

The difference between the common mode value during the idle bus condition and the common mode value during a burst on transitions to and from the idle bus condition.

7.2.2.2.16 TX Rise/Fall Imbalance

The match in the rise of TX+ and fall of TX- determined by the functions: absolute value (TX+,rise – TX-,fall)/average where average is (TX+,rise + TX-,fall)/2 and all rise and fall times are 20-80%.

The match in the fall of TX+ and rise of TX- determined by the function: absolute value (TX+,fall – TX-,rise)/average where average is (TX+,fall + TX-,rise)/2 and all rise and fall times are 20-80%.

7.2.2.2.17 TX Amplitude Imbalance (Gen2i, Gen1x, Gen2x)

The match in the amplitudes of TX+ and TX- determined by the function: absolute value (TX+ amplitude - TX- amplitude)/average where average is (TX+ amplitude + TX- amplitude)/2 and all amplitudes are determined by mode (most prevalent) voltage.

7.2.2.2.18 Clock-to-Data Transmit Jitter (Gen1i, Gen1m, Gen1x, Gen2i, Gen2m, Gen2x, Gen3i)

Transmitters shall meet the jitter specifications for the Reference Clock characteristics specified in each case.

Table 31 shows the maximum amount of jitter that a transmitter may generate and still be SATA compliant and section 7.4.8 describes the measurement. Since this specification places the compliance point afterthe connector, any jitter generated at the package connection, on the printed circuit board, and at the board connector shall be included in the measurement.

7.2.2.3 Receiver Specification Details

This section contains the details on Table 32 entries.

7.2.2.3.1 RX Pair Differential Impedance (Gen1i)

As seen by a differential TDR with 100 ps (max) edge looking into connector (20%-80%). Measured with TDR in differential mode.

7.2.2.3.2 RX Single-Ended Impedance (Gen1i)

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.

This requirement shall be met during all possible power and electrical conditions of the Phy including power off and power ramping.

7.2.2.3.3 RX Differential Mode Return Loss (Gen2i, Gen2m)

Receiver differential mode return loss is measured similar to transmitter differential mode return loss. See details in section 7.2.2.2.3.

7.2.2.3.4 RX Common Mode Return Loss (Gen2i, Gen2m)

Receiver common mode return loss is measured similar to transmitter common mode return loss. See details in section 7.2.2.2.4.

7.2.2.3.5 RX Impedance Balance (Gen2i, Gen2m)

Receiver impedance balance is measured similar to transmitter impedance balance. See details in section 7.2.2.2.5.

7.2.2.3.6 RX Differential Mode Return Loss (Gen3i)

Receiver differential mode return loss is measured similar to transmitter differential mode return loss. See details in section 7.2.2.2.6.

7.2.2.4 Lab Load Details

The Lab Load is an electrical test system connected to the unit under test. The serial transmitter signals from the UUT are connected through a “mated SATA connector pair” module consisting of connectors and cables to a HBWS terminated into two 50 Ohms (plus and minus 5 Ohms) loads. The cables shall be 50 Ohms (plus and minus 5 Ohms) impedance. The inputs of the Laboratory Load (from the back of the mated SATA connector to the 50 Ohm load within the HBWS) shall have an individual return loss greater than 20 dB over a bandwidth of 100 MHz to 5.0 GHz, and greater than 10 dB from 5 GHz to 8 GHz. The skew between the channels under test shall have 10 Picoseconds or less after compensation. The LL consists of this total assembly. The LL does not include the “other half of the mated connector” which is considered part of the UUT but is physically located on the LL. The LL is shown in Figure 124.

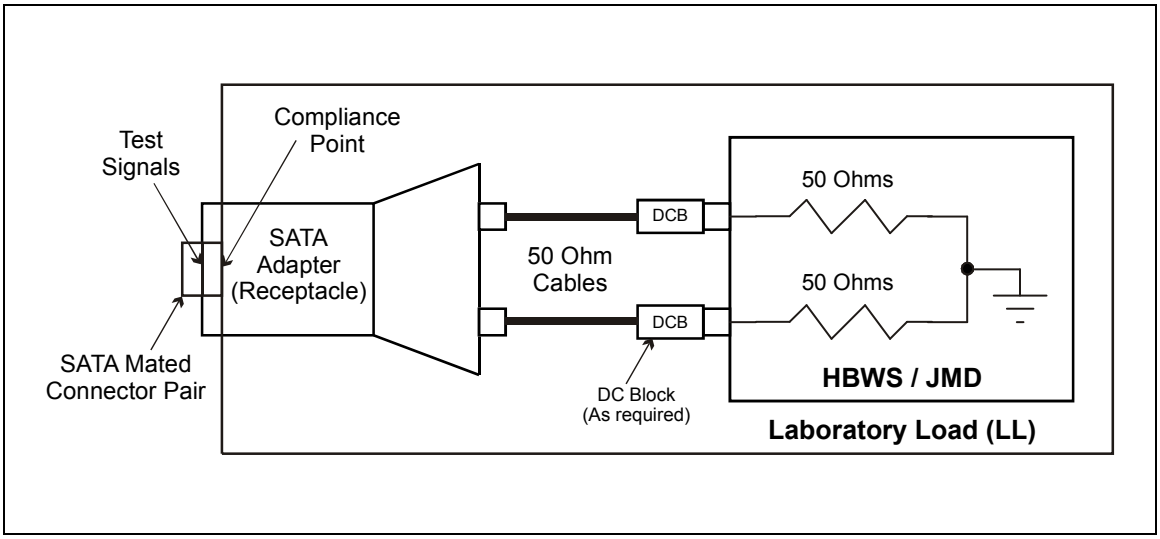


Figure 124 – LL Laboratory Load

The electrical characteristics of the LL shall be greater than the required performance of the parameter being measured such that the LL effects of the on the parameter under test may be successfully compensated for, or de-embedded, in the measured data.

7.2.2.5 Lab-Sourced Signal Details

This section contains the details on Table 33 entries.

The Laboratory Sourced Signal or Lab-Sourced Signal is an instrument and electrical test system connected to the unit under test. The LSS provides a signal to the UUT at the defined impedance level of 100 Ohms differential and 25 Ohms common mode. The LSS may also provide a SATA signal with impairments such as jitter and common mode noise. The LSS may consist of several instruments in combination with fixturing to create a signal with impairments.

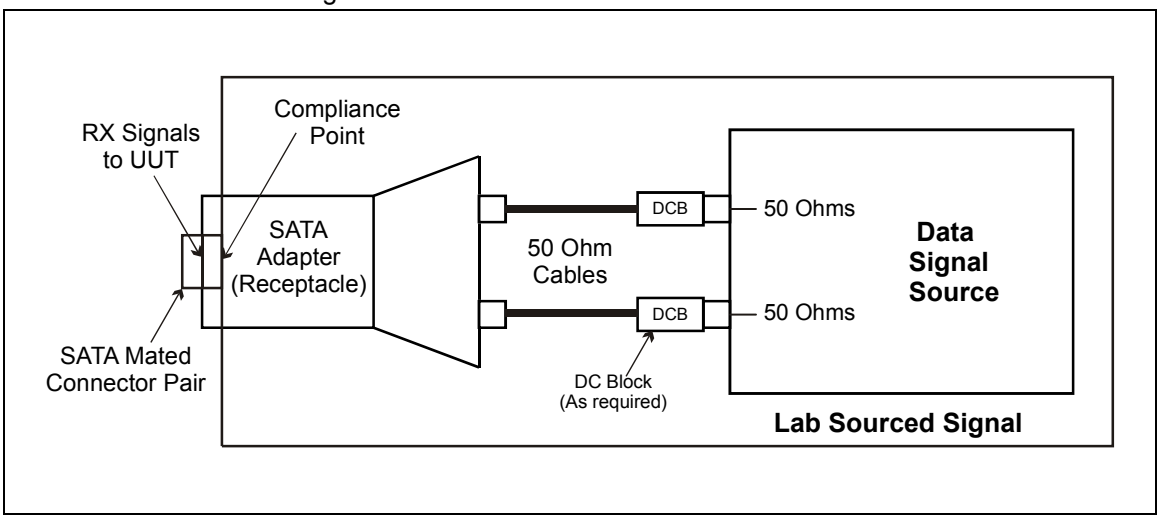


Figure 125 – LSS Lab-Sourced Signal

The Lab-Sourced signal is a laboratory generated signal which is calibrated into an impedance matched load of 100 Ohms differential and 25 Ohms common mode and then applied to the RX+ and RX- signals of the Receiver Under Test.In the case of Gen3i, the Gen3i CIC is inserted in the signal path applied to the RX. (see section 7.4.12) The load used to calibrate the LSS shall have an individual return loss greater than 20 dB over a bandwidth of 100 MHz to 5.0 GHz, and greater than 10 dB from 5 GHz to 8 GHz. During calibration, the characteristics of the Lab-Sourced signal shall comply with the specifications of Table 33. When this signal is then applied to the Receiver Under Test the Frame Error Rate specifications of Table 29 shall be met.

7.2.2.5.1 RX Differential Input Voltage

The RX Differential Input Voltage is the range of input voltage under compliance test conditions that a receiver shall operate to the required link performance level. This is one range of input conditions a receiver shall tolerate (see section 7.4.11).

The Serial ATA system has a transmitter and receiver with impedances near the nominal system impedance of 100 Ohms. The voltage at compliance points is strongly dependent on the transmitter, receiver, and interconnect impedances. The RX differential input voltage is delivered from an impedance matched signal source into a matched load (see Figure 126). When the actual receiver is substituted for the matched load, the voltage changes by an amount that is receiver design dependent. This change is part of the receiver design burden.

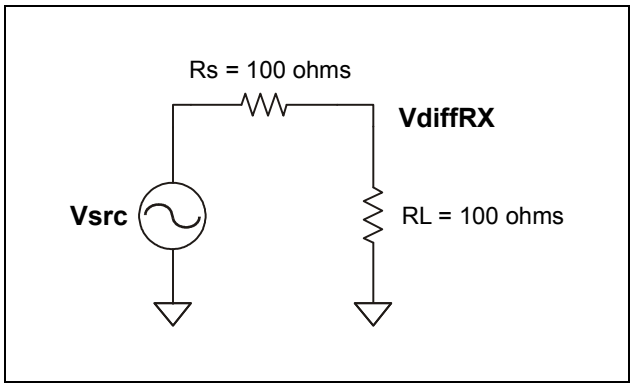


Figure 126 – RX Differential Input Voltage Conditions

The RX differential input voltage does not describe the voltage delivered from the interconnect. The interconnect output impedance is not equal to the nominal system impedance over the entire frequency range. It is not the voltage at a matched load delivered from interconnects, nor is it the voltage at a receiver delivered from the interconnect. Example calculations demonstrating this are given in section 7.4.5.

7.2.2.5.2 RX Rise/Fall Times

Rise times and fall times are measured between 20% and 80% of the signal. The rise and fall time requirement $t_{20\text{-}80RX}$ applies to differential transitions (applied to RX+ and RX- for Gen1 and Gen2). For Gen3i the maximum and minimum RX rise time and fall time requirements are applied to the Data Signal Source before the Gen3i CIC. (see Section 7.4.12)

7.2.2.5.3 RX Minimum Voltage Measurement Interval

The point in a UI that the signal shall meet minimum levels.

7.2.2.5.4 RX Differential Skew (Gen2i, Gen1x, Gen2x, Gen3i)

RX Differential Skew is the time difference between the single-ended mid-point of the RX+ signal rising/falling edge, and the single-ended mid-point of the RX- signal falling/rising edge, as measured at the RX connector. The receiver should tolerate the RX skew levels per Table 33, as generated by a Lab-Sourced Signal.

The receiver differential skew is an important parameter to consider, as excessive skew may result in increased high frequency jitter and high frequency common mode noise seen at the high-speed differential receiver. Figure 127 depicts how late and early skew signaling affect the time at which the differential receiver resolves the differential input signals. For the minimal skew case, when the single-ended slew rate is at maximum, at the crossover, the UI width is also maximized. However, this is not the case for the early and late skew cases. The high frequency common mode noise is a result of the rapid changing of the operating point of the high-speed receiver. Section 7.4.15 describes the applicable measurement method that should be used to calibrate the intentionally skewed Lab-Sourced Signal output into the receiver.

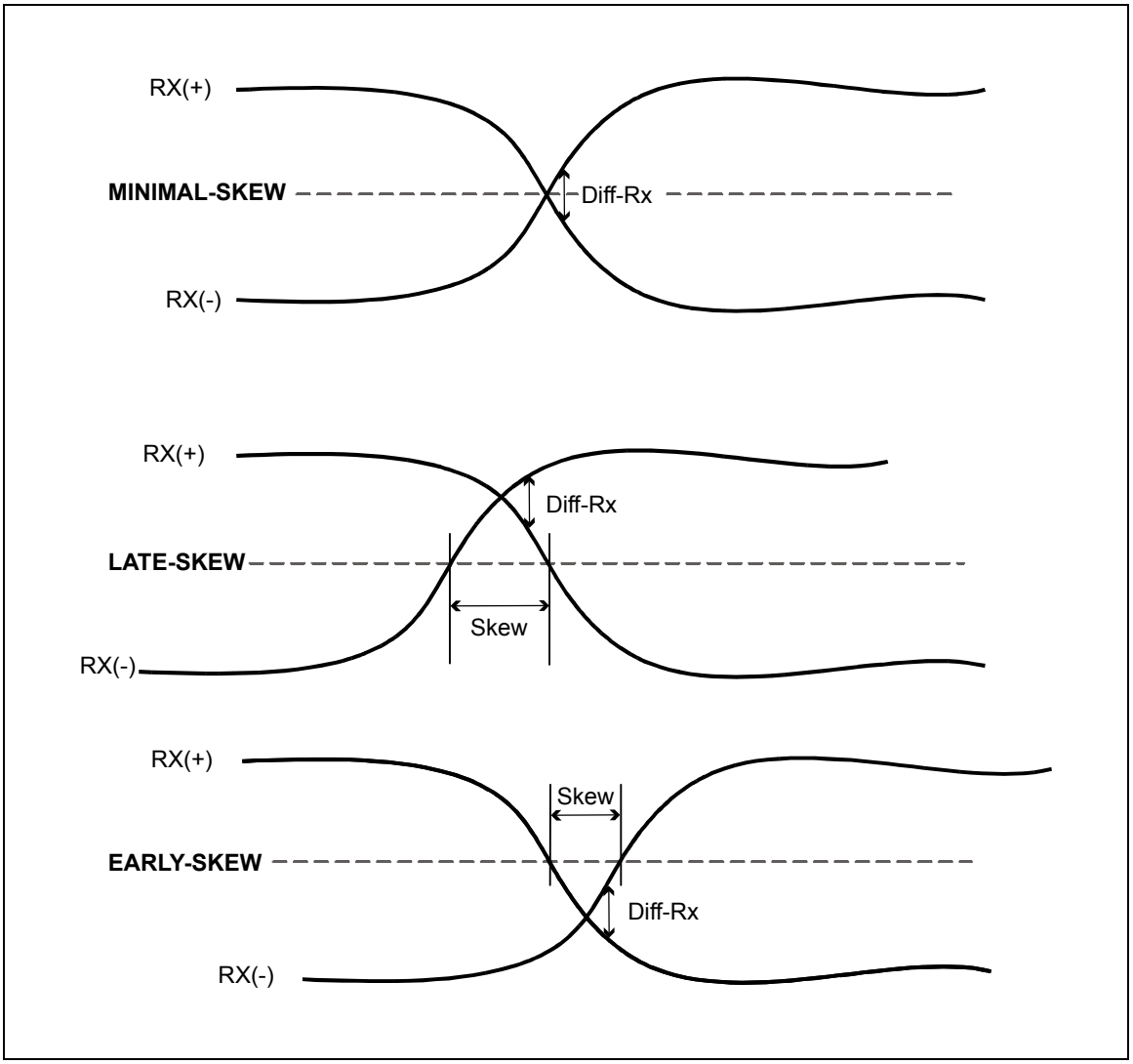


Figure 127 – RX Intra-Pair Skew

7.2.2.5.5 RX AC Common Mode Voltage (Gen2i, Gen1x, Gen2x)

Max peak-to-peak sinusoidal amplitude of AC common mode signal [(RX+) + (RX-)]/2.

The Receiver shall operate to within the frame error rate cited in Table 29, when subjected to a sinusoidal common mode interfering signal with peak-to-peak voltage $V_{cmRX,ac}$ defined in Table 33 and swept across the frequency range, $f_{cm,acRX}$, defined in Table 33 at a sweep rate period no shorter than 33.33 us.

7.2.2.5.6 AC Common Mode Frequency

All receivers shall be able to tolerate sinusoidal common-mode noise components inside this frequency range with amplitude of $V_{cm,acRX}$.

7.2.2.5.7 Clock-Data Receiver Jitter Tolerance (Gen1i, Gen1m, Gen2i, Gen2m, Gen1x, Gen2x)

Jitter tolerance is the ability of the receiver to recover data in the presence of jitter. The minimum amount of jitter that a receiver shall be able to operate is the jitter tolerance specification provided in Table 33 and section 7.4.11 describes the measurement for Gen1 and Gen2.

7.2.2.5.8 Clock-Data Receiver Jitter Tolerance (Gen3i)

See 7.4.12 for Gen3i jitter tolerance measurement details.

7.2.2.6 OOB Specifications Details

This section provides details on Table 34**.**

7.2.2.6.1 OOB Signal Burst Generation

Out-Of-Band (OOB) signals are groupings of waveforms made up of low frequency, waveform bursts, interspersed with idle gaps. These do not appear during normal data stream transfers, but are used to communicate low frequency identification information during initial notification and calibration periods, before data transfers begin. The OOB signal consists of a defined amount of idle time followed by a defined amount of burst time, this combination repeated for multiple iterations. During the idle time, the physical link carries D.C. idle. During the burst time, the physical link carries low frequency signal transitions intended to be repetitive waveshapes for envelope detection means. These bursts are translated into ON/OFF times as a means for very low speed communication. The OOB signal OFF time is determined by the length of idle time between the waveform bursts.

The signal patterns used during the OOB bursts shall be comprised of D24.3 characters (preferred) or ALIGN primitives (allowed), transmitted at the Gen1 rate. The OOB burst is only required to generate an envelope for detection by A.C. coupled detection circuitry. A burst of D24.3 characters at Gen1 speed is equivalent to a square wave pattern that is a '1' for 2 UIoob periods and then a '0' for 2 UIoob periods, or simply, a squarewave with a period of 2.66 nsec. All data speed generations shall use the Gen1 OOB burst speed, establishing a singular requirement for all OOB detection circuitry.

7.2.2.6.2 OOB Signal Detection Threshold

Differential signal amplitude detected as activity by the squelch detector during OOB signaling.

V_{diffRX} signals less than the minimum V_{thresh} defined in Table 34 shall not be detected as activity. Signal levels greater than the maximum V_{thresh} defined in Table 34 shall be detected as activity.

7.2.2.6.3 UI During OOB Signaling (UI_{OOB})

Average data period during OOB burst transmission (at Gen1 speed +/- 3%).

7.2.2.6.4 COMINIT/COMRESET and COMWAKE Transmit Burst Length

Burst length in ns, as measured from the first crossing point (+100mV or –100mV) of the burst to the last crossing point (+100mV or –100mV) of the burst.

7.2.2.6.5 COMINIT/COMRESET Transmit Gap Length

Gap length in ns, as measured from the last crossing point (+100mV or –100mV) of one COMINIT/COMRESET burst to the first crossing point (+100mV or –100mV) of the following COMINIT/COMRESET burst.

7.2.2.6.6 COMWAKE Transmit Gap Length

Gap length in ns, as measured from the last crossing point (+100mV or –100mV) of one COMWAKE burst to the first crossing point (+100mV or –100mV) of the following COMWAKE burst.

7.2.2.6.7 COMWAKE Gap Detection Windows

Three timing ranges defining the validation and invalidation of COMWAKE gaps; see Table 34.

Any OOB gap between bursts falling in the defined “may detect” range may be recognized as a valid COMWAKE gap.

Any OOB gap between bursts falling in the “shall detect” range shall be recognized as a valid COMWAKE gap.

Any OOB gap between bursts falling in the “shall not detect” ranges shall be recognized as an invalid COMWAKE gap (shall not be recognized as a valid COMWAKE gap).

7.2.2.6.8 COMINIT/COMRESET Gap Detection Windows

Three timing ranges defining the validation and invalidation of COMINIT and COMRESET gaps; see Table 34.

Any OOB gap between bursts falling in the defined “may detect” range may be recognized as a valid COMINIT or COMRESET gap.

Any OOB gap between bursts falling in the “shall detect” range shall be recognized as a valid COMINIT or COMRESET gap.

Any OOB gap between bursts falling in the “shall not detect” ranges shall be recognized as an invalid COMINIT or COMRESET gap (shall not be recognized as a valid COMINIT or COMRESET gap).

7.2.3 Loopback

In addition to meeting all electrical specifications in Table 29 through Table 34, all Hosts and Devices shall provide Far-End Retimed Loopback mode. Two other loopback modes are optional but if implemented shall comply with sections 7.2.3.2 and 7.2.3.3.

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

7.2.3.1 Far-End Retimed

Figure 128 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 section 10.3.9.

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

As a minimum, Far-End Retimed Loopback shall involve far-end circuitry such that the data stream, at the Far-End interface, is extracted by the deserializer and data recovery circuit (DRC) before being sent back through the serializer and transmitter with appropriately inserted retiming $ALIGN_P$ primitives as described in section 7.6. The data may be decoded and descrambled in order to provide testing coverage for those portions of the 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.

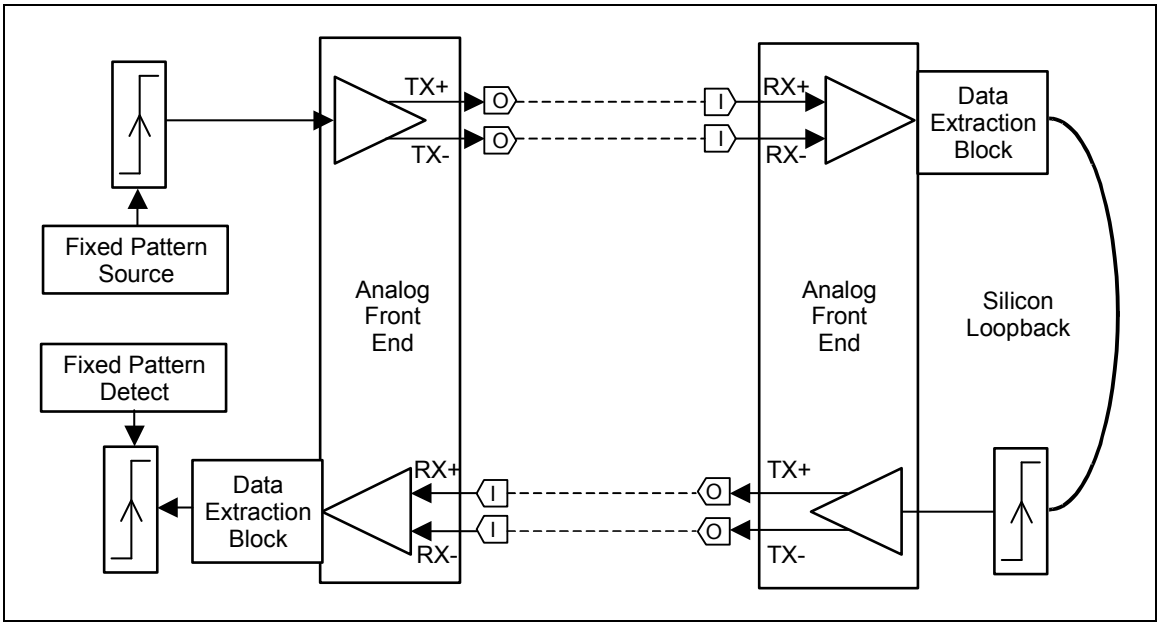


Figure 128 – Far-End Retimed Loopback

7.2.3.2 Far-End Analog (Optional)

Figure 129 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 shall be entered by way of the BIST Activate FIS described in section 10.3.9.

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

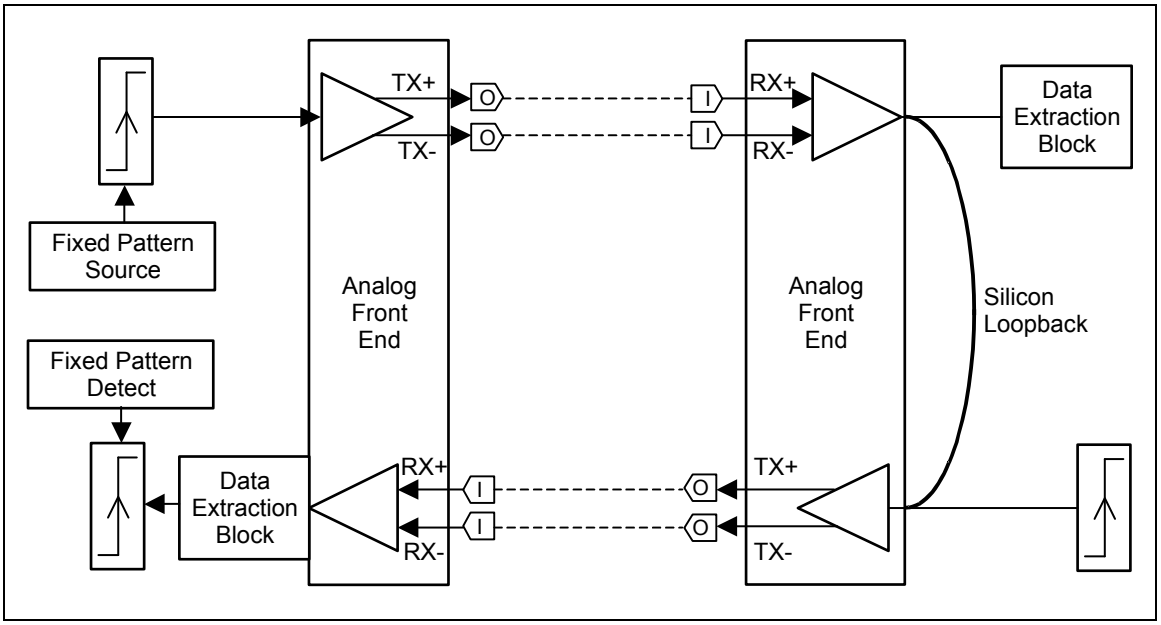


Figure 129 – Far-End Analog Loopback

7.2.3.3 Near-End Analog (Optional)

Figure 130 illustrates the scope, at the architectural block diagram level, of the Near-End Analog Loopback. This loopback scheme needs the far-end connected interface to be in a non-transmitting mode, such as Slumber, or Partial interface power management states. Entry to and exit from this mode is vendor specific.

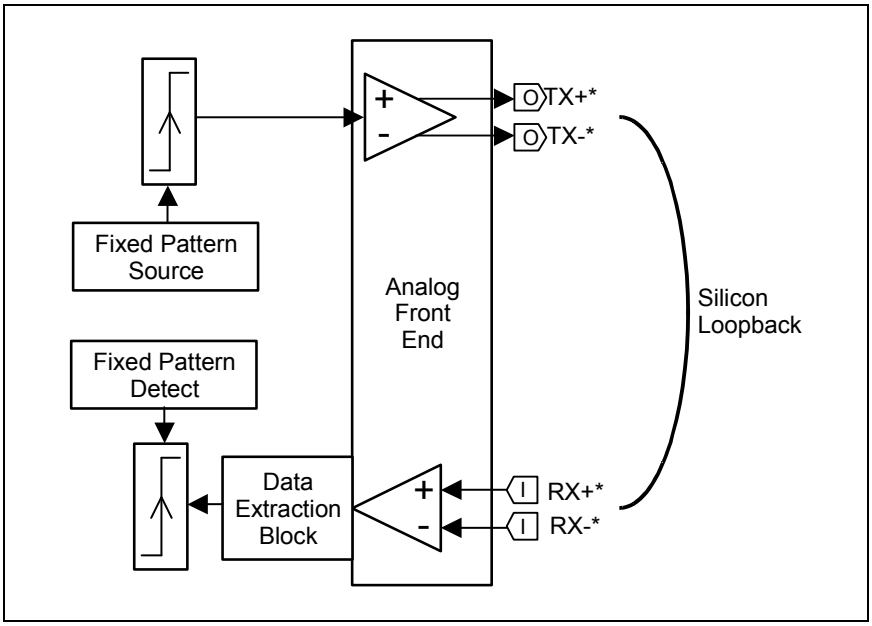


Figure 130 – Near-End Analog Loopback

7.2.4 Test Pattern Requirements

Test patterns shall be used for compliance testing of the Serial ATA interfaces. This section defines various patterns to be used in compliance testing. Individual sections within section 7.4 define which patterns are to be used for specific tests. 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 with the required FIS formats, but are just a repeated selected set of 8b/10b characters.

Compliant patterns are those specified patterns that contain the leading SOF_P primitive, the specified pattern as data content, and trailing CRC_P and EOF_P primitives. There is no suppression of the dual-consecutive $ALIGN_P$ primitive during stimulus with this class of pattern.

Test patterns cited in this section are used as stimulus to verify interface 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.

7.2.4.1 Non-Compliant Patterns

Electrical parameters of section 7.2 shall be verified using the patterns identified in the measurement method section 7.4.

a) Lone Bit Patterns as per section 7.2.4.3.5.
b) High Frequency Test Pattern as per section 4.1.59.
c) Mid Frequency Test Pattern as per section 4.1.78.
d) Low Frequency Test Pattern as per section 4.1.75.

7.2.4.2 Compliant Frame Patterns

The frame error rates specified in section 7.4.1.2 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 Lone Bit Patterns as per section 7.2.4.3.5.
b) Compliant composite patterns as per section 7.2.4.3.6.

Where the qualifying prefix term "compliant" signifies transmission of the cited pattern encapsulated in payload of a Data FIS, and used in a Serial ATA operational transmission context.

Note that the cited patterns should appear on the wire, and the N parameters of the reference patterns shall be extended to achieve the maximum frame length. These compliant patterns contain the necessary SOF_P leading primitive, the Dword header containing the FIS Type indicating a Data FIS, the specified test pattern, the calculated CRC, and the trailing EOF_P, as shown in Figure 131. To generate these patterns on the SATA link, scrambling needs to be taken into account.

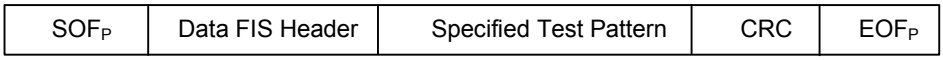


Figure 131 – Compliant Test Patterns

7.2.4.3 Test Bit Patterns and Sequence Characteristics

There are various types of bit sequence patterns that emphasize low/high transition density patterns, as well as low/high frequency patterns.

a) Low transition density patterns (LTDP) 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 (HTDP) are those patterns containing short runs of ones and zeroes, also intended to create inter-symbol interference.

c) Low frequency spectral content patterns (LFSCP) 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 (SSOP) are achieved by transmitting alternating ones 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 lone-bit patterns (LBP) are comprised of the consecutive combination of certain 10b patterns that result in a lone-bit. These patterns create a condition where the preceding 4-bit run-length results in minimum amplitude of the lone-bit as well as its time-width in comparison to its surrounding segments. This is often the worst-case condition that the receiving data recovery circuits may encounter.

f) The intent of random bit patterns is to provide those patterns containing sufficiently broad spectral content, and minimal peaking, that should 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.

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

7.2.4.3.1 Low Transition Density Patterns (LTDP)

Low transition density bit patterns (LTDP), as shown in Table 35 and Table 36 below, contain long runs of ones and zeroes. These patterns create jitter due to inter-symbol interference. This is aggravated when part of the composite pattern described in section 7.2.4.3.6. Bit sequences are shown for both cases, where the starting running disparity is negative or positive.

Table 35 – Low Transition Density Pattern (LTDP) Starting with RD-

Transmission Order →

	D17.7(F1h)-		D30.7(FEh)+			D7.1(27h)+		D14.7(EEh)+			
-	1000	1101	1110	0001	1110	0001	1110	0101	1100	1000	-
	8	D	E	1	E	1	E	5	C	8	
	D30.7(FEh)-		D7.6(C7h)-			D30.3(7Eh)-		D30.3(7Eh)+			
-	0111	1000	0111	1000	0110	0111	1000	1110	0001	1100	-
	7	8	7	8	6	7	8	E	1	C	
	D30.3(7Eh)-		D30.3(7Eh)+			D30.3(7Eh)-		D30.3(7Eh)+			
-	0111	1000	1110	0001	1100	0111	1000	1110	0001	1100	-
	7	8	E	1	C	7	8	E	1	C	
	Above Dword is repeated a total of 2045 times for long version. Above Dword is repeated a total of 125 times for short version.										
	D3.7(E3h)-		D28.7(FCh)+			D3.7(E3h)-		D28.7(FCh)+			
-	1100	0111	1000	1110	0001	1100	0111	1000	1110	0001	-
	C	7	8	E	1	C	7	8	E	1	

Long version total: 3 + 2045 = 2048 Dwords.
Short version total: 3 + 125 = 128 Dwords.

Table 36 – Low Transition Density Pattern (LTDP) starting with RD+

Transmission Order →

+	D14.7(EEh)+		D30.7(FEh)-			D7.6(C7h)+		D17.7(F1h)-			+
	0111	0010	0001	1110	0001	1110	0001	1010	0011	0111	
	7	2	1	E	1	E	1	A	3	7	
+	D30.7(FEh)+		D7.1(27h)+			D30.3(7Eh)+		D30.3(7Eh)-			+
	1000	0111	1000	0111	1001	1000	0111	0001	1110	0011	
	8	7	8	7	9	8	7	1	E	3	
+	D30.3(7Eh)+		D30.3(7Eh)-			D30.3(7Eh)+		D30.3(7Eh)-			+
	1000	0111	0001	1110	0011	1000	0111	0001	1110	0011	
	8	7	1	E	3	8	7	1	E	3	
	Above Dword is repeated a total of 2045 times for long version. Above Dword is repeated a total of 125 times for short version.										
+	D28.7(FCh)+		D3.7(E3h)-			D28.7(FCh)+		D3.7(E3h)-			+
	0011	1000	0111	0001	1110	0011	1000	0111	0001	1110	
	3	8	7	1	E	3	8	7	1	E	

Long version total: 3 + 2045 = 2048 Dwords
Short version total: 3 + 125 = 128 Dwords

7.2.4.3.2 High Transition Density Patterns (HTDP)

High transition density patterns are those patterns containing short runs of ones and zeroes, as shown in Table 37 and Table 38. These patterns create jitter due to inter-symbol interference, becoming more pronounced when part of the composite pattern described in section 7.2.4.3.6. There are two types of high-transition density patterns of interest:

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

Both types are used in the high transition density test pattern. Bit sequences are shown for both cases, where the starting running disparity is negative or positive.

Table 37 – High Transition Density Pattern (HTDP) Starting with RD–

	Transmission Order →										
-	D21.5(B5h)-				D21.5(B5h)-				D21.5(B5h)-				D21.5(B5h)-				-
	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	
	A	A	A	A	A	A	A	A	A	A	
	Above Dword is repeated a total of 512 times for long version. Above Dword is repeated a total of 32 times for short version.										
-	D24.3(78h)-				D24.3(78h)+				D24.3(78h)-				D24.3(78h)+				-
	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	
	C	C	C	C	C	C	C	C	C	C	
	Above Dword is repeated a total of 512 times for long version. Above Dword is repeated a total of 32 times for short version.										
-	D10.2(4Ah)-				D10.2(4Ah)-				D10.2(4Ah)-				D10.2(4Ah)-				-
	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	
	5	5	5	5	5	5	5	5	5	5	
	Above Dword is repeated a total of 512 times for long version. Above Dword is repeated a total of 32 times for short version.										
-	D25.6(D9h)-				D6.1(26h)+				D25.6(D9h)-				D6.1(26h)+				-
	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	
	9	9	9	9	9	9	9	9	9	9	
	Above Dword is repeated a total of 512 times for long version. Above Dword is repeated a total of 32 times for short version.										

Long version total: 4 * 512 = 2048 Dwords
Short version total: 4 * 32 = 128 Dwords

Table 38 – High Transition Density Pattern (HTDP) Starting with RD+

	Transmission Order →										
+	D21.5(B5h)+		D21.5(B5h)+			D21.5(B5h)+		D21.5(B5h)+			+
	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	
	A	A	A	A	A	A	A	A	A	A	
	Above Dword is repeated a total of 512 times for long version. Above Dword is repeated a total of 32 times for short version.										
+	D24.3(78h)+		D24.3(78h)+			D24.3(78h)+		D24.3(78h)+			+
	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	
	3	3	3	3	3	3	3	3	3	3	
	Above Dword is repeated a total of 512 times for long version. Above Dword is repeated a total of 32 times for short version.										
+	D10.2(4Ah)+		D10.2(4Ah)+			D10.2(4Ah)+		D10.2(4Ah)+			+
	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	
	5	5	5	5	5	5	5	5	5	5	
	Above Dword is repeated a total of 512 times for long version. Above Dword is repeated a total of 32 times for short version.										
+	D25.6(D9h)+		D6.1(26h)+			D25.6(D9h)+		D6.1(26h)+			+
	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	
	9	9	9	9	9	9	9	9	9	9	
	Above Dword is repeated a total of 512 times for long version. Above Dword is repeated a total of 32 times for short version.										

Long version total: 4 * 512 = 2048 Dwords
Short version total: 4 * 32 = 128 Dwords

7.2.4.3.3 Low Frequency Spectral Content Pattern (LFSCP)

Bit patterns that contain low frequency spectral components, as shown in Table 39 and Table 40, are a good test of the interconnect transmission, especially any AC coupling capacitors. Poor transmission through these components introduces signal distortion shown by this test pattern. Bit sequences are shown for both cases, where the starting running disparity is negative or positive.

Table 39 – Low Frequency Spectral Content Pattern (LFSCP) Starting with RD–

Transmission Order →

-	D20.2(54h)-		D20.2(54h)-			D20.2(54h)-		D20.2(54h)-			-
	0010	1101	0100	1011	0101	0010	1101	0100	1011	0101	
	2	D	4	B	5	2	D	4	B	5	

Above Dword is repeated a total of 1023 times for long version.
Above Dword is repeated a total of 63 times for short version.

-	D20.2(54h)-		D20.7(F4h)-			D11.5(ABh)+		D11.5(ABh)+			+
	0010	1101	0100	1011	0111	1101	0010	1011	0100	1010	
	2	D	4	B	7	D	2	B	4	A	

+	D11.5(ABh)+		D11.5(ABh)+			D11.5(ABh)+		D11.5(ABh)+			+
	1101	0010	1011	0100	1010	1101	0010	1011	0100	1010	
	D	2	B	4	A	D	2	B	4	A	

Above Dword is repeated a total of 1023 times for long version.
Above Dword is repeated a total of 63 times for short version.

+	D11.5(ABh)+		D11.7(EBh)+			D20.2.(54h)-		D20.2(54h)-			-
	1101	0010	1011	0100	1000	0010	1101	0100	1011	0101	
	D	2	B	4	8	2	D	4	B	5	

Long version total: 2 + 2 * 1023 = 2048 Dwords.
4095 bytes of D11.5, 4095 bytes of D20.2.
1 D11.7 transitional byte including 0000010 run,
1 D20.7 transitional byte including 1111101 run.

Short version total: 2 + 2 * 63 = 128 Dwords
255 bytes of D11.5, 255 bytes of D20.2
1 D11.7 transitional byte including 0000010 run,
1 D20.7 transitional byte including 1111101 run

Table 40 – Low Frequency Spectral Content Pattern (LFSCP) Starting with RD+

Transmission Order →

	D11.5(ABh)+			D11.5(ABh)+		D11.5(ABh)+			D11.5(ABh)+		
+	1101	0010	1011	0100	1010	1101	0010	1011	0100	1010	+
	D	2	B	4	A	D	2	B	4	A	

Above Dword is repeated a total of 1023 times.
Above Dword is repeated a total of 63 times for short version.

	D11.5(ABh)+			D11.7(EBh)+		D20.2.(54h)-			D20.2(54h)-		
+	1101	0010	1011	0100	1000	0010	1101	0100	1011	0101	-
	D	2	B	4	8	2	D	4	B	5	

	D20.2(54h)-			D20.2(54h)-		D20.2(54h)-			D20.2(54h)-		
-	0010	1101	0100	1011	0101	0010	1101	0100	1011	0101	-
	2	D	4	B	5	2	D	4	B	5	

Above Dword is repeated a total of 1023 times.
Above Dword is repeated a total of 63 times for short version.

	D20.2(54h)-			D20.7(F4h)-		D11.5(ABh)+			D11.5(ABh)+		
-	0010	1101	0100	1011	0111	1101	0010	1011	0100	1010	+
	2	D	4	B	7	D	2	B	4	A	

Long version total: 2 + 2 * 1023 = 2048 Dwords
4095 bytes of D11.5, 4095 bytes of D20.2
1 D11.7 transitional byte including 0000010 run,
1 D20.7 transitional byte including 1111101 run

Short version total: 2 + 2 * 63 = 128 Dwords
255 bytes of D11.5, 255 bytes of D20.2

7.2.4.3.4 Simultaneous Switching Outputs Pattern (SSOP)

The simultaneous switching outputs bit pattern (SSOP), shown in Table 41 and Table 42, induces inductive switching (Ldi/dt) noise into substrate supply of a receiver providing a good test of noise control. The SSOP pattern, alternating 1's complement bit patterns (10-bits), are applied to a receiver. Bit sequences are shown for both cases, where the starting running disparity is negative or positive.

Table 41 – Simultaneous Switching Outputs Pattern (SSOP) Starting with RD–

	Transmission Order →										
	D31.3(7Fh)-			D31.3(7Fh)+		D31.3(7Fh)-			D31.3(7Fh)+		
-	1010	1100	1101	0100	1100	1010	1100	1101	0100	1100	-
	A	C	D	4	C	A	C	D	4	C	
	Above Dword is repeated a total of 2048 times for long version. Above Dword is repeated a total of 128 times for short version.										

Long version total: 1 * 2048 = 2048 Dwords.
Short version total: 1 * 128 = 128 Dwords.

Table 42 – Simultaneous Switching Outputs Pattern (SSOP) Starting with RD+

	Transmission Order →										
	D31.3(7Fh)+			D31.3(7Fh)-		D31.3(7Fh)+			D31.3(7Fh)-		
+	0101	0011	0010	1011	0011	0101	0011	0010	1011	0011	+
	5	3	2	B	3	5	3	2	B	3	
	Above Dword is repeated a total of 2048 times for long version. Above Dword is repeated a total of 128 times for short version.										

Long version total: 1 * 2048 = 2048 Dwords.
Short version total: 1 * 128 = 128 Dwords.

7.2.4.3.5 Lone-Bit Pattern (LBP)

The lone-bit patterns, shown in Table 43 and Table 44, are comprised of the combination of adjacent 10B patterns, resulting in a lone one bit prefixed by a run length of four zeros, and suffixed by a run length of three zeros. It also results in a lone zero bit prefixed by a run length of two ones, one zero, two ones, one zero, four ones, and suffixed by a single one. This is a good test of the receiver jitter tolerance under adverse signaling conditions. The lone bit may be attenuated and narrower than expected. Bit sequences are shown for both cases, where the starting running disparity is negative or positive.

Table 43 – Lone-Bit Pattern (LBP) Starting with RD–

Transmission Order →

	D12.0(0Ch)-		D11.4(8Bh)+			D12.0(0Ch)-		D11.3(6Bh)+			
-	0011	0110	1111	0100	0010	0011	0110	1111	0100	0011	+
	3	6	F	4	2	3	6	F	4	3	

	D12.0(0Ch)+		D11.4(8Bh)-			D12.0(0Ch)+		D11.3(6Bh)-			
+	0011	0101	0011	0100	1101	0011	0101	0011	0100	1100	-
	3	5	3	4	D	3	5	3	4	C	

Long version total: 1 * 2048 = 2048 Dwords.
Short version total: 1 * 128 = 128 Dwords.

Table 44 – Lone-Bit Pattern (LBP) Starting with RD+

Transmission Order →

	D12.0(0Ch)+		D11.4(8Bh)-			D12.0(0Ch)+		D11.3(6Bh)-			
+	0011	0101	0011	0100	1101	0011	0101	0011	0100	1100	-
	3	5	3	4	D	3	5	3	4	C	

	D12.0(0Ch)-		D11.4(8Bh)+			D12.0(0Ch)-		D11.3(6Bh)+			
-	0011	0110	1111	0100	0010	0011	0110	1111	0100	0011	+
	3	6	F	4	2	3	6	F	4	3	

Long version total: 1 * 2048 = 2048 Dwords.
Short version total: 1 * 128 = 128 Dwords.

7.2.4.3.6 Composite Pattern (COMP)

For the measurement of jitter, the composite patterns, as shown in Table 45 and Table 46**,** should combine low frequency, low transition density, and high transition density patterns. All these combinations, but the low frequency spectral content class may be performed for relatively short test time intervals, for good jitter 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.

The composite pattern (COMP) 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.

Note that for the sequence that totals only 128 Dwords, the 128-Dword composite pattern is too short to get a sufficient number of continuous repeats for each pattern type.

Table 45 – Composite-Bit Pattern (COMP) Starting with RD–

	Transmission Order →										
-	D31.3(7Fh)-		D31.3(7Fh)+			D31.3(7Fh)-		D31.3(7Fh)+			-
	1010	1100	1101	0100	1100	1010	1100	1101	0100	1100	
	A	C	D	4	C	A	C	D	4	C	
	Above Dword is repeated a total of 256 times for long version. Above Dword is repeated a total of 16 times for short version.										
-	D21.5(B5h)-		D21.5(B5h)-			D21.5(B5h)-		D21.5(B5h)-			-
	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	
	A	A	A	A	A	A	A	A	A	A	
	Above Dword is repeated a total of 64 times for long version. Above Dword is repeated a total of 4 times for short version.										
-	D24.3(78h)-		D24.3(78h)+			D24.3(78h)-		D24.3(78h)+			-
	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	
	C	C	C	C	C	C	C	C	C	C	
	Above Dword is repeated a total of 64 times for long version. Above Dword is repeated a total of 4 times for short version.										
-	D10.2(4Ah)-		D10.2(4Ah)-			D10.2(4Ah)-		D10.2(4Ah)-			-
	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	
	5	5	5	5	5	5	5	5	5	5	
	Above Dword is repeated a total of 64 times for long version. Above Dword is repeated a total of 4 times for short version.										
-	D25.6(D9h)-		D6.1(26h)+			D25.6(D9h)-		D6.1(26h)+			-
	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	
	9	9	9	9	9	9	9	9	9	9	
	Above Dword is repeated a total of 64 times for long version. Above Dword is repeated a total of 4 times for short version.										
-	D17.7(F1h)-		D30.7(FEh)+			D7.1(27h)+		D14.7(EEh)+			-
	1000	1101	1110	0001	1110	0001	1110	0101	1100	1000	
	8	D	E	1	E	1	E	5	C	8	
-	D30.7(FEh)-		D7.6(C7h)-			D30.3(7Eh)-		D30.3(7Eh)+			-
	0111	1000	0111	1000	0110	0111	1000	1110	0001	1100	
	7	8	7	8	6	7	8	E	1	C	

	Transmission Order										
-	D30.3(7Eh)-		D30.3(7Eh)+			D30.3(7Eh)-		D30.3(7Eh)+			-
	0111	1000	1110	0001	1100	0111	1000	1110	0001	1100	
	7	8	E	1	C	7	8	E	1	C	
	Above Dword is repeated a total of 509 times for long version. Above Dword is repeated a total of 29 times for short version.										
-	D3.7(E3h)-		D28.7(FCh)+			D3.7(E3h)-		D28.7(FCh)+			-
	1100	0111	1000	1110	0001	1100	0111	1000	1110	0001	
	C	7	8	E	1	C	7	8	E	1	
-	D12.0(0Ch)-		D11.4(8Bh)+			D12.0(0Ch)-		D11.3(6Bh)+			+
	0011	0110	1111	0100	0010	0011	0110	1111	0100	0011	
	3	6	F	4	2	3	6	F	4	3	
+	D12.0(0Ch)+		D11.4(8Bh)-			D12.0(0Ch)+		D11.3(6Bh)-			-
	0011	0101	0011	0100	1101	0011	0101	0011	0100	1100	
	3	5	3	4	D	3	5	3	4	C	
	Above 2 Dwords are repeated a total of 128 times for long version. Above 2 Dwords are repeated a total of 8 times for short version.										
-	D20.2(54h)-		D20.2(54h)-			D20.2(54h)-		D20.2(54h)-			-
	0010	1101	0100	1011	0101	0010	1101	0100	1011	0101	
	2	D	4	B	5	2	D	4	B	5	
	Above Dword is repeated a total of 255 times for long version. Above Dword is repeated a total of 15 times for short version.										
-	D20.2(54h)-		D20.7(F4h)-			D11.5(ABh)+		D11.5(ABh)+			+
	0010	1101	0100	1011	0111	1101	0010	1011	0100	1010	
	2	D	4	B	7	D	2	B	4	A	
+	D11.5(ABh)+		D11.5(ABh)+			D11.5(ABh)+		D11.5(ABh)+			+
	1101	0010	1011	0100	1010	1101	0010	1011	0100	1010	
	D	2	B	4	A	D	2	B	4	A	
	Above Dword is repeated a total of 255 times for long version. Above Dword is repeated a total of 15 times for short version.										
+	D11.5(ABh)+		D11.7(EBh)+			D20.2.(54h)-		D20.2.(54h)-			-
	1101	0010	1011	0100	1000	0010	1101	0100	1011	0101	
	D	2	B	4	8	2	D	4	B	5	
-	D21.5(B5h)-		D21.5(B5h)-			D21.5(B5h)-		D21.5(B5h)-			-

	Transmission Order →										
	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	
	A	A	A	A	A	A	A	A	A	A	
	Above Dword is repeated a total of 64 times for long version. Above Dword is repeated a total of 4 times for short version.										
-	D24.3(78h)-		D24.3(78h)-		D24.3(78h)-		D24.3(78h)-				-
	1100	1100	1100	1100	1100	1100	1100	1100	1100	1100	
	C	C	C	C	C	C	C	C	C	C	
	Above Dword is repeated a total of 64 times for long version. Above Dword is repeated a total of 4 times for short version.										
-	D10.2(4Ah)-		D10.2(4Ah)-		D10.2(4Ah)-		D10.2(4Ah)-				-
	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	
	5	5	5	5	5	5	5	5	5	5	
	Above Dword is repeated a total of 64 times for long version. Above Dword is repeated a total of 4 times for short version.										
-	D25.6(D9h)-		D6.1(26h)+		D25.6(D9h)-		D6.1(26h)+				-
	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	
	9	9	9	9	9	9	9	9	9	9	
	Above Dword is repeated a total of 64 times for long version. Above Dword is repeated a total of 4 times for short version.										

Long version total: 2048 Dwords total
256DW SSOP
256DW HTDP (64DW, 64DW, 64DW, 64DW)
512DW LTDP (1DW, 1DW, 509DW, 1DW)
256DW LBP ((1DW,1DW) x 128)
512DW LFSCP (255DW, 1DW, 255DW, 1DW)
256DW HTDP (64DW, 64DW, 64DW, 64DW)

Short version total: 128 Dwords total
16DW SSOP
16DW HTDP (4DW, 4DW, 4DW, 4DW)
32DW LTDP (1DW, 1DW, 29DW, 1DW)
16DW LBP ((1DW,1DW) x 8)
32DW LFSCP (15DW, 1DW, 15DW, 1DW)
16DW HTDP (4DW, 4DW, 4DW, 4DW)

Table 46 – Composite-Bit Pattern (COMP) Starting with RD+

Transmission Order →

	D31.3(7Fh)+		D31.3(7Fh)-			D31.3(7Fh)+			D31.3(7Fh)-		
+	0101	0011	0010	1011	0011	0101	0011	0010	1011	0011	+
	5	3	2	B	3	5	3	2	B	3	

Above Dword is repeated a total of 256 times for long version.
Above Dword is repeated a total of 16 times for short version.

	D21.5(B5h)+		D21.5(B5h)+			D21.5(B5h)+			D21.5(B5h)+		
+	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	+
	A	A	A	A	A	A	A	A	A	A	

Above Dword is repeated a total of 64 times for long version.
Above Dword is repeated a total of 4 times for short version.

	D24.3(78h)+		D24.3(78h)+			D24.3(78h)+			D24.3(78h)+		
+	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	+
	3	3	3	3	3	3	3	3	3	3	

Above Dword is repeated a total of 64 times for long version.
Above Dword is repeated a total of 4 times for short version.

	D10.2(4Ah)+		D10.2(4Ah)+			D10.2(4Ah)+			D10.2(4Ah)+		
+	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	+
	5	5	5	5	5	5	5	5	5	5	

Above Dword is repeated a total of 64 times for long version.
Above Dword is repeated a total of 4 times for short version.

	D25.6(D9h)+		D6.1(26h)+			D25.6(D9h)+			D6.1(26h)+		
+	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	+
	9	9	9	9	9	9	9	9	9	9	

Above Dword is repeated a total of 64 times for long version.
Above Dword is repeated a total of 4 times for short version.

	D14.7(EEh)+		D30.7(FEh)-			D7.6(C7h)+			D17.7(F1h)-		+
+	0111	0010	0001	1110	0001	1110	0001	1010	0011	0111	
	7	2	1	E	1	E	1	A	3	7	

	D30.7(FEh)+		D7.1(27h)+			D30.3(7Eh)+			D30.3(7Eh)-		+
+	1000	0111	1000	0111	1001	1000	0111	0001	1110	0011	
	8	7	8	7	9	8	7	1	E	3	

	D30.3(7Eh)+		D30.3(7Eh)-			D30.3(7Eh)+			D30.3(7Eh)-		+
+	1000	0111	0001	1110	0011	1000	0111	0001	1110	0011	
	8	7	1	E	3	8	7	1	E	3	

Above Dword is repeated a total of 509 times for long version.
Above Dword is repeated a total of 29 times for short version.

Transmission Order →

+	D28.7(FCh)+		D3.7(E3h)-			D28.7(FCh)+		D3.7(E3h)-			+
	0011	1000	0111	0001	1110	0011	1000	0111	0001	1110	
	3	8	7	1	E	3	8	7	1	E	

+	D12.0(0Ch)+		D11.4(8Bh)-			D12.0(0Ch)+		D11.3(6Bh)-			-
	0011	0100	0011	0100	1101	0011	0101	0011	0101	1100	
	3	5	3	4	D	3	5	3	4	C	

-	D12.0(0Ch)+		D11.4(8Bh)-			D12.0(0Ch)+		D11.3(6Bh)-			+
	0011	0110	1111	0100	0010	0011	0110	1111	0100	0011	
	3	6	F	4	2	3	6	F	4	3	

Above 2 Dwords are repeated a total of 128 times for long version.
Above 2 Dwords are repeated a total of 8 times for short version.

+	D11.5(ABh)+		D11.5(ABh)+			D11.5(ABh)+		D11.5(ABh)+			+
	1101	0010	1011	0100	1010	1101	0010	1011	0100	1010	
	D	2	B	4	A	D	2	B	4	A	

Above Dword is repeated a total of 255 times for long version.
Above Dword is repeated a total of 15 times for short version.

+	D11.5(ABh)+		D11.7(EBh)+			D20.2.(54h)-		D20.2(54h)-			-
	1101	0010	1011	0100	1000	0010	1101	0100	1011	0101	
	D	2	B	4	8	2	D	4	B	5	

-	D20.2(54h)-		D20.2(54h)-			D20.2(54h)-		D20.2(54h)-			-
	0010	1101	0100	1011	0101	0010	1101	0100	1011	0101	
	2	D	4	B	5	2	D	4	B	5	

Above Dword is repeated a total of 255 times for long version.
Above Dword is repeated a total of 15 times for short version.

-	D20.2(54h)-		D20.7(F4h)-			D11.5(ABh)+		D11.5(ABh)+			+
	0010	1101	0100	1011	0111	1101	0010	1011	0100	1010	
	2	D	4	B	7	D	2	B	4	A	

Transmission Order →

+	D21.5(B5h)+		D21.5(B5h)+		D21.5(B5h)+		D21.5(B5h)+				+
	1010	1010	1010	1010	1010	1010	1010	1010	1010	1010	
	A	A	A	A	A	A	A	A	A	A	

Above Dword is repeated a total of 64 times for long version.
Above Dword is repeated a total of 4 times for short version.

+	D24.3(78h)+		D24.3(78h)+		D24.3(78h)+		D24.3(78h)+				+
	0011	0011	0011	0011	0011	0011	0011	0011	0011	0011	
	3	3	3	3	3	3	3	3	3	3	

Above Dword is repeated a total of 64 times for long version.
Above Dword is repeated a total of 4 times for short version.

+	D10.2(4Ah)+		D10.2(4Ah)+		D10.2(4Ah)+		D10.2(4Ah)+				+
	0101	0101	0101	0101	0101	0101	0101	0101	0101	0101	
	5	5	5	5	5	5	5	5	5	5	

Above Dword is repeated a total of 64 times for long version.
Above Dword is repeated a total of 4 times for short version.

+	D25.6(D9h)+		D6.1(26h)+		D25.6(D9h)+		D6.1(26h)+				+
	1001	1001	1001	1001	1001	1001	1001	1001	1001	1001	
	9	9	9	9	9	9	9	9	9	9	

Above Dword is repeated a total of 64 times for long version.
Above Dword is repeated a total of 4 times for short version.

Long version total: 2048 Dwords total
256DW SSOP
256DW HTDP (64DW, 64DW, 64DW, 64DW)
512DW LTDP (1DW, 1DW, 509DW, 1DW)
256DW LBP ((1DW,1DW) x 128)
512DW LFSCP (255DW, 1DW, 255DW, 1DW)
256DW HTDP (64DW, 64DW, 64DW, 64DW)

Short version total: 128 Dwords total
16DW SSOP
16DW HTDP (4DW, 4DW, 4DW, 4DW)
32DW LTDP (1DW, 1DW, 29DW, 1DW)
16DW LBP ((1DW,1DW) x 8)
32DW LFSCP (15DW, 1DW, 15DW, 1DW)
16DW HTDP (4DW, 4DW, 4DW, 4DW)

Note that only 128 Dwords total for the composite pattern is too short to get a sufficient number of continuous repeats for each pattern type.

7.2.5 Hot Plug Considerations

7.2.5.1 Hot Plug Overview

The purpose of this section is to provide the minimum set of normative requirements necessary for a Serial ATA Host or Device to be declared as “Hot-Plug Capable”. As there exists various Hot-Plug events, there are relevant electrical and operational limitations for each of those types of events. The events are defined below, and the Hot-Plug Capability is further classified into:

a) Surprise Hot-Plug capable
b) OS-Aware Hot-Plug capable

When a Host or Device is declared Hot-Plug Capable without any qualifier, this shall imply that the SATA interface is Surprise Hot-Plug Capable.

For the purposes of this specification, Hot-Plug operations are defined as insertion or removal operations, between SATA hosts and devices, when either side of the interface is powered.

Gen1x / Gen 2x / Gen1m and Gen2m interfaces shall meet the requirements to be classified as Hot-Plug Capable. These requirements are not applicable to Gen1i and Gen2i cabled interfaces, however, Gen1i/Gen2i Devices used in Short Backplane applications shall be Hot-Plug Capable.

Hot-Plug Capable Hosts/Devices shall not suffer any electrical damage, or permanent electrical degradation, and shall resume compliant Tx/Rx operations after the applicable OOB operations, following the Hot-Plug Events.

- **Asynchronous Signal Hot Plug / Removal:** A signal cable is plugged / unplugged at any time. Power to the Host/Device remains on since it is sourced through an alternate mechanism, which is not associated with the signal cable. This applies to External Single-Lane and Multilane Cabled applications.
- **Unpowered OS-Aware Hot Plug / Removal:** This is defined as the insertion / removal of a Device into / from a backplane connector (combined signal and power) that has power shutdown. Prior to removal, the Host is placed into a quiescent state (not defined here) and power is removed from the backplane connector to the Device. After insertion, the backplane is powered; both the Device and Host initialize and then operate normally. The mechanism for powering the backplane on/off and transitioning the Host into/out of the “quiescent” state is not defined here. During OS-Aware events, the Host is powered. This applies to “Short” and “Long” Backplane applications.
- **Powered OS-Aware Hot Plug / Removal:** This is defined as the removal of a Device into / from a backplane connector (combined signal and power) that has power on. After insertion, both the Device and Host initialize and then operate normally. Prior to insertion or removal, the Host is placed into a quiescent state (not defined here) but the backplane connector to the Device is powered at all times. The mechanism for transitioning the Host into/out of the “quiescent” state is not defined here. During OS-Aware events, the Host is powered. This applies to “Short” and “Long” Backplane applications.
- **Surprise Hot-Plug / Removal:** This is defined as the insertion / removal of a Host or Device into / from a backplane connector (combined signal and power) that has power on. After insertion, both the Device and Host initialize and then operate normally. The powered Host or Device is not in a quiescent state.

NOTE: This does not imply transparent resumption of system-level operation since data may be lost, the device may have to be re-discovered and initialized, etc. Regardless of the above definitions, the removal of a device, which is still rotating, is not recommended and should be prevented by the system designer.

7.2.5.2 Electrical Requirements

AC coupling shall be required. Additional hot plug electrical characteristics should include considerations for: common-mode transients, ESD, and drive body discharge.

7.2.5.3 Common-Mode Transients (Informative)

It is a requirement that the Hot-Plug Capable SATA component is designed to handle hot-plug events; this informative section highlights the maximum transient events encountered during hot-plug operations. An example is presented depicting some of the Hot-Plug relevant specifications of Table 29 (Sequencing Transient Voltage and Common Mode Transient Settle Time), where the impact on hosts/devices is shown.

The maximum current induced by a common mode transient is limited by Vcm and the minimum single-ended impedance of 42.5 Ohms. Hence the worst possible surge current would be 2 V / 42.5 Ohms = 47 mA. The duration of this current is limited by the time constant, C * Rtx ~ 0.5 microseconds. This current should be further reduced by supplying common mode termination at the victim end, assuming ESD diodes do not turn on.

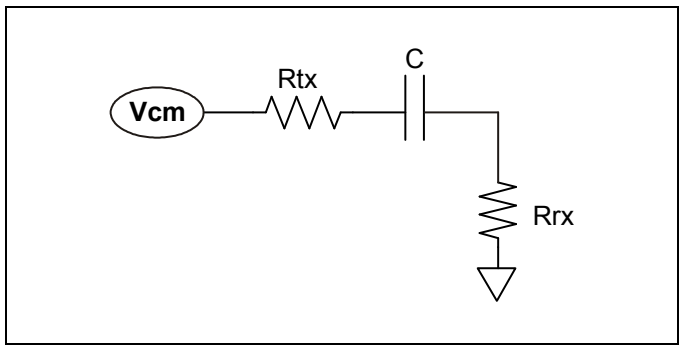


Figure 132 – Example Circuit for Common Mode Transients

0 < Vcm < 2 V
21.25 < Rtx < 40 Ohms
21.25 < Rrx < 40 Ohms
C < 0.024 uF (Two 0.012 uF capacitors in parallel)

The maximum voltage step that may be transmitted to the "victim" end by a transient at the "aggressor" end is the maximum Vcm. This voltage is added to the existing bias voltage at the victim end. Since terminators have no maximum, single-ended limit, this step is not guaranteed to be reduced by any resistive divider. Voltage transients at the "victim" end, however, may be limited by clamping action of ESD diode structures.

7.2.5.4 ESD (Informative)

There is no ESD requirement on the SATA connector interface pins. However, it is recommended that the semiconductors used in the Hot Plug Capable Hosts and Devices meet the following ESD specifications.

Receiver and Transmitter semiconductor signal pins and power pins should tolerate a minimum of 2000 V using test methods per JEDEC EIA-JESD22-A114-B, Electrostatic Discharge (ESD) Sensitivity Testing Human Body Model (HBM).

Receiver and transmitter semiconductor signal pins should tolerate 500 V per JESD22-C101-A, Field-Induced Charged-Device Model Test Method for Electrostatic-Discharge-Withstand Thresholds of Microelectronic Components.

7.2.5.5 Drive Body Discharge (Informative)

For all Serial ATA backplane systems, the device canister or enclosure should provide sufficient electrical bonding such that electrostatic potential is discharged from the device body ground to the enclosure ground prior to the connector mating. It is strongly advised for all Serial ATA backplane systems, and device canisters for hot-plug capable devices, that the guide-rails are designed to be electrically conductive. The device canister should be designed to have an electrical ground connection to device ground, and the guide-rails within the canister system should be connected to system ground.

7.2.6 Mated Connector Pair Definition

The compliance point for receiver and transmitter is at the device/host I/O including the mated connector pair.

Figure 133 shows the mated connector pair detail. The compliance point includes the "tails" of the receptacle pins. The physical description of the receptacle pin tails is shown in Figure 134.

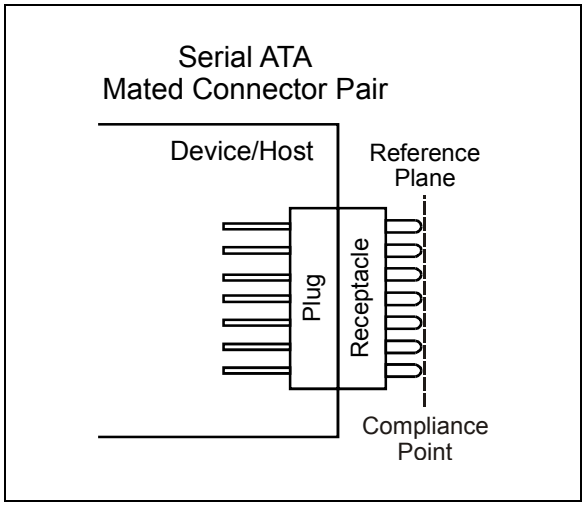


Figure 133 – Mated Connector Pair

The signal interface to the mated pair connector pin tails should be done with care to minimize parasitic capacitance or inductance. The connector pin, tails are a coplanar waveguide transmission line in a ground, signal, signal, ground (GSSG) configuration. The signal interface to the pin, tails should maintain the GSSG configuration.

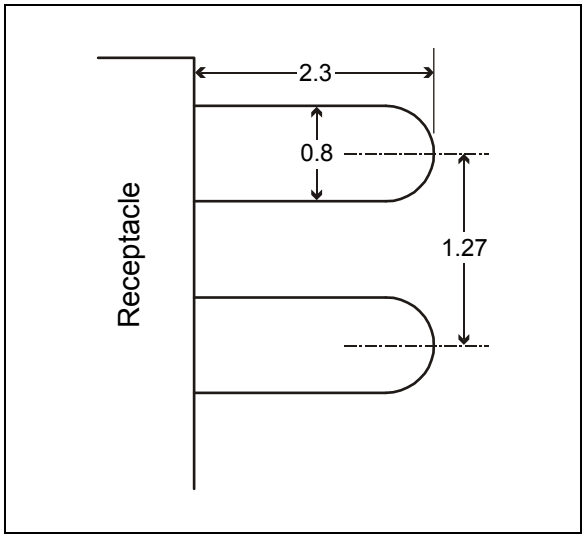


Figure 134 – Mated Connector Pair, Pin Tail Detail

7.2.7 Compliance Interconnect Channels (Gen1x, Gen2x, Gen3i)

Compliance Interconnect Channels are defined as a set of calibrated physical test circuits applied to the Transmitter mated connector, intended to be representative of the highest-loss interconnects.

The Compliance Interconnect Channel (CIC) is used to verify that the signal electrical characteristics at the Transmitter mated connector are sufficient to ensure compliance to the input electrical specifications for Gen1x, Gen2x and Gen3i receivers as delivered through worst-case media. The magnitude of this worst-case loss as a function of frequency is defined mathematically as a Transmitter Compliance Transfer Function (TCTF). There is a Gen3i TCTF, Gen2x TCTF and a Gen1x TCTF. Any linear, passive, differential two-port (e.g., a SATA cable) with loss greater than the TCTF at all frequencies and which meets the ISI loss constraint (defined below) is defined to be a CIC. (See also section 7.2.7.1.1.)

A combination of a zero-length test load (i.e., the Laboratory Load) plus the applicable CIC (Gen1x/Gen2x/Gen3i) is used for the specification of the host-controller or device transmitter characteristics.

A Gen1x/Gen2x/Gen3i transmitter signal is specified by:

1. Meeting all parameters in Table 31[1] for Gen1x, Gen2x or Gen3i when transmitting into a Laboratory Load.
2. Meeting Table 31 input swing (V_{diffTX}) and jitter (TJ after CIC and DJ after CIC) requirements for Gen1x or Gen2x when transmitting through the appropriate Gen1x or Gen2x CIC into a Laboratory Load while using the same transmitter settings (emphasis, amplitude, etc.) as in the first test.[2] (see sections 7.4.5 and 7.4.9)
3. Meeting Table 31 input swing (VdiffTx) and total jitter (TJ after CIC) requirements for Gen3i when transmitting through the appropriate Gen3i CIC into a Laboratory Load while using the same transmitter settings (emphasis, amplitude, etc.) as in the first test.

The transmission magnitude response, $|S_{21}|$, of the Gen3i TCTF satisfies the following two inequalities[3]:

$$|S_{21}| \leq -20 \log_{10}(e) \{[3.0 \times 10^{-6} (f^{0.5})] + [1.1 \times 10^{-10} (f)]\} \text{ dB}$$

for 50 MHz < f < 9.0 GHz, (f expressed in Hz),

$$|S_{21}| \text{ at } 600 \text{ MHz} - |S_{21}| \text{ at } 3000 \text{ MHz} > 2.7 \text{ dB}$$

---

[1] Note that the Transmitter Compliance Specifications are defined and measured into a Laboratory Load. Received signal attenuation or amplification due to actual receiver terminator tolerance as well as additional received signal ISI due to the actual receiver return loss may further degrade the actual receiver's input signal. Transmitter Compliance Specifications are expected to be only slightly tighter than Receiver Specifications.

[2] While not permitted in this specification, this second requirement can be approximated by *mathematically* processing through a TCTF the signal captured by the HBWS using only the Laboratory Load in the first requirement.

[3] Please note that "e" in the first expression is the base of the natural logarithms, approximately 2.71828. Hence, the first factor, 20 $\log_{10}(e)$, evaluates to approximately 8.6859. This value is the conversion factor from nepers (defined as the natural logarithm of a power ratio) to decibels.

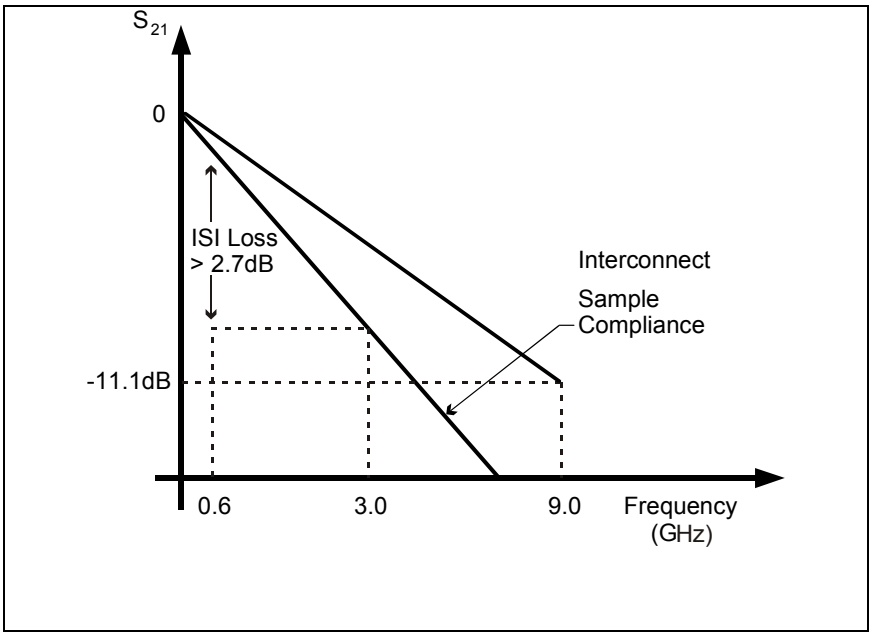


Figure 135 – Compliance Channel Loss for Gen3i

The second constraint, termed ISI loss, may be motivated as follows: $|S_{21}|$ at one tenth the data rate is the attenuation of the fundamental component of a repeating five-ones-five-zeroes pattern, the longest possible run lengths in 8b/10b encoded data. Similarly, $|S_{21}|$ at one half the data rate is the attenuation of the fundamental component of a repeating …010101… pattern, the shortest possible run lengths in 8b/10b encoded data. Hence, for an output waveform of this TCTF, ISI loss approximates the ratio between a) the peak-peak voltage (established by the long run lengths) and b) the inside vertical eye opening (established by the high frequency pattern). Any TCTF with a flatter loss characteristic (i.e., with more broadband attenuation) would generate less inter-symbol interference (ISI) and therefore less output jitter. This constraint prohibits such a TCTF.

The transmission magnitude response, |S21|, of the Gen2x TCTF satisfies the following three inequalities[4]:

$$\mathbf{|S_{21}| \leq -20 \log_{10}(e) \{[1.7 \times 10^{-5} (f^{0.5})] + [1.0 \times 10^{-10} (f)]\}\ dB}$$

for 50 MHz < f < 3.0 GHz, (f expressed in Hz),

$$\mathbf{|S_{21}| \leq -10.7\ dB}$$

for 3.0 GHz < f < 5.0 GHz, and

$$\mathbf{|S_{21}| \text{ at } 300\ MHz - |S_{21}| \text{ at } 1500\ MHz > 3.9\ dB}$$

[4] Please note that “e” in the first expression is the base of the natural logarithms, approximately 2.71828. Hence, the first factor, 20 $\log_{10}(e)$, evaluates to approximately 8.6859. This value is the conversion factor from nepers (defined as the natural logarithm of a power ratio) to decibels.

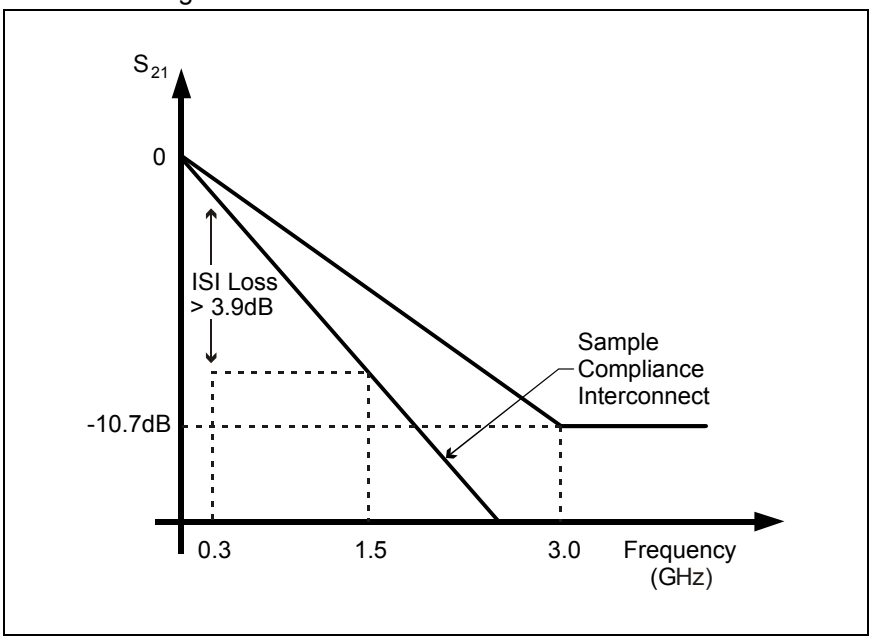


Figure 136 – Compliance Channel Loss for Gen2x

The third constraint, termed ISI loss, may be motivated as follows: $|S_{21}|$ at one tenth the data rate is the attenuation of the fundamental component of a repeating five-ones-five-zeroes pattern, the longest possible run lengths in 8b/10b encoded data. Similarly, $|S_{21}|$ at one half the data rate is the attenuation of the fundamental component of a repeating …010101… pattern, the shortest possible run lengths in 8b/10b encoded data. Hence, for an output waveform of this TCTF, ISI loss approximates the ratio between a) the peak-peak voltage (established by the long run lengths) and b) the inside vertical eye opening (established by the high frequency pattern). Any TCTF with a flatter loss characteristic (i.e., with more broadband attenuation) would generate less inter-symbol interference (ISI) and therefore less output jitter. This constraint prohibits such a TCTF.

The transmission magnitude response, $|S21|$, of the Gen1x TCTF satisfies the following three inequalities:

$| S_{21} | \leq -20 \log_{10} (e) \{[1.7 \times 10^{-5} (f^{0.5})] + [1.0 \times 10^{-10} (f)] \}$ dB
for 50 MHz < f < 1.5 GHz, (f expressed in Hz),

$| S_{21} | \leq -7.0$ dB
for 1.5 GHz < f < 5.0 GHz, and

$| S_{21} |$ at 150 MHz - $| S_{21} |$ at 750 MHz > 2.0 dB

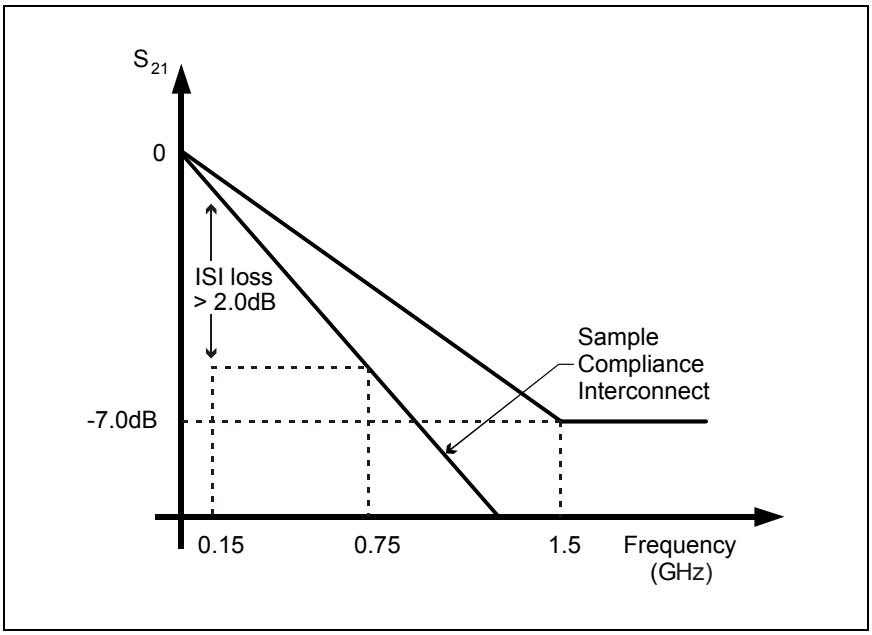


Figure 137 – Compliance Channel Loss for Gen1x

7.2.7.1.1 Calibration of Compliance Interconnect Channels

The TCTF defines the worst-case loss exclusive of the two SATA connectors in the path from transmitter to receiver. However, the loss due to these two connectors shall be included in the transmitter characterization. That is, the transmitter shall be tested with the TCTF-defined loss plus two mated SATA connector pairs.

For a CIC implemented with SMA connectors, it is seen in Figure 145 that the addition of a SATA adapter (plug) following the CIC and driving into a Laboratory Load provides the required total loss of TCTF (embodied in the CIC loss) plus the loss of two SATA connectors.

7.2.8 Impedance Calibration (Optional)

Hosts and devices may employ on-chip adaptive impedance matching circuits to ensure best possible termination for both its transmitter and receiver.

The host, 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 controller should utilize a separate reference to perform calibration. In a desktop system, 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 controller should make use of a separate reference (such as an accurate off-chip resistor) for the calibration phase.

The device, on the other hand, may assume that the termination on the far side (host side) of the cable is fully calibrated, and may make use of this as the reference. Using the host termination

as the calibration reference allows the devices operating in both the desktop and the mobile system environment to use the same hardware.

Signals generated for the impedance calibration process shall not duplicate the OOB signals, COMWAKE, COMINIT, or COMRESET. Signals generated for the impedance calibration process shall not exceed the normal operating voltage levels, cited in section 7.2. See the power management section for suggested times to perform calibration during power-on.

7.3 Jitter

Jitter is the short-term variations of the zero crossings from ideal positions in time. A "Reference Clock" (defined in section 7.3.2) defines the ideal positions in time. The Reference Clock method provides for the separation of jitter from SSC, tracking SSC and other low frequency modulation but not jitter. There are several types of jitter separated into two classes: deterministic and random. Deterministic jitter is bounded and random jitter is not. The amount of tolerable jitter is limited by the desired bit error rate performance of the channel. Two classes of jitter are used in analysis because they accumulate differently.

The Serial ATA data stream employs an embedded clock; no clock signal is separately sent. At the receiver, the Serial ATA data stream is re-clocked to form a parallel digital signal. Adequate timing margin is required for this process to function properly. Jitter analysis is the timing analysis used in systems with an embedded clock.

In SATA systems, random jitter is a significant portion of the total jitter causing occasional errors to occur. When a bit error occurs, the error is detected when an entire frame of bits is received. The bit error is corrected by retransmitting the frame. If two bit errors occur within a single frame, the corrective action is the same. The data throughput on the channel is diminished when frames are retransmitted. Frame Error Rate is the channel performance measure.

Since a portion of the jitter is random, a measurement of jitter also has a random nature. That is, repeated measurements yield results that are somewhat different. As the sample size of each measurement increases, the spread of the measurement results decreases. A measured value of random jitter is determined to a known confidence level.

A frame error rate test is a system performance test done on a combination of SATA compliant components. To achieve a statistically significant estimate of the frame error rate a large sample size is necessary. A frame error rate test on a SATA channel is lengthy requiring about an hour at Gen2 rates.

Jitter tests are compliance tests done on an individual SATA component, a device, host, or interconnect to ensure system performance. Compliance tests help to predict the performance of combinations of compliant components. It is often desirable to make jitter measurements in a short period of time rather than hours. Consequently, jitter measurements are done with small sample sizes and the results are extrapolated to predict results with larger sample size.

Extrapolation of results from small sample size to large sample size involves assumptions. This specification defines two assumptions as normative. First, the random jitter has a Gaussian distribution. Second, the total jitter is the sum of the deterministic jitter plus 14 times the standard deviation of the random jitter. These allow the separation of deterministic from random jitter, and an estimate of the total jitter for an equivalent BER of 10^{-12} from a much smaller sample size.

7.3.1 Jitter Definition

Jitter is defined as the difference in time between a data transition and the associated Reference Clock event. The jitter at the receiver is the result of the aggregate jitter in the transmission path. First, jitter is generated during clocking of the data in the transmitter. Then, each element in the channel between the transmitter and the receiver influences the jitter. Finally, the receiver shall be able to recover the data despite the jitter, otherwise errors occur. The receiver jitter tolerance shall be greater than the transmitter's generated jitter and the expected jitter accumulation through the channel.

Jitter budgets are dependent on the desired bit error rate (BER). SATA assumes a BER target of less than 10^{-12}. Jitter levels are defined as Reference Clock to data. The Reference Clock is extracted from a serial data stream using either a PLL (hardware) or a clock recovery algorithm (software).

The Reference Clock to data jitter methodology allows for jitter measurements to be made on a device or host using a Spread Spectrum Clock or a non-spreading clock.

7.3.2 Reference Clock Definition

The Reference Clock is defined as that clock recovered from a Serial ATA data stream. The Reference Clock provides the distinction between Spread Spectrum Clocking (SSC) and jitter. The Reference Clock tracks SSC and wander, but not jitter. In addition, it provides a definition for determining the SSC profile. Reference Clock extraction is performed using either hardware or software PLLs

7.3.2.1 Gen1i, Gen1m, Gen2i and Gen2m Normative Requirements

For Gen1i, Gen1m, Gen2i and Gen2m, the Reference Clock characteristics are controlled by the resulting JTF (Jitter Transfer Function) characteristics obtained by taking the time difference between the Type 2 PLL output (the Reference Clock) and the data stream sourced to the PLL. The PLL CLTF -3 dB corner frequency, and other adjustable CLTF parameters such a peaking, are determined by the value required to meet the requirements of the JTF. (See section 7.4.7 for JTF information)

The JTF for Gen1i and Gen1m shall have the following characteristics for an encoded Gen1 D24.3 pattern (1100110011 0011001100). This is the Gen1 MFTP which is a test pattern that has clock-like characteristics and a transition density of 0.5.

The JTF for Gen2i and Gen2m shall have the following characteristics for an encoded Gen2 D24.3 pattern (1100110011 0011001100). This is the Gen2 MFTP which is a test pattern that has clock-like characteristics and a transition density of 0.5.

1) 1) The -3 dB corner frequency of the JTF shall be as shown in Table 31 Jitter Transfer Function Bandwidth (D24.3, high pass -3dB).
2) The magnitude peaking of the JTF shall be as shown in Table 31 Jitter Transfer Function Peaking.
3) The attenuation at Jitter Transfer Function Low Frequency Attenuation Measurement Frequency in Table 31 shall be as shown in Table 31 Jitter Transfer Function Low Frequency Attenuation.

The JTF -3dB corner frequency and the magnitude peaking requirements shall be measured with sinusoidal PJ applied, with a peak-to-peak amplitude of 0.3 UI +/- 10%. The attenuation at 30 KHz shall be measured with sinusoidal phase (time) modulation applied, with a peak-to-peak amplitude of 20.8 ns +/- 10%.

7.3.2.2 Gen1i, Gen1m, Gen2i, and Gen2m Informative Comments

Typically a CLTF -3 dB corner frequency of f_{BAUD} /500 could provide a JTF with characteristics close to the requirements, but due to differences in Type 2 PLL designs, the actual CLTF settings required to meet the required JTF can vary widely.

It is desired that the phase response of the JTF of a JMD (Reported Jitter / Applied Jitter) be that of the JTF of the time difference of the output of a Type 2 PLL to the Data stream applied to the PLL. This is the reference design. In the presence of multiple jitter component frequencies, the relative phase at these frequencies determines how they are combined to construct the final reported jitter value. In the case of discrepancies between the reported jitter levels, between JMDs with the same JFT magnitude response, the JMD with the JTF phase characteristics closest to that of the reference design, is to be considered correct. The JTF phase response of a JMD is important, but it is not always possible to determine this without proprietary information concerning the JMD processing methods, and it is not externally observable in some classes of JMDs.

The JTF of the time difference of the output of a Type 2 PLL to the Data stream applied to the PLL, or the reference design, is defined with a pattern that has a transition density of 0.5. Since this Type 2 PLL contains a sampled data mode phase detector, with a gain that varies proportionally with transition density, the JTF -3 dB corner frequency will change with the transition density of the applied pattern. For a well designed PLL, with significant phase margin in the open loop response, the JTF -3 dB corner frequency, with shift proportionally with the change of pattern transition density. For example, the 2.1 MHz JTF -3dB corner frequency, set with a pattern with a transition density of 0.5, will shift to 4.2 MHz when a pattern with a transition

density of 1.0, such as the D10.2 pattern, is applied. A proportional decrease of the JTF -3dB corner frequency will also be observed for a decrease in pattern transition density compared to a 0.5 transition density. This is the expected JMD response to changes in pattern transition density as the reference design would exhibit. If a JMD shifts the JTF -3dB corner frequency in a manner that does not match this characteristic, or does not shift at all, measurements of jitter with patterns with transition densities significantly different than 0.5 may lead to discrepancies in reported jitter levels. In the case of reported jitter discrepancies between JMDs, the JMD with the shift of the -3dB corner frequency, closest to the proportional characteristic of the reference design, it to be considered correct. This characteristic may be measured using the conditions defined above for measuring the -3dB corner frequency, using multiple patterns with different transition densities.

7.3.2.3 Gen1x and Gen2x Normative Requirements

For Gen1x and Gen2x, the Reference Clock PLL is defined as type 2 PLL with a -3 dB corner frequency $f_{c3dB} = f_{BAUD}$ /1667 given a transition density of 1.0 (corresponding to a 1010101010 clock-like pattern) and damping factor ξ = 0.707 min to 1.00 max.

7.3.2.4 Gen3i Normative Requirements

For Gen3i the Reference Clock characteristics are controlled by the resulting JTF (Jitter Transfer Function) characteristics obtained by taking the time difference between the Type 2 PLL output (the Reference Clock) and the data stream sourced to the PLL. The PLL CLTF -3 dB corner frequency, and other adjustable CLTF parameters such a peaking, are determined by the value required to meet the requirements of the JTF. (See section 7.4.8 for JTF information)

The JTF for Gen3i shall have the following characteristics for an encoded Gen3 D24.3 pattern (1100110011 0011001100). This is the Gen3 MFTP, which is a test pattern that has clock-like characteristics and a transition density of 0.5.

1) The -3 dB corner frequency of the JTF shall be 4.2 MHz +/- 2 MHz. as shown in Table 31 Jitter Transfer Function Bandwidth (D24.3, high pass -3dB)(Gen3).
2) The magnitude peaking of the JTF shall be 3.5 dB maximum. as showna in Table 31 Jitter Transfer Function Peaking (Gen3).
3) The attenuation at 420 KHz +/- 1% Jitter Transfer Function Low Frequency Attenuation Measurement Frequency (Gen3) in Table 31 shall be 38.2 dB +/- 3 dB. as shown in Table 31 Jitter Transfer Function Low Frequency Attenuation (Gen3).

The JTF -3dB corner frequency and the magnitude peaking requirements shall be measured with sinusoidal PJ applied, with peak-to-peak amplitude of 0.3 UI +/- 10%. The attenuation at 420 KHz shall be measured with sinusoidal phase (time) modulation applied, with peak-to-peak amplitude of 1.0 ns +/- 10%. The attenuation is measured on the 40 dB/Dec slope of the JTF at 1/10 the -3 dB corner frequency nominal target value. This is equivalent to 72 dB at 60 KHz for an ideal 40 dB/Dec slope that corresponds to a 2X increase of JTF BW for Gen3i compared to Gen2i. This shift in measurement point allows for improved practical measurements and lower test signal phase modulation level requirements.

7.3.3 Spread Spectrum Clocking

Serial ATA allows the use of spread spectrum clocking, or intentional low frequency modulation of the transmitter clock. The purpose of this modulation is to spread the spectral energy to mitigate the unintentional interference to radio services. The modulation frequency of SSC shall be in the range defined for f_{SSC} in Table 29.

The modulation frequency deviation shall be in the prescribed range for SSC_{tol} in Table 29. The instantaneous frequency (each period) of the Reference Clock shall fall within the prescribed T_{UI} range. If the rate of change of the instantaneous frequency is excessive jitter is increased.

The SSC modulation only moves the frequency below the nominal frequency. This technique is often called “down-spreading”.

7.3.3.1 Example SSC Profile (Informative)

An example triangular frequency modulation profile is shown in Figure 138. The modulation profile in a modulation period is expressed as:

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

where f_{nom} is the nominal frequency in the non-SSC mode, f_m is the modulation frequency, δ is the modulation amount, and t is time.

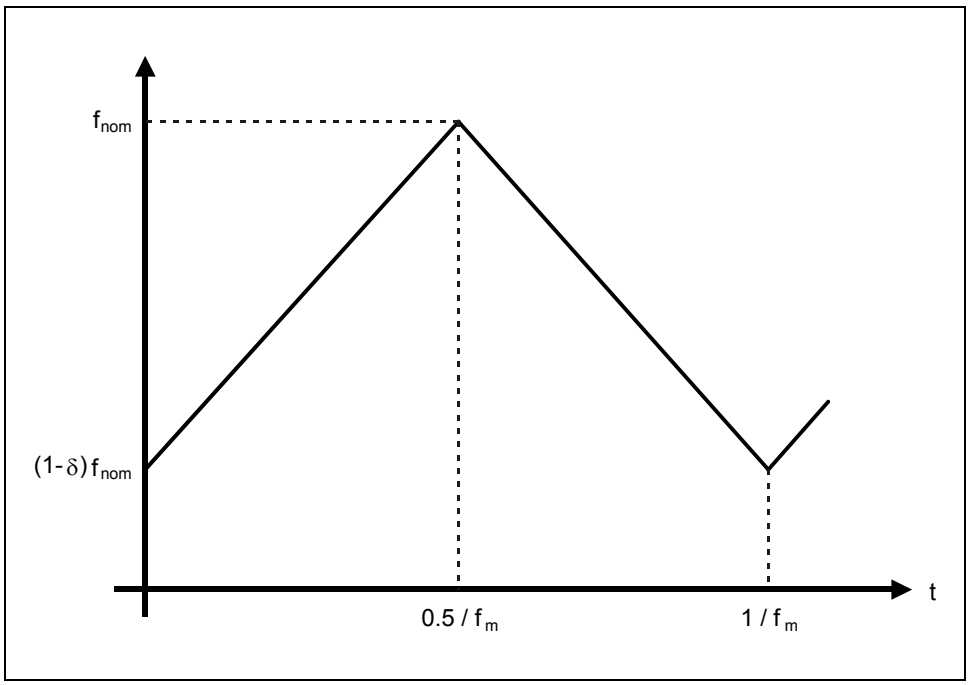


Figure 138 – SSC Profile Example: Triangular

As an example, for triangular modulation, the absolute spread amount at the fundamental frequency is shown in Figure 139, as the width of its spectral distribution.

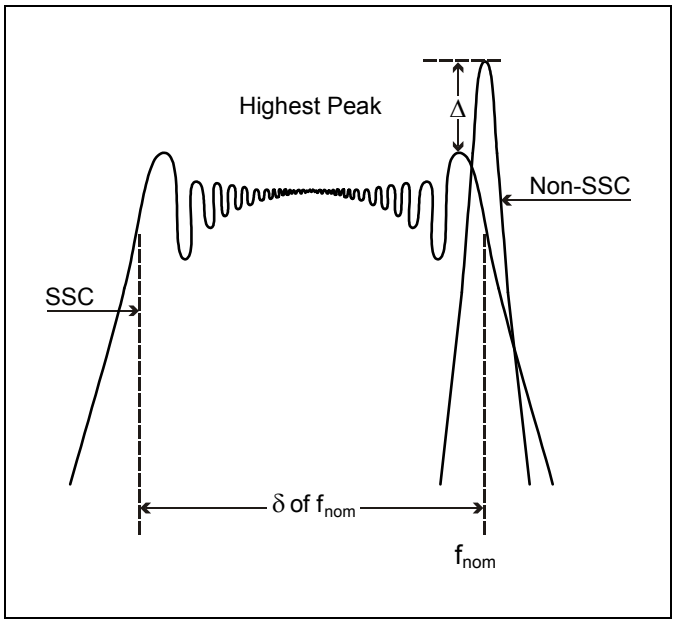


Figure 139 – Spectral Fundamental Frequency Comparison

7.3.4 Jitter Budget

There are two types of jitter, random jitter (RJ) and deterministic jitter (DJ). Random jitter is Gaussian and unbounded. For ease, the standard deviation (RJ_σ) is multiplied by a factor, which corresponds to the target BER. For a target BER = 10^{-12} , the associated multiplication factor for Serial ATA is 14.

Total jitter (TJ) is peak-to-peak and defined as:

$$TJ = (14 * RJ_\sigma) + DJ$$

Table 31 and Table 33 show the compliance jitter values. The measurement of jitter is described in section 7.4.8.

7.4 Measurements

The performance of a SATA system with host and device connected together is measured by the frame error rate, using a set of reference frames, defined by a specific set of ordered test patterns within the frame. A host or device is commanded to generate the various test patterns through the use of the BIST Activate FIS or other vendor unique commands to the device under test.

Measurements of devices and hosts are done to determine compliance with this specification. Compliance tests are done with a device or host connected to test equipment. Compliance tests are not done with devices and hosts connected together. Unless otherwise specified, all compliance measurements shall be taken through the mated connector pair.

The values specified in Table 31 refer to the output signal from the unit under test at the mated connector into a Laboratory Load. The signals are not specified when attached to a system cable or backplane.

The values specified in Table 33 refer to the input signal from any signal source as measured at the unit under test using a Laboratory Load.

The components that make up the system each affect the system's performance, not simply by summing up the low-level parameters. There are many interactions as well as protocol effects such as the retry algorithms.

Fundamental to this specification is a clear definition of the UUT (unit under test). The UUT consists of the host/device and the receptacle side of the mated pair connector. This places the signals at a point in the test measurement setup where all specifications of the UUT are defined at an impedance level of 50 Ohms each signal line to ground which is 100 Ohms differential and 25 Ohms common mode impedance. This is the compliance point for hosts and devices.

The Serial ATA Phy layer compliance shall be tested using the Parametric Method. This method uses repetitive patterns and a Laboratory Load to allow accurate, repeatable measurements to be performed on the unit under test.

Additional measurement methods for several parameters are aimed at providing quick No-Go testing. These use any valid data pattern and a Laboratory Load to quickly produce a visual "picture" of the performance of the unit under test. For example, one measurement method uses Data Eyes to quickly understand jitter. Another uses a mode measurement for identifying a potentially complex signal with a single amplitude value. However, none of these No-Go measurement methods may be used for testing compliance to electrical specifications. These measurement methods are valuable for gaining useful information about the performance of the unit under test, which goes beyond specification compliance issues.

Both methods produce measurements of electrical performance, however, the parametric method shall be used for validation of the unit under test to the Serial ATA requirements in section 7.2 while other methods may be used as general No-Go tests.

7.4.1 Frame Error Rate Testing

Frame error rate is the measure of link performance, a system level test. Since bit errors are ignored except during frames, frame error rate testing is used as the method of measuring channel performance during system operation.

Serial ATA error detection at the frame level uses the CRC (Cyclic Redundancy Check) error detection mechanism, and respective reporting to the higher layer levels. Since all frames include a header and CRC field, the calculation includes these overhead bytes in the Frame Error specification.

The bit error rate is a measurement of raw channel performance and is closely related to the Phy parameters. The 8b/10b encoding and SATA protocol complicate the measurement of bit error rate. A single bit 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. 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. A missing or an extra bit detected by the receiver translates into a series of errors that spans across multiple byte-boundaries until bit re-alignment via $ALIGN_P$ primitives.

Under the following condition, if BER = 0 then FER = 0 as well. BER = 0 therefore implies that FER = 0, and any product that achieves a receiver performance with zero BER, at the required confidence level, satisfies the FER requirement as well. The opposite is not necessarily the case

7.4.1.1 Frame Error Rate Patterns

Frame Error Rate patterns contain the elements of the Bit Error Rate test bit patterns and sequence of patterns, so as to thoroughly stress the serial interface in the system, while using the higher level CRC error detection and reporting from the lower protocol level layers to the Application layer.

The frame patterns shall be comprised of the set of the Composite Patterns, cited in section 7.2.4.3.6, but with the parameters extended so as to achieve the maximum frame length.

Note that the compliant patterns shown are the patterns that are expected on the wire. When sent using a normal FIS payload mechanism the data within a FIS is scrambled. For the correct patterns to appear on the wire "pre-scrambling" needs to be performed so that the specified patterns appear on the link after payload scrambling is performed by the Transport layer.

7.4.1.2 Frame Error Rate Measurements

The Frame Error Rate (FER) shall be measured and computed to be no greater than $8.200*10^{-8}$ at a 95% confidence level when tested with any given 8b/10b pattern, including the Frame Error Rate reference patterns cited in section 7.4.1.1. The Serial ATA CRC error detection mechanism is used to measure FER.

The Frame Error Rate is calculated based on the maximum size of a Data FIS, plus overhead for the FIS header and CRC Dwords. The Frame Error Rate assumes a target bit error rate of 10^{-12}.

$$FER = (8192 + 8) \times 10 \times 10^{-12} = 8.200 \times 10^{-8}$$

Note that the cited patterns should appear on the wire, and the parameters of the reference patterns shall be extended to achieve the maximum frame length of 8192 user payload bytes.

7.4.1.3 Amount of Data to Transfer to Achieve Target Confidence Level (Informative)

As this is a statistical process, there is a confidence level associated with each measurement that is related to the number of frames transferred. For example, one should only declare that the interface Frame-Error-Rate performance has been achieved with a confidence level for that given sample size, and Error-thresholds as shown in Table 47.

Table 47 – Frame Error Rate Confidence Levels Versus Sample Size

Sample Size (Frames)	Number of Frame-Errors										
	0	1	2	3	4	5	6	7	8	9	10
$1.22*10^{7}$	63.21%	26.42%	8.03%	1.90%	0.37%	0.06%	0.01%	<0.01%	<0.01%	<0.01%	<0.01%
$1.22*10^{8}$	>99.99%	99.95%	99.72%	98.97%	97.07%	93.29%	86.99%	77.98%	66.72%	54.21%	41.70%

The sample size is taken as ten times the total number of frames transmitted for a given error rate.

$$\frac{1}{8.2 \times 10^{-8}} \times 10 = 1.22 \times 10^{8}$$

When a test is conducted where $1.22*10^{8}$ frames are passed, the maximum number of frame errors to declare a Frame Error Rate of $8.200*10^{-8}$ with confidence level of >95% is four.

7.4.1.4 Bit Error Rate Testing (Informative)

There are two basic classes of errors that affect the bit-error rate performance: bit-errors, and burst-errors.

In order to get a fair assessment of bit-error-rate performance, bit-errors, as well as burst errors, are considered separately. This is because a missing or an extra bit detected by the receiver translates into a series of errors that spans across multiple byte boundaries until re-alignment via an alignment sequence. This series of errors are defined as burst errors.

Another type of byte-wise error exists when an entire byte is not received. As viewed by the higher-level protocol it appears as a loss of word synchronization. It causes a burst error whose span may be limited by higher-layer protocol transmission conventions at the next alignment sequence.

Any of these errors may result in several related errors occurring closely together which in turn may result in multiple apparent bit-error events. For example, 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.

All of these events eventually are recognized during the decoding process and result in a frame error.

NOTE: Burst Error Rate measurements shall not be used for Compliance testing.

7.4.1.4.1 Bit Error Rate Measurements

The Bit Error Rate, if measured and computed, byte-wise, should be no greater than 10^{-12} bit-errors when tested with the reference test patterns, cited in section 7.4.1.1.

Frame Error Rate measurements constitute the basis for the applicable test requirements for this specification. See 7.4.1 for an explanation of the relationship between FER and BER.

7.4.1.4.2 Amount of Data to Transfer to Achieve Target Error Rate

As this is a statistical process, there are confidence levels associated to each sample size. For example, one should only declare that the interface Bit Error Rate performance has been achieved with a confidence level of 95% for that given sample size, and error thresholds as shown in Table 48.

Table 48 – Bit Error Rate Confidence Levels Versus Sample Size

Sample Size (Bits)	Number of Bit-Error Events - Threshold										
	0	1	2	3	4	5	6	7	8	9	10
$1.00*10^{12}$	63.21%	26.42%	8.03%	1.90%	0.37%	0.06%	0.01%	<0.01%	<0.01%	<0.01%	<0.01%
$1.00*10^{13}$	>99.99%	99.95%	99.72%	98.97%	97.07%	93.29%	86.99%	77.98%	66.72%	54.21%	41.70%

When a test is conducted where 10^{13} bits are passed, the maximum number of error events to declare a Bit Error Rate of 10^{-12} with confidence level of >95% is four.

7.4.2 Measurement of Differential Voltage Amplitudes (Gen1, Gen2)

The differential voltage amplitude, V_{diffTX}, shall be measured for bits in representative data patterns. It is necessary to use patterns that are DC balanced for this testing (otherwise, an offset is introduced that shifts the measured mean values).

The test setup shown in Figure 140 below shows the connections:

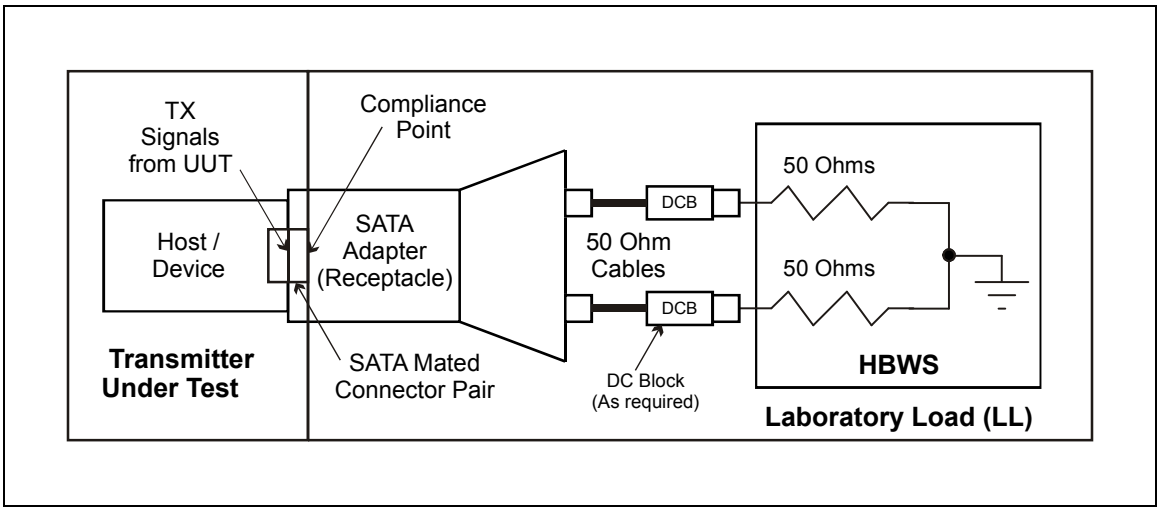


Figure 140 – Differential Voltage Amplitude Measurement

The transmitter under test sends the test pattern to a HBWS. The differential voltage waveform corresponding to one complete cycle of the N bit pattern has some unit intervals corresponding to zero bits and some unit intervals corresponding to ones bits. Figure 141 illustrates an example of a display on an equivalent time scope:

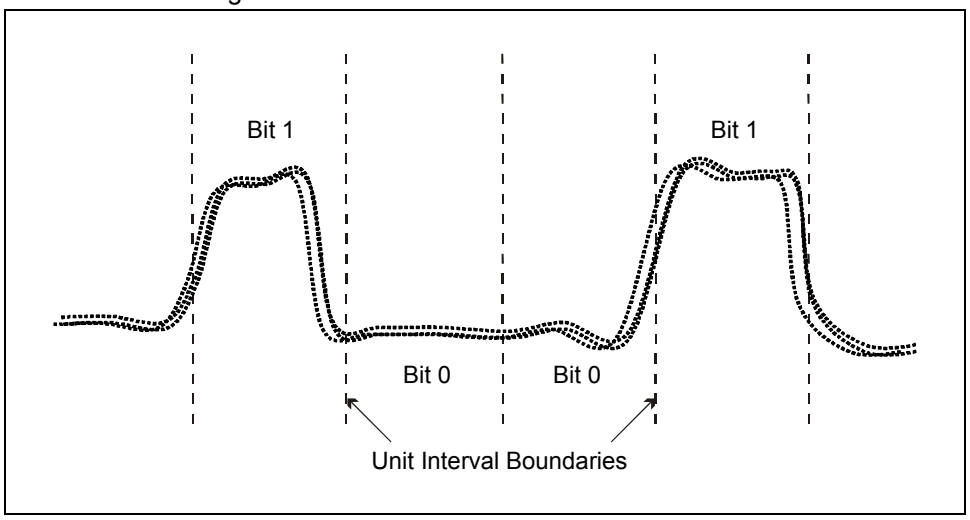


Figure 141 – Differential Voltage Amplitude Measurement Pattern Example

7.4.2.1 Testing for Minimum Differential Voltage Amplitude

There are two separate procedures for this testing. However, each of the two procedures requires a common set of steps to be performed. These are labeled below as “Common Steps”. Following these steps are those that describe the rest of the procedure for one of two options (labeled as “Option 1 or Option 2 Steps”).

7.4.2.1.1 Common Steps

Common Step 1: Transmitting a HFTP pattern, for a unit interval (UI) corresponding to a 1 bit, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

The inequality above is based on a requirement that enough samples are collected to define a confidence interval with at least 95% probability and with a width no greater than 10% of the sample mean.

Compute the following value:

$$UH = [\bar{x} - \frac{1.96s}{\sqrt{n}}]$$

Common Step 2: Transmitting a HFTP pattern, for a unit interval (UI) corresponding to a 0 bit, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Compute the following value:

$$LH = [\bar{x} + \frac{1.96s}{\sqrt{n}}]$$

Common Step 3: Transmitting a MFTP pattern, for a unit interval (UI) corresponding to the second 1 bit of a string of two consecutive 1 bits, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Compute the following value:

$$UM = [\bar{x} - \frac{1.96s}{\sqrt{n}}]$$

Common Step 4: Transmitting a MFTP pattern, for a unit interval (UI) corresponding to the second 0 bit of a string of two consecutive 0 bits, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s / \bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Compute the following value:

$$LM = [\bar{x} + \frac{1.96s}{\sqrt{n}}]$$

Common Step 5: Compute the minimum of the following two differences:

DH = UH – LH
DM = UM – LM

That is, compute DHM = min (DH, DM).

This value is used in the final step of each of the following two options.

7.4.2.1.2 Lone Bit Pattern Measurements, Option 1

If the test environment allows for the creation of a pattern trigger, the LBP pattern is used to make the following measurements. Continue the procedure from Common Step 5 above with the following steps for Option 1.

If the test environment does not allow for the creation of a pattern trigger, then continue the procedure from Common Step 5 above with the steps beginning with Option 2 Step 6 below.

Option 1 Step 6: Transmitting a LBP pattern, for a unit interval (UI) corresponding to a lone 1 bit, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for

the UI. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Compute the following value:

$$A = [\bar{x} - \frac{1.96s}{\sqrt{n}}]$$

Option 1 Step 7: Transmitting a LBP pattern, for a unit interval (UI) corresponding to a lone 0 bit, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Compute the following value:

$$B = [\bar{x} + \frac{1.96s}{\sqrt{n}}]$$

Option 1 Step 8: From A and B obtained in steps 1 and 2, compute:

VTestLBP = A – B

Then take the minimum of VTestLBP and the previously computed DHM (from Common Step 5), that is,

VTest = min (VTestLBP, DHM)

The test for minimum amplitude is passed if:

VTest > V_{diffTX}(Min)

See Table 31 section 1.1.1 for V_{diffTX}(Min). Otherwise, the test for minimum differential voltage amplitude has not been passed. If the test for minimum voltage amplitude is failed, the number of samples, n, is to be increased and the test shall be executed again for this larger number of samples. Failure to arrive at a value n, for which the test passes, means that the requirement of the specification, for minimum differential voltage amplitude, has not been met.

7.4.2.1.3 Approximation to Lone Bit Pattern Measurements, Option 2

To test for minimum differential voltage amplitude without the ability to create a pattern trigger, continue the procedure from Common Step 5 above with the following steps for option 2:

Option 2 Step 6: Transmitting a LFTP pattern, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI of the first 1 bit that follows either a string of three preceding 0 bits or a string of four preceding 0 bits. It is required that the histogram samples be the union of the samples collected for both cases. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Compute the following value:

$$A = [\bar{x} - \frac{1.96s}{\sqrt{n}}]$$

Option 2 Step 7: Transmitting a LFTP pattern, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI of the first 0 bit that follows either a string of three preceding 1 bits or a string of four preceding 1 bits. It is required that the histogram samples be the union of the samples collected for both cases. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Compute the following value:

$$B = [\bar{x} + \frac{1.96s}{\sqrt{n}}]$$

Option 2 Step 8: Transmitting a LFTP pattern, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI of the last '1' bit in a string of three or four '1' bits. It is required that the histogram samples be the union of the samples collected for both cases. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Call the mean, $\bar{x} = C$

Option 2 Step 9: Transmitting a LFTP pattern, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI of the last '0' bit in a string of three or four '0' bits. It is required that the histogram samples be the union of the samples collected for both cases. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Call the mean, $\bar{x} = D$

Option 2 Step 10: Transmitting a LFTP pattern, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI of the last '1' bit in a string of four '1' bits. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Call the mean, $\bar{x} = E$

Option 2 Step 11: Transmitting a LFTP pattern, construct a histogram based on n samples collected in the waveform epoch [0.45 UI, 0.55 UI] for the UI of the last '0' bit in a string of four '0' bits. The number of samples in a histogram (n) for the UI shall be greater than or equal to 100 and shall meet the requirement that:

$$1537(s/\bar{x})^2 \leq n$$

where:

$\bar{x} =$ the mean of the voltage samples in the histogram that may be read from the HBWS in histogram measurement mode

s = the standard deviation of the voltage samples in the histogram which may also be read from the HBWS

n = the number of samples that contribute to the histogram – this may also be read from the HBWS

Call the mean, $\bar{x} = F$

Option 2 Step 12: From A and B obtained in steps 1 and 2, compute:

VTestAPP = (A + C + F) – (B + D + E)

Then take the minimum of VTestAPP and the previously computed DHM, that is,

VTest = min (VTestAPP, DHM)

The test for minimum amplitude is passed if:

VTest > V_{diffTX}(Min)

See Table 31, section 1.1.1 for V_{diffTX}(Min). Otherwise, the test for minimum differential voltage amplitude has not been passed. If the test for minimum voltage amplitude is failed, the number of samples, n, is to be increased and the test shall be executed again for this larger number of samples. Failure to arrive at a value n, for which the test passes, means that the requirement of the specification for minimum differential voltage amplitude has not been met.

Figure 142 illustrates the locations of the sections of the LFTP as displayed on a scope from which the measurements of the values for A, B, C, D, E, and F (see steps 1 through 6 above) are to be made.

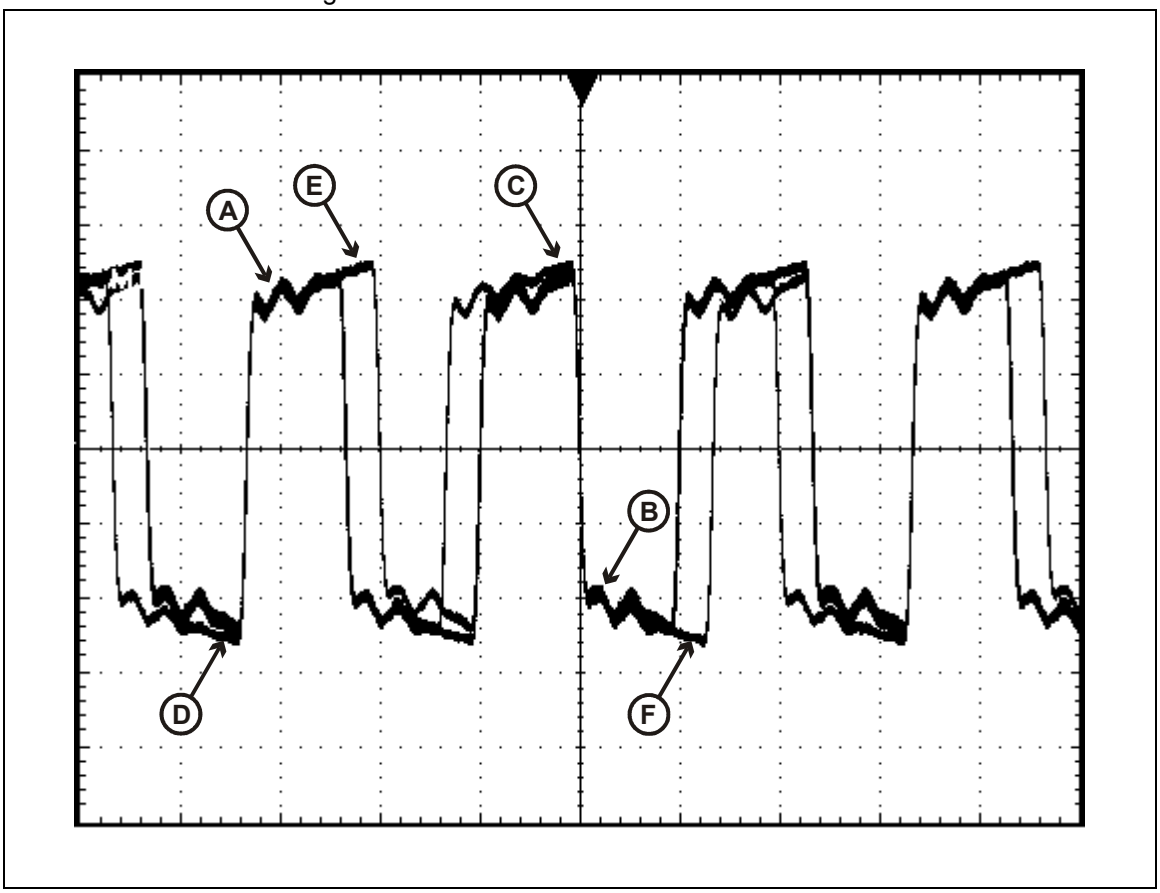


Figure 142 – LFTP Pattern on High BW Scope (HBWS)

7.4.2.2 Test for Maximum Differential Voltage Amplitudes

To test for maximum differential voltage amplitude for a given data pattern, perform the following steps using the LFTP and the MFTP as the data patterns. The waveform sections to be examined are defined as:

High Test UI:

For the LFTP, use the fourth ‘1’ bit in a string of four ‘1’ bits.
For the MFTP, use the first ‘1’ bit in a string of two ‘1’ bits.

Low Test UI:

For the LFTP, use the fourth ‘0’ bit in a string of four ‘0’ bits.
For the MFTP, use the first ‘0’ bit in a string of two ‘0’ bits.

Step 1: For the High Test UI, construct a histogram in the waveform epoch [0.0 UI, 1.0 UI] for the UI. Position the upper edge of the histogram window at VU mV where:

$$VU = \frac{1}{2} V_{DIFFTX}(\max)$$

See Table 31, Section 1.1.1 for V_{diffTX}(Max).

Position the lower edge of the histogram window at 0mV. Let the histogram acquire hits for a fixed time duration, T, such that the number of hits acquired is at least 10000. Note the number of histogram hits as NU. This histogram may be based on data stored in the waveform database.

For the same High Test UI, construct a histogram in the waveform epoch [0.0 UI, 1.0 UI]. Position the upper edge of the histogram window at VU + 300 mV. Position the lower edge of the histogram window at VU mV. Note the number of histogram hits as nu.

Step 2: For the Low Test UI, construct a histogram in the waveform epoch [0.0 UI, 1.0 UI]. Position the upper edge of the histogram window at 0 mV. Position the lower edge of the histogram window at VL mV where:

$$VL = -\frac{1}{2} V_{DIFFTX}(\max)$$

Let the histogram acquire hits for the same fixed time duration, T, as used in step 1. (In practice, using the same waveform database as that collected in Step 1 insures that the same time duration is examined.) Note the number of histogram hits as NL.

For the same Low Test UI, construct a histogram in the waveform epoch [0.0 UI, 1.0 UI]. Position the lower edge of the histogram window at VL – 300 mV. Position the upper edge of the histogram window at VL mV. Note the number of histogram hits as nl.

Step 3: Compute the values:

$$pu = \frac{nu}{nu + NU}$$

$$pl = \frac{nl}{nl + NL}$$

(Note: There are two values of pu and pl computed; one for the use of the LFTP and one for the use of the MFTP).

Step 4: The test for maximum amplitude is passed if:

pu < 0.05

and

pl < 0.05

(Note: Since there are two values of pl and pu, the test needs to be applied to each pair.)

Otherwise, the test for maximum differential voltage amplitude has not been passed.

7.4.3 Measurement of Differential Voltage Amplitudes (Gen3i)

The amplitude measurement of differential signals for Gen3i uses different methods for the maximum amplitude and the minimum amplitude compliance tests. The maximum amplitude is a peak-to-peak value measured at the TX compliance point into a Lab Load. This limits the magnitude of signals present in the interface. The minimum amplitude is a measurement of the minimum eye opening, using the specified method, after the Gen3i CIC, into a Lab Load. This provides a minimum signal level for the receiver, measured in a manner that is representative of how a typical receiver would process the signal.

Achieving both the maximum and minimum differential amplitude compliance limits as listed in Table 31 shall be required, using the same transmitter settings for both tests.

The same methods and patterns are used for setting up the Lab-Sourced Signal for Receiver Tolerance Testing, with the compliance limits specified in **Table 33**.

7.4.3.1 Maximum Differential Voltage Amplitude (Gen3i)

The maximum differential amplitude shall be measured at the TX Compliance point into a Lab Load. Figure 156 shows a drawing of this test connection. A Gen3 MFTP shall be used for this compliance measurement, although it is possible that with other patterns and signal path characteristics, additional peak-to-peak maximum amplitude values maybe present in the actual system. The MFTP will contain emphasis due to its run length, if the transmitter supports this signal conditioning, and allows for simple edge triggering for the signal capture.

The maximum amplitude is defined as the peak to peak value of the average of 500 waveforms measured over a time span of 4 Gen3 UI, using the HBWS.

7.4.3.2 Minimum Differential Voltage Amplitude (Gen3i)

The minimum TX differential amplitude shall be measured through the Gen3i CIC as specified in section 7.2.7 terminated into the Lab Load. Figure 157 shows a drawing of this test connection. A Gen3 LBP shall be used for this compliance measurement, although it is possible that with other patterns and signal path characteristics, lower amplitudes may be present in the actual system.

The minimum amplitude is defined as the vertical eye opening of the 1E-12 BER contour at the 50% point of the UI, when the data is captured using the Gen3i Reference Clock JTF defined in section 7.3.2.

The test equipment or JMDs typically used for this measurement include a BERT or a HBWS with appropriate hardware and software to measure or extrapolate a Vertical Bathtub Curve, Eye BER Contour plot, or statistical Eye Opening characteristics. Since some JMDs extrapolate the Eye Opening at the BER target from a smaller population size capture, that measurement is an approximation of the actual Eye Opening. If there are discrepancies between test equipment in the reported Eye Opening at the target BER, the value obtained from a full population vertical BERT scan shall be the standard for this measurement. This is analogous to the horizontal full population BERT scan as being the standard for TJ when measuring jitter.

All test equipment requires a minimum signal amplitude to be able to measure the 1E-12 BER contour or equivalent data. This level varies with instrumentation type, hardware and software. This minimum required instrumentation amplitude introduces errors in the reported minimum amplitude measurement value. This error results in the minimum amplitude reported by the test equipment to be smaller than the actual signal minimum amplitude. This instrumentation error shall be corrected for, to determine the actual minimum amplitude value using recommended methods provided by the test equipment manufacturer. If no such recommended correction

procedure is available for one piece of test equipment, alternate test equipment could be selected.

An approximate correction method could be used in the case of unsupported test equipment. The amplitude of a low noise Lab Source Gen3 MFTP test pattern with the fastest allowed Gen3i rise and fall times is reduced in several steps using passive calibrated attenuators. If the reported amplitude is plotted on the y-axis and the ideal amplitude calculated using the calibrated attenuators and source is plotted on the x-axis, the y-axis intercept represents the theoretical reported amplitude for a zero amplitude input. A linear curve fit to the measured data can extrapolate the measured data to the y-axis intercept. (a negative value) The absolute value of this y-axis intercept is then added as a positive number to the instrument reported minimum Eye Opening values to correct for this error term. Since this is a statistical measurement and the test equipment may contain significant random amplitude variations. This correction method can be in error since it does not convolve the random amplitude variation sources, but it reduces the error magnitude below an uncorrected measurement. This may result in a possible over correction of the instrumentation error term.

7.4.4 Rise and Fall Times

The rise and fall times of the waveform under test are defined over a 20%-80% output level change from the High and Low reference levels. High Reference level of the waveform under test is the “mode” of the top portion while the Low Reference level is the “mode” of the bottom portion. Mode is measured using Statistical Methods of the desired waveform and is the most common value of the probability density function. The minimum time span of the analysis zone for measuring the mode amplitude shall be 8 UI.

Therefore, Rise Time = X2 - X1; where X2 is the mean horizontal time value corresponding to 80% of the distance between the Low and High value and X1 is the mean horizontal time value position corresponding to 20% of the distance between the Low and High value.

And Fall Time = X1 - X2; where X1 is the mean horizontal time value corresponding to 20% of the distance between the Low and High value and X2 is the mean horizontal time value position corresponding to 80% of the distance between the Low and High value.

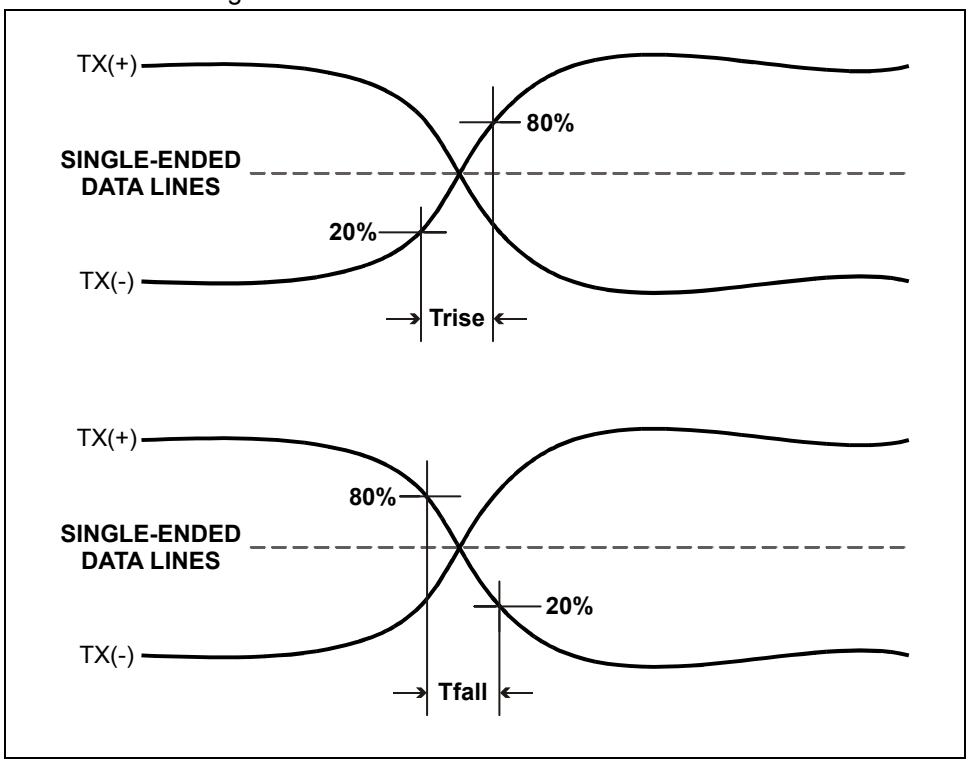


Figure 143 – Single Ended Rise and Fall Time

Rise and Fall values are measured using the LFTP Pattern previously defined, for all Gen1 and Gen2 Electrical Specifications. The average rise time of all rising edges and the separate average fall time of all falling edges within the 8 UI analysis zone shall both meet the required rise and fall time compliance limits.

For Gen3i, the Rise and Fall time values, between 20% and 80%, are measured using only the Gen3 LFTP. This minimizes errors in determining the 0% and 100% reference levels using the Mode Amplitude measurement method. The analysis zone of the measurement shall be made over a minimum time length of 8 UI. This is a Lab Load measurement. The Rise and Fall time compliance limits, for the differential TX test pattern are listed in Table 31. The average Rise time of all rising edges and the separate average Fall time of all falling edges within the analysis zone shall meet the Rise and Fall time compliance limits respectively. The Rise and Fall time compliance limits for the differential Data Signal Source (see Figure 162), for Receiver Tolerance testing, are also set with this method and pattern. The compliance limits for the Lab-Sourced Signal are listed in Table 33.

The rise and fall times for transmitter differential buffer lines are measured with the load fixture shown in Figure 144. The rise and fall times shall be measured with an HBWS.

7.4.5 Transmitter Amplitude

The transmitter amplitude values specified in Table 31 refer to the output signal from the unit under test (UUT) at the mated connector into a Laboratory Load (LL) (for Gen1i, Gen2i, Gen1m, Gen2m, Gen1x, Gen2x, and Gen3i), or from the unit under test through a Compliance Interconnect Channel (CIC) into a Laboratory Load (for Gen1x, Gen2x, and Gen3i only). The signals are not specified when attached to a system cable or backplane.

7.4.5.1 Transmitter Amplitude (Gen1 and Gen2)

Transmitter minimum amplitude is measured with each of three waveforms: HFTP, MFTP, and the Lone Bit Pattern (LBP). Amplitude specifications shall be met according to the measurement method outlined in section 7.4.2
The minimum amplitude value is measured during the TX minimum voltage measurement interval defined in Table 31. The Reference Clock (defined in section 7.3.2) defines the ideal (zero jitter) zero crossing times. The maximum amplitude is measured according to the measurement method outlined in section 7.4.2.2 using waveforms LFTP and MFTP.

The transmit DC offset voltage (for Gen1i only) should be measured with the setup in Figure 144. The HBWS is measuring a DC voltage and the DC blocks shall not be present.

Figure 144 and Figure 145 show test setups for measuring transmitter amplitude. The HBWS is the standard for measuring amplitude. The losses in the test connections may be significant so it is prudent to minimize and estimate these. Several methods may be used to estimate the cabling losses. The first is to use two cables of different lengths and compare the losses of each. The second is to rely on published data for the cables. The third is to obtain a separate means for measuring the cable loss such as characterization with a network analyzer or power meter.

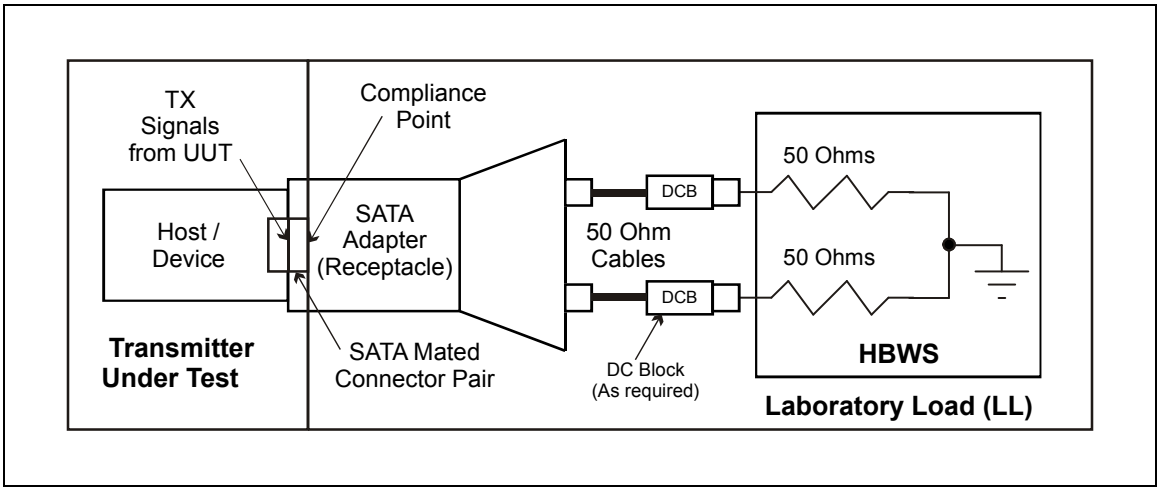


Figure 144 – Transmit Amplitude Test with Laboratory Load

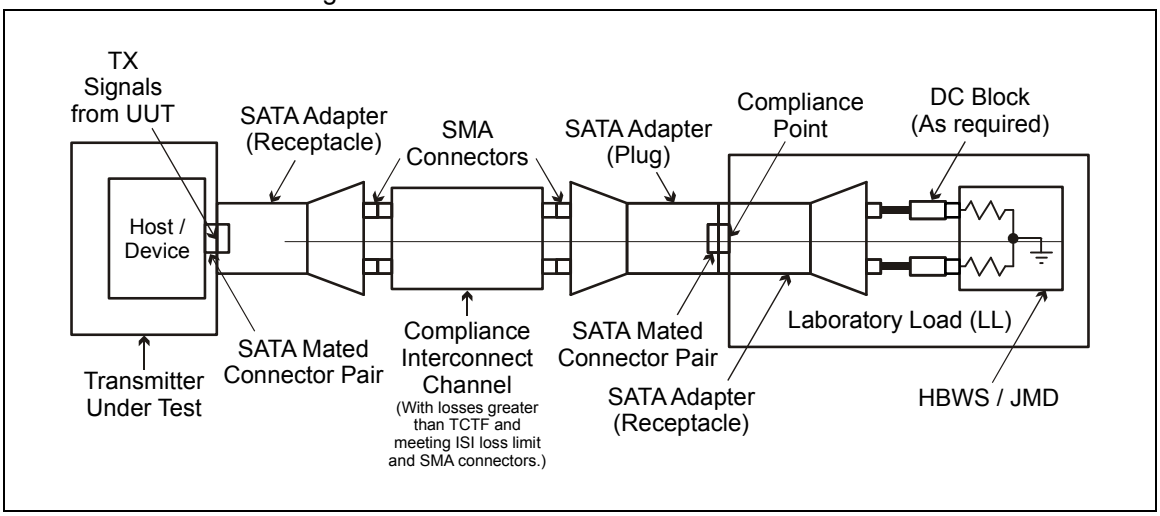


Figure 145 – Transmit Amplitude Test with Compliance Interconnect Channel

This specification describes transmitter levels in terms of voltage when driving a test load of 100 Ohms differential (the Laboratory Load, LL) and 50 Ohms single ended to ground. To relate the specified maximum levels to the maximum values seen in a system requires a calculation. An example of this calculation is in section 7.4.6.

7.4.5.2 Transmitter Amplitude (Gen3i)

Transmitter minimum amplitude is measured with the Lone Bit Pattern (LBP). Amplitude specifications shall be met according to the measurement method outlined in section 7.4.3.

The minimum amplitude value is measured during the TX minimum voltage measurement interval defined in Table 31. The Reference Clock (defined in section 7.3.2) defines the ideal (zero jitter) zero crossing times. The maximum amplitude is measured according to the measurement method outlined in section 7.4.3 using the MFTP waveform.

Figure 156 and Figure 157 show test setups for measuring transmitter amplitude. See section 7.4.5.1 for suggestions on compensating for losses in the test connections.

7.4.6 Receive Amplitude

This section describes setting the receive amplitude, a test condition common to many tests. The proper operation of the receiver is its ability to receive a signal. An example of this testing is described in section 7.4.11. The values as specified in Table 33 refer to the input signal from any signal source as measured at the device under test using a Laboratory Load.

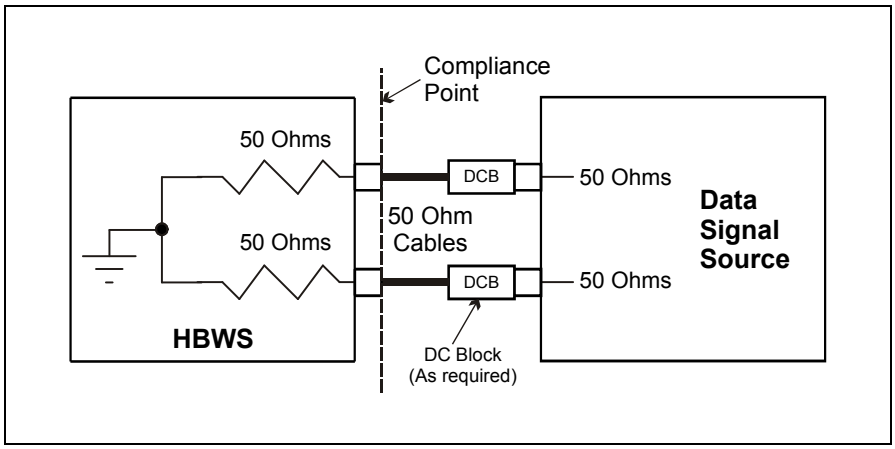


Figure 146 – Receiver Amplitude Test--Setting Levels

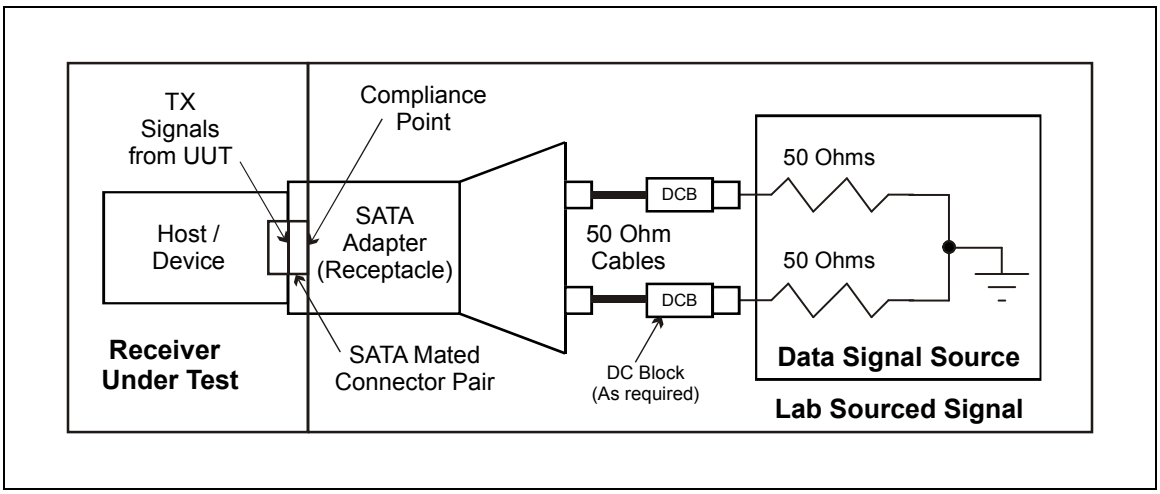


Figure 147 – Receiver Amplitude Test

Figure 146 shows an example to account for loss in the cabling and error in the signal source for receiver level testing. The loss in the SATA adapter is not accounted for here, but may be separately measured. The HBWS is used as the standard for amplitude when setting the levels for testing receivers. Equivalent methods to account for loss in the cabling are acceptable.

This specification describes receiver levels in terms of voltage driven from a differential source of 100 Ohms impedance. A calculation is required to relate the specified maximum receiver level to the maximum receiver level in a system. The maximum receiver level is set at a HBWS by driving with a signal source impedance of 100 Ohms. With the signal generator level set, it is then applied to the receiver under test. The voltage actually seen at the receiver inputs depends on the input impedance of the receiver. The maximum voltage at the receiver occurs when the receiver input impedance is at its maximum value.

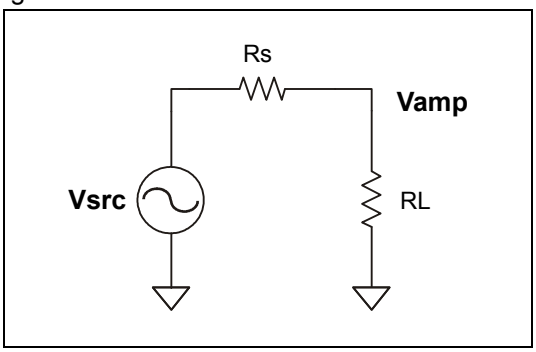


Figure 148 - Voltage at Receiver Input

$$V_{AMP} = \frac{V_{SRC}}{\left(1+\frac{R_S}{R_L}\right)} = \frac{V_A\left(1+\frac{R_S}{R_{LL}}\right)}{\left(1+\frac{R_S}{R_L}\right)}$$

The values of the receiver and transmitter resistor termination are set by the return loss specification at low frequency. Return loss is given by the following equation:

$$RL = -20\log\left[\left|\frac{Z-Z_0}{Z+Z_0}\right|\right]$$

And solving this for the resistance, the real part of the impedance gives two solutions

$$R = Z_0\frac{1-10^{-RL/20}}{10^{-RL/20}+1} = (100)\frac{1-10^{-18/20}}{10^{-18/20}+1} = 77.64$$

$$R = Z_0\frac{10^{-RL/20}+1}{1-10^{-RL/20}} = (100)\frac{10^{-18/20}+1}{1-10^{-18/20}} = 128.8$$

$$R = Z_0\frac{1-10^{-RL/20}}{10^{-RL/20}+1} = (100)\frac{1-10^{-14/20}}{10^{-14/20}+1} = 66.73$$

$$R = Z_0\frac{10^{-RL/20}+1}{1-10^{-RL/20}} = (100)\frac{10^{-14/20}+1}{1-10^{-14/20}} = 149.9$$

The highest amplitude that may be seen at the receiver occurs when the receiver input resistance is highest.

$$V_{AMP} = \frac{0.7\left(1+\frac{100}{100}\right)}{\left(1+\frac{100}{128.8}\right)} = 0.7881$$

The lowest amplitude at the receiver occurs when the receiver input resistance is lowest.

$$V_{AMP} = \frac{0.4\left(1+\frac{100}{100}\right)}{\left(1+\frac{100}{77.64}\right)} = 0.3497$$

7.4.7 Long Term Frequency Accuracy

There are several considerations for choosing instruments to measure long-term frequency accuracy. The long-term frequency accuracy of the instrument time base needs to be significantly better than the 350 ppm limit in this specification; many oscilloscopes do not have this frequency accuracy.

A method to measure the long-term frequency accuracy is to use a frequency counter. The test setup shown in Figure 149 below shows the connections. The transmitter under test sends a HFTP (D10.2) signal to the frequency counter. The signal shall not have SSC modulation. The frequency counter should have a gating period set long enough to reduce the effects of noise; this may be done by setting the counter resolution to 10 Hz or better (350 ppm at 1.5 GHz is 525 kHz). The counter reads the long-term frequency of the transmitter; the accuracy is a percentage.

When SSC is present, long term frequency accuracy specification is not applicable, instead the SSC profile is measured (see section 7.4.11)

There are other instruments that contain a frequency counter with an accuracy significantly better than 350 ppm. For example some BERT equipment has a frequency counter on the clock input.

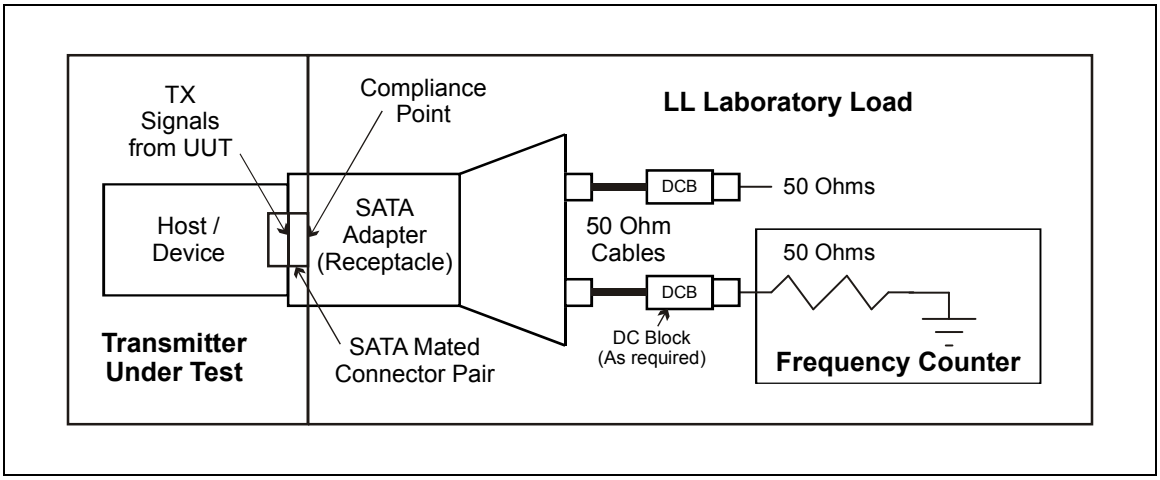


Figure 149 – TX Long Term Frequency Measurement

7.4.8 Jitter Measurements

Jitter is the difference in time between a data transition and the associated Reference Clock event, taken as the ideal point for a transition. The causes of jitter are categorized into random sources (RJ) and deterministic sources (DJ). Although the total jitter (TJ) is the convolution of the probability density functions for all the independent jitter sources, this specification defines the random jitter as Gaussian and the total jitter as the deterministic jitter plus 14 times the random jitter. The TJ specifications of Table 31 and Table 33 were chosen at a targeted BER of 10^{-12}. The BERT scan method described in section 7.4.8.1 is the only method that measures the actual TJ and is used as the reference for all TJ estimation methods. The method for estimating TJ is unique to each measurement instrument.

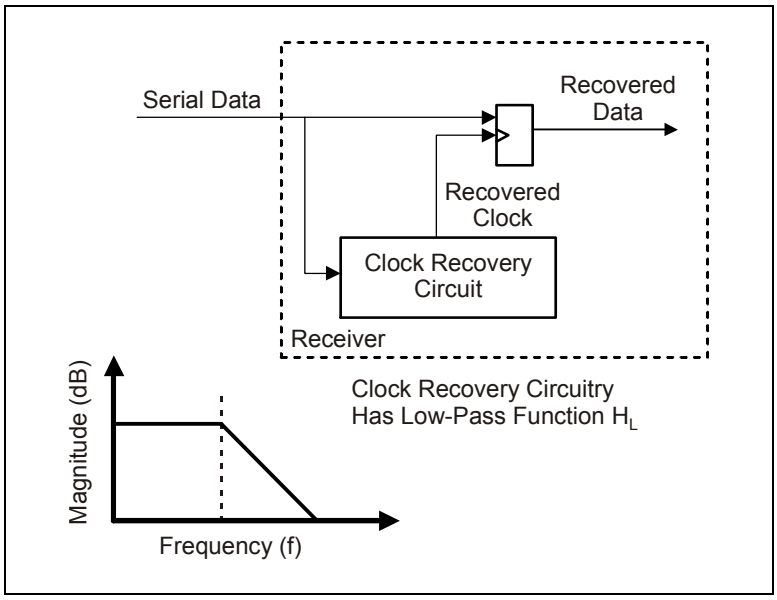


Figure 150- Receiver Model for Jitter

The jitter measurement methodology is defined as a clock to data jitter measurement. Figure 150 shows a block diagram of a deserializer input. The serial data is split into two paths. One path feeds clock recovery circuitry, which becomes the reference signal used to latch the data bits of the serial data stream. This clock recovery circuitry has a low pass transfer function H_L. This low pass function is the CLTF of the PLL or clock recovery circuit. The jitter seen by the receiver is the time difference of the recovered clock edge to the data edge position. This time difference function is shown in Figure 151. The resulting jitter seen by the receiver has a high pass function H_H shown in Figure 152. This high pass function is the JTF (Jitter Transfer Function) of the system. This defines the measurement function required by all jitter measurement methodologies. The required characteristics for the JTF (Gen1i, Gen1m, Gen2i, Gen2m) and the CLTF corner frequency f_c (Gen1x, Gen2x) are provided in section 7.3.2. In the case of a JMD, the JTF may be simply viewed as the ratio of the reported jitter to the applied jitter, for a sinusoidal PJ input.
Both the CLTF and the JTF are uniquely defined by the open loop transfer function G(s). Defining a CLTF does not uniquely define the G(s) and subsequently the JTF due to the level of cancellation of G(s) in the numerator and denominator of the CLTF especially when G(s) is much greater than 1, which is necessary for jitter tracking by the clock recovery circuit. This is the rationale for Gen1i, Gen1m, Gen2i and Gen2m directly specifying the JTF rather than the CLTF

of the clock recovery circuit. When the JTF of a JMD meets the requirements specified, the JMD reported jitter levels will closer represent the jitter applied to the receiver in this reference design.

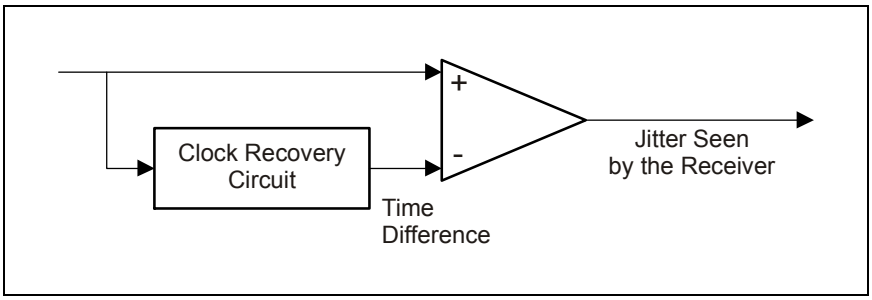


Figure 151 – Jitter at Receiver

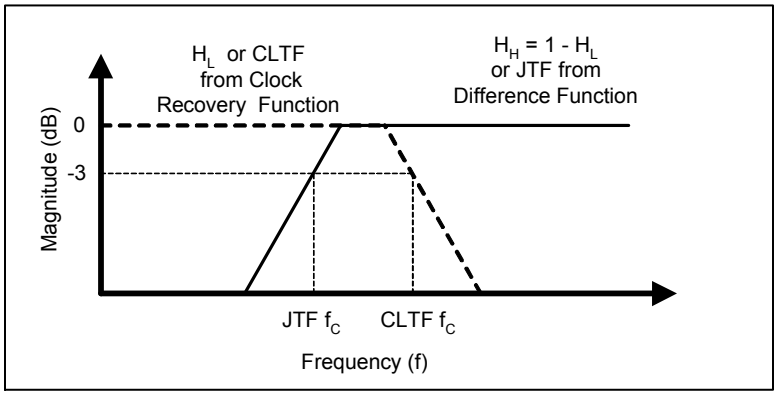


Figure 152 – Jitter at Receiver, High Pass Function

This JTF (H_H in Figure 152) mimics the receiver's ability to track lower frequency jitter components (wander, SSC) and not include them in the jitter measurement. This measurement methodology enables any measurement instrument to accurately measure the jitter seen by a receiver and produce measurements that correlate from measurement instrument to measurement instrument.

It should be noted that the corner frequency of the JTF is not the corner frequency of the clock recovery CLTF. This may not be obvious until one considers the phase shift caused by the clock recovery circuit. In general the vector sum $H_L(f) + H_H(f) = 1$. All quantities consist of changing magnitude and phase as a function of frequency. This accounts for differences in corner frequencies and peaking in the two frequency dependant functions.

Figure 153 shows more detail into how the JTF and CLTF relate to the jitter that would be applied to a receiver. The subfigure A) represents a generic control system block diagram for a feedback loop based clock recovery system. Subfigure B) translates the same complex variables to the combined system of the clock recovery circuit and the time difference function. It can be seen that E(s) is the jitter seen by the receiver, as well as being the error signal in the clock recovery circuit. Subfigure C) provides the defining equations for the clock recovery circuit CLTF and the combined system JTF function.

Both the CLTF and the JTF are uniquely defined by the open loop transfer function G(s). Defining a CLTF does not uniquely define the G(s) and subsequently the JTF due to the level of cancellation of G(s) in the numerator and denominator of the CLTF especially when G(s) is much greater than 1, which is necessary for jitter tracking by the clock recovery circuit. This is the rational for Gen2i, Gen2m, and Gen3i: directly specifying the JTF rather than the CLTF of the clock recovery circuit. When the JTF of a JMD meets the requirements specified, the JMD reported jitter levels will closer represent the jitter applied to the receiver in this reference design.

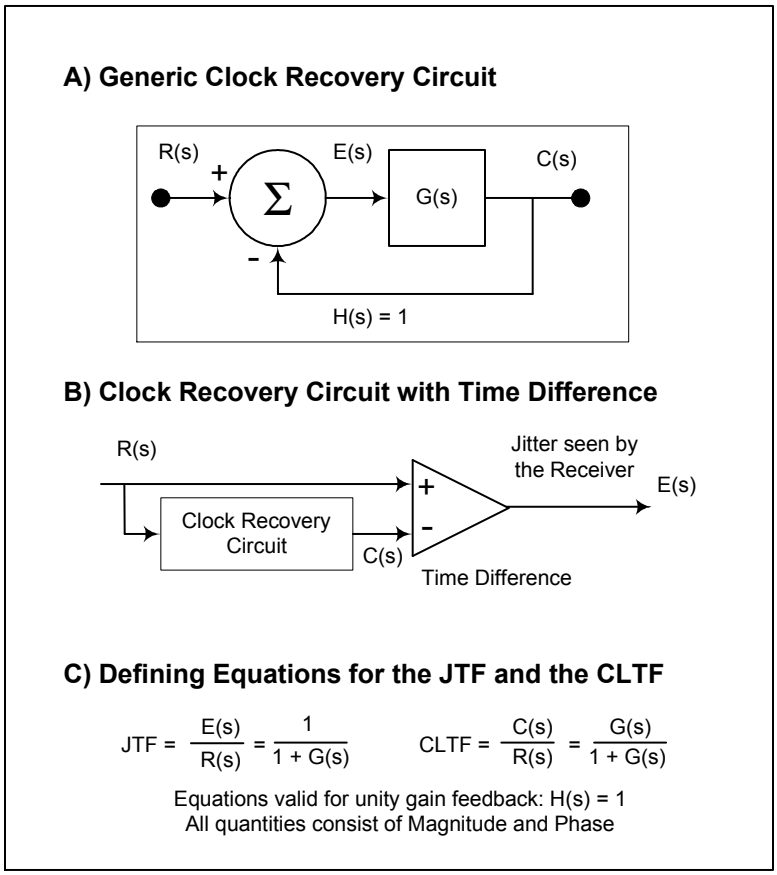


Figure 153 – JTF and CLTF Definition

7.4.8.1 Jitter Measurements with a Bit Error Rate Tester (BERT)

Most instruments used to measure jitter are unable to directly measure TJ at very low bit error rates like 10^{-12} due to the time it would take to capture sufficient transitions for a statistically significant direct measurement. Instead, these instruments capture a smaller sample size and extrapolate TJ using complex, and in some cases proprietary, algorithms. The determination of TJ through extrapolation may greatly reduce the amount of time required to measure jitter but experience has shown different extrapolation-based methods may produce different results. An alternate method to measure TJ is through the use of a BERT scan method. Since a BERT scan may directly measure jitter to 10^{-12} and even lower rates in a reasonable amount of time, it also provides a means of reconciling any differences in the extrapolated value of TJ.

A BERT scan method utilizes the variable clock-to-data timing path available on a BERT. In addition to this, a PLL inside or outside the BERT that meets the requirements described in section 7.3.2 shall be used to generate the clock reference. The BERT scan systematically increases or decreases the clock-to-data timing and directly measures the BER performance at each increment of time. This is done until the time skew is found for the desired BER rate on each

side of a Unit Interval. BER rates directly measured at each timing point on the left and right of a UI may be plotted to produce what is known as a bathtub curve. The time for one UI minus the time between the curves at the desired bit error rate is TJ. If those points on the bathtub curve are from directly measured data and not extrapolated, the TJ is a direct measure of TJ. Alternatively, the BERT scan is done to the left and right of the nominal zero crossing time relative to the Reference Clock to directly measure the tails of the Cumulative Distribution Function (CDF) histogram. The width of this histogram at the desired BER is TJ at that BER.

Methods do exist to extrapolate TJ on a BERT from time scan values at higher rates. While such methods may be used to predict TJ at a desired BER, only a direct measure all the way to the desired BER shall be used when using the BERT as a jitter standard for comparison to extrapolation methods.

The standard also requires a measure of DJ for compliance testing. All measures of DJ are statistically based including the estimations of DJ from a BERT. If a BERT is being used to measure TJ, the TJ values determined by that BERT may be used to estimate the DJ. Methods of estimating DJ from BERT TJ values are described in the public domain.[5] These methods involve the measurements of TJ at different BER levels using the BERT scan.

When measuring TJ and extracting the DJ and RJ components, it is common to encounter RJ measurements that are higher than actual random jitter. This is often encountered in systems where noise from the system causes jitter that is not correlated with the Serial ATA channel activity. For example, power supply noise from a system, which contaminates a transmitter's bit clock generator, may cause variations in the bit clock which impact jitter directly.

When making random jitter measurements, this non-correlated DJ is often included in the result which, when multiplied by 14 may lead to non-compliance to jitter specifications. This is inappropriate since non-correlated DJ is bounded, non-Gaussian and should not be multiplied by 14. Furthermore, non-correlated DJ is included in normal DJ measurements.

Extracting non-correlated DJ from RJ measurement lies beyond the scope of this document since it usually requires in-depth knowledge of the characteristics of the non-correlated DJ and an appropriate algorithm for its measurement/extraction. Consequently, it is the readers' responsibility to characterize and then extract non-correlated DJ from their RJ measurements.

7.4.9 Transmit Jitter (Gen1i, Gen2i, Gen1m, Gen2m, Gen1x, and Gen2x)

The transmit jitter values specified in Table 31 refer to the output signal from the unit under test (UUT) at the mated connector into a Laboratory Load (LL) (for Gen1i, Gen2i, Gen1m, Gen2m, Gen1x, and Gen2x), or from the unit under test through a Compliance Interconnect Channel (CIC) into a Laboratory Load (for Gen1x and Gen2x). The signals are not specified when attached to a system cable or backplane. All the interconnect characteristics of the transmitter, package, printed circuit board traces, and mated connector pair are included in the measured transmitter jitter. Since the SATA adapter is also included as part of the measurement, good matching and low loss in the adapter are desirable to minimize its contributions to the measured transmitter jitter.

Transmit jitter is measured with each of the specified patterns in section 7.2.4.1. The measurement of jitter is described in section 7.4.8. Transmit jitter is measured in one of the

[5] "Estimation of Small Probabilities by Linearization of the Tail of a Probability Distribution Function" by S.B. Weinstein, IEEE Transactions on Communications Technology, Vol. COM-19, No. 6, December 1971.

following two setups for Gen2i and both setups for Gen1x and Gen2x. For Gen1i, Gen2i, Gen1x, and Gen2x the transmitter is connected directly into the Laboratory Load (LL) shown in Figure 154. Additionally, for Gen1x and Gen2x the transmitter is connected through the Compliance Interconnect Channel (see section 7.2.7) into the Laboratory Load shown in Figure 155.

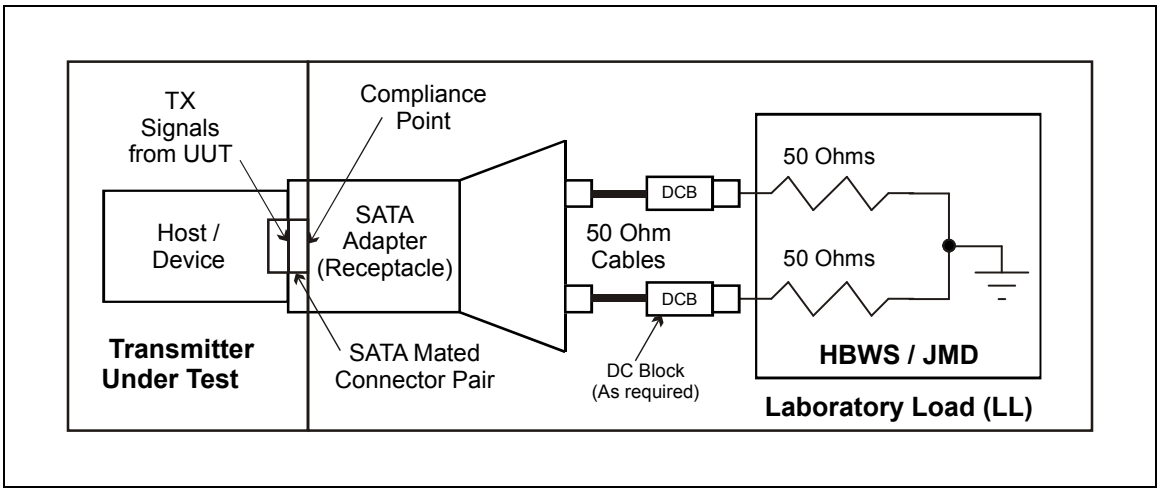


Figure 154 – Transmitter Jitter Test (Gen1i, Gen2i)

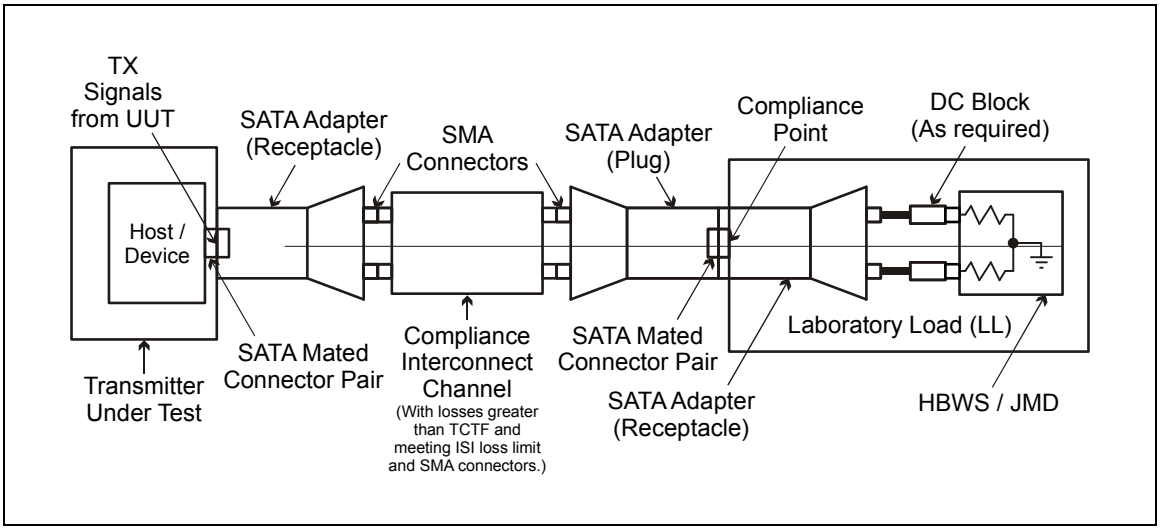


Figure 155 – Transmit Jitter Test with Compliance Interconnect Channel (Gen1x, Gen2x)

Transmitter jitter is measured into the Laboratory Load (LL), or in conjunction with the Compliance Interconnect Channel; both have very good impedance matching. The jitter in an actual system is higher since load and interconnect mismatch results in reflections and additional data dependent jitter. It is generally not possible to remove the effects of the SATA adapter on jitter since jitter due to mismatch depends on the entire test setup.

7.4.10 Transmit Jitter (Gen3i)

The Transmit Jitter values specified in Table 31 refer to the output signal from the unit under test (UUT) at the mated connector into a Laboratory Load (LL) when measuring Random Jitter (RJ) or from the unit under test through a Compliance Interconnect Channel (CIC) into a Laboratory Load when measuring Total Jitter (TJ). The signals are not specified when attached to a system cable or backplane. All the interconnect characteristics of the transmitter, package, printed circuit board traces, and mated connector pair are included in the measured transmitter jitter. Since the SATA adapter is also included as part of the measurement, good matching and low loss in the adapter are desirable to minimize its contributions to the measured transmitter jitter.

The Random Jitter is measured with a MFTP pattern and the Total Jitter is measured with each of the specified patterns in section 7.2.4.1 and section 7.2.4.3.4. The measurement of jitter is described in section 7.4.8.

First, the Random Jitter of a Gen3 MFTP pattern is measured directly into the Laboratory Load as is shown in Figure 156. The measured value shall meet the Transmit Jitter level for Random Jitter (RJ) for Gen3i in Table 31. The actual measured value of Random Jitter during this test is referred to as the Measured Random Jitter (RJ_{meas}) and is used to calculate the allowable TJ level for the second Transmitter Jitter Test, so this value needs to be recorded.

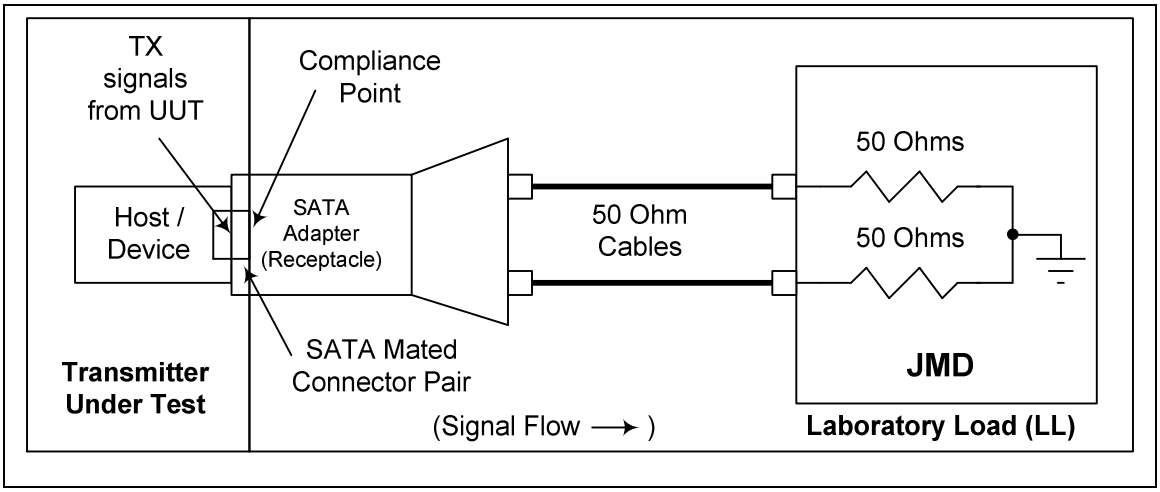


Figure 156 – Transmitter Random Jitter Test (Gen3i)

The second Transmit Jitter test measures the Total Jitter (TJ) through the Gen3i Compliance Interconnect Channel (see Section 7.2.7) into the Laboratory Load as is shown in Figure 157.

The allowable Total Jitter (TJ) level is calculated based on the equation in Section 7.4.8 that defines Total Jitter as the sum of Deterministic Jitter (DJ) plus 14 times the standard deviation of the Random Jitter (RJ). The maximum DJ level to be added to the RJ_{meas} value obtained in the first Transmit Jitter test is listed in Table 31. The sum of the RJmeas and the allowable DJ defines the maximum TJ level that the Transmit Jitter shall have.

The measured RJ from a JMD is typically reported as a standard deviation value (one sigma). The RJ must be multiplied by 14 to convert it to a peak-to-peak value before adding it to the allowable DJ level, which is a peak-to-peak value.

To clarify the maximum allowable TJ calculation for the second test, the following example is provided. The values in Table 31 provide the Gen3i TX Jitter compliance requirements. The following values are only an example. If the maximum allowable RJ is 0.18 UI p-p and the actual measured RJ (RJmeas) from the first Transmit Jitter test is 0.10 UI p-p (1.19 ps 1 sigma) and the allowable DJ addition is 0.34 UI p-p, then the maximum allowable TJ for the second Transmit Jitter test is: TJ (UI p-p) = DJ (UIp-p) + RJ_{meas} (UI p-p) = 0.34 (UI p-p) + 0.10 (UI p-p) = 0.44 (UI p-p). This is the maximum allowed TJ for the transmitter under test. It can be seen that if the RJmeas from the first test is at the maximum limit of 0.18 (UI p-p) and the allowable DJ addition is 0.34 (UI p-p), then the maximum allowable TJ is 0.52 (UI p-p).

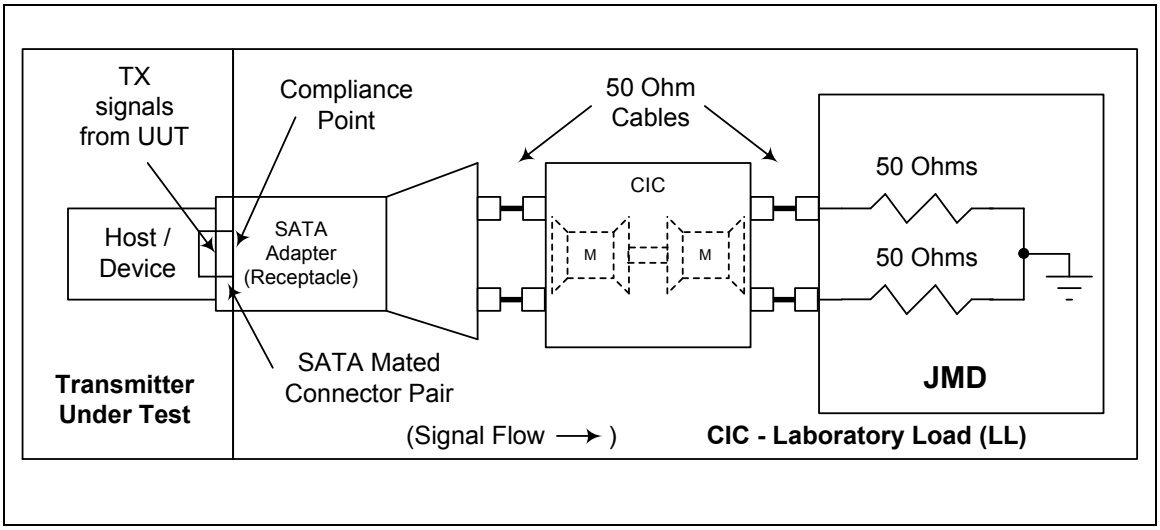


Figure 157 – Transmitter Total Jitter Test (Gen3i)

The Transmit Jitter shall meet both requirements as listed in Table 31 using the methods of the first test for RJ and the second test for TJ defined in this section. The methods used for Gen3i Transmit Jitter testing are intended to minimize RJ measurement error and allow for the TJ to be verified by a full population BERT scan as described in Section 7.4.8. This method also puts an upper limit on both DJ and RJ so neither may dominate the TJ.

Transmitter jitter is measured into the Laboratory Load (LL), or in conjunction with the Compliance Interconnect Channel; both have very good impedance matching. The jitter in an actual system is higher since load and interconnects mismatch results in reflections and additional data dependent jitter. It is generally not possible to remove the effects of the SATA adapter on jitter since jitter due to mismatch depends on the entire test setup.

7.4.11 Receiver Tolerance (Gen1i, Gen2i, Gen1m, Gen2m, Gen1x, and Gen2x)

The performance measure for receiver tolerance and common mode interference rejection is the correct detection of data by the receiver. When measuring receiver and Common Mode tolerance it is necessary to set the maximum allowable jitter and common mode interference on the signal sent to the receiver and monitor data errors.

The data signal source provides a data signal with jitter, and a controlled rise/fall time, with matched output impedance. The sine wave source provides common mode interference with matched output impedance. The two sources are combined with resistive splitters into the receiver under test (see Figure 159). Equivalent signal generation methods that provide the data with jitter, common mode interference, and an impedance-matched output are allowed. All the

interconnect characteristics of the receiver, mated connector pair, printed circuit board traces, and package are included in the measured receiver jitter tolerance.

Figure 158 shows a setup to set the level of jitter and common mode signal at the compliance point, on the cable side of the mated pair connector. The JMD is used as the standard for measuring jitter, and the HBWS is used as the standard for measuring the common mode interference. Since the SATA adapter is not included when setting the level of jitter, good matching and low loss in the adapter are desirable to minimize contributions to the amount of receiver jitter used in testing. Unlike other measurements, it is generally not possible to remove the effects of the SATA adapter on jitter since jitter due to mismatch depends on the entire test setup. Figure 159 shows one example approach to generate the Lab-Sourced signal.

The receiver tolerance test shall be conducted over variations in parameters SSC on and off, maximum and minimum rise and fall times, minimum and maximum amplitude, common mode interference over the specified frequency range, the test patterns LBP and the full payload COMP described in section 7.2.4.3, and jitter which includes random and deterministic jitter of various types: data dependent, periodic, duty cycle distortion. The receiver tolerance to the impairments is required over all signal variations.

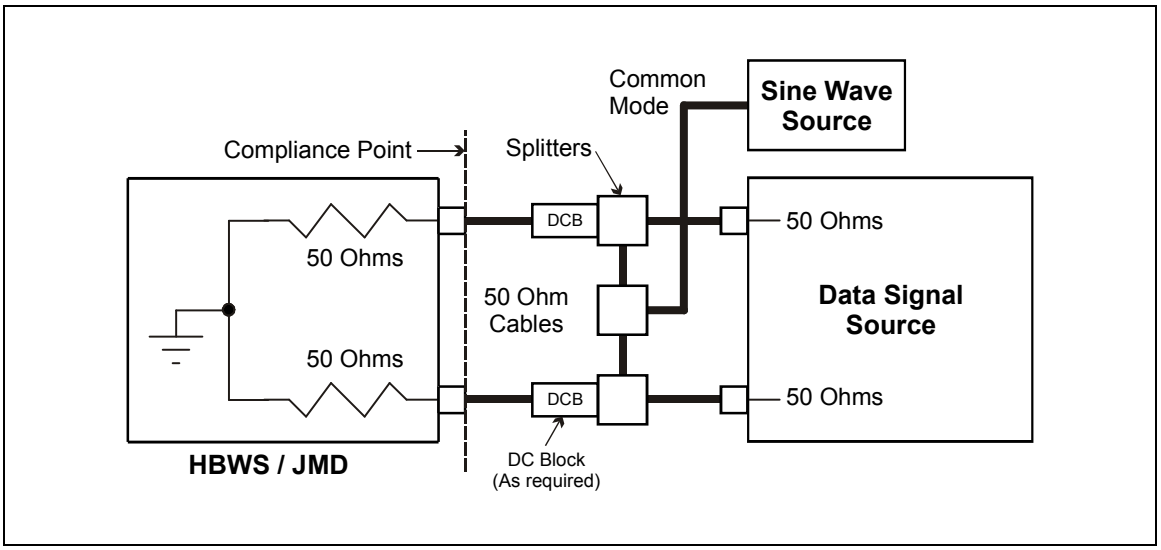


Figure 158 – Receiver Jitter and CM Tolerance Test—Setting Levels

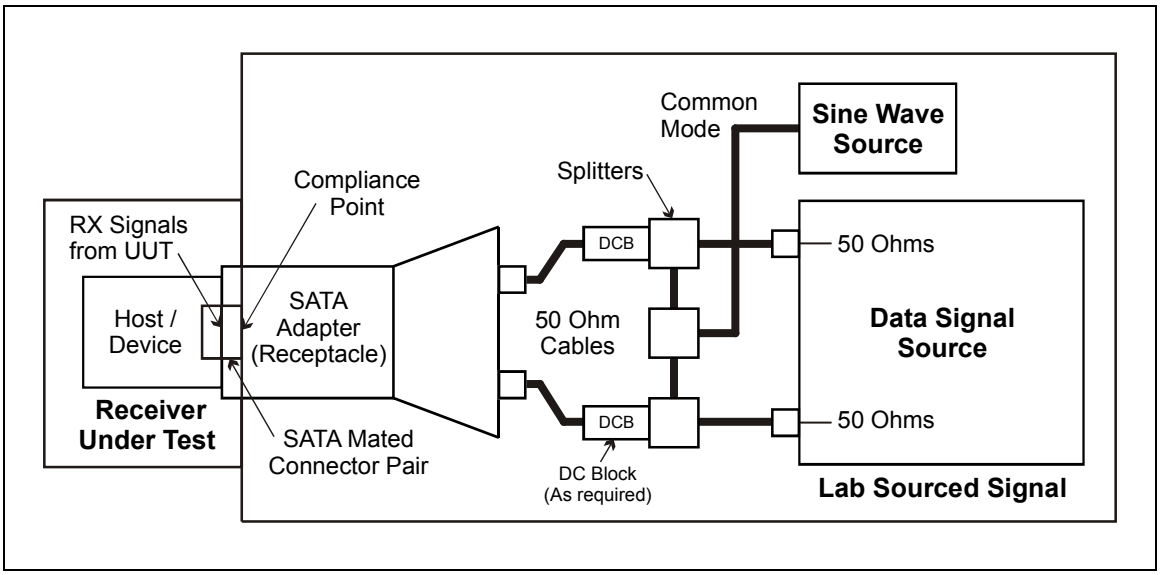


Figure 159 – Receiver Jitter and CM Tolerance Test

7.4.12 Receiver Tolerance (Gen3i)

The performance measure for receiver tolerance and Common Mode interference rejection is the correct detection of data by the receiver. When measuring receiver and Common Mode tolerance it is necessary to set the maximum allowable jitter and Common Mode interference on the signal sent to the receiver and monitor data errors.

The data signal source provides a data signal with jitter, and a controlled rise/fall time with a matched output impedance. Additional DJ (ISI) is added by the CIC. The sine wave source provides common mode interference with a matched output impedance. The two sources are combined with resistive splitters into the receiver under test (see Figure 162). Equivalent signal generation methods that provide the data with jitter, common mode interference, and an impedance-matched output are allowed. All the interconnect characteristics of the receiver, mated connector pair, printed circuit board traces, and package are included in the measured receiver jitter tolerance.

To calibrate the test signal for Receiver Tolerance testing, the Data Signal Source is measured using two procedures, one for Random Jitter (RJ) and a second for Total Jitter (TJ) and the common mode signal content.

The rise time and fall time of the Data Signal Source in the following figures shall meet the requirements listed in Table 33 for the Gen3i Lab Sourced Signal. This defines the signal rise time and fall time characteristics in the signal path before the CIC. This requirement shall be met using the rise time and fall time methods described in Section 7.4.4

Figure 160 show the test configuration for setting the Random Jitter (RJ) level as is defined in Table 33 for the Gen3i Lab Sourced Signal. The RJ level is set using a Gen3i MFTP pattern. This method minimizes the measurement errors of RJ, compared to the case when other signal degradations are present, and shall be done before adding additional jitter components and common mode signals.

This second procedure is performed after the RJ level of the Data Signal Source is set, as described above. Figure 161 shows one example approach for setting the Total Jitter (TJ) and the common mode signal level. The actual calibration plane is at the SMA connectors that will be applied to the SATA to SMA adaptor during the Receiver Tolerance test. This is shown in Figure

161 as a dotted vertical line. The JMD is used as the standard for measuring jitter, and the HBWS is used as the standard for measuring the common mode interference and signal amplitude. Since the SATA adapter is not included when setting the level of jitter, good matching and low loss in the adapter are desirable to minimize contributions to the amount of receiver jitter used in testing. Unlike other measurements, it is generally not possible to remove the effects of the SATA adapter on jitter since jitter due to mismatch depends on the entire test setup.

The measurement of the minimum and maximum amplitude levels of the test signal at the calibration plane, are performed in the same method used for these parameters for the TX amplitude tests. (see Section 7.4.3 for required method) In general the maximum peak-to-peak amplitude of a Gen3 MFTP pattern is the maximum limit, and the minimum eye opening of a Gen3 LBP at a BER of 1E-12 is the minimum limit.

Figure 162 shows the calibrated Lab Sourced Signal applied to the Receiver Under Test.

The receiver tolerance test shall be conducted over variations in parameters SSC on and off, minimum and maximum amplitude, common mode interference over the specified frequency range, the test patterns LBP and the full payload COMP described in section 7.2.4.3, and jitter which includes the maximum random and deterministic jitter of various types: data dependent, periodic, duty cycle distortion. The receiver tolerance to the impairments is required over all signal variations.

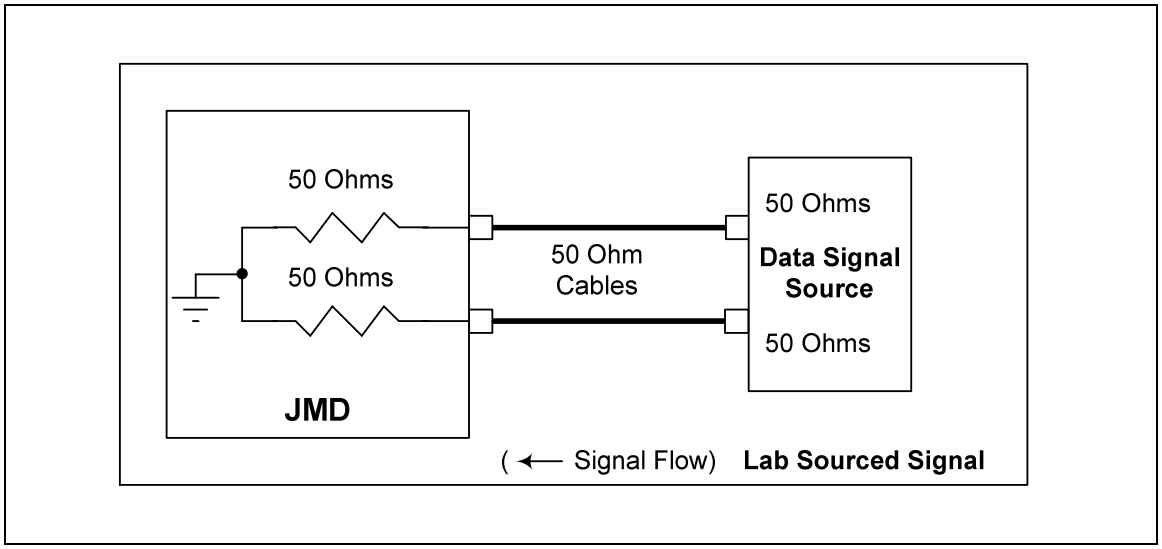


Figure 160 – Receiver Jitter and CM Tolerance Test – Setting RJ Level (Gen3i)

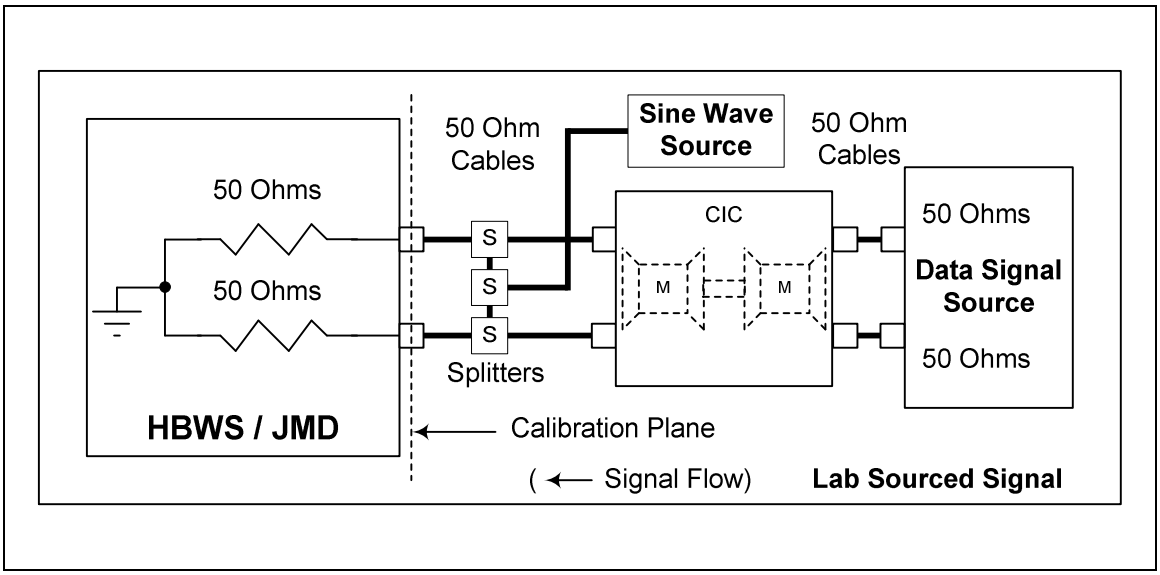


Figure 161 – Receiver Jitter and CM Tolerance Test – Setting TJ and CM Levels (Gen3i)

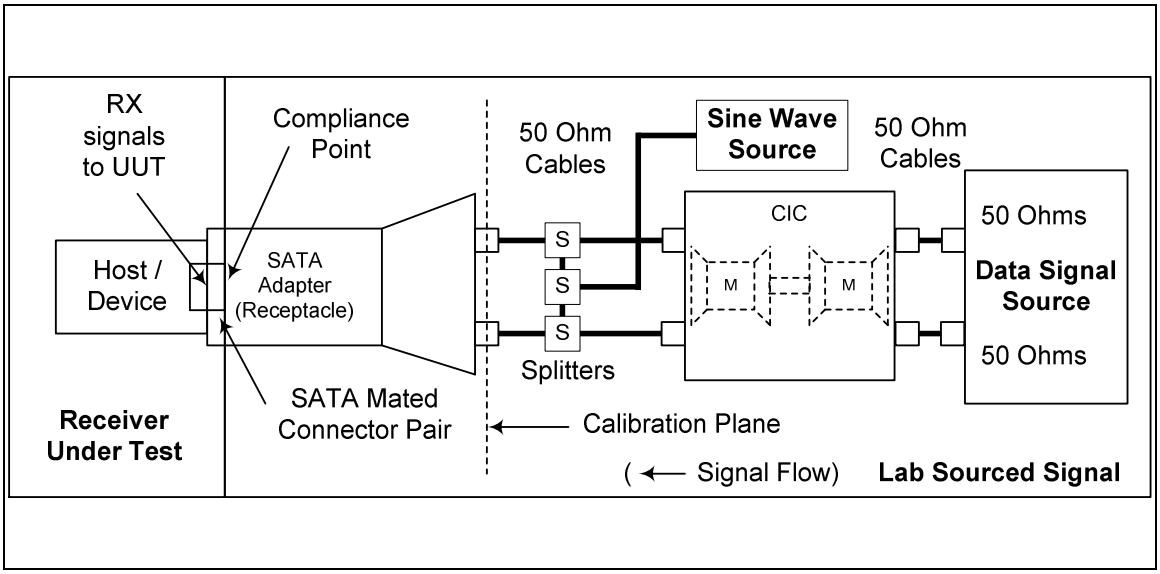


Figure 162 – Receiver Jitter and CM Tolerance Test (Gen3i)

7.4.13 Return Loss and Impedance Balance

The purpose of the return loss and impedance balance specifications (for rates above Gen1) is to bound the additional data dependent jitter incurred when attaching a host/device into a system. The test setup for hosts and devices is impedance matched in both differential and common modes and has good impedance balance whereas the system environment may not. Additional data dependent jitter occurs in a system from these imperfections. The return loss of a host/device quantifies the effect on the level of reflections in the system and the impedance balance controls the conversion between differential and common modes.

The differential return loss is defined as the magnitude of the differential mode reflection given a differential mode excitation, expressed in decibels. The common mode return loss is defined as the magnitude of the common mode reflection given a common mode excitation. The impedance balance is defined as the magnitude of the differential mode reflection given a common mode excitation. Each of these contributes to additional data dependent jitter in a system beyond that in a test setup.

The differential mode signal is defined by

$$v_{dm} = v_2 - v_1$$
$$i_{dm} = i_2 - i_1$$

The common mode signal is defined by

$$v_{cm} = \frac{v_2 + v_1}{2}$$
$$i_{cm} = \frac{i_2 + i_1}{2}$$

The return loss is defined by the magnitude of the reflection coefficient

$$RL = -20\log|\rho|$$

This specification describes transmitter output impedance and receiver input impedance in terms of the magnitude of a reflection of a sine wave. In a lossless line, the return loss remains constant over position. Attenuation loss in the test setup causes the measured return loss to appear higher (better matched) than actual. Figure 163 shows a setup to compensate for the loss in the cables. The short and load are assumed standards with RF connectors, for example SMA type connectors.

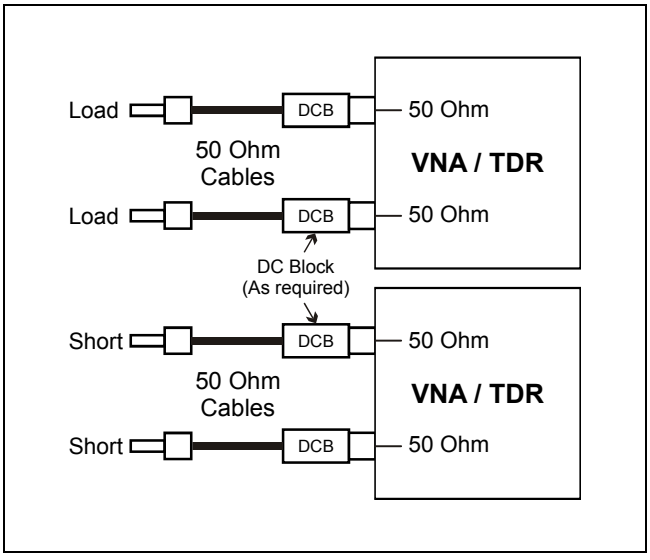


Figure 163 – Return Loss Test-Calibration

A reflection test set allows the measurement of reflections in the differential mode, or in the common mode. It may consist of a TDR with processing software, a multiport vector network analyzer, hybrid couplers and directional bridges, or a 2-port vector network analyzer and processing software.

Differential return loss, common mode return loss, and impedance balance may be measured with a 2-port vector network analyzer. The VNA is connected to the host/device and the S parameters are measured (with a 50 Ohm reference impedance). The differential return loss in terms of the mixed mode S parameters as well as the 2-port S parameters is given by

$$RL_{DD11} = -20\log\left|S_{DD11}\right| = -20\log\left|\frac{s_{11} + s_{22} - s_{12} - s_{21}}{2}\right|$$

The common mode return loss is given by

$$RL_{CC11} = -20\log\left|S_{CC11}\right| = -20\log\left|\frac{s_{11} + s_{22} + s_{12} + s_{21}}{2}\right|$$

The impedance balance is given by

$$RL_{DC11} = -20\log\left|S_{DC11}\right| = -20\log\left|\frac{s_{11} - s_{22} + s_{12} - s_{21}}{2}\right|$$

where the mixed mode S parameters are measured with a 4-port VNA, or alternatively the 2-port S parameters are measured with a 2-port VNA.

For the above equations, the mapping from single-ended to differential s-parameters assumes 50 Ohm single ended reference sources to 100 Ohm differential reference sources and 25 Ohm common-mode reference sources.

Figure 164 shows a test setup for measuring differential return loss. Since the SATA adapter is not included, good matching and low loss in the adapter are desirable to minimize its contributions to the measured return loss. If measurements and SATA adapter are characterized with S parameters, it is possible to remove adapter and test setup effects through a de-embedding process.

Test adapter imperfections affect the measurement of the unit under test; they introduce measurement uncertainty. For example, the attenuation loss in the test adapter reduces the reflection from the unit under test making the measured return loss higher than actual. An attenuation loss of 0.5 dB (about 1 inch of PCB trace on FR4 at 5 GHz) causes the measured return loss to increase by 1 dB over actual. The return loss of the adapter may affect the measured return loss higher or lower as the reflection from the adapter either adds or subtracts from the reflection from the unit under test. A well-matched adapter with 20 dB return loss may affect the measurements of a unit under test with return loss of 5 dB by ± 1 dB. These effects are most pronounced at higher frequencies.

The adapter affects the measured reflection by the following measurement uncertainty equation

$$s_{11m} = \varepsilon_{00} + \varepsilon_{01}\varepsilon_{10}s_{11a} + \varepsilon_{01}\varepsilon_{10}\varepsilon_{11}\left(s_{11a}\right)^2$$

where ε_{10} and ε_{01} are the attenuation loss, ε_{00} is the input reflection, and ε_{11} is the output reflection of the adapter. s_{11m} is the measured reflection, and s_{11a} is the actual reflection from the unit under test. The return loss is related to the reflection amplitude.

For the most accurate measurements, the adapter effects should be characterized or calibrated, and then de-embedded or removed from the measurements.

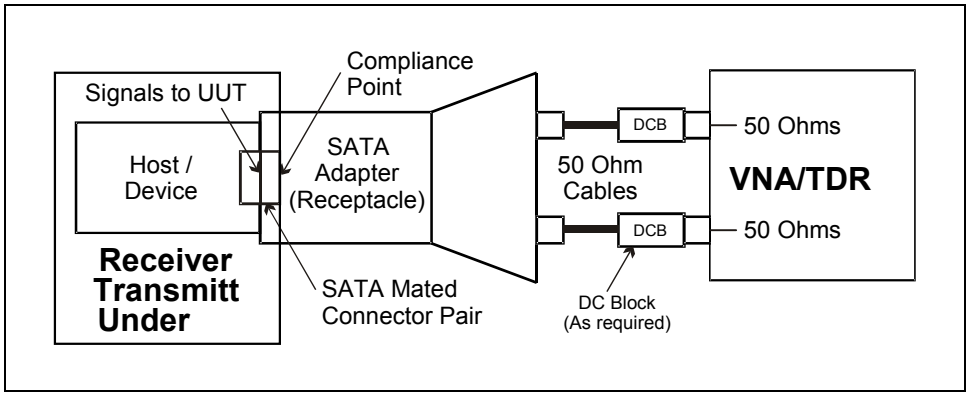


Figure 164 – Return Loss Test

When measuring output impedance of transmitters the operating condition shall be during transmission of MFTP. This is to assure the measurement is performed during a mode of operation that represents normal operation. The amplitude of external excitation applied shall not exceed -13.2 dBm 50 Ohms (139 mVpp) single ended on each differential port of the transmitter. This number is derived from the maximum reflected signal that may be present at a transmitter. A maximum transmitted signal of 700mVppd (-5.14 dBm 50 Ohms, each differential side, HFTP maximum rise/fall time); reflecting off a receiver with a differential return loss of 8 dB and direct attach.

When measuring input impedance of receivers the operating condition shall be during a PHYRDY Interface Power State (see section 8.1). The amplitude of external excitation applied shall not exceed –6.48 dBm 50 Ohms (300 mVpp) on each differential port of the receiver. This number is derived from the maximum signal that may be present at a receiver, 600 mVppd (Gen1i, -6.48 dBm 50 Ohms, each differential side).

7.4.14 SSC Profile

Spread Spectrum Clocking is intentional low frequency modulation of the bit clock. The SSC profile is the modulation on the bit clock. To measure the SSC profile, a frequency demodulator and low pass filter are necessary. There are many possible realizations of this, in hardware and software. The low pass filter is necessary to reject undesired post-demodulation frequency components from bit patterns and jitter. To minimize these undesired signals the HFTP bit pattern shall be used. This may be produced using the BIST Activate FIS to invoke the Transmit-Only option. The SSC Profile measurement is also used to determine the Unit Interval values.

The Reference Clock as defined in section 7.3.2 should be used with an additional low pass filter in the phase detector output to measure the SSC profile. The output is DC coupled and should be calibrated with a signal source with sufficient long-term frequency accuracy.

A single shot capture oscilloscope should be used to measure the times of zero crossings (through interpolation) and perform the FM demodulator and low pass filter function. The memory record of the oscilloscope shall be long enough to achieve the low pass filter cutoff frequency. The long term frequency accuracy of the oscilloscope time base should be significantly better than the 350 ppm limit in this specification; oscilloscopes that do not have this frequency accuracy may be calibrated using a separate signal source of sufficient accuracy into a separate channel.

Modulation analysis tools with sufficient bandwidth provide alternative methods of measuring the SSC profile. These exist in some spectrum analyzers, modulation analyzers, or could be implemented as a separate frequency modulation receiver. Calibration is easier when the FM receiver has a DC coupled modulation path.

The low pass filter 3 dB cutoff frequency shall be 60 times the modulation rate. The filter stopband rejection shall be greater or equal to a second order low-pass of 20 dB per decade.

7.4.15 Intra-Pair Skew

Intra-pair skew measurements are important measurements of transmitters and receivers. For transmitters they are a measure of the symmetry of the SATA transmitter silicon (see Table 31).

For receivers, they are a measure of the ability to handle signal degradation due to the interconnect. At a receiver, intra-pair skew adversely affects jitter levels. In a system, intra-pair skew has a direct impact on radiated emission levels. As the measurement values are typically just a few picoseconds, care should be taken to minimize measurement error.

Figure 165 illustrates a test setup for a measurement method using a HBWS and its built-in processing. Each single-ended channel of a transmitter is measured into a Laboratory Load with DC blocks. Use HFTP and MFTP as the test patterns when measuring transmitter skew. A new displayed signal is formed by mathematically changing the polarity (arithmetic sign) and displayed with the original signal. This creates crossover points for each single ended signal, one displayed on the upper and the other on the lower part of the display. The example shown in Figure 168 of a transmitter, Ch5 and Ch6 are the two single-ended signals of the differential pair, the M5-trace is the inverted Ch5 (-Ch5), and M6 is the inverted Ch6 (-Ch6). Vertical cursors are used to measure the time between crossovers as the intra-pair skew.

Receivers shall be tested to show required performance with the RX Differential Skew set to maximum as specified in Table 33. Skew may be created using test cables of differing propagation delay or active control by the data signal source within the Lab-Sourced Signal generator. Receiver skew may be setup at the same time as receiver amplitude as seen in Figure 166. Use the HFTP as the pattern when setting the skew. The skew measurement is performed as described above for the transmitter.

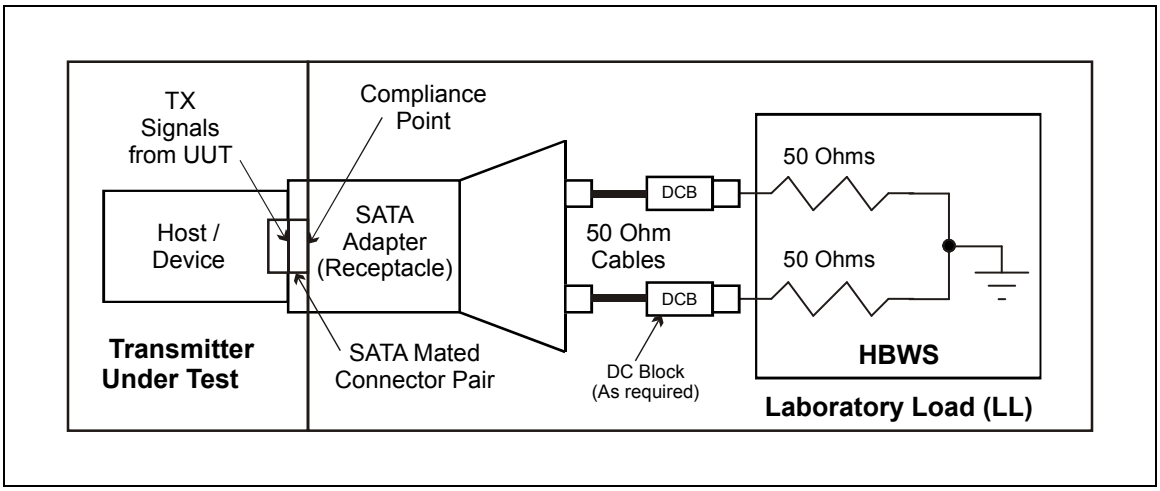


Figure 165 – Intra-Pair Skew Test for a Transmitter

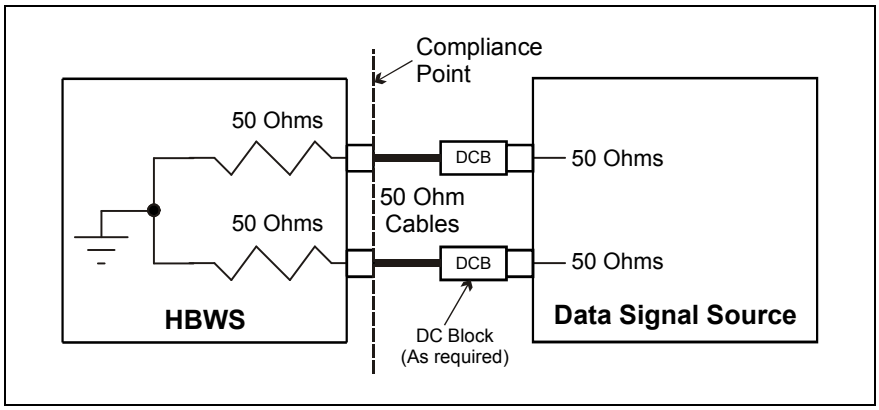


Figure 166 – Receiver Intra-Pair Skew Test—Setting Levels

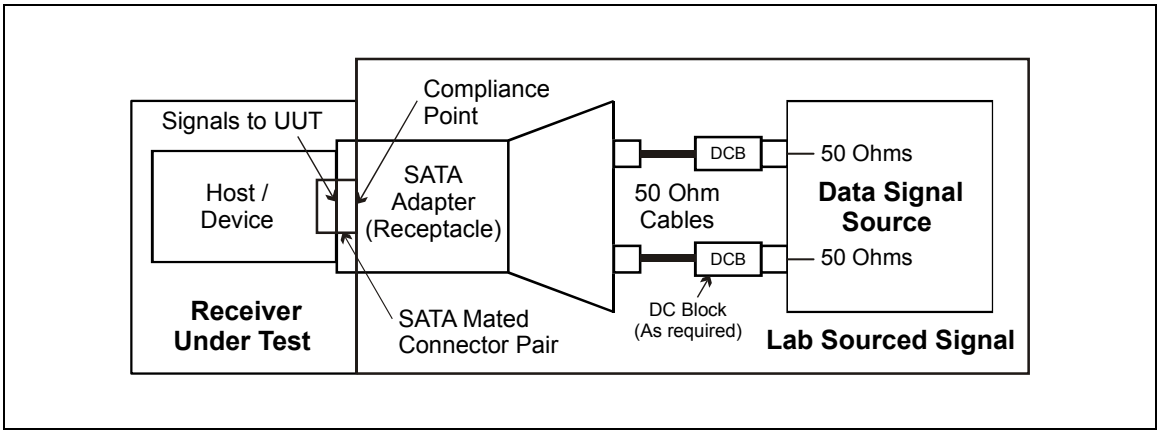


Figure 167 – Receiver Intra-Pair Skew Test

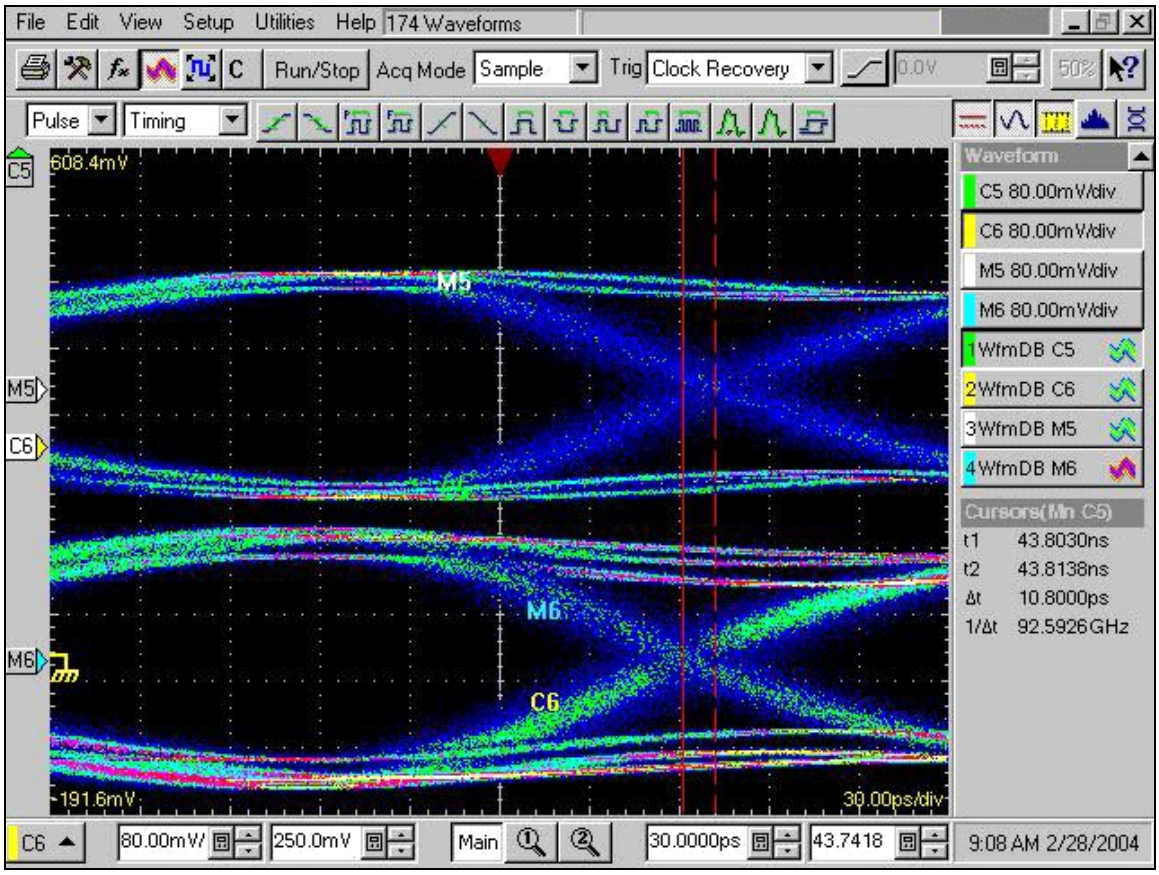


Figure 168 – Example Intra-Pair Skew test for Transmitter (10.8 pS)

7.4.16 Sequencing Transient Voltage

Figure 169 shows the connections to the receiver or transmitter under test. Each RX or TX line is terminated to ground with a minimum impedance of 10 megohms that includes the probe and any external load. The value of the voltage transients during power on or power off sequencing, or power state changes seen at Vp or Vn, shall remain in the voltage range specified.

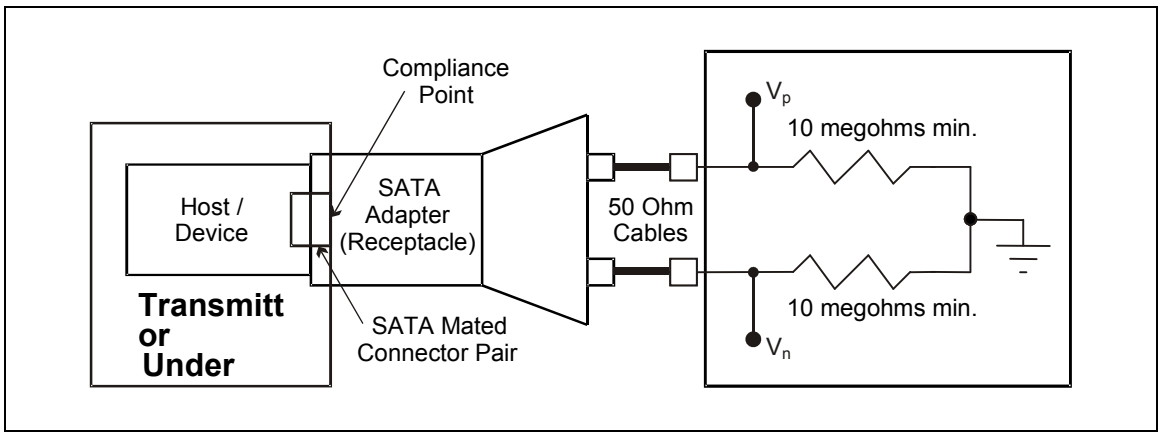


Figure 169 – TX/RX Sequencing Transient Voltage Measurement

7.4.17 AC Coupling Capacitor

This measurement is only applicable to AC coupled transmitters and receivers. The AC coupling capacitor value is not directly observable at the SATA connector.

In order to measure this capacitance, each signal shall be probed on both sides of the AC coupling capacitor. The unit under test is powered off and nothing is plugged into the SATA connector. In the case of coupling within the IC or where there is no access to the signals between the IC and external coupling capacitors, this parameter is not measurable as shown.

Figure 170 shows the connections to each coupling capacitor. Each coupling capacitor shall be lower than the specified maximum.

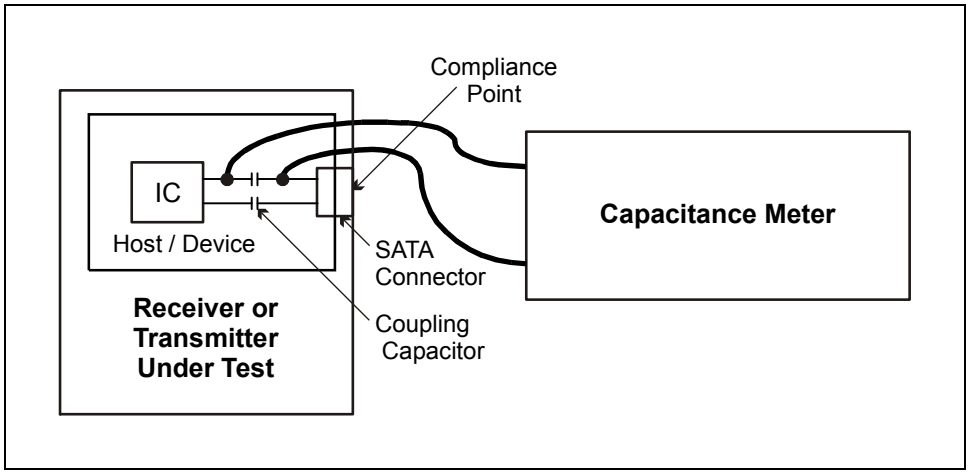


Figure 170 – AC Coupled Capacitance Measurement

7.4.18 TX Amplitude Imbalance

This parameter is a measure of the match in the single-ended amplitudes of the TX+ and TX- signals. The test setup shown Figure 144 shall be used for this measurement. This parameter shall be measured and met with both the HFTP and MFTP patterns. Clock-like patterns are used here to enable the use of standard mode-based amplitude measurements for the sole purpose of determining imbalance. The measurement of differential amplitude uses a different method.

In order to determine the amplitude imbalance, single ended mode high and mode low based amplitudes of both TX+ and TX- over 10 to 20 cycles of the clock-like pattern being used shall be determined. The amplitude imbalance value for that pattern is then determined by the equation:

absolute value(TX+ amplitude - TX- amplitude)/average
where average is (TX+ amplitude + TX- amplitude)/2

The amplitude imbalance value for each pattern shall be less than the maximum listed in Table 31.

7.4.19 TX Rise/Fall Imbalance

This parameter is a measure of the match in the simultaneous single-ended rise/fall or fall/rise times of the Transmitter. The test setup shown in Figure 144 shall be used for this measurement. This parameter shall be measured and met with both the HFTP and MFTP patterns.

In order to determine the imbalance, the single ended 20-80% rise and fall times of both TX+ and TX- shall be determined for a given pattern. Two imbalance values for that pattern are then determined by the two equations:

absolute value(TX+,rise – TX-,fall)/average, where average is (TX+,rise + TX-,fall)/2

absolute value(TX+,fall – TX-,rise)/average, where average is (TX+,fall + TX-,rise)/2

Both values for each pattern shall be less than the maximum listed in Table 31.

7.4.20 TX AC Common Mode Voltage (Gen2i, Gen2m)

This parameter is a measure of common mode noise other than the CM spikes during transitions due to TX+/TX- mismatch and skews which are limited by the rise/fall mismatch and other requirements. Measurement of this parameter is achieved by transmitting through a mated connector into a Lab Load as shown in Figure 144. The transmitter shall use an MFTP (mid-frequency test pattern). The measurement instrument may be a HBWS or other instrument with analog bandwidth of at least 3 * bitrate / 2.

Separate channels shall be used for TX+ and TX- and the common mode is (TX+ + TX-) / 2. This raw common mode shall be filtered with a first order filter having a cutoff equal to the bitrate / 2 to remove the noise contribution from the edge mismatches. The peak-to-peak voltage of the filter output is the AC Common Mode Voltage and shall remain below the specified limit.

7.4.21 Tx AC Common Mode Voltage (Gen3i)

A method to measure the common mode voltage is attaching a metrology-grade power combiner between the Tx+ and Tx- outputs of the transmitter; at the Device or Host transmit connector. Both outputs shall be joined to the combiner with phase matched cables having <= +/- 3 ps of mismatch, diminishing phase distortion during the measurement. The power combiner's output is connected to a spectrum analyzer having sufficient bandwidth to measure the fundamental and 2nd harmonic frequencies of the data speed. (Example: 6.0 Gbps, fundamental = 3 GHz.)
The measurement shall be made using a 1 MHz resolution bandwidth. The transmitter shall output the HFTP (D10.2) pattern during the test.

Equivalent methods may be used.

The dBmV level is understood to be: 0 dBmV = 1mV into 50 ohms.

7.4.22 OOB Common Mode Delta

This parameter is a measure of the offset between the common mode voltage of idle times during OOB generation and the common mode voltage during the OOB bursts. The test setup shown in Figure 144 shall be used for this measurement. A HBWS or single-shot scope may be used for this measurement, the DUT shall be configured to send an OOB sequence or multiple OOB sequences, and the instrument shall be configured so that at least 40 Gen 1 UI worth of idle time before the first OOB burst in a sequence and at least 40 Gen 1 UI worth of burst activity in the first OOB burst of a sequence are observed.

The common mode signal is (TX+ + TX-)/2 and the common mode voltage during idle for this parameter is determined by averaging the common mode voltage of a 40 Gen1 UI span of idle time within the last 60 Gen1 UI worth of time prior to the first OOB burst in a sequence. The average common mode voltage during active time for this parameter is determined by averaging the common mode voltage of a 40 Gen1 UI span of time within the first 60 Gen1 UI of the first burst in a sequence. The reason that the active span is taken within the first 60 Gen1 UI of the

first burst in a sequence is to minimize the affect of AC coupling RC time constant on the resulting common mode offset if one exists.

7.4.23 OOB Differential Delta

This parameter is a measure of the offset between the differential voltage of idle times during OOB generation and the average differential voltage during the OOB bursts. The test setup shown in Figure 144 shall be used for this measurement. A HBWS or single-shot scope may be used for this measurement, the DUT shall be configured to send an OOB sequence or multiple OOB sequences, and the instrument shall be configured so that at least 40 Gen 1 UI worth of idle time before the first OOB burst in a sequence and at least 40 Gen 1 UI worth of burst activity in the first OOB burst of a sequence are observed.

The differential signal is TX+ - TX- and the differential voltage during idle for this parameter is determined by averaging the differential voltage of a 40 Gen1 UI span of idle time within the last 60 Gen1 UI worth of time prior to the first OOB burst in a sequence. The average differential voltage during active time for this parameter is determined by averaging the differential voltage of a 40 Gen1 UI span of time within the first 60 Gen1 UI of the first burst in a sequence. The use of a span of 40 Gen1 UI ensures that no matter what the starting time within the burst, the signal is DC balanced and the average represents the differential mean. The reason that the active span is taken within the first 60 Gen1 UI of the first burst in a sequence is to minimize the affect of AC coupling RC time constant on the resulting differential offset if one exists.

7.4.24 Squelch Detector Tests

The squelch detector is an essential function in receiving OOB signaling. There are two conditions to test: when above the maximum threshold the detector shall detect, and when below the minimum threshold the detector shall not detect. Figure 171 shows the test setup to set the proper level of the OOB signal. Note the same method is used to calibrate the Lab-Sourced signal amplitude as in section 7.4.6. To ensure the proper detection, multiple tests shall be done and the statistics of the results presented to show compliance.

Note: the pattern content in the OOB may affect the detection.

The timing of the gaps in the OOB bursts shall be varied to ensure compliance to the OOB timing specification (see Table 34).

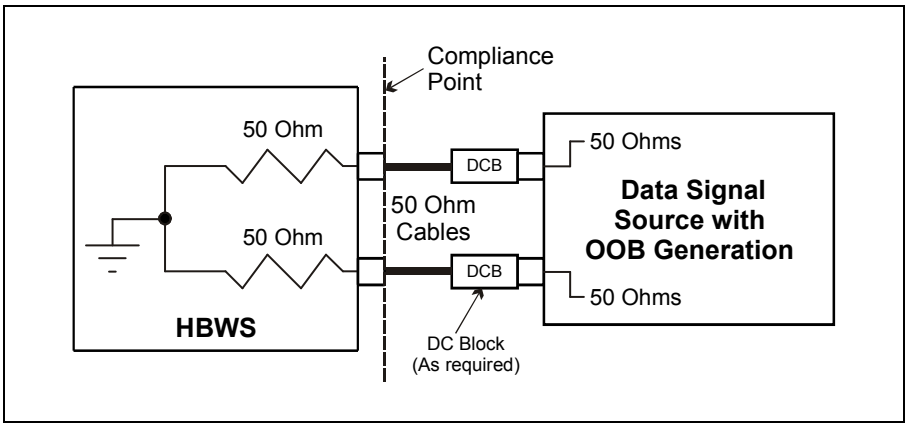


Figure 171 – Squelch Detector Threshold Test—Setting Levels

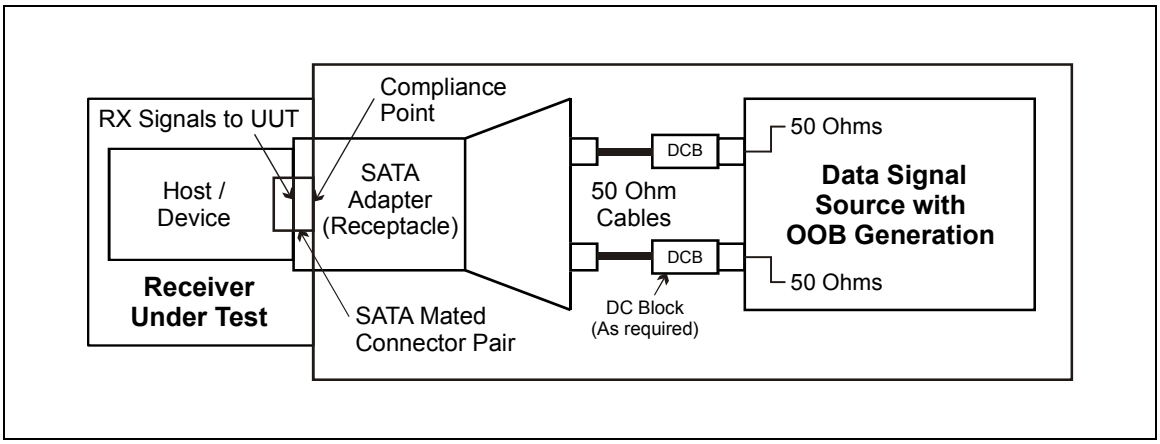


Figure 172 – Squelch Detector Threshold Test

7.4.25 OOB Signaling Tests

OOB signaling is used to signal specific actions during conditions where the receiving interface is in an active mode, a low interface power state, or a test mode.

This section specifies the set of test requirements to ensure that the OOB detector circuits comply with the OOB signaling sequences under various conditions.

7.4.25.1 Power-On Sequence

7.4.25.1.1 Calibration

When the host controller performs impedance calibration, it shall adjust its own impedance such that the electrical requirements of section 7.2 are satisfied.

7.4.25.1.2 Speed Negotiation

Speed negotiation and transition to lower serial interface data rates shall be implemented for higher data speed compatible interfaces, negotiating and transitioning down to lower data speed, as required. There is no requirement for speed negotiation and transition to lower speeds than Gen1.

7.4.25.1.3 Interface Power Management Sequences

7.4.25.1.4 Partial

The interface shall detect the OOB signaling sequence COMWAKE and COMRESET when in the Partial Interface power management state.

While in the Partial state, the interface shall be subjected to the low-transition density bit pattern (LTDP) sequences of section 7.2.4.3; the interface shall remain in the Partial state until receipt of a valid COMWAKE (or COMRESET) OOB signaling sequence.

Power dissipation in this Partial state shall be measured or calculated to be less than the Phy Active state, but more than the Slumber state defined in section 8.1.

The requirement for a "not-to-exceed" power dissipation limit in the Partial interface power management state is classified as vendor specific, and should be documented as part of the implementation performance specifications.

7.4.25.1.5 Slumber

The interface shall detect the OOB signaling sequence COMWAKE and COMRESET when in the Slumber Interface power management state.

While in the Slumber state, the interface shall be subjected to the low-transition density bit pattern (LTDP) sequences of section 7.2.4.3; the interface shall remain in the Slumber state until receipt of a valid COMWAKE (or COMRESET) OOB signaling sequence.

Power dissipation in this Slumber state shall be measured or calculated to be less than the Phy Ready state, and less than the Partial state defined in section 8.1.

The requirement for a "not-to-exceed" power dissipation limit in the Slumber interface power management state is classified as vendor specific, and should be documented as part of the implementation performance specifications.

7.4.26 TDR Differential Impedance (Gen1i / Gen1m)

This specification describes transmitter output impedance and receiver input impedance in terms of both the peak value of a reflection given an incident step of known risetime and also in terms of return loss. The return loss measurement shall be sufficient to verify compliance with Gen1 and Gen2 requirements. In order to ensure replacement of the test outlined below does not invalidate Gen1 designs passing the TDR differential impedance, that method shall be sufficient to verify compliance with Gen 1 requirements. Verification of compliance by both methods shall not be required.

To achieve consistent measurements it is important to control the test conditions at the compliance point. These conditions include the signal launch (see section 7.2.6), the source match looking back into the test setup and TDR, the risetime and shape of the TDR edge, and the attenuation loss on the reflection return path to the TDR. There are various methods to control and remove the test setup effects.

When measuring output impedance of transmitters the operating condition shall be during transmission of MFTP. This is to assure the measurement is performed during a mode of operation that represents normal operation. The amplitude of a TDR pulse or excitation applied to an active transmitter shall not exceed 139 mVpp (-13.2 dBm 50 ohms) single ended. This number is derived from the maximum reflected signal that may be present at a transmitter. A maximum transmitted signal of 700 mVppd reflecting off a receiver with a differential return loss of 8 dB and direct attach.

Source match is a constant 100 Ohms differential impedance level on the TDR trace preceding the compliance point. This may be achieved by impedance controlled test setup or a calibration procedure.

Figure 173 shows the setup to set the risetime at the device under test. The risetime shall be set accurately at the compliance point. The shape of the TDR edge at the compliance point is affected by the edge shape of the TDR generator, the attenuation loss in the test setup, and averaging done on the received signal at the TDR.

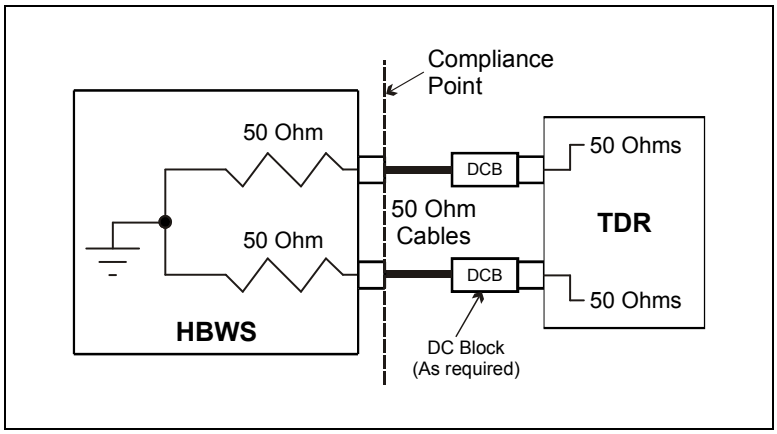


Figure 173 – TDR Impedance Test—Setting Risetime

Since the SATA adapter is not included when setting risetime, good matching and low loss are necessary in the adapter to minimize errors in the measured TDR impedance.

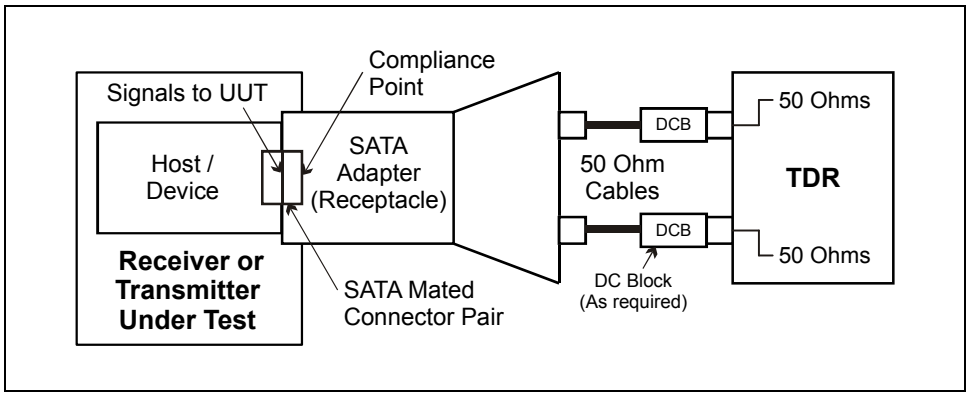


Figure 174 – TDR Impedance Test

7.4.27 TDR Single-Ended Impedance (Gen1i / Gen1m)

This specification describes transmitter single-ended output impedance and receiver single-ended input impedance in terms of the peak value of a reflection given an incident step of known risetime. To achieve consistent measurements it is important to control the test conditions at the compliance point. These conditions include the signal launch (see section 7.2.6), the source match looking back into the test setup and TDR, the risetime and shape of the TDR edge, and the attenuation loss on the reflection return path to the TDR. There are various methods to control and remove the test setup effects.

Source match is a constant 50 Ohms single-ended impedance level on the TDR trace preceding the compliance point. This may be achieved by impedance controlled test setup or a calibration procedure.

Figure 175 shows the setup to set the risetime at the device under test. The risetime shall be set accurately at the compliance point. The shape of the TDR edge at the compliance point is affected by the edge shape of the TDR generator, the attenuation loss in the test setup, and averaging done on the received signal at the TDR.

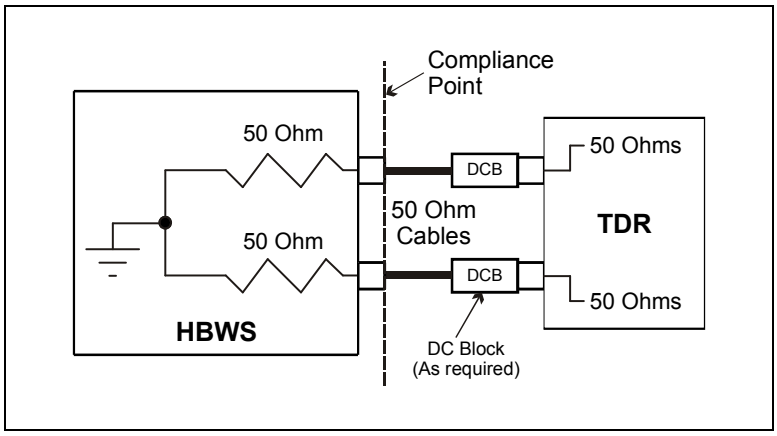


Figure 175 – TDR Impedance Test—Setting Risetime

Since the SATA adapter is not included when setting risetime, good matching and low loss are necessary in the adapter to minimize errors in the measured TDR impedance.

Figure 178 shows the connections to the receiver or transmitter under test. For single-ended measurements, the TDR shall be set to produce simultaneous positive pulses on both signals of the pair. Single-ended impedance is the resulting (even mode) impedance of each signal observed independently. Both signals shall meet the single-ended impedance requirement.

7.4.28 DC Coupled Common Mode Voltage (Gen1i / Gen1m)

This measurement is only applicable to DC coupled transmitters and receivers. The following measurement on an AC coupled signal or with AC coupled probing results in a value near or at 0 V. Figure 176 shows the connections to the receiver or transmitter under test. Each RX or TX line is terminated to ground with a minimum impedance of 10 megohms that includes the probe and any external load. The common mode is (Vp + Vn)/2 and this term shall be in the range specified.

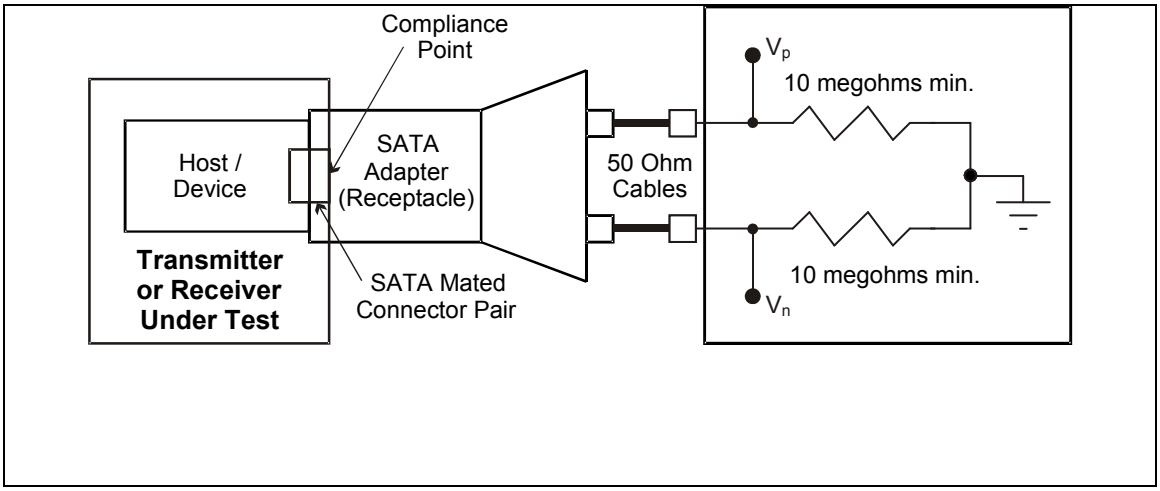


Figure 176 – DC Coupled Common Mode Voltage Measurement

7.4.29 AC Coupled Common Mode Voltage (Gen1i / Gen1m)

This measurement is only applicable to AC coupled transmitters and receivers. The AC coupled common mode voltage is not directly observable at the SATA connector.

In order to measure this voltage, each RX or TX signal shall be probed between the IC and AC coupling capacitor. In the case of coupling within the IC or where there is no access to the signals between the IC and external coupling capacitors, it is not measurable.

Figure 177 shows the connections to the receiver or transmitter under test. Each RX or TX line is terminated to ground with a minimum impedance of 10 megohms that includes the probe and any external load. The common mode is (Vp + Vn)/2 and this term shall be in the range specified.

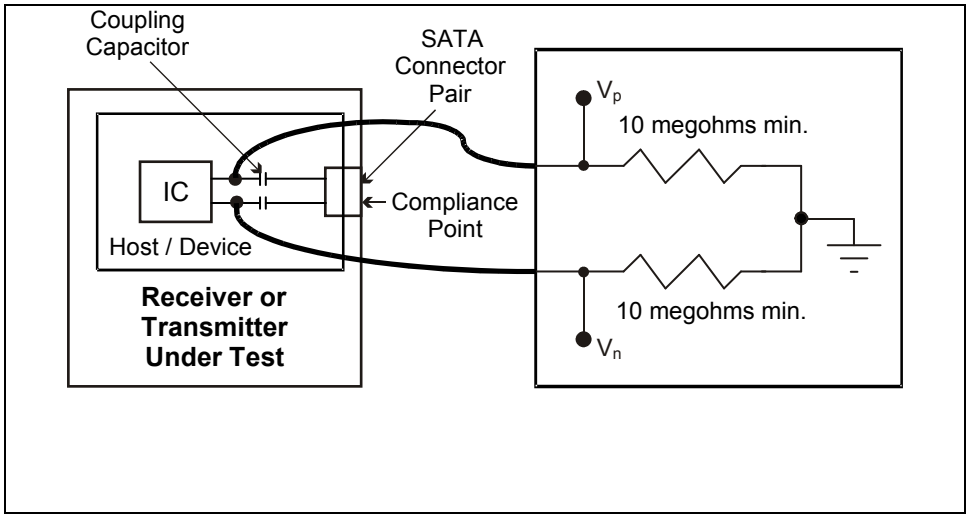


Figure 177 – AC Coupled Common Mode Voltage Measurement

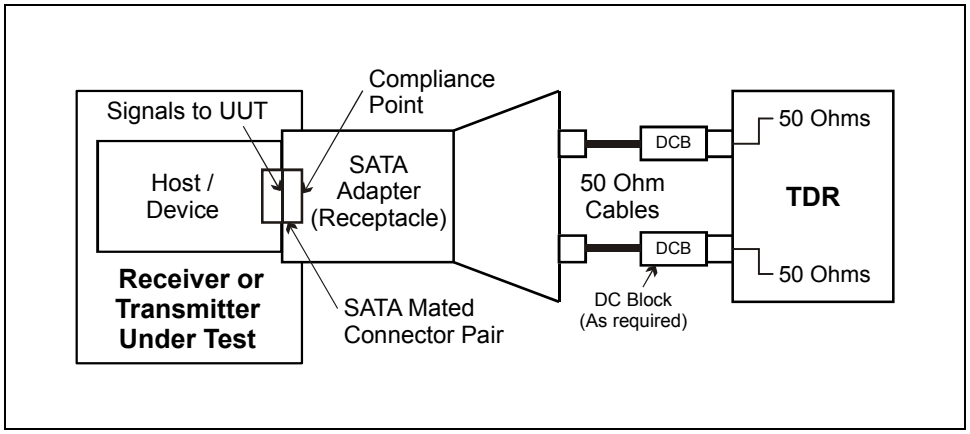


Figure 178 – TDR Impedance Test

7.4.30 Sequencing Transient Voltage - Laboratory Load (Gen3i)

Error! Reference source not found. shows the connections to the receiver or transmitter under test. Each RX or TX line is terminated to ground using a laboratory load (see 7.2.2.4 for lab load definition). The value of the voltage transients during power on or power off sequencing, or power state changes seen at Vp or Vn, shall remain in the voltage range specified.

In some laboratory load configurations additional DC blocking components are added. For this measurement there shall not be any additional DC blocking components added in the laboratory load.

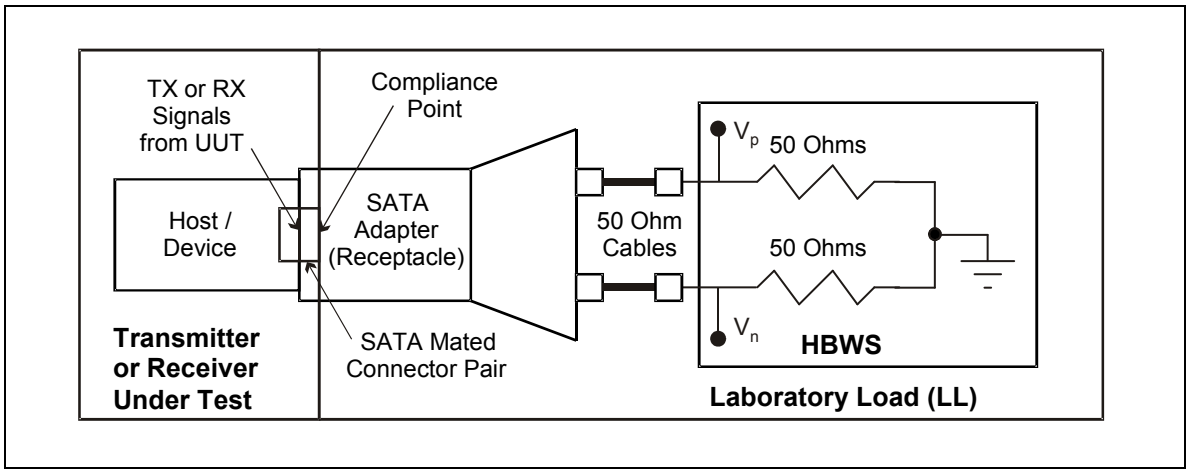


Figure 179 - Sequencing Transient Voltage Laboratory Load

7.5 Interface States

7.5.1 Out Of Band Signaling

There shall be three Out Of Band (OOB) signals used/detected by the Phy: COMRESET, COMINIT, and COMWAKE. COMINIT, COMRESET and COMWAKE OOB signaling shall be achieved by transmission of either a burst of four Gen1 $ALIGN_P$ primitives or a burst composed of four Gen1 Dwords with each Dword composed of four D24.3 characters, each burst having a duration of 160 UI_{OOB}. Each burst is followed by idle periods (at common-mode levels), having durations as depicted in Figure 180 and Table 49.

Previous versions of Serial ATA allow only for the ALIGN sequence as legitimate OOB signal content. The alternate OOB sequence defined in this section has different characteristics than the ALIGN sequence in both the time and frequency domains. The use of alternate OOB signal content may lead to backwards incompatibility with Gen1 Phys designed to previous Serial ATA specification versions. Interoperability issues with Gen1 Phys designed to the earlier SATA specification arising from the use of alternate OOB signal content are the sole responsibility of the Phy transmitting this alternate content.

During OOB signaling transmissions, the differential and common mode levels of the signal lines shall comply with the same electrical specifications as for normal data transmission, specified in section 7.2. In Figure 180 below, COMRESET, COMINIT, and COMWAKE are shown. OOB signals are 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.

Even though they are transmitted with apparent Gen1 timings, the OOB burst transmissions may be transmitted using Gen2 rise / fall times.

Any spacing less than or greater than the COMWAKE detector off threshold in Table **34** shall negate the COMWAKE detector output. The COMWAKE OOB signaling is used to bring the Phy out of a power-down state (Partial or Slumber) as described in section 8.4.3.2. The interface shall be held inactive for at least the maximum COMWAKE detector off threshold in Table **34** after the last burst to ensure far-end detector detects the negation properly. The device shall hold the interface inactive no more than the maximum COMWAKE detector off threshold plus two Gen1 Dwords (approximately 228.3 ns) at the end of a COMWAKE to prevent susceptibility to crosstalk.

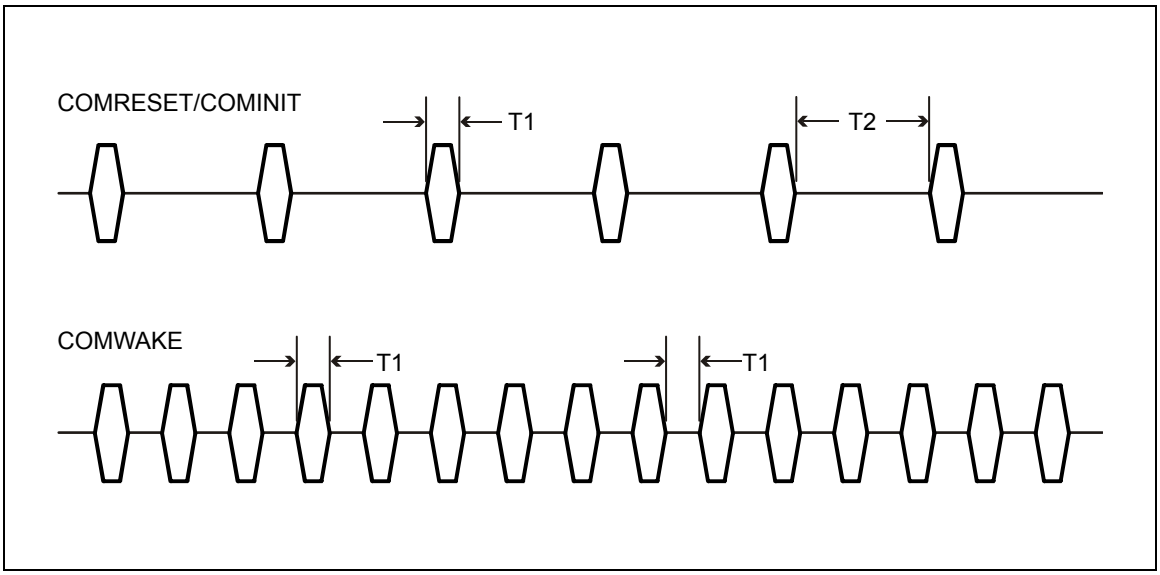


Figure 180 – OOB Signals

Table 49 – OOB Signal Times

Time	Value
T1	160 UI_{oob} (106.7 ns nominal)
T2	480 UI_{oob} (320 ns nominal)

7.5.1.1 Idle Bus Status

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

Common-mode transients, shall not exceed the maximum amplitude levels (Vcm,ac) cited in section 7.2, and shall settle to within 25 mV of the previous state common mode voltage within Tsettle,cm, cited in section 7.2. The following figure shows several transmitter examples, and how the transition to and from the idle state may be implemented.

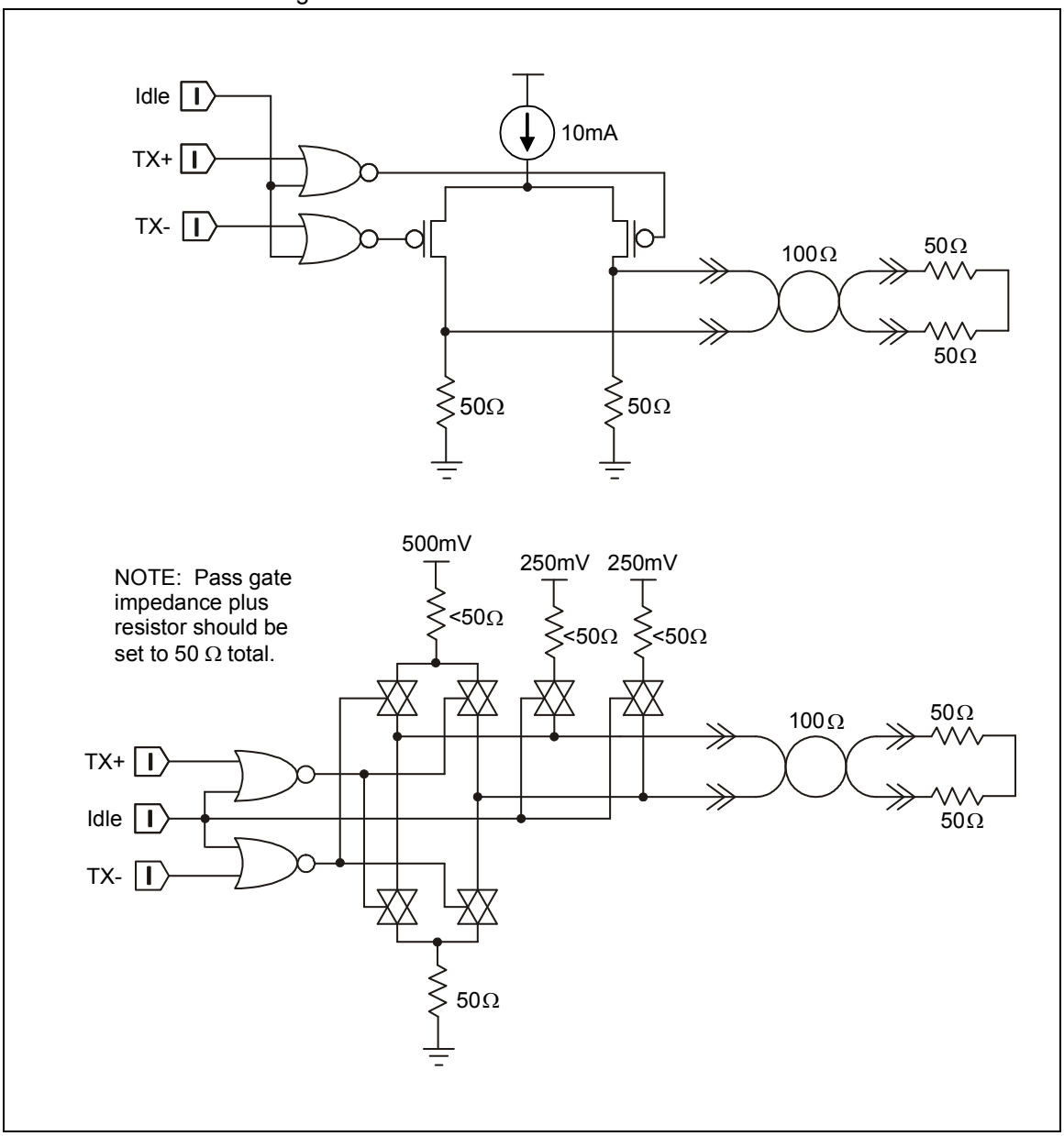


Figure 181 – Transmitter Examples

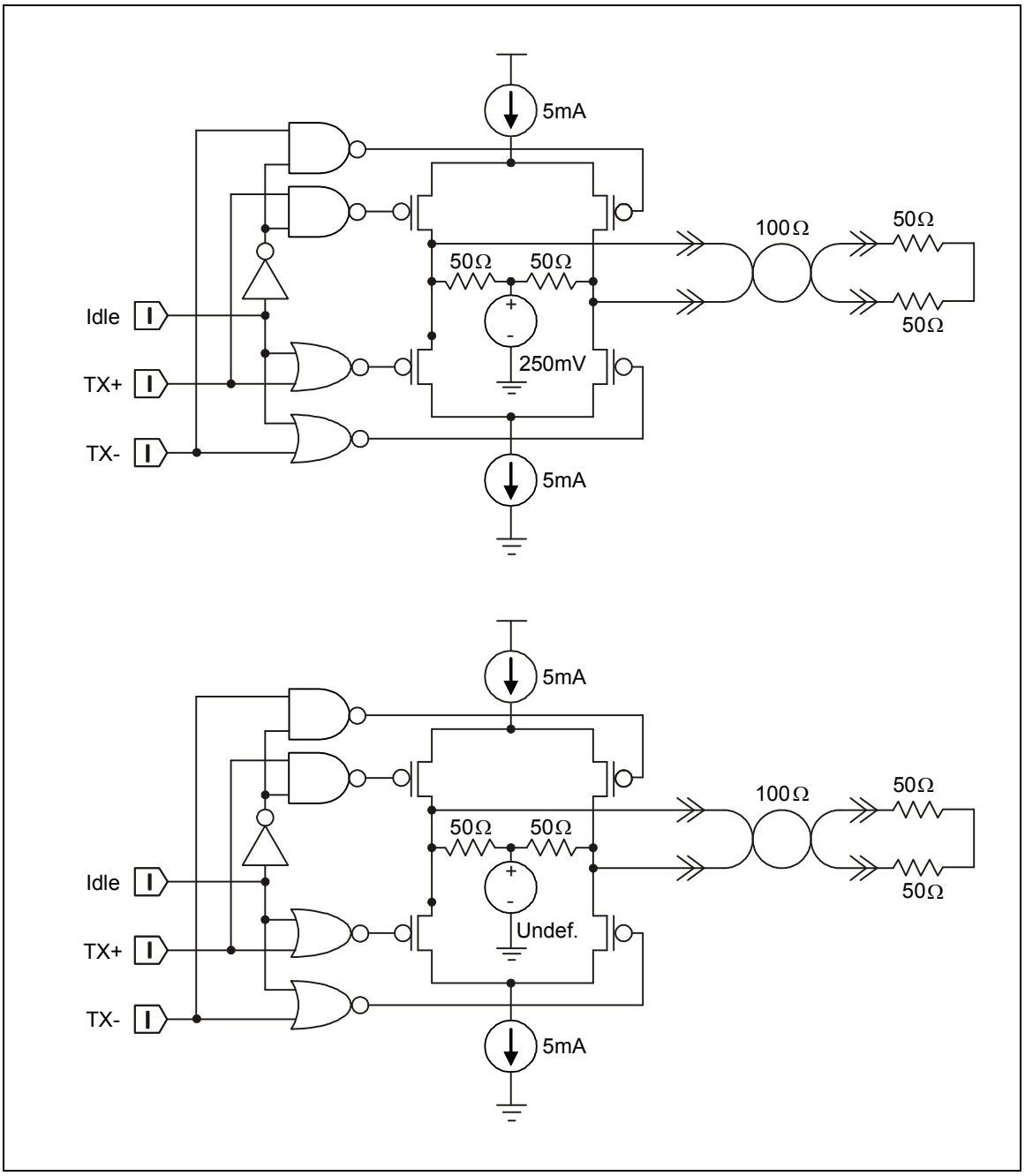


Figure 182 – Transmitter Examples (Concluded)

7.5.1.2 COMRESET

COMRESET always originates from the host controller, and forces a hardware reset in the device. It is indicated by transmitting bursts of data separated by an idle bus condition.

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

1) Sustained/continued uninterrupted as long as the system hard reset is asserted, or

2) Started during the system hardware reset and ended some time after the negation of system hardware reset, or
3) Transmitted immediately following the negation of the system hardware reset signal.

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

Each burst shall be 160 Gen1 UI's long (106.7 ns) and each inter-burst idle state shall be 480 Gen1 UI's long (320 ns). A COMRESET detector looksfor four consecutive bursts with 320 ns spacing (nominal).

Any spacing less than 175 ns or greater than 525 ns shall invalidate the COMRESET detector output. The COMRESET interface signal to the Phy layer shall initiate the Reset sequence shown in Figure 183 below. The interface shall be held inactive for at least 525 ns after the last burst to ensure far-end detector detects the negation properly.

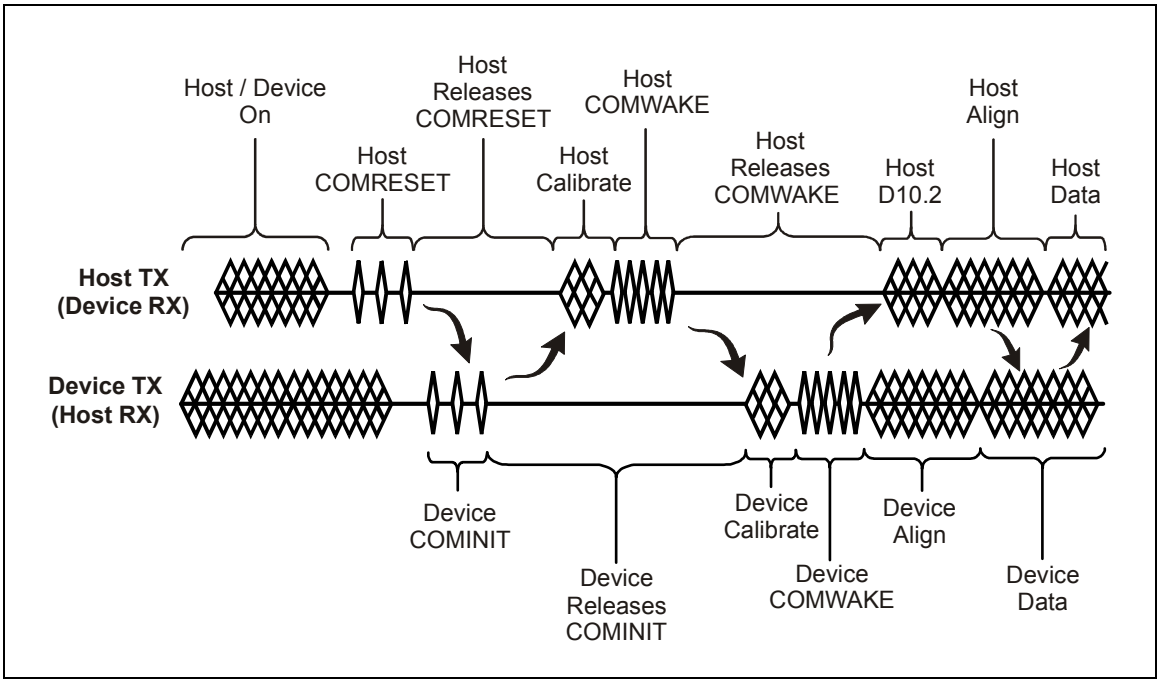


Figure 183 – COMRESET Sequence

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 issue COMRESET
3. Host releases COMRESET. Once the condition causing the COMRESET is released, the host releases the COMRESET signal and puts the bus in a quiescent condition.
4. Device issues COMINIT – When the device detects the release of COMRESET, it responds with a COMINIT. This is also the entry point if the device is late starting. The device may initiate communications at any time by issuing a COMINIT.
5. Host calibrates and issues a COMWAKE.
6. Device responds – The device detects the COMWAKE sequence on its RX pair and calibrates its transmitter (optional). Following calibration the device sends a six burst COMWAKE sequence and then sends a continuous stream of the ALIGN sequence starting at the device's highest supported speed. After $ALIGN_P$ Dwords have been sent

for 54.6us (2048 nominal Gen1 Dword times) without a response from the host as determined by detection of $ALIGN_P$ 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 $ALIGN_P$ Dwords at that speed for 54.6 us (2048 nominal Gen1 Dword times.) This step is repeated for as many slower speeds as are supported. Once the lowest speed has been reached without response from the host, the device enters an error state.

7. Host locks – after detecting the COMWAKE, the host starts transmitting D10.2 characters (see 7.6) at its lowest supported speed. Meanwhile, the host receiver locks to the ALIGN sequence and, when ready, returns the ALIGN sequence to the device at the same speed as received. A host shall be designed such that it acquires lock in 54.6us (2048 nominal Gen1 Dword times) at any given speed. The host should allow for at least 873.8 us (32768 nominal Gen1 Dword times) after detecting the release of COMWAKE to receive the first $ALIGN_P$. This ensures interoperability with multi-generational and synchronous designs. If no $ALIGN_P$ is received within 873.8 us (32768 nominal Gen1 Dword times) the host restarts the power-on sequence – repeating indefinitely until told to stop by the Application layer.
8. Device locks – the device locks to the ALIGN sequence and, when ready, sends $SYNC_P$ indicating it is ready to start normal operation.
9. Upon receipt of three back-to-back non-$ALIGN_P$ primitives, the communication link is established and normal operation may begin.

7.5.1.3 COMINIT

COMINIT always originates from the device and requests a communication initialization. It is electrically identical to the COMRESET signal except that it originates from the device and is sent to the host. It is used by the device to request a reset from the host in accordance to the sequence shown in Figure 184, below.

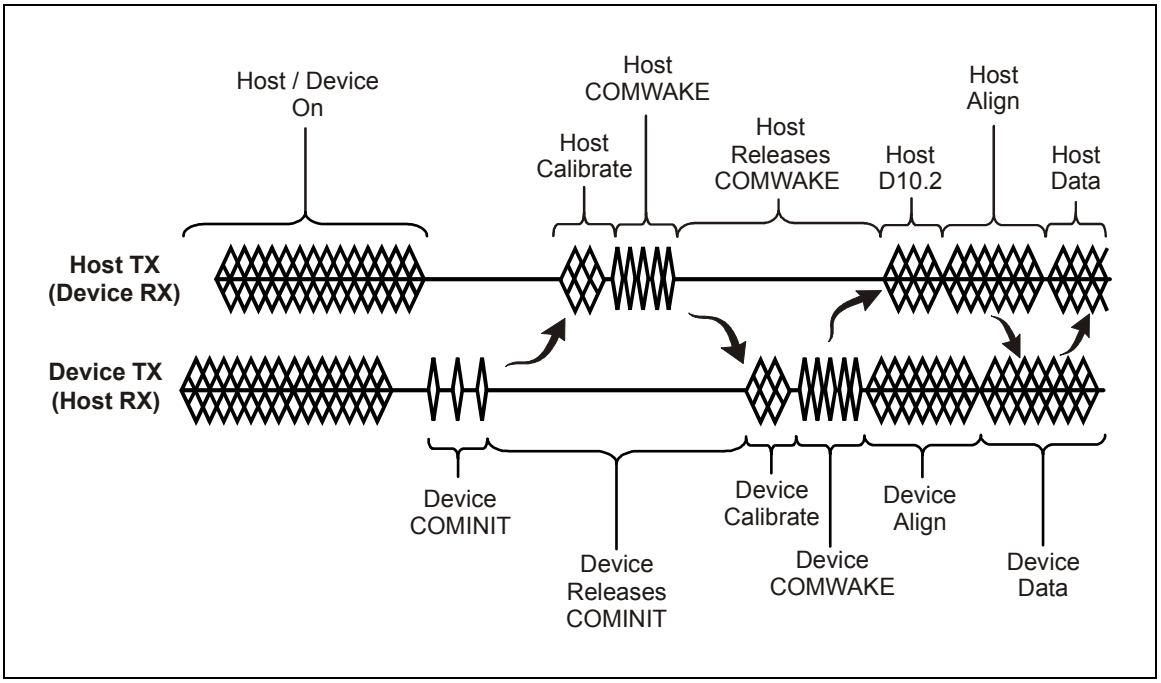


Figure 184 – COMINIT Sequence

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 issues a COMINIT
3. Host calibrates and issues a COMWAKE.
4. Device responds – The device detects the COMWAKE sequence on its RX pair and calibrates its transmitter (optional). Following calibration the device sends a six burst COMWAKE sequence and then sends a continuous stream of the ALIGN sequence starting at the device's highest supported speed. After $ALIGN_P$ Dwords have been sent for 54.6 us (2048 nominal Gen1 Dword times) without a response from the host as determined by detection of $ALIGN_P$ 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 $ALIGN_P$ Dwords at that speed for 54.6 us (2048 nominal Gen1 Dword times.) This step is repeated for as many slower speeds as are supported. Once the lowest speed has been reached without response from the host, the device enters an error state.
5. Host locks – after detecting the COMWAKE, the host starts transmitting D10.2 characters (see section 7.6) at its lowest supported speed. Meanwhile, the host receiver locks to the ALIGN sequence and, when ready, returns the ALIGN sequence to the device at the same speed as received. A host shall be designed such that it acquires lock in 54.6 us (2048 nominal Gen1 Dword times) at any given speed. The host should allow for at least 873.8 us (32768 nominal Gen1 Dword times) after detecting the release of COMWAKE to receive the first $ALIGN_P$. This ensures interoperability with multi-generational and synchronous designs. If no $ALIGN_P$ is received within 873.8 us (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 sequence and, when ready, sends $SYNC_P$ indicating it is ready to start normal operation.
7. Upon receipt of three back-to-back non-$ALIGN_P$ primitives, the communication link is established and normal operation may begin.

7.5.1.4 COMWAKE

COMWAKE may originate from either the host controller or the device. It is signaled by transmitting six bursts of data separated by an idle bus condition.

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

Each burst shall be 160 Gen1 UI long and each inter-burst idle state shall be 160 Gen1 UI long. A COMWAKE detector looks for four consecutive burst with a 106.7 ns spacing (nominal).

Any spacing less than 35 ns or greater than 175 ns shall invalidate the COMWAKE detector output. The COMWAKE OOB signaling is used to bring the Phy out of a power-down state (Partial or Slumber) as described in section 8.1. The interface shall be held inactive for at least 175 ns after the last burst to ensure far-end detector detects the negation properly. The device shall hold the interface inactive no more then 228.3 ns (175 ns + two Gen1 Dwords) at the end of a COMWAKE to prevent susceptibility to crosstalk.

7.5.1.5 Design Example (Informative)

This section includes 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 specification.

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.

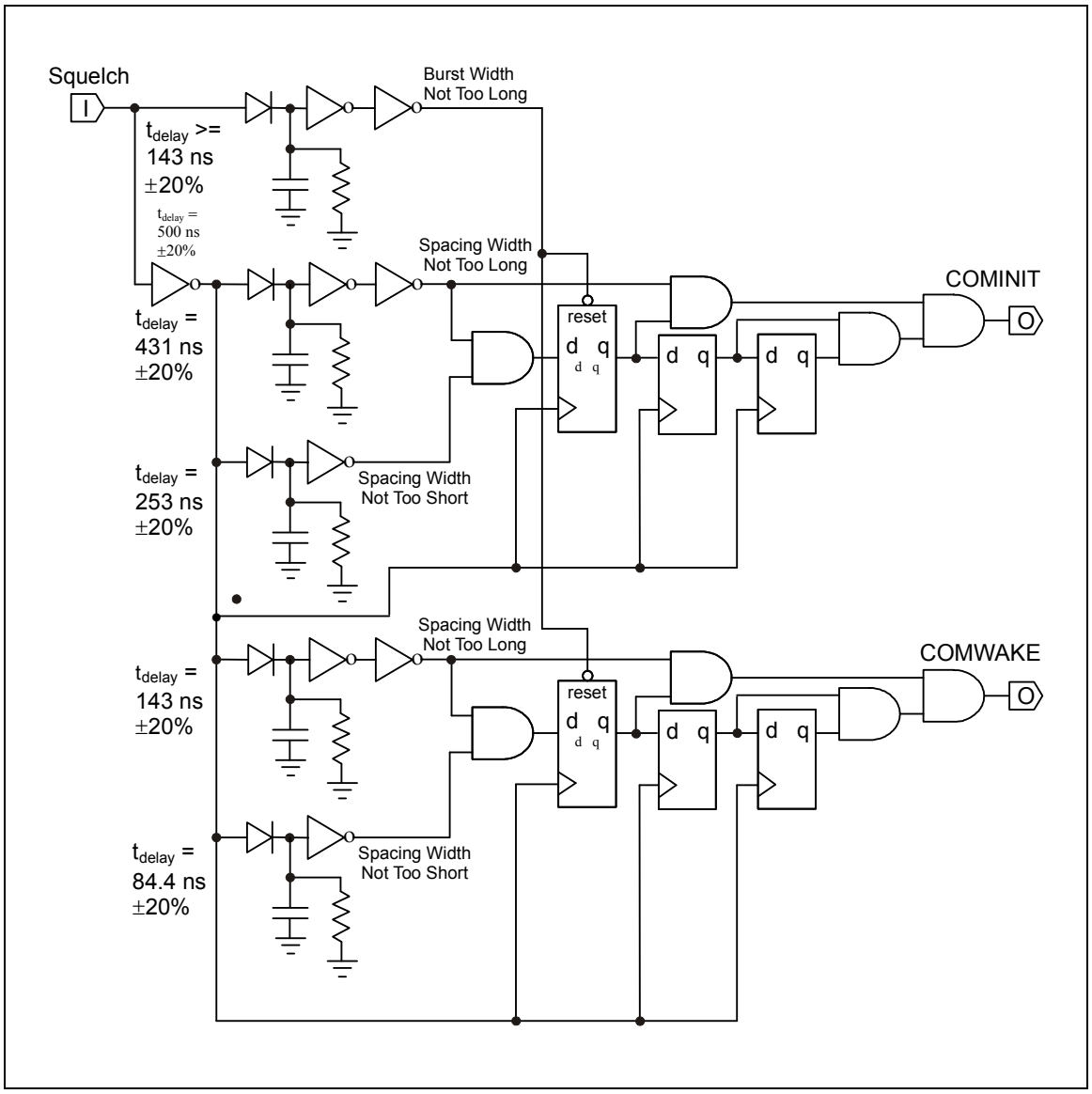


Figure 185 – OOB Signal Detector

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 shall 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 186, like the OOB Signal Detector figure shown in Figure 185, 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 section 7.2, for the worst case noise environment (common-mode) conditions.

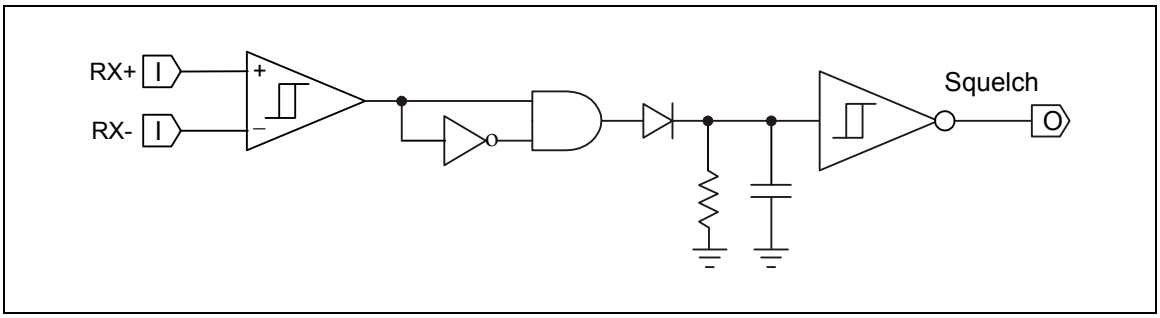


Figure 186 – Squelch Detector

7.5.2 Idle Bus Condition

During power management states (Partial and Slumber), the electrical interface shall maintain the proper common-mode levels, as cited in section 7.2, with zero differential on both signal pairs (all four conductors at 250 mV) for all interface scenarios, except for the case where both, the device and the host-controller, are AC-coupled and the conductor pairs are allowed to float.

All transmitter designs shall ensure that transition to and from the idle bus condition does not result in a disturbance in the differential baseline on the conductors. To accomplish this, an AC-coupled transmitter shall hold its outputs at zero differential with the same common-mode level as normal operation when in the Partial power management mode. When operating in the Slumber power management mode, the common mode level of the AC coupled transmitter is allowed to float (while maintaining zero differential) as long as it remains within the limits cited in section 7.2.

It is unacceptable to hold the TX outputs at a logical zero or one state during the idle bus condition since this results in a baseline shift when communications are resumed.

7.6 Elasticity Buffer Management

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

The Serial ATA specification is written to support both tracking and non-tracking architectures. A non-tracking architecture shall contain the elasticity buffer within the Phy layer.

Note that since this elasticity buffer is designed to have finite length, there needs to be a mechanism at the Phy 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 Phy layer protocol not only 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 $ALIGN_P$ 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$_P$ primitive sequence, followed by at most 254 non-ALIGN$_P$ Dwords. The cycle repeats starting with another dual-consecutive ALIGN$_P$ primitive sequence. The Link may issue more than one dual ALIGN$_P$ primitive sequence but shall not send an unpaired ALIGN$_P$ primitive (i.e. ALIGN$_P$ primitives are always sent in pairs) except as noted for retimed loopback.

ALIGN$_P$ consists of the following four characters:

(rd+)	(rd-)	
1100000101	0011111010	Align1 (K28.5)
0101010101	0101010101	Align2 (D10.2)
0101010101	0101010101	Align3 (D10.2)
1101100011	0010011100	Align4 (D27.3)

8 OOB and Phy Power States

8.1 Interface Power States

Serial ATA interface power states are controlled by the device and host controller. The interface power states are defined as described in Table 50.

Table 50 – Interface Power States

State	Description
PHYRDY	The Phy logic and main PLL are both on and active. The interface is synchronized and capable of receiving and sending data.
Partial	The Phy logic is powered, but is in a reduced power state. Both signal lines on the interface are at a neutral logic state (common mode voltage). The exit latency from this state shall be no longer than 10 us unless when Automatic Partial to Slumber transitions is supported. When Automatic Partial to Slumber Transitions are enabled the exit latency from this state shall be no longer than the maximum Slumber exit latency.
Slumber	The Phy logic is powered but is in a reduced power state. 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 Table 29 entry AC coupled common mode voltage. The exit latency from this state shall be no longer than 10 ms.

8.2 Asynchronous Signal Recovery (Optional)

Phys may support asynchronous signal recovery for those applications where the usage model of device insertion into a receptacle (power applied at time of insertion) does not apply.

When signal is lost, both the host and the device may attempt to recover the signal. A host or device shall determine loss of signal as represented by a transition from PHYRDY to PHYRDYn, which is associated with entry into states LSI: NoCommErr or LS2:NoComm within the Link layer. Note that negation of PHYRDY does not always constitute a loss of signal (e.g., Phy transition to Partial/Slumber). Recovery of the signal is associated with exit from state LS2:NoComm. If the device attempts to recover the signal before the host by issuing a COMINIT, the device shall return its signature following completion of the OOB sequence which included COMINIT. If a host supports asynchronous signal recovery, when the host receives an unsolicited COMINIT, the host shall issue a COMRESET to the device. An unsolicited COMINIT is a COMINIT that was not in response to a preceding COMRESET, as defined by the host not being in the HP2:HR_AwaitCOMINIT state when the COMINIT signal is first received.

When a COMRESET is sent to the device in response to an unsolicited COMINIT, the host shall set the Status register to 7Fh and shall set all other Shadow Command Block Registers to FFh. When the COMINIT is received in response to the COMRESET which is associated with entry into state HP2B:HR_AwaitNoCOMINIT, the Shadow Status register value shall be updated to either FFh or 80h to reflect that a device is attached.

8.2.1 Unsolicited COMINIT Usage (Informative)

Issuing a COMRESET to the device causes the device to lose software settings, other than the cases where software settings preservation is supported as described in section 13.5. If the COMRESET was due to asynchronous signal recovery and legacy mode (see section 4.1.72) software is in use, software does not replace the lost software settings. Issuing a non-

commanded COMRESET to the device should be minimized in order to ensure robust operation with legacy mode software and avoid inadvertent loss of critical software settings.

The use of unsolicited COMINIT was originally intended to only be used when the signal is lost between host and device. Based on the Host Phy Initialization state machine, the host shall assume that when receiving an unsolicited COMINIT that either a new device was connected or that the cable was unplugged and communication was lost to the device. The proper host response to an unsolicited COMINIT is to issue a COMRESET, putting the device into a known state. The device issuing an unsolicited COMINIT leads to a COMRESET from the host which could change the software settings of the device in such a way that legacy mode software cannot recover. To minimize potential for exposure to such indeterminate behavior, the device should only issue an unsolicited COMINIT when the Phy voltage threshold falls below the minimum value or as a last resort in error recovery.

8.3 OOB and Signature FIS return (Informative)

After an OOB sequence, some devices compliant to older revisions of the Serial ATA specification may send a Register – Device to Host FIS with the device signature only if the device recognized COMRESET during the OOB. To ensure a robust host solution for compatibility with these older devices, the host may ensure at a system power-on event that the device always receives a valid COMRESET after power is determined good at the device. Hot plug aware software shall ensure that the device always receives a COMRESET on a hot plug event.

One mechanism as a host workaround is to implement the following software procedure when determining device presence:

1. Wait for SError.DIAG.X to be set to one.
2. Clear SError.DIAG.X to zero by writing a one to that bit location.
3. Issue a COMRESET to the device (a valid COMINIT was received to set the X bit to one, thus power at the device is known to be good).
4. Wait up to 10 milliseconds for SError.DIAG.X to be set to one.
5. If SError.DIAG.X is not set after 10 milliseconds, go back to step 3 or exit if number of retries is exceeded.
6. At this point, the device is now required to transmit a Register FIS with the device signature.

Other methods for ensuring that the device receives a COMRESET in these conditions are possible.

8.4 Power-On Sequence State Machine

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

In those states where the Phy relies on detection of received $ALIGN_P$ primitives or comma sequences for state transitions, the Phy shall ensure accurate detection of the $ALIGN_P$ primitives at the compatible signaling speed, with adequate implementation safeguards to ensure that there is no misdetection of $ALIGN_P$ in the HP6:HR_AwaitAlign state in light of aliasing effects given the different data rates of $ALIGN_P$ primitives and D10.2's in the incoming data streams.

8.4.1 Host Phy Initialization State Machine

As described in section 7.5.1.3, reception of a COMINIT signal shall cause the host to reinitialize communications with the device. Implementations that do not support asynchronous signal recovery shall unconditionally force the Host Phy Initialization state machine to transition to the HP2B:HR_AwaitNoCOMINIT state when a COMINIT is received regardless of other conditions. Implementations that do support asynchronous signal recovery shall unconditionally force the

Host Phy Initialization state machine to transition to the HP1:HR_Reset state when an unsolicited COMINIT is received regardless of other conditions; if the COMINIT is not unsolicited the implementation shall force the Host Phy Initialization state machine to transition to the HP2B:HR_AwaitNoCOMINIT state regardless of other conditions. Reception of COMINIT is effectively an additional transition into the HP2B:HR_AwaitNoCOMINIT or HP1:HR_Reset state that appears in every Host Phy state. For the sake of brevity, this implied transition has been omitted from all the states.

A state variable called ResumePending is used to track whether the Host Phy has been to a power management state such that re-establishing communications is as a result of a resume from a low power state. If a COMWAKE signal is not received when resuming from a low power state, the Host Phy shall allow the device to retransmit COMWAKE and shall not transmit a COMRESET to the device unless a COMRESET is explicitly triggered from a higher layer. If a COMWAKE signal is not received from the device when resuming from a low power state, the host may retransmit COMWAKE to the device.

Designs that support asynchronous signal recovery have a state variable referred to as RetryInterval that determines the speed at which optional signal recovery polling is attempted. The value for RetryInterval shall be no shorter than 10 ms. Implementations that do not implement optional retry polling may consider the RetryInterval value to be infinite.

Table 51 – State Diagram Host Phy Initialization State Machine

HP1: HR_Reset	Transmit COMRESET[2,3,4]. If asynchronous signal recovery is supported then clear ResumePending to 0.	
1. Power-on reset and explicit reset request negated.	→	HR_AwaitCOMINIT
2. Power-on reset or explicit reset request asserted.	→	HR_Reset

NOTES:
1. This state is entered asynchronously any time in response to power-on reset or an explicit reset request. For hosts supporting asynchronous signal recovery, this state is entered in response to receipt of a COMINIT signal from any state other than the HP2:HR_AwaitCOMINITor the HP2B:HR_AwaitNoCOMINIT state.
2. Shall transmit COMRESET for a minimum of 6 bursts (and a multiple of 6)
3. As described in section 7.5.1.2, COMRESET may be transmitted for the duration of this state, or it may be transmitted starting in this state and cease transmission after departure of this state, or it may be transmitted upon departure of this state.
4. Hosts that support asynchronous signal recovery shall complete transmission of COMRESET in response to a received COMINIT that causes a transition to this state within 10 ms of the de-qualification of the received COMINIT signal.

HP2: HR_AwaitCOMINIT	Interface quiescent.	
1. COMINIT detected from device.	→	HR_AwaitNoCOMINIT
2. COMINIT not detected from device and (asynchronous signal recovery not supported or RetryInterval not elapsed since entry into the HP2:HR_AwaitCOMINIT state) .	→	HR_AwaitCOMINIT
3. COMINIT not detected from device and asynchronous signal recovery supported and RetryInterval elapsed since entry into the HP2:HR_AwaitCOMINIT state.	→	HR_Reset

HP2B: HR_AwaitNoCOMINIT	Interface quiescent.		
1. COMINIT not detected from device.		→	HR_Calibrate
2. COMINIT detected from device.		→	HR_AwaitNoCOMINIT
NOTES: 1. For hosts that do not support asynchronous signal recovery, this state is entered asynchronously any time in response to COMINIT unless during a power-on reset or an explicit reset request (in which case HP1 is entered).			

HP3: HR_Calibrate	Perform calibration[1].		
1. Calibration complete or bypass not implemented.		→	HR_COMWAKE
2. Calibration not complete.		→	HR_Calibrate
NOTES: 1. Calibration is optional. If bypassed or not implemented, proceed directly to HR_COMWAKE.			

HP4: HR_COMWAKE	Transmit COMWAKE.	
1. COMWAKE not detected from device.	→	HR_AwaitCOMWAKE
2. COMWAKE detected from device.	→	HR_AwaitNoCOMWAKE

HP5: HR_AwaitCOMWAKE	Interface quiescent.	
1. COMWAKE detected from device.	→	HR_AwaitNoCOMWAKE
2. COMWAKE not detected from device and (asynchronous signal recovery not supported or RetryInterval not elapsed since entry into the HP5:HR_AwaitCOMWAKE state).	→	HR_AwaitCOMWAKE
3. COMWAKE not detected from device and asynchronous signal recovery supported and RetryInterval elapsed since entry into the HP5:HR_AwaitCOMWAKE state and ResumePending = 0.	→	HR_Reset
4. COMWAKE not detected from device and asynchronous signal recovery supported and RetryInterval elapsed since entry into the HP5:HR_AwaitCOMWAKE state and ResumePending = 1.	→	HR_COMWAKE

HP5B: HR_AwaitNoCOMWAKE	Interface quiescent	
1. COMWAKE not detected from device.	→	HR_AwaitAlign
2. COMWAKE detected from device.	→	HR_AwaitNoCOMWAKE

HP6: HR_AwaitAlign	Host transmits D10.2 characters at lowest supported rate[2]	
1. $ALIGN_P$ detected from device (at any supported speed)[3].	→	HR_AdjustSpeed
2. $ALIGN_P$ not detected from device and 873.8 us (32768 Gen1 Dwords) has elapsed since entry to HR_AwaitAlign.	→	HR_Reset[1,4]
3. $ALIGN_P$ not detected from device and less than 873.8 us (32768 Gen1 Dwords) has elapsed since entry to HR_AwaitAlign.	→	HR_AwaitAlign

NOTES:
1. Host retries the power-on sequence indefinitely unless explicitly turned off by the Application layer.
2. Host shall start transmitting D10.2 characters no later than 533 ns (20 Gen1 Dwords) after COMWAKE is negated as specified in the OOB signaling section.
3. Host designers should be aware that the device is allowed 53.3 ns (2 Gen1 Dwords) after releasing COMWAKE (by holding the idle condition for more than 175 ns) to start sending characters. Until this occurs, the bus is at an idle condition and may be susceptible to crosstalk from other devices. Care should be taken so that crosstalk during this window doesn't result in a false detection of an $ALIGN_P$. For example: a compliant host may detect the negation of COMWAKE in as little as 112 ns, such a host should wait at least 116.3 ns (175+53.3-112) after detecting the release of COMWAKE to start looking for $ALIGN_P$ primitives.
4. The Host Phy Initialization state machine may use the transition to HR_Reset as a method of speed negotiation.

HP7: HR_SendAlign	Transmit $ALIGN_P$ at speed detected	
1. Three back-to-back non-$ALIGN_P$ primitives[2] detected from device.	→	HR_Ready
2. Three back-to-back non-$ALIGN_P$ primitives not detected from device.	→	HR_SendAlign[1]

NOTES:
1. Host retries indefinitely unless explicitly turned off by the Application layer
2. Non-$ALIGN_P$ primitives may be detected by the presence of the K28.3 control character in the Byte 0 position.

HP8: HR_Ready	Transmit word from Link[1].	
1. Partial signal from Link asserted.	→	HR_Partial
2. Slumber signal from Link asserted.	→	HR_Slumber
3. No power management request received and (asynchronous signal recovery not supported or signal recovery poll not initiated[2] or received signal detected).	→	HR_Ready
4. No power management request received and asynchronous signal recovery supported and received signal not detected and signal recovery poll initiated[2].	→	HR_Reset
NOTES: 1. PHYRDY asserted only when in the HR_Ready state and the Phy is maintaining synchronization with the incoming signal to its receiver and is transmitting a valid signal on its transmitter. 2. The latency at which a host elects to initiate an optional signal recovery poll is implementation specific but shall be greater than the $ALIGN_P$ transmit interval.		

HP9: HR_Partial	Interface quiescent. If asynchronous signal recovery is supported then set ResumePending = 1.	
1. Partial signal from Link negated and no COMWAKE detected from device[1,2].	→	HR_COMWAKE
2. Partial signal from Link negated and COMWAKE detected from device[1,2].	→	HR_AwaitNoCOMWAKE
3. Slumber signal from the Link asserted and host Automatic Partial to Slumber transitions are supported[3]	→	HR_Slumber
4. Partial signal from Link asserted.	→	HR_Partial
NOTES: 1. Host Phy shall remember if COMWAKE was detected during Partial to determine if the wakeup request originated from the host or the Phy. 2. The host Phy may take this transition only after it has recovered from Partial mode and the Phy is prepared to initiate communications. If Phy has not yet recovered from the Partial mode it shall remain in this state. 3. The host Phy may transition to HR_SLumber if host Automatic Partial to Slumber transitions are supported by the host and device. Please refer to 13.16 for more information regarding Automatic Partial to Slumber transitions.		

HP10: HR_Slumber	Interface quiescent. If asynchronous signal recovery is supported then set ResumePending = 1.		
	1. Slumber signal from Link negated and no COMWAKE detected from device[1,2].	→	HR_COMWAKE
	2. Slumber signal from Link negated and COMWAKE detected from device[1,2].	→	HR_AwaitNoCOMWAKE
	3. Slumber signal from Link asserted.	→	HR_Slumber
	NOTES: 1. Host Phy shall remember if COMWAKE was detected during Slumber to determine if the wakeup request originated from the host or the Phy. 2. The host Phy may take this transition only after it has recovered from Slumber mode and the Phy is prepared to initiate communications. If Phy has not yet recovered from the Slumber mode it shall remain in this state.		

HP11: HR_AdjustSpeed	Interface undefined but not quiescent[1]		
	1 Transition to appropriate speed completed.	→	HR_SendAlign
	2 Transition to appropriate speed not completed.	→	HR_AdjustSpeed
	NOTES: 1. 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 ns (two Gen1 Dwords) 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.		

8.4.2 Device Phy Initialization State Machine

As described in section 7.5.1.2, reception of a COMRESET signal shall be treated by the device as a hardware reset signal and shall unconditionally force the Device Phy Initialization state machine to transition to the DP1:DR_Reset initial state regardless of other conditions. Reception of COMRESET 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.

Table 52 – State Diagram Device Phy Initialization State Machine

DP1: DR_Reset[1]	Interface quiescent		
	1. COMRESET not detected and power-on reset negated.	→	DR_COMINIT
	2. COMRESET detected or power-on reset asserted.	→	DR_Reset
	NOTES: 1. This state is entered asynchronously any time in response to power-on reset or receipt of a COMRESET signal from the host.		

DP2: DR_COMINIT	Transmit COMINIT[1, 2]		
	1. Unconditional	→	DR_AwaitCOMWAKE
	NOTES: 1. COMINIT transmitted for a 6 bursts duration 2. As indicated in section 13.1, devices shall respond with a COMINIT signal within 10 ms of		

DP2: DR_COMINIT	Transmit COMINIT[1, 2]
the negation of power-on reset or de-qualification of a received COMRESET signal.	

DP3: DR_AwaitCOMWAKE	Interface quiescent	
1. COMWAKE detected from host	→	DR_AwaitNoCOMWAKE
2. COMWAKE not detected from host and (asynchronous signal recovery not implemented or RetryInterval not elapsed since entry into the DP3:DR_AwaitCOMWAKE state).	→	DR_AwaitCOMWAKE
3. COMWAKE not detected from host and asynchronous signal recovery implemented and RetryInterval elapsed since entry into the DP3:DR_AwaitCOMWAKE state.	→	DR_Reset

DP3B: DR_AwaitNoCOMWAKE	Interface quiescent	
1. COMWAKE not detected from host and part of power-on reset sequence[1].	→	DR_Calibrate
2. COMWAKE not detected from host and part of Partial/Slumber awake sequence[1].	→	DR_COMWAKE
3. COMWAKE detected from host.	→	DR_AwaitNoCOMWAKE
NOTES: 1. Device shall remember if it was sent to Partial or Slumber mode for proper wakeup action.		

DP4: DR_Calibrate	Perform calibration[1]	
1. Calibration complete or bypass not implemented.	→	DR_COMWAKE
2. Calibration not complete.	→	DR_Calibrate
NOTES: 1. Calibration is optional. If bypassed or not implemented, proceed directly to DR_COMWAKE.		

DP5: DR_COMWAKE	Transmit COMWAKE	
1. Unconditional	→	DR_SendAlign

DP6: DR_SendAlign	Transmit $ALIGN_P$[1,2,3,5]	
1. $ALIGN_P$ detected from host (device locked to incoming data)[4].	→	DR_Ready
2. $ALIGN_P$ not detected from host and $ALIGN_P$ primitives transmitted for 54.6 us (2048[5] Gen1 $ALIGN_P$ primitives) at speed other than lowest[6].	→	DR_ReduceSpeed
3. $ALIGN_P$ not detected from host and $ALIGN_P$ primitives transmitted for 54.6 us (2048[5] Gen1 $ALIGN_P$ primitives) at lowest speed[6].	→	DR_Error
4. $ALIGN_P$ not detected from host and $ALIGN_P$ primitives transmitted for less than 54.6 us (2048 Gen1 $ALIGN_P$ primitives).	→	DR_SendAlign

NOTES:
1. If this is part of a recovery from a Slumber or Partial power management state, the device shall send $ALIGN_P$ at the previously negotiated speed. For all other cases, $ALIGN_P$ should be sent at the device's fastest supported speed.
2. $ALIGN_P$ primitives should be sent only at valid frequencies (if PLL not locked, send D10.2).
3. After COMWAKE is released as specified in the OOB signaling section, the device shall ensure the interface is active (not quiescent).
4. Device designers should be aware that the host is allowed 533 ns (20 Gen1 Dwords) after detecting the negation of COMWAKE to start sending D10.2 characters. Until this occurs, the bus is in an idle condition and may be susceptible to crosstalk from other devices. Care should be taken so that crosstalk during this window doesn't result in a false detection of an $ALIGN_P$. Devices may extend this timeout up to an additional 54.6 us (2048 Gen1 Dwords) (for a max total of 109.2 us), as necessary to allow their receiver time to lock to the host $ALIGN_P$.
5. Device shall not leave the bus idle more than 53.3 ns (2 Gen1 Dwords) longer than the required 175 ns to negate COMWAKE.
6. If this is part of a recovery from the Slumber or Partial power management state, the device Phy shall not reduce its speed in response to failure to establish communications. Upon failing to establish communications it should instead transition directly to the DR_Error state to initiate a retry of the COMWAKE sequence.

DP7: DR_Ready[1]	Transmit word from Link	
1. Partial signal from Link asserted.	→	DR_Partial
2. Slumber signal from Link asserted.	→	DR_Slumber
3. No power management request received and (asynchronous signal recovery not supported or received signal detected).	→	DR_Ready
4. No power management request received and asynchronous signal recovery supported and received signal not detected.	→	DR_Error

NOTES:
1. PHYRDY asserted only when in the DR_Ready state and the Phy is maintaining synchronization with the incoming signal to its receiver and is transmitting a valid signal on its transmitter.

DP8: DR_Partial	Interface quiescent		
	1. Partial signal from Link negated[1]	→	DR_COMWAKE
	2. Partial signal from Link negated and COMWAKE detected from host[1]	→	DR_AwaitNoCOMWAKE
	3. Slumber signal from Link asserted and device Automatic Partial to Slumber transitions enabled[2]	→	DR_Slumber
	4. Partial signal from Link asserted	→	DR_Partial
	NOTES: 1. The device Phy may take this transition only after it has recovered from Partial mode and the Phy is prepared to initiate communications. If Phy has not yet recovered from the Partial mode it shall remain in this state. 2. The device Phy may transition to DR_Slumber if device Automatic Partial to Slumber transitions are enabled. Please refer to 13.16 for more information regarding Automatic Partial to Slumber transitions.		

DP9: DR_Slumber	Interface quiescent		
	1. Slumber signal from Link negated[1]	→	DR_COMWAKE
	2. Slumber signal from Link negated and COMWAKE detected from host[1]	→	DR_AwaitNoCOMWAKE
	3. Slumber signal from Link asserted	→	DR_Slumber
	NOTES: 1. The device Phy may take this transition only after it has recovered from Slumber mode and the Phy is prepared to initiate communications. If Phy has not yet recovered from the Slumber mode it shall remain in this state.		

DP10: DR_ReduceSpeed	Interface quiescent		
	1. Transition toa slower speed complete	→	DR_SendAlign[1]
	2. Transition to a slower speed not complete	→	DR_ReduceSpeed
	NOTES: 1. 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.		

DP11: DR_Error	Interface quiescent		
	1. Error not due to failure to resume and ((asynchronous signal recovery not supported) or (asynchronous signal recovery supported and RetryInterval not elapsed since entry into the DP11:DR_Error state))	→	DR_Error
	2. Resume from Slumber or Partial failed	→	DR_COMWAKE
	3. Error not due to failure to resume and asynchronous signal recovery supported and RetryInterval elapsed since entry into the DP11:DR_Error state.	→	DR_Reset

8.4.3 Speed Negotiation

In state HP6:HR_AwaitAlign, it is possible for the host to receive a signal at a speed different than what the host is awaiting (i.e. a Gen1 host may receive a Gen2 signal from the device or a Gen2 host may receive a Gen1 signal from the device). Some data recovery circuits may return unpredictable recovered data when presented with an incoming signaling speed higher than supported. Conversely, signal aliasing effects may impact the accuracy of decoded signals when a lower signaling speed than expected is received. Because the recovered data may be invalid, implementations shall insure that $ALIGN_P$ primitives are accurately decoded in the HP6:HR_AwaitAlign state in light of the possibility of recovered data in this state being the result of falsely decoding a signal at a speed different than the host is anticipating.

To reduce susceptibility to false $ALIGN_P$ detection/handshake, receivers should fully qualify the entire received ALIGN sequence instead of relying on qualifying only a portion of it (such as just the comma sequence). Additional means for ensuring that the transition from the HP6:HR_AwaitAlign is accurately traversed and not traversed in response to a spurious signal from the data recovery circuit is to ensure that a series of contiguous $ALIGN_P$ primitives are successfully decoded. Other possible means for ensuring accuracy of the $ALIGN_P$ detection are also possible.

It is the responsibility of the designs to ensure that the conditions and state transitions associated with the Phy initialization state machines are accurately performed and are not susceptible to false decoding/transition as a result of receiving a signal at a speed different than currently selected or at a speed that is not supported.

Devices shall not rely on the host transmission of D10.2 as a means for determining host communication speed since the D10.2 transmission is done at the lowest supported communication speed and not necessarily at the highest mutually supported data speed being negotiated. The D10.2 transmission is only for the purpose of crosstalk suppression and for providing a reference clock; there is no protocol interlock on the D10.2 reception in the Device Phy Initialization state machine.

8.4.3.1 Power-On Sequence Timing Diagram

The following timing diagrams and descriptions are provided for clarity and are informative. The state diagrams provided in section 8.4 comprise the normative behavior specification and is the ultimate reference.

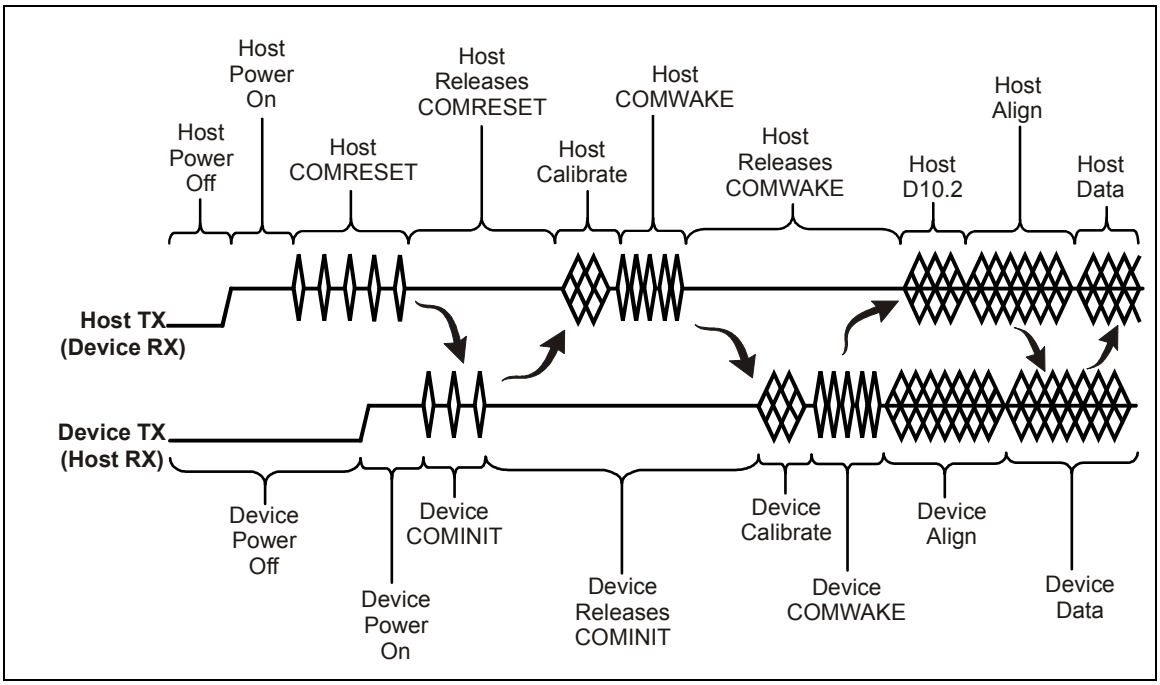


Figure 187 – Power-On Sequence

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 COMRESET
4. Host releases COMRESET. Once the power-on reset is released, the host releases the COMRESET signal and puts the bus in a quiescent condition.
5. Device issues COMINIT – When the device detects the release of COMRESET, it responds with a COMINIT. This is also the entry point if the device is late starting. The device may initiate communications at any time by issuing a COMINIT.
6. Host calibrates and issues a COMWAKE.
7. Device responds – The device detects the COMWAKE sequence on its RX pair and calibrates its transmitter (optional). Following calibration the device sends a six burst COMWAKE sequence and then sends a continuous stream of the ALIGN sequence starting at the device's highest supported speed. After $ALIGN_P$ primitives have been sent for 54.6 us (2048 nominal Gen1 Dword times) without a response from the host as determined by detection of $ALIGN_P$ 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 $ALIGN_P$ primitives at that speed for 54.6 us (2048 nominal Gen1 Dword times.) This step is repeated for as many slower speeds as are supported. Once the lowest speed has been reached without response from the host, the device shall enter an error state.
8. Host locks – after detecting the COMWAKE, the host starts transmitting D10.2 characters (see 7.6) at its lowest supported speed. Meanwhile, the host receiver locks to the ALIGN sequence and, when ready, returns the ALIGN sequence to the device at the same speed as received. A host shall be designed such that it acquires lock in 54.6 us (2048 nominal Gen1 Dword times) at any given speed. The host should allow for at least 873.8 us (32768 nominal Gen1 Dword times) after detecting the release of COMWAKE to receive the first $ALIGN_P$. This insures interoperability with multi-generational and synchronous designs. If no $ALIGN_P$ is received within 873.8 us (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 sequence and, when ready, sends the $SYNC_P$ primitive indicating it is ready to start normal operation.
10. Upon receipt of three back-to-back non-$ALIGN_P$ primitives, the communication link is established and normal operation may begin.

8.4.3.2 Partial/Slumber to PHYRDY

8.4.3.2.1 Host Initiated

The host may initiate a wakeup from the Partial or Slumber states by entering the power-on sequence at the “Host COMWAKE” point in the state machine. 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, shall transmit $ALIGN_P$ primitives at the speed determined at power-on.

8.4.3.2.2 Device Initiated

The device may initiate a wakeup from the Partial or Slumber states by entering the power-on sequence at the “Device COMWAKE” point in the state machine. 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, shall transmit $ALIGN_P$ primitives at the speed determined at power-on.

8.4.3.3 PHYRDY to Partial/Slumber

8.4.3.3.1 Host Initiated

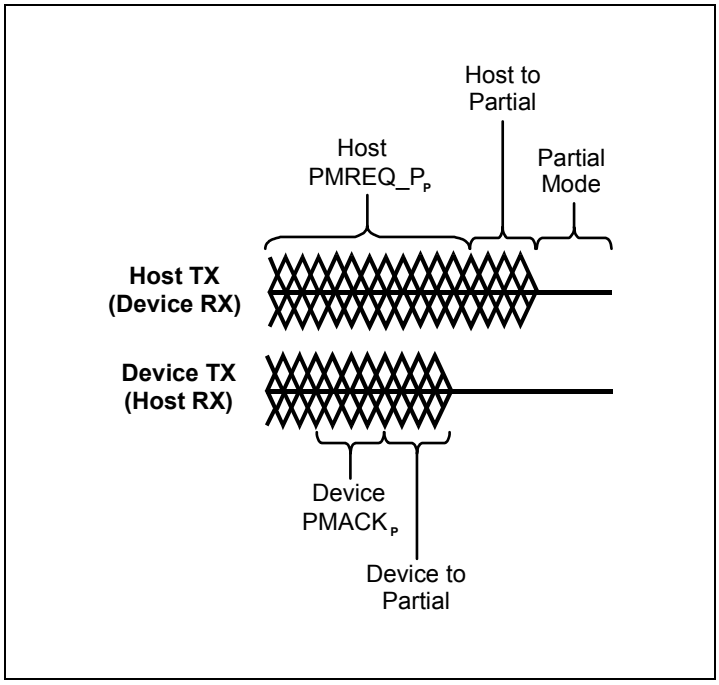


Figure 188 – PHYRDY to Partial—Host Initiated

Note: For Slumber, the same sequence applies except PMREQ_P_P is replaced with PMREQ_S_P and Partial is replaced with Slumber.

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_P_P primitive and transmits it to the host Phy layer.
4. Host Phy layer serializes PMREQ_P_P primitives and transmits them to device Phy layer.
5. Device Phy de-serializes PMREQ_P_P primitives and transmits them to device Link layer.
6. Device Link layer decodes PMREQ_P_P 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 $PMACK_P$ primitive and transmits it four times to device Phy layer.
11. Device Phy layer transmits between four and sixteen $PMACK_P$ primitives to host Phy layer.
12. Device Link layer places device Phy layer in Partial state.
13. Host Phy layer de-serializes $PMACK_P$ primitives and transmits them to host Link layer.
14. Host Link layer decodes $PMACK_P$ primitives and transmits acceptance to host Transport layer.
15. Host Link layer places host Phy layer in Partial State.

16. Host Transport layer transmits acceptance to host Application layer.

8.4.3.3.2 Device Initiated

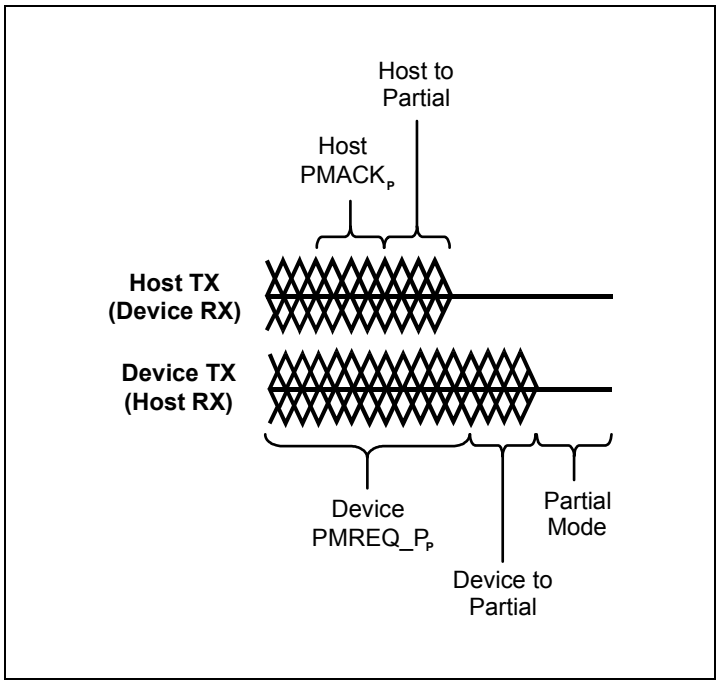


Figure 189 – PHYRDY to Partial—Device Initiated

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_P$_P$ primitive and transmits it to device Phy layer.
4. Device Phy layer serializes PMREQ_P$_P$ primitives and transmits them to host Phy layer.
5. Host Phy de-serializes PMREQ_P$_P$ primitives and transmits them to host Link layer.
6. Host Link layer decodes PMREQ_P$_P$ primitives and transmits request to host Transport layer.
7. Host Transport layer transmits request to host Application layer.
   NOTE – In this context, the host Application layer does not necessarily imply BIOS or other host CPU programming. Rather, the Application layer is the intelligent control section of the chipset logic.
8. Host Application layer processes and accepts request. Issues accept to host Transport layer.
9. Host Transport layer transmits acceptance to host Link layer.
10. Host link layer encodes acceptance as PMACK$_P$ and transmits it four times to host Phy layer.
11. Host Phy layer transmits between four and sixteen PMACK$_P$ primitives to device Phy layer.
12. Host Link layer asserts Partial signal and places host Phy layer in Partial state.
13. Host Phy layer negates PHYRDY signal.
14. Device Phy layer de-serializes PMACK$_P$ primitives and transmits them to device Link layer.

15. Device Link layer decodes $PMACK_P$ primitives and transmits acceptance to device Transport layer.
16. Device Link layer asserts Partial signal and places device Phy layer in Partial State.
17. Device Phy layer negates PHYRDY signal.
18. Device Transport layer transmits acceptance to device Application layer.

9 Link Layer

9.1 Overview

The Link layer transmits and receives frames, transmits primitives based on control signals from the Transport layer, and receives primitives from the Phy 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 are similar, however the device is given precedence when both the host and device request ownership for transmission.

9.1.1 Frame Transmission

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

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

9.1.2 Frame Reception

When data is received from the Phy layer, the Link layer provides the following services:

- Acknowledges to the peer Link layer readiness to receive a frame.
- Receives data in the form of encoded characters from the Phy layer.
- Decodes the encoded 8b/10b character stream into aligned Dwords of data.
- Removes the envelope around frames (i.e., SOF_P, CRC, EOF_P).
- Calculates CRC on the received Dwords.
- Provides frame flow control in response to requests from the FIFO or the peer Link layer.
- Compares the calculated CRC to the received CRC.
- Reports good reception or Link/Phy layer errors to Transport layer and the peer Link layer.
- Descrambles data Dwords received from a peer Link layer.

9.2 Encoding Method

Information to be transmitted over Serial ATA 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 Serial ATA 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 subset of the control character set is utilized by Serial ATA.

9.2.1 Notation and Conventions

Serial ATA uses a letter notation for describing data bits and control variables. A description of the translation process between these notations follows. This section 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.

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.

Table 53 below illustrates the association between the numbered unencoded bits in a byte, the control variable, and the letter-labeled bits in the encoding scheme:

Table 53 – Bit Designations

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

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 190 below, shows the relationship between the various representations.

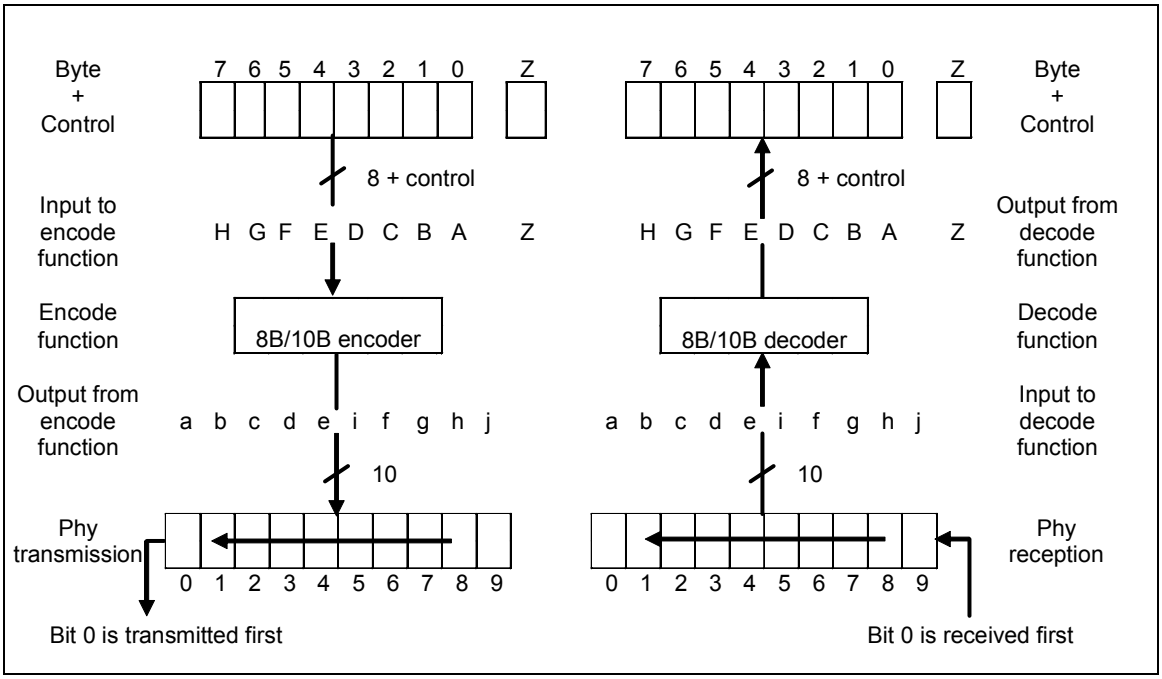


Figure 190 – Nomenclature Reference

Table 54 below shows conversions from byte notation to character notation for a control and data byte. The examples chosen have special significance and are also used during the conversion from data notation to the 8b/10b code values.

Table 54 – Conversion Examples

Byte Notation	BCh, Control Character		4Ah, Data Character	
Bit notation	76543210 10111100	Control variable K	76543210 01001010	Control variable D
Unencoded bit notation	HGF EDCBA 101 11100	Z K	HGF EDCBA 010 01010	Z D
Bit notation reordered to conform with Zxx.y convention	Z EDCBA HGF K 11100 101		Z EDCBA HGF D 01010 010	
Character name	K 28 .5		D 10 .2	

9.2.2 Character Code

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

Serial ATA uses a subset of the 8b/10b coding method described by Widmer and Franaszek (see references). The Serial ATA code 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.

9.2.2.1 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.

9.2.2.2 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, 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 will also be negative.

The rules for Running Disparity will 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 are of alternating disparity.

9.2.2.3 Data Encoding

Table 55 and Table 56 describe the code and running disparity generation rules for each of the sub-blocks. The results may be used to generate the data in the data character tables.

The digital logic which is used to generate the results may be found in Franaszek and Widmer [2] (see references). The generation of control characters is also covered in the patent but not here.

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 55 – 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		rd	D19	10011	110010		
D4	00100	001010	110101	-rd	D20	10100	001011		
D5	00101	101001		rd	D21	10101	101010		
D6	00110	011001			D22	10110	011010		
D7	00111	000111	111000		D23	10111	000101	111010	-rd
D8	01000	000110	111001	-rd	D24	11000	001100	110011	
D9	01001	100101		rd	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	-rd	D31	11111	010100	101011	

Table 56 – 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	
Dx.4	100	0010	1101	-rd
Dx.5	101	1010		rd
Dx.6	110	0110		
Dx.P7	111	0001	1110	-rd
Dx.A7	111	1000	0111	
NOTES: A7 replaces P7 if[(rd>0) and (e=i=0)] or [(rd<0) and (e=i=1)]				

9.2.2.4 Encoding Examples

The encoding examples in Table 57 below 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 may 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 110100b, with a resulting Running Disparity of positive for the 6-bit sub-block. Encoding the 4-bit sub-block triggers the substitution clause of Dx.A7 for Dx.P7 since [(rd>0) AND (e=i=0)].

Table 57 – Encoding Examples

Initial rd	Character Name	abcdei Output	6-Bit Sub-Block rd	fghj Output	4-bit Sub-block rd	Encoded Character	Ending rd
-	D10.2	010101	-	0101	-	010101 0101	-
+	D11.7	110100	+	1000	-	110100 1000	-

9.2.2.5 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.

9.2.2.5.1 Data Characters

Table 58 – 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 58 – 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 58 – 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	AEh	011100 1010	011100 1010
D15.4	8Fh	010111 0010	101000 1101	D15.5	AFh	010111 1010	101000 1010
D16.4	90h	011011 0010	100100 1101	D16.5	B0h	011011 1010	100100 1010
D17.4	91h	100011 1101	100011 0010	D17.5	B1h	100011 1010	100011 1010
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 58 – 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)

9.2.2.5.2 Control Characters

Table 59 – Valid Control Characters

Name	Byte	abcdei fghj Output		Description
		Current rd-	Current rd+	
K28.3	7Ch	001111 0011	110000 1100	Occurs only at Byte 0 of all primitives except for $ALIGN_P$
K28.5	BCh	001111 1010	110000 0101	Occurs only at Byte 0 of $ALIGN_P$

In Serial ATA only the K28.3 and K28.5 control characters are valid and are always used as the first byte in a four-byte primitive. The K28.3 control character is used to prefix all primitives other than $ALIGN_P$, while the K28.5 control character is used to prefix $ALIGN_P$. 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.

$ALIGN_P$ primitives are of neutral disparity, that is the running disparity at the end of $ALIGN_P$ is the same as the running disparity at the beginning of $ALIGN_P$.

9.2.3 Transmission Summary

9.2.3.1 Transmission Order

9.2.3.1.1 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.

9.2.3.1.2 Bytes Within a Dword

For all transmissions and receptions, Serial ATA organizes all values as Dwords. Even when representing a 32-bit value, the Dword shall be considered a set of four bytes. The transmission order of the bytes within the Dword shall be from the least-significant byte (byte 0) to the most-significant byte (byte 3). This right-to-left transmission order differs from Fibre Channel. Figure 191 illustrates how the bytes are arranged in a Dword and the order in which bits are sent.

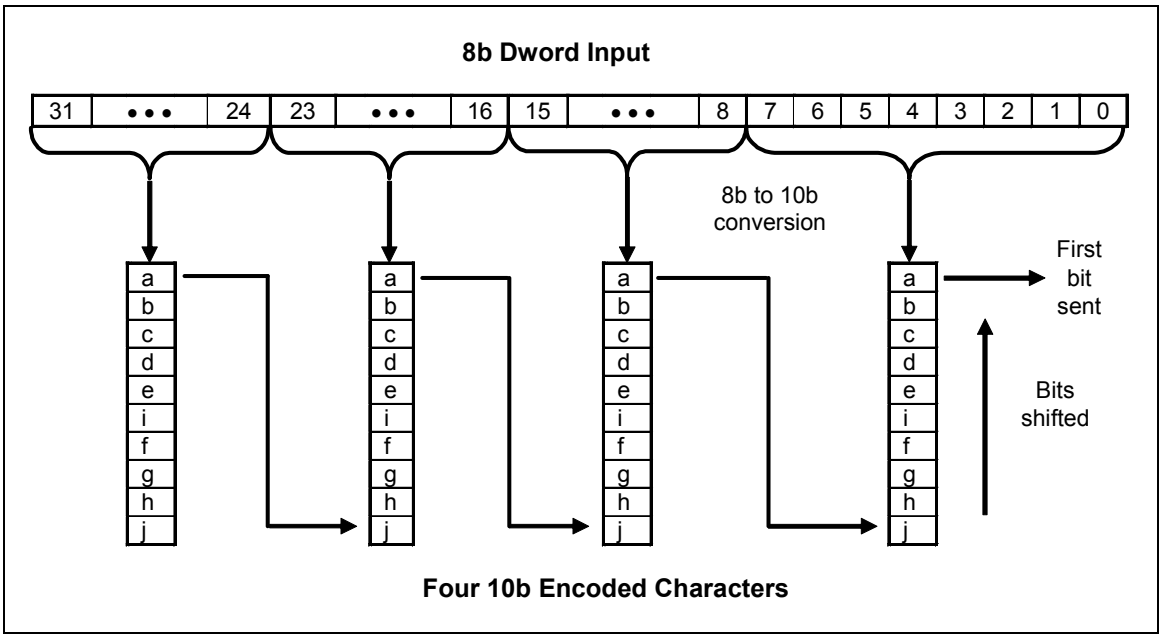


Figure 191 – Bit Ordering and Significance

9.2.3.1.3 Dwords Within a Frame

A frame (as described in section 10.2.1) shall be transmitted sequentially in ascending Dword order starting with the SOF_P delimiter, followed by the Dwords of the frame contents, followed by the CRC, and ending with the EOF_P delimiter.

NOTE – While this specification discusses 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 shall be preserved.

9.2.4 Reception Summary

Upon reception of an encoded character the column corresponding to the receiver's current Running Disparity shall be searched for the encoded character value. If the encoded character value is found in the table the received encoded character shall be considered 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.

9.2.4.1 Disparity and the Detection of a Code Violation

Due to the propagation characteristics of the 8b/10b code, it is possible that although most errors are detected, a single bit error might not be detected until several characters after the error occurred. 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 Serial ATA sends data in Dword increments, but the transmitter and receiver operate in units of a byte (character). The examples don't show Dword boundaries, so it is possible that an error in either of these cases could be deferred 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.

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

Table 60 – Single Bit Error with Two Character Delay

	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 10<u>1</u>1[a]	+	010101 0101	+	111010[b] 1010	+
Decoded character stream.	-	D21.0	+	D10.2	+	Code violation[b]	+[c]

NOTES:
[a] Bit error introduced: 1001b => 1011b
[b] 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.
[c] Running disparity shall be computed on the received character regardless of the validity of the encoded character.

Table 61 – Single Bit Error with One 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 10<u>1</u>1[a]	+	111010[b] 0010	-	111010 1010	+
Decoded character stream	-	D21.0	+	Code violation[b]	-	D23.5	+[c]

NOTES:
[a] Bit error introduced: 1001b => 1011b
[b] 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.
[c] Running disparity shall be computed on the received character regardless of the validity of the encoded character.

9.3 Transmission Overview

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 special character. Since all of the bytes are constants, a primitive cannot be used to convey variable information. Later sections describe the exact contents of the primitives used by Serial ATA.

A frame consists of multiple Dwords, and always starts with SOF_P, followed by a user payload called a Frame Information Structure (FIS), a CRC, and ends with EOF_P. The CRC is defined to be the last non-primitive Dword immediately preceding EOF_P. Some number of flow control primitives ($HOLD_P$ or $HOLDA_P$, or a $CONT_P$ stream to sustain a $HOLD_P$ or $HOLDA_P$ state) are allowed between the SOF_P and EOF_P primitives to throttle data flow for speed matching purposes.

Figure 192 shows an example of a sequence of transmissions.

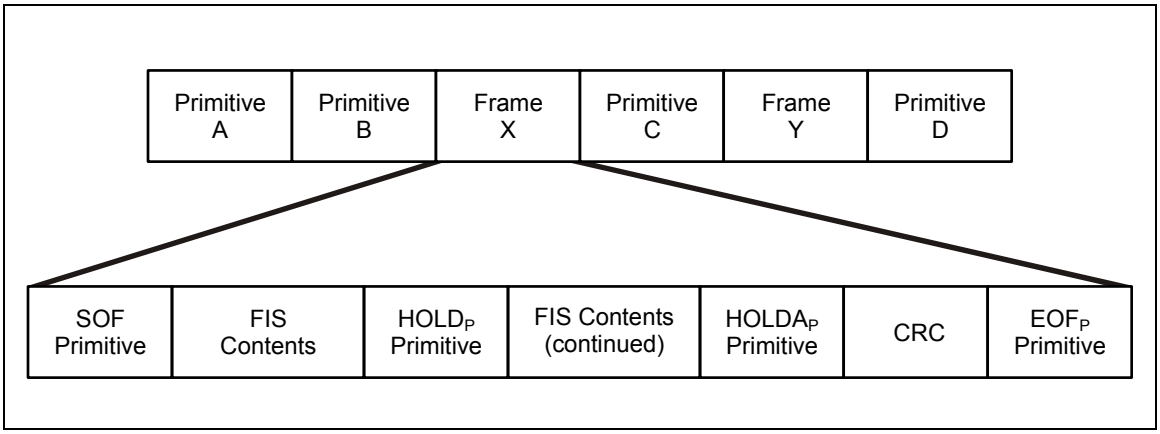


Figure 192 – Transmission Structures

9.4 Primitives

9.4.1 Overview

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

Primitives always begin with a control character; all primitives use the K28.3 control character to signify the beginning of a primitive except for $ALIGN_P$ which begins with the K28.5 control character. $ALIGN_P$ thus represents the only primitive that contains the comma character. Following the control character, three additional characters are encoded to complete the Dword. Table 62 is a summary of the character combinations that make up each primitive.

9.4.1.1 Primitive Disparity

Primitives may 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.

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

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

9.4.1.2 Primitive Handshakes

Some primitives are transmitted in response to receipt of other primitives to acknowledge receipt. For example, $HOLDA_P$ is transmitted in response to the receipt of $HOLD_P$ primitives and R_OK_P or R_ERR_P is transmitted in response to $WTRM_P$. Due to the different clock domains between to two ends of the cable, the number of response primitives may not match the number of primitives to which they are responding. For example, a device may send five $HOLD_P$ primitives but receive six $HOLDA_P$ primitives in response. Neither the transmitter nor receiver 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 $HOLDA_P$ may be sent.

9.4.2 Primitive Descriptions

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

Table 62 – Description of Primitives

Primitive	Name	Description
$ALIGN_P$	Phy layer control.	Upon receipt of an $ALIGN_P$, the Phy layer re-adjusts 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 $ALIGN_P$ primitives shall be sent (except as noted for retimed loopback).
$CONT_P$	Continue repeating previous primitive.	$CONT_P$ allows long strings of repeated primitives to be eliminated. $CONT_P$ implies that the previously received primitive be repeated as long as another primitive is not received.
$DMAT_P$	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 EOF_P. The transmitter context is assumed to remain stable after EOF_P has been sent.
EOF_P	End of frame.	EOF_P marks the end of a frame. The previous non-primitive Dword is the CRC for the frame.
$HOLD_P$	Hold data transmission.	$HOLD_P$ is transmitted in place of payload data within a frame when the transmitter does not have the next payload data ready for transmission. $HOLD_P$ is also transmitted by the receiver when a receiver is not ready to receive additional payload data.
$HOLDA_P$	Hold acknowledge.	This primitive is sent while $HOLD_P$ is received.
$PMACK_P$	Power management acknowledge.	Sent in response to a $PMREQ\_S_P$ or $PMREQ\_P_P$ when a receiving node is prepared to enter a power mode state.
$PMNAK_P$	Power management denial.	Sent in response to a $PMREQ\_S_P$ or $PMREQ\_P_P$ when a receiving node is not prepared to enter a power mode state or when power management is not supported.

Primitive	Name	Description
$PMREQ\_P_P$	Power management request to Partial.	This primitive is sent continuously until $PMACK_P$ or $PMNAK_P$ is received. When $PMACK_P$ is received, the current node (host or device) stops transmitting $PMREQ\_P_P$ and enters the Partial power management state.
$PMREQ\_S_P$	Power management request to Slumber	This primitive is sent continuously until $PMACK_P$ or $PMNAK_P$ is received. When $PMACK_P$ is received, the current node (host or device) stops transmitting $PMREQ\_S_P$ and enters the Slumber power management state.
R_ERR_P	Reception error.	Current node (host or device) detected error in received payload.
R_IP_P	Reception in Progress.	Current node (host or device) is receiving payload.
R_OK_P	Reception with no error.	Current node (host or device) detected no error in received payload.
R_RDY_P	Receiver ready.	Current node (host or device) is ready to receive payload.
SOF_P	Start of frame.	Start of a frame. Payload and CRC follow until EOF_P.
$SYNC_P$	Synchronization	Synchronizing primitive.
$WTRM_P$	Wait for frame termination	After transmission of an EOF_P, the transmitter sends $WTRM_P$ while waiting for reception status from receiver.
X_RDY_P	Transmission data ready.	Current node (host or device) has payload ready for transmission.

9.4.3 Primitive Encoding

The following table defines the encoding for each primitive.

Table 63 – Primitive Encoding

Primitive Name	Byte 3 Contents	Byte 2 Contents	Byte 1 Contents	Byte 0 Contents
$ALIGN_P$	D27.3	D10.2	D10.2.	K28.5
$CONT_P$	D25.4	D25.4	D10.5	K28.3
$DMAT_P$	D22.1	D22.1	D21.5	K28.3
EOF_P	D21.6	D21.6	D21.5	K28.3
$HOLD_P$	D21.6	D21.6	D10.5	K28.3
$HOLDA_P$	D21.4	D21.4	D10.5	K28.3
$PMACK_P$	D21.4	D21.4	D21.4	K28.3
$PMNAK_P$	D21.7	D21.7	D21.4	K28.3
$PMREQ\_P_P$	D23.0	D23.0	D21.5	K28.3
$PMREQ\_S_P$	D21.3	D21.3	D21.4	K28.3
R_ERR_P	D22.2	D22.2	D21.5	K28.3
R_IP_P	D21.2	D21.2	D21.5	K28.3
R_OK_P	D21.1	D21.1	D21.5	K28.3
R_RDY_P	D10.2	D10.2	D21.4	K28.3
SOF_P	D23.1	D23.1	D21.5	K28.3
$SYNC_P$	D21.5	D21.5	D21.4	K28.3
$WTRM_P$	D24.2	D24.2	D21.5	K28.3
X_RDY_P	D23.2	D23.2	D21.5	K28.3

9.4.4 $DMAT_P$ Primitive

No consistent use of the $DMAT_P$ facility is defined, and its use may impact software compatibility. Implementations should tolerate reception of $DMAT_P$ as defined in this specification but should avoid transmission of $DMAT_P$ in order to minimize potential interaction problems. One valid response to reception of $DMAT_P$ is to treat the $DMAT_P$ as R_IP_P and complete the transfer.

For example, in the case of a DMA read from device, a Serial ATA device may terminate the transfer with an EOF_P, and send a Register Device to Host FIS to the host, with Error and Status registers updated appropriately. In the case of a DMA write to device, the device sends a DMA Activate FIS to the host, and then after receiving an SOF_P, has to accept all data until receiving an EOF_P from the host. Since the device cannot terminate such a transfer once started, a special abort primitive is used.

The DMA Terminate ($DMAT_P$) primitive may be sent on the back channel during reception 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. Reception of the $DMAT_P$ signal shall cause the recipient to close the current frame by inserting the CRC and EOF_P, and return to the idle state.

For host to device data transfers, upon receiving the $DMAT_P$ signal the host shall terminate the transfer in progress by deactivating its DMA engine and closing the frame with valid CRC and EOF_P. 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 Activate 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 $DMAT_P$ signal by the device results in permanent termination of the transfer and is not resumable. The device shall terminate the transmission in progress and close the frame with a valid CRC and EOF_P, 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 $DMAT_P$ signal is provided.

Some implementations may have an implementation-dependent latency associated with closing the affected Data FIS in response to the $DMAT_P$ signal. For example, a host controller 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. Conservative designs should minimize the $DMAT_P$ response latency while being tolerant of other devices having a long latency.

9.4.5 $CONT_P$ Primitive

In order to accommodate EMI reductions, scrambling of data is incorporated in Serial ATA as described in section 9.5. The scrambling of data is simple, with a linear feedback shift register (LFSR) used in generating the scrambling pattern being reset at each SOF_P. 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 suppressed through the use of $CONT_P$.

Any repetitive primitive may be implied to continue repeating through the use of $CONT_P$. The recipient of $CONT_P$ shall ignore all data received after $CONT_P$ until the reception of any primitive, excluding $ALIGN_P$. After transmitting $CONT_P$, the transmitter may send any sequence of data characters to the recipient provided that no primitives are included. The reception of $CONT_P$ shall cause the last valid primitive to be implied as repeated until the reception of the next valid primitive.

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 $CONT_P$ is transmitted. The first primitive correctly received is the initiator of any action within the receiver. This avoids scenarios, for example, where X_RDY_P is sent from the host, followed by a $CONT_P$, and the X_RDY_P is received improperly resulting in the device not returning an R_RDY_P and causing the system to deadlock until a timeout/reset condition occurs.

The transmission of $CONT_P$ is optional, but the ability to receive and properly process $CONT_P$ is required. The insertion of a single, or two repetitive primitives not followed by $CONT_P$ is valid (i.e. data, data, $HOLD_P$, data).

The following primitives may be followed by a $CONT_P$: $HOLD_P$, $HOLDA_P$, $PMREQ\_P_P$, $PMREQ\_S_P$, R_ERR_P, R_IP_P, R_OK_P, R_RDY_P, $SYNC_P$, $WTRM_P$ and X_RDY_P.

The host Phy initialization state machine consumes the first few received primitives before communications between the host and device have been established (see state HP7:HR_SendAlign in section 8.4.1). 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 section 9.6.2 and section 9.6.5), the use of $CONT_P$ 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-$ALIGN_P$ primitives have been transmitted or until receipt of a primitive other than $SYNC_P$ or $ALIGN_P$ has been detected.

Table 64 illustrates use of $CONT_P$ in the transmission of a FIS.

Table 64 – $CONT_P$ Usage Example

Transmitter	Receiver
XXXX	XXXX
XXXX	XXXX
X_RDY_P	XXXX
X_RDY_P	XXXX
$CONT_P$	XXXX
XXXX	XXXX
XXXX	R_RDY_P
XXXX	R_RDY_P
XXXX	$CONT_P$
SOF_P	XXXX
DATA (FIS Type)	XXXX
DATA	XXXX
DATA	R_IP_P
DATA	R_IP_P
DATA	$CONT_P$
$HOLD_P$	XXXX
$HOLD_P$	XXXX
$CONT_P$	XXXX
XXXX	XXXX
XXXX	$HOLDA_P$
XXXX	$HOLDA_P$

Transmitter	Receiver
$HOLD_P$	$CONT_P$
DATA	XXXX
DATA	XXXX
DATA	XXXX
CRC	XXXX
EOF_P	R_IP_P
$WTRM_P$	R_IP_P
$WTRM_P$	$CONT_P$
$WTRM_P$	XXXX
$CONT_P$	XXXX
XXXX	R_OK_P
XXXX	R_OK_P
XXXX	$CONT_P$
XXXX	XXXX
$SYNC_P$	XXXX
$SYNC_P$	XXXX
$CONT_P$	XXXX
XXXX	XXXX
XXXX	$SYNC_P$
XXXX	$SYNC_P$
XXXX	$CONT_P$
XXXX	XXXX
XXXX	XXXX

NOTES:
XXXX Scrambled data values (non-primitives)
DATA FIS payload data.

9.4.5.1 Scrambling of Data Following the $CONT_P$ Primitive

The data following the $CONT_P$ shall be the output of an LFSR which implements the same polynomial as is used to scramble FIS contents. That polynomial is defined in section 9.5.

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 $CONT_P$ shall be reset to the initial value upon detection of a COMINIT or COMRESET event.

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

The sequence of LFSR values used to scramble the payload contents of a 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 data payload character that is scrambled using the data payload LFSR. See section 7.5 for additional information on scrambling and repeated primitive suppression.

9.4.5.2 Periodic Retransmission of Sustained Primitives (Informative)

In order to be able to determine the state that a bus is in, it is recommended that a sustained primitive periodically be retransmitted. The only requirement is that the interval at which the retransmit occurs is large enough that EMI is not substantially affected. Since the ALIGN sequence is required to be sent at an interval of at most 256 Dwords, one solution to providing visibility to a suppressed primitive stream is to retransmit the suppressed primitive sequence immediately after the $ALIGN_P$ primitives are inserted. For example, if the original sequence was PRIM / PRIM / $CONT_P$ / junk. . . $ALIGN_P$ / $ALIGN_P$ / junk. . . the new sequence could look like: PRIM / PRIM / $CONT_P$ / junk . . . $ALIGN_P$ / $ALIGN_P$ / PRIM / PRIM / $CONT_P$ / junk.

The actual interval chosen by an implementation could be longer. The goal is to make visible the primitive stream being sustained within the normal depth of a logic analyzer.

9.4.6 $ALIGN_P$ Primitive

The Link layer shall ignore reception of $ALIGN_P$ primitives. The Phy layer is free to consume received $ALIGN_P$ primitives. Implementations where the Phy does not consume received $ALIGN_P$ primitives shall effectively drop received $ALIGN_P$ 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 $ALIGN_P$ primitives are consumed by the Phy and not presented to the Link.

9.4.7 Flow Control Signaling Latency

There is a finite pipeline latency in a round-trip handshake across the Serial ATA interface. In order to avoid buffer overflow in flow control situations, the maximum tolerable latency from when a receiver issues a $HOLD_P$ signal until it receives the $HOLDA_P$ signal from the transmitter is specified. This allows the high-water mark 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 $HOLD_P$ characters on the back channel. Some number of received Dwords later, valid data ceases, and $HOLDA_P$ characters are received. The larger the latency between transmitting $HOLD_P$ until receiving $HOLDA_P$, the larger the receive FIFO needs to be. Within a single $HOLD_P$/ $HOLDA_P$ sequence, the maximum allowed latency from the time the MSB of the initial $HOLD_P$ is on the wire, until the MSB of the initial $HOLDA_P$ 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 $HOLD_P$ flow control character to the transmitter, and the transmitter shall respond with a $HOLDA_P$ in response to receiving a $HOLD_P$ within 20 Dword times. The 20 Dword latency specification is not applicable to any subsequent transmissions of the $HOLD_P$ flow control character within the same sequence. Upon each new instantiation of a $HOLD_P$/ $HOLDA_P$ sequence, the receiver and transmitter shall meet the 20 Dword latency specification.

There is no reference design in this specification. 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. The figures represent an almost literal interpretation of the spec into logic design. A synchronizer is assumed to be a worst case of 2.99 clocks of any clock domain and is rounded to three whole clocks. The Serial ATA cable contains less than half of a Dword of content at Gen1i and Gen2i speeds with 1m internal cables, and is therefore negligible. For longer cable lengths, the effect of the cable should not be ignored. 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.

The following figure outlines the origin of the 20 Dword latency specification. The example illustrates the components of a round trip delay when the receiver transmits $HOLD_P$ on the link until reception of the $HOLDA_P$ from the transmitter. This corresponds to the number of Dwords that the receiver shall be able to accept after transmitting a $HOLD_P$.

Table 65 – Example of Components of a Round Trip Delay

Receiver Sends $HOLD_P$	
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 $HOLDA_P$.
1 Dword	Link acts on notification and inserts $HOLDA_P$ 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
$HOLDA_P$ on the Cable	

9.4.7.1 Cable Length and Flow Control Latency (Informative)

For hosts and devices which are designed for use in data center applications in which cables longer than 1 meter are used, it is advised that these designs accommodate potential increased effects on overflow latencies. When operating at 3.0 Gbps with a 1 meter cable, the cable contains less than half a Dword of content at any point in time and thus the latency effect from the cable is ignored in flow control calculations. However, when operating at 3.0 Gbps with an 8 meter cable as an example, the cable contains almost 3 Dwords of content. When cables longer than 1 meter are used, the effect of the cable on flow control latency should be accounted for in the system design.

A system design may account for these effects in a multitude of ways. Some examples include:

- Hosts and/or devices may be selected that meet more stringent flow control requirements.
- Hosts and/or devices may be selected that have a larger flow control buffer and absorb more than 20 Dwords of latency.
- Do not select excessive cables lengths over what is required for the environment as it impacts flow control latencies.

9.4.8 Examples of Primitive Usage (Informative)

Table 66, Table 67, and Table 68 are examples that illustrate basic primitive usage. They do not show detailed lengthy sequences which invoke the use of $CONT_P$.

Table 66 – SRST Write from Host to Device Transmission Breaking Through a Device to Host Data FIS

Host	Device	Description
…	…	Previous activity abbreviated for clarity.
R_IP_P	DATA n	Device transmitting data.
R_IP_P	DATA n+1	
R_IP_P	$HOLD_P$	Device transmit FIFO empty, and flow control applied.
R_IP_P	$HOLD_P$	Host receives and decodes $HOLD_P$ flow control.
$HOLDA_P$	$HOLD_P$	Host acknowledges flow control. Device internally deadlocked and no more data forthcoming (device hung) .
$HOLDA_P$	$HOLD_P$	
…	…	System in this state until host decides to reset device.
$HOLDA_P$	$HOLD_P$	Host detects SRST write to Device Control register, needs to break deadlock.
$SYNC_P$	$HOLD_P$	Host transmits $SYNC_P$ to abort current transmission.
$SYNC_P$	$HOLD_P$	Device receives and decodes $SYNC_P$, abandons transmission in progress.
$SYNC_P$	$SYNC_P$	Host sends $SYNC_P$ / Device sends $SYNC_P$ (both returned to idle state)
$SYNC_P$	$SYNC_P$	Host receives and decodes $SYNC_P$, may now initiate new FIS transmission
X_RDY_P	$SYNC_P$	Host ready to send Shadow register block registers for SRST write
X_RDY_P	$SYNC_P$	Device decodes X_RDY_P
X_RDY_P	R_RDY_P	Device indicates ready to receive
X_RDY_P	R_RDY_P	Host decodes R_RDY_P
SOF_P	R_RDY_P	Host starts a frame
etc	etc	etc

Table 67 – Command Shadow Register Block Register Transmission Example

Host	Device	Description
$SYNC_P$	$SYNC_P$	Idle condition.
$SYNC_P$	$SYNC_P$	Idle condition.
X_RDY_P	$SYNC_P$	Host ready to send Shadow register block registers.
X_RDY_P	$SYNC_P$	Device decodes X_RDY_{Pc}.
X_RDY_P	R_RDY_P	Device indicates ready to receive.
X_RDY_P	R_RDY_P	Host decodes R_RDY_P.
SOF_P	R_RDY_P	Host starts a frame.
DATA 0	R_RDY_P	Host sends Register FIS Dword 0 / device decodes SOF_P.
DATA 1	R_IP_P	Host sends Register FIS Dword 1 / device stores DATA Dword 0.

Host	Device	Description
…	…	…
DATA n	R_IP$_P$	Host sends Register FIS Dword n / device stores DATA Dword n-1.
CRC	R_IP$_P$	Host sends CRC / device stores DATA Dword n.
EOF$_P$	R_IP$_P$	Host sends EOF$_P$ / device stores CRC.
WTRM$_P$	R_IP$_P$	Device decodes EOF$_P$.
WTRM$_P$	R_IP$_P$	Device computes good CRC and releases TF contents.
WTRM$_P$	R_OK$_P$	Device sends good end.
WTRM$_P$	R_OK$_P$	Host decodes R_OK$_P$ as good results.
SYNC$_P$	R_OK$_P$	Host releases interface.
SYNC$_P$	R_OK$_P$	Device decodes release by host - is allowed to release.
SYNC$_P$	SYNC$_P$	Idle condition.

Table 68 – Data from Host to Device Transmission Example

Host	Device	Description
SYNC$_P$	SYNC$_P$	Idle condition.
SYNC$_P$	SYNC$_P$	Idle condition.
X_RDY$_P$	SYNC$_P$	Host ready to send Shadow register block registers.
X_RDY$_P$	SYNC$_P$	Device decodes X_RDY$_P$.
X_RDY$_P$	R_RDY$_P$	Device indicates ready to receive.
X_RDY$_P$	R_RDY$_P$	Host decodes R_RDY$_P$.
SOF$_P$	R_RDY$_P$	Host starts a frame.
DATA 0	R_RDY$_P$	Host sends DATA Dword 0 / device decodes SOF$_P$.
DATA 1	R_IP$_P$	Host sends DATA Dword 1 / device stores DATA Dword 0.
...	..	...
DATA x	R_IP$_P$	Host sends DATA Dword x / device stores DATA Dword (x-1) .
HOLD$_P$	R_IP$_P$	Host sends HOLD / device stores DATA Dword (x) and decodes HOLD$_P$.
HOLD$_P$	HOLDA$_P$	Device acknowledges HOLD$_P$.
HOLD$_P$	HOLDA$_P$	Host decodes HOLDA$_P$ – host may release HOLD$_P$ at any time.
DATA(n-2)	HOLDA$_P$	Host sends (n-2)th DATA Dword / device decodes DATA Dword.
DATA(n-1)	R_IP$_P$	Host sends (n-1)th data Dword / device stores (n-2)th DATA Dword.
DATA(n)	R_IP$_P$	Host sends nth data Dword / device stores (n-1)th DATA Dword.
CRC	R_IP$_P$	Host sends CRC / device stores nth DATA Dword.
EOF$_P$	R_IP$_P$	Host sends EOF$_P$ / device stores CRC.
WTRM$_P$	R_IP$_P$	Device decodes EOF$_P$.
WTRM$_P$	R_IP$_P$	Device computes good CRC and releases data contents.
WTRM$_P$	R_OK$_P$	Device sends good end.
WTRM$_P$	R_OK$_P$	Host decodes R_OK$_P$ as good results.
SYNC$_P$	R_OK$_P$	Host releases interface.
SYNC$_P$	R_OK$_P$	Device decodes release by host - is allowed to release.
SYNC$_P$	SYNC$_P$	Idle condition.

9.5 CRC and Scrambling

The CRC (Cyclic Redundancy Check) of a frame is a Dword (32-bit) field that shall follow the last Dword of the contents of a FIS and precede EOF_P. The CRC calculation covers all of the FIS transport data between the SOF_P and EOF_P primitives, and excludes any intervening primitives and $CONT_P$ 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 a 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 SOF_P to EOF_P shall not exceed 2064 Dwords including the FIS type and CRC..

The contents of a frame shall be scrambled before transmission by the Phy 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 the seed value of FFFFh before the first shift of the LFSR. The shift register shall be initialized to the seed value before SOF_P is transmitted. All data words between the SOF_P and EOF_P shall be scrambled, including the CRC.

9.5.1 Relationship Between Scrambling of FIS Data and Repeated Primitives

There are two separate scramblers used in Serial ATA. One scrambler is used for the data payload encoding and a separate scrambler is used for repeated primitive suppression. The scrambler used for data payload encoding shall maintain consistent and contiguous context over the scrambled payload characters of a frame (between SOF_P and EOF_P), 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 $ALIGN_P$, 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 $CONT_P$ primitive). Since it is possible for a repeated primitive stream to occur in the middle of a data frame – multiple $HOLD_P$/$HOLDA_P$ primitives are likely – care should be taken to insure that the data payload LFSR is only advanced for each data payload character that it scrambles and that it is not advanced for primitives or for data characters transmitted as part of repeated primitive suppression which uses a separate scrambler.

9.5.2 Relationship Between Scrambling and CRC

The order of application of scrambling shall be as follows. For a Dword of data following SOF_P 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.

9.5.3 Scrambling Disable (Informative)

Hosts and devices should provide a vendor-specific means of disabling the transmission/reception of scrambled data. Three independent controls are recommended – one to disable the scrambling of transmitted FIS payload data, the second to disable the $CONT_P$/junk method of repeated primitive suppression, and the third to disable the unscrambling of received FIS payload data.

Using the scrambling disable capabilities is intended for testability and design debug, and not recommended as an end-user feature. It is the responsibility of the engineer/operator to ensure that both ends of the cable are configured in such a way that the host and device may communicate (i.e. if scrambled transmission is disabled on the device then scrambled reception shall be disabled on the host). Devices that disable payload scrambling may not interoperate with other devices that do not implement this recommendation. Systems that disable scrambling may not meet EMI regulatory requirements.

9.6 Link Layer State Machine

9.6.1 Terms Used in Link Layer Transition Tables

1. LRESET: Link layer COMRESET or COMINIT signal
2. PHYRDYn: The negation of the PHYRDY signal.
3. PHYRDY: Phy status as defined in section 7.1.2.
4. DecErr: Bad decode of a 32 bit Dword transferred from Phy to Link
   - Invalid 10b pattern
   - Disparity error
   - Primitive with a control character in the first byte but not an allowed control character
   - Any control character in other than the first byte of the Dword
5. DatDword: A 32 bit pattern that is formed correctly, but does not have the primitive leading 10b pattern (K28.5 or K28.3).
6. COMWAKE: Signal from the OOB detector in the Phy indicating that the COMWAKE OOB signal is being detected.
7. AnyDword: A 32 bit pattern of any type - even one with DecErr received from Phy

9.6.2 Link Idle State Diagram

Table 69 – State Diagram Link Idle

L1: L_IDLE[4]	Transmit $SYNC_P$.	
1. Transport layer requests frame transmission and PHYRDY[2].	→	HL_SendChkRdy or DL_SendChkRdy[1]
2. Transport layer requests transition to Partial and PHYRDY[2,5].	→	L_TPMPartial
3. Transport layer requests transition to Slumber and PHYRDY[2,5].	→	L_TPMSlumber
4. X_RDY_P received from Phy.	→	L_RcvWaitFifo
5. Phy layer forwards ($PMREQ\_P_P$ or $PMREQ\_S_P$) and power modes are enabled and acceptable.	→	L_PMOff
6. Phy layer forwards ($PMREQ\_P_P$ or $PMREQ\_S_P$) and power modes are disabled or are unacceptable.	→	L_PMDeny
7. Phy layer forwards AnyDword other than (X_RDY_P or $PMREQ\_P_P$ or $PMREQ\_S_P$) and no transmit request from Transport layer[2,3].	→	L_IDLE
8. PHYRDYn	→	L_NoCommErr

NOTES:
1. The host Link layer makes a transition to the HL_SendChkRdy state; the device Link layer makes a transition to the DL_SendChkRdy state.
2. This transition is taken even if errors such as 10b decoding errors are detected.
3. This statement also ignores any unrecognized sequences or commands not defined in this specification.
4. Upon entry to this state from the LS3:L_SendAlign state or the LPM8:L_WakeUp2 state, use of $CONT_P$ is not allowed until either a minimum of 10 non-$ALIGN_P$ primitives have been transmitted or until receipt of a primitive other than $SYNC_P$ or $ALIGN_P$ has been detected.
5. Hosts shall not attempt initiating an interface power state transition between an issued reset and the receipt of the device reset signature. Hosts should not attempt initiating an interface power management request without first verifying the device has such capabilities as determined by the information in the device's IDENTIFY DEVICE (or IDENTIFY PACKET DEVICE) data structure.

L2: L_SyncEscape[1]	Transmit $SYNC_P$.	
1. AnyDword other than X_RDY_P or $SYNC_P$ received from Phy.	→	L_SyncEscape
2. X_RDY_P or $SYNC_P$ received from Phy.	→	L_IDLE
3. PHYRDYn	→	L_NoCommErr[2]

NOTES:
1. This state is entered asynchronously from any other Link layer state where the Link layer has transmitted $SYNC_P$ to escape a FIS transfer, also known as a SYNC Escape.
2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.

LS1: L_NoCommErr	Post Phy not ready error to Transport layer.	
1. Unconditional	→	L_NoComm

LS2: L_NoComm	Transmit $ALIGN_P$[1].	
1. PHYRDYn	→	L_NoComm
2. PHYRDY	→	L_SendAlign
NOTES: 1. Also deactivate any signal for Phy layer to abort operation.		

LS3: L_SendAlign	Transmit $ALIGN_P$.	
1. PHYRDYn	→	L_NoCommErr
2. PHYRDY	→	L_IDLE

LS4: L_RESET[1]	Reset Link state to initial conditions.	
1. LRESET Link reset signal asserted.	→	L_RESET
2. LRESET Link reset signal negated.	→	L_NoComm
NOTES: 1. This state is entered asynchronously when the link reset control is active.		

L1: L_IDLE state: This state is entered when a frame transmission has been completed by the Link layer.

When in this state, the Link layer transmits $SYNC_P$ and waits for X_RDY_P from the Phy layer or a frame transmission request from the Transport layer.

Transition L1:1a: When the host Link layer receives a request to transmit a frame from the Transport layer and the Phy layer is ready, the Link layer shall make a transition to the LT1: HL_SendChkRdy state.

Transition L1:1b: When the device Link layer receives a request to transmit a frame from the Transport layer and the Phy layer is ready, the Link layer shall make a transition to the LT2: DL_SendChkRdy state.

Transition L1:2: When the Link layer receives a request to enter the Partial power mode from the Transport layer and the Phy layer is ready, the Link layer shall make a transition to the L_TPMPartial state.

Transition L1:3: When the Link layer receives a request to enter the Slumber power mode from the Transport layer and the Phy layer is ready, the Link layer shall make a transition to the L_TPMSlumber state.

Transition L1:4: When the Link layer receives an X_RDY_P from the Phy layer, the Link layer shall make a transition to the LR2: L_RcvWaitFifo state.

Transition L1:5: When the Link layer receives a $PMREQ\_P_P$ or $PMREQ\_S_P$ from the Phy layer,is enabled to perform power management modes, and in a state to accept power mode requests, the Link layer shall make a transition to the LPM3: L_PMOff state.

Transition L1:6: When the Link layer receives a PMREQ_P_P or a PMREQ_S_P from the Phy layer and is not enabled to perform power management modes or is not in a state to accept power mode requests, the Link layer shall make a transition to the LR0: L_PMDeny state. This transition is still valid if interface power states are supported and enabled as verified by Word 76 bit 9 set to one in IDENTIFY (PACKET) DEVICE data.

Transition L1:7: 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_P from the Phy layer or does not receive a PMREQ_x from the Phy layer the Link layer shall make a transition to the L1: L_IDLE state.

Transition L1:8: If the Phy layer becomes not ready even if the Transport layer is requesting an operation, the Link layer transitions to the L_NoCommErr state.

L2: L_SyncEscape state: This state is entered when the Link layer transmits $SYNC_P$ to escape a FIS transmission. The Link layer may choose to escape a FIS transmission due to a request from the Transport layer or due to an invalid state transition. This state is only entered by the initiator of the SYNC Escape.

When in this state, the Link layer transmits $SYNC_P$ and waits for a $SYNC_P$ from the Phy layer before proceeding to L_IDLE. The Link layer also transitions to L_IDLE if X_RDY_P is received in order to avoid a deadlock condition.

Transition L2:1: When the Link layer receives any Dword from the Phy that is not X_RDY_P or $SYNC_P$, the Link layer shall make a transition to the L2: L_SyncEscape state.

Transition L2:2: When the Link layer receives X_RDY_P or $SYNC_P$ from the Phy, the Link layer shall make a transition to the L1: L_IDLE state.

Transition L2:3: When the host Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LS1: L_NoCommErr state: This state is entered upon detection of a non ready condition of the Phy layer while attempting to process another state. The entry into this state heralds a relatively serious error condition in the Link layer. This state is executed only once so as to pass on the error condition up to the Transport layer.

Transition LS1:1: The transition is made to LS1:L_NoComm unconditionally.

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

Transition LS2:1: For as long as the Phy layer stays not ready, the transition is made to LS2: L_NoComm.

Transition LS2:2: When the Phy layer signals it is ready, a transition is made to LS3: L_SendAlign.

LS3: L_SendAlign state: This state is entered whenever an $ALIGN_P$ needs to be sent to the Phy layer.

Transition LS3:1: If the Phy layer becomes not ready, then a transition is made to LS1: L_NoCommErr.

Transition LS3:2: If the Phy layer indicates that it is ready, a transition is made to the L1: L_IDLE state.

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

Transition LS4:1: While the RESET control is active a transition is made back to the LS4: L_RESET state.

Transition LS4:2: When the RESET control goes inactive a transition is made to the LS2: L_NoComm state.

9.6.3 Link Transmit State Diagram

Table 70 – State Diagram Link Transmit

LT1: HL_SendChkRdy	Transmit X_RDY_P.	
1. R_RDY_P received from Phy.	→	L_SendSOF
2. X_RDY_P received from Phy.	→	L_RcvWaitFifo
3. AnyDword other than (R_RDY_P or X_RDY_P)[1] received from Phy layer.	→	HL_SendChkRdy
4. PHYRDYn	→	L_NoCommErr[2]
NOTES: 1. Any received errors such as 10b decoding errors and invalid primitives are ignored. 2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LT2: DL_SendChkRdy	Transmit X_RDY_P.	
1. R_RDY_P received from Phy.	→	L_SendSOF
2. AnyDword other than R_RDY_P received from Phy.	→	DL_SendChkRdy
3. PHYRDYn	→	L_NoCommErr

LT3: L_SendSOF	Transmit SOF_P	
1. PHYRDY[1]	→	L_SendData
2. PHYRDYn	→	L_NoCommErr[2]
3. $SYNC_P$ received from Phy.	→	L_IDLE[2]
NOTES: 1. Any received errors such as 10b decoding errors and invalid primitives are ignored. 2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LT4: L_SendData	Transmit data Dword	
1. More data to transmit and AnyDword other than ($HOLD_P$ or $DMAT_P$ or $SYNC_P$) received from Phy[1,2].	→	L_SendData
2. More data to transmit and $HOLD_P$ received from Phy.	→	L_RcvrHold
3. Data transmit not complete and data not ready to transmit and AnyDword other than $SYNC_P$ received from Phy.	→	L_SendHold
4. $DMAT_P$ received from Phy or data transmit complete and AnyDword other than $SYNC_P$ received from Phy.	→	L_SendCRC[5]
5. $SYNC_P$ received from Phy.	→	L_IDLE[3]
6. PHYRDYn	→	L_NoCommErr[3]
7. Transport layer indicates request to escape current frame[4].	→	L_SyncEscape

NOTES:
1. Any received errors such as 10b decoding errors and invalid primitives are ignored.
2. This makes possible a back channel during this time.
3. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.
4. When this condition is true, the associated transition has priority over all other transitions exiting this state.
5. The $DMAT_P$ signal is advisory and data transmission should be halted at the earliest opportunity but is not required to cease immediately. It is allowable to stay in the LT4: L_SendData state when there is more data to transmit and $DMAT_P$ is received.

LT5: L_RcvrHold	Transmit $HOLDA_P$.	
1. More data to transmit and AnyDword other than ($HOLD_P$ or $SYNC_P$ or $DMAT_P$) received from Phy with no DecErr.	→	L_SendData
2. More data to transmit and $HOLD_P$ received from Phy or DecErr.	→	L_RcvrHold
3. More data to transmit and $SYNC_P$ received from Phy.	→	L_IDLE[1]
4. More data to transmit and $DMAT_P$ received from Phy	→	L_SendCRC[3]
5. PHYRDYn.	→	L_NoCommErr[1]
6. Transport layer indicates request to escape current frame[2].	→	L_SyncEscape
7. $SYNC_P$ received from Phy.	→	L_IDLE[1]

NOTES:
1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.
2. When this condition is true, the associated transition has priority over all other transitions exiting this state.
3. The $DMAT_P$ signal is advisory and data transmission should be halted at the earliest opportunity but is not required to cease immediately. It is allowable to stay in the LT5: L_RcvrHold state when there is more data to transmit and $DMAT_P$ is received.

LT6: L_SendHold	Transmit $HOLD_P$.		
	1. More data ready to transmit and AnyDword other than ($HOLD_P$ or $SYNC_P$) received from Phy.	→	L_SendData
	2. More data ready to transmit and $HOLD_P$ received from Phy.	→	L_RcvrHold
	3. Data transmit not complete and data not ready to transmit and AnyDword other than ($SYNC_P$ or $DMAT_P$) received from Phy.	→	L_SendHold
	4. $DMAT_P$ received from Phy or data transmit complete and AnyDword other than $SYNC_P$ received from Phy.	→	L_SendCRC[3]
	5. $SYNC_P$ received from Phy.	→	L_IDLE[1]
	6. PHYRDYn	→	L_NoCommErr[1]
	7. Transport layer indicates request to escape current frame[2].	→	L_SyncEscape
	NOTES: 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer. 2. When this condition is true, the associated transition has priority over all other transitions exiting this state. 3. The $DMAT_P$ signal is advisory and data transmission should be halted at the earliest opportunity but is not required to cease immediately. It is allowable to stay in the LT6: L_SendHold state when there is more data to transmit and $DMAT_P$ is received.		

LT7: L_SendCRC	Transmit CRC.		
	1. PHYRDY and $SYNC_P$ not received from Phy.	→	L_SendEOF
	2. PHYRDYn	→	L_NoCommErr[1]
	3. PHYRDY and $SYNC_P$ received from Phy.	→	L_IDLE[1]
	NOTES: 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LT8: L_SendEOF	Transmit EOF_P.		
	1. PHYRDY and $SYNC_P$ not received from Phy.	→	L_Wait
	2. PHYRDYn	→	L_NoCommErr[1]
	3. PHYRDY and $SYNC_P$ received from Phy.	→	L_IDLE[1]
	NOTES: 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LT9: L_Wait	Transmit $WTRM_P$.	
1. R_OK_P received from Phy.	→	L_IDLE (good status)
2. R_ERR_P received from Phy.	→	L_IDLE (bad status)
3. $SYNC_P$ received from Phy.	→	L_IDLE[1]
4. AnyDword other than (R_OK_P or R_ERR_P or $SYNC_P$) received from Phy.	→	L_Wait
5. PHYRDYn	→	L_NoCommErr[1]
NOTES: 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

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 X_RDY_P and waits for X_RDY_P or R_RDY_P from the Phy layer.

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

Transition LT1:1: When the host Link layer receives R_RDY_P from the Phy layer, the Link layer shall make a transition to the LT3: L_SendSOF state.

Transition LT1:2: When the host Link layer receives X_RDY_P from the Phy layer, the Link layer shall make a transition to the LR2: L_RcvWaitFifo state.

Transition LT1:3: When the host Link layer receives any Dword other than R_RDY_P or X_RDY_P from the Phy layer, the Link layer shall make a transition to the LT1: HL_SendChkRdy state.

Transition LT1:4: When the host Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

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 X_RDY_P and waits for R_RDY_P from the Phy layer.

Transition LT2:1: When the device Link layer receives R_RDY_P from the Phy layer, the Link layer shall make a transition to the LT3: L_SendSOF state.

Transition LT2:2: When the device Link layer does not receive R_RDY_P from the Phy layer, the Link layer shall make a transition to the LT2: DL_SendChkRdy state.

Transition LT2:3: When the device Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LT3: L_SendSOF state: This state is entered when R_RDY_P has been received from the Phy layer.

When in this state, the Link layer transmits SOF_P.

Transition LT3:1: When the device Link layer has transmitted SOF_P, the Link layer shall make a transition to the LT4: L_SendDATA state.

Transition LT3:2: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT3:3: When the Link layer receives $SYNC_P$ from the Phy 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 SOF_P 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 LT4:1: When the Link layer receives any Dword other than a $HOLD_P$, $DMAT_P$, or $SYNC_P$ primitive from the Phy layer and the Transport layer indicates a Dword is available for transfer, the Link layer shall make a transition to the LT4: L_SendData state. The $DMAT_P$ 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 $DMAT_P$ is received.

Transition LT4:2: When the device Link layer receives $HOLD_P$ from the Phy layer, the Link layer shall make a transition to the LT5: L_RcvrHold state.

Transition LT4:3: When the Transport layer indicates that the next Dword is not available to transfer and any Dword other than $SYNC_P$ has been received from the Phy layer, the Link layer shall make a transition to the LT6: L_SendHold state.

Transition LT4:4: When the Transport layer indicates that all data for the frame has been transferred and any Dword other than $SYNC_P$ has been received from the Phy layer, the Link layer shall make a transition to the LT7: L_SendCRC state. When the Link layer receives $DMAT_P$ from the Phy layer, it shall notify the Transport layer and terminate the transmission in progress as described in section 9.4.4 and shall transition to the LT7: L_SendCRC state.

Transition LT4:5: When the Link layer receives $SYNC_P$ from the Phy 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
Transition LT4:6: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT4:7: When the Link layer receives notification from the Transport layer that the current frame transfer should be escaped, a transition to the L_SyncEscape state shall be made.

LT5: L_RcvrHold state: This state is entered when $HOLD_P$ has been received from the Phy layer.

When in this state, the Link layer shall transmit $HOLDA_P$.

Transition LT5:1: When the Link layer receives any Dword other than a $HOLD_P$, $SYNC_P$, or a $DMAT_P$ primitive from the Phy layer with no decoding error detected, 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 LT5:2: When the device Link layer receives $HOLD_P$ from the Phy layer or a decoding error was detected, the Link layer shall make a transition to the LT5: L_RcvrHold state.

Transition LT5:3: When the Link layer receives $SYNC_P$ from the Phy 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.

Transition LT5:4: When the Link layer receives $DMAT_P$ from the Phy layer, it shall notify the Transport layer and terminate the transmission in progress as described in section 9.4.4 and shall transition to the LT7: L_SendCRC state.

Transition LT5:5: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT5:6: When the Link layer receives notification from the Transport layer that the current frame should be escaped, a transition to the L_SyncEscape state shall be made.

Transition LT5:7: When the Link layer receives $SYNC_P$ from the Phy 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.

LT6: L_SendHold state: This state is entered when the Transport layer indicates a Dword is not available for transfer and $HOLD_P$ has not been received from the Phy layer.

When in this state, the Link layer shall transmit $HOLD_P$.

Transition LT6:1: When the Link layer receives any Dword other than a $HOLD_P$ or $SYNC_P$ primitive from the Phy 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 LT6:2: When the Link layer receives $HOLD_P$ from the Phy 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 LT6:3: When the Transport layer indicates that a Dword is not available for transfer and any Dword other than $SYNC_P$ is received from the Phy layer, the Link layer shall make a transition to the LT6: L_SendHold state.

Transition LT6:4: When the Transport layer indicates that all data for the frame has been transferred and any Dword other than $SYNC_P$ has been received from the Phy layer, the Link layer shall make a transition to the LT7: L_SendCRC state. When the Link layer receives $DMAT_P$ from the Phy layer, it shall notify the Transport layer and terminate the transmission in progress as described in section 9.4.4 and shall transition to the LT7:L_SendCRC state.

Transition LT6:5: When the Link layer receives $SYNC_P$ from the Phy 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.

Transition LT6:6: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT6:7: When the Link layer receives notification from the Transport layer that the current frame should be escaped, a transition to the L_SyncEscape state shall be made.

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 LT7:1: When the CRC has been transmitted, the Link layer shall make a transition to the LT8: L_SendEOF state.

Transition LT7:2: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT7:3: When the Link layer receives $SYNC_P$ from the Phy 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.

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 EOF_P.

Transition LT8:1: When EOF_P has been transmitted, the Link layer shall make a transition to the LT9: L_Wait state.

Transition LT8:2: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LT8:3: When the Link layer receives $SYNC_P$ from the Phy 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.

LT9: L_Wait state: This state is entered when EOF_P has been transmitted.

When in this state, the Link layer shall transmit $WTRM_P$.

Transition LT9:1: When the Link layer receives R_OK_P from the Phy layer, the Link layer shall notify the Transport layer and make a transition to the L1: L_IDLE state.

Transition LT9:2: When the Link layer receives R_ERR_P from the Phy layer, the Link layer shall notify the Transport layer and make a transition to the L1: L_IDLE state.

Transition LT9:3: When the Link layer receives $SYNC_P$ from the Phy layer, the Link layer shall notify the Transport layer and make a transition to the L1: L_IDLE state.

Transition LT9:4: When the Link layer receives any Dword other than an R_OK_P, R_ERR_P, or $SYNC_P$ primitive from the Phy layer, the Link layer shall make a transition to the LT9: L_Wait state.

Transition LT9:5: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

9.6.4 Link Receive State Diagram

Table 71 – State Diagram Link Receive

LR1: L_RcvChkRdy	Transmit R_RDY_P.	
1. X_RDY_P received from Phy.	→	L_RcvChkRdy
2. SOF_P received from Phy.	→	L_RcvData
3. Any Dword other than (X_RDY_P or SOF_P) received from Phy.	→	L_IDLE
4. PHYRDYn	→	L_NoCommErr[1]
NOTES: 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR2: L_ RcvWaitFifo	Transmit $SYNC_P$.	
1. X_RDY_P received from Phy and FIFO space available.	→	L_RcvChkRdy
2. X_RDY_P received from Phy and FIFO space not available.	→	L_RcvWaitFifo
3. Any Dword other than X_RDY_P received from Phy.	→	L_IDLE
4. PHYRDYn	→	L_NoCommErr[1]
NOTES: 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR3: L_RcvData	Transmit R_IP_P or $DMAT_P$[1].	
1. (DatDword received from Phy and FIFO space) or $HOLDA_P$ received from Phy.	→	L_RcvData
2. DatDword received from Phy and insufficient FIFO space.	→	L_Hold
3. $HOLD_P$ received from Phy.	→	L_RcvHold
4. EOF_P received from Phy.	→	L_RcvEOF
5. $WTRM_P$ received from Phy.	→	L_BadEnd
6. $SYNC_P$ received from Phy.	→	L_IDLE
7. AnyDword other than ($HOLD_P$ or EOF_P or $HOLDA_P$ or $SYNC_P$ or $WTRM_P$) received from Phy.	→	L_RcvData
8. PHYRDYn	→	L_NoCommErr[2]
9. Transport layer indicates request to escape current frame.	→	L_SyncEscape
NOTES: 1. If the Transport layer signals that it wishes to terminate the transfer, $DMAT_P$ is transmitted in place of R_IP_P. 2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR4: L_Hold	Transmit $HOLD_P$.	
1. FIFO space available and AnyDword other than $HOLD_P$ or EOF_P received from Phy.	→	L_RcvData
2. FIFO space available and $HOLD_P$ received from Phy.	→	L_RcvHold
3. EOF_P received from Phy.	→	L_RcvEOF
4. No FIFO space available and EOF_P not received from Phy and $SYNC_P$ not received from Phy and PHYRDY	→	L_Hold
5. PHYRDYn	→	L_NoCommErr[1]
6. $SYNC_P$ received from Phy.	→	L_IDLE
7. Transport layer indicates request to escape current frame.	→	L_SyncEscape
NOTES: 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR5: L_RcvHold	Transmit $HOLDA_P$ or $DMAT_P$ [1].	
1. AnyDword other than ($HOLD_P$ or EOF_P or $SYNC_P$) received from Phy.	→	L_RcvData
2. $HOLD_P$ received from Phy.	→	L_RcvHold
3. EOF_P received from Phy.	→	L_RcvEOF
4. $SYNC_P$ received from Phy.	→	L_IDLE
5. PHYRDYn	→	L_NoCommErr[2]
6. Transport layer indicates request to escape current frame.	→	L_SyncEscape
NOTES: 1. If the Transport layer signals that it wishes to terminate the transfer, $DMAT_P$ is transmitted in place of $HOLDA_P$. 2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR6: L_RcvEOF	Transmit R_IP_P.	
1. CRC check not complete.	→	L_RcvEOF
2. CRC good.	→	L_GoodCRC
3. CRC bad.	→	L_BadEnd
4. PHYRDYn	→	L_NoCommErr[1]
NOTES: 1. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR7: L_GoodCRC	Transmit R_IP$_P$.		
	1. Transport layer indicated good result.	→	L_GoodEnd
	2. Transport layer indicates unrecognized FIS.	→	L_BadEnd
	3. Transport layer has yet to respond.	→	L_GoodCRC
	4. PHYRDYn	→	L_NoCommErr[2]
	5. Transport or Link layer indicated error detected during reception of recognized FIS.	→	L_BadEnd
	6. SYNC$_P$ received from Phy.	→	L_IDLE
	NOTES: 1. Upon entering this state for the first time, the Link layer shall notify the Transport layer that the CRC for this frame is valid. 2. The Link layer shall notify the Transport layer of the condition and fail the attempted transfer.		

LR8: L_GoodEnd	Transmit R_OK$_P$.		
	1. SYNC$_P$ received from Phy.	→	L_IDLE
	2. AnyDword other than SYNC$_P$ received from Phy.	→	L_GoodEnd
	3. PHYRDYn	→	L_NoCommErr

LR9: L_BadEnd	Transmit R_ERR$_P$.		
	1. SYNC$_P$ received from Phy.	→	L_IDLE
	2. AnyDword other than SYNC$_P$ received from Phy.	→	L_BadEnd
	3. PHYRDYn	→	L_NoCommErr

LR1: L_RcvChkRdy state: This state is entered when X_RDY$_P$ has been received from the Phy layer.

When in this state, the Link layer shall transmit R_RDY$_P$ and wait for SOF$_P$ from the Phy layer.

Transition LR1:1: When the Link layer receives X_RDY$_P$ from the Phy layer, the Link layer shall make a transition to the LR1: L_RcvChkRdy state.

Transition LR1:2: When the Link layer receives SOF$_P$ from the Phy layer, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR1:3: When the Link layer receives any Dword other than an X_RDY$_P$ or SOF$_P$ primitive from the Phy layer, the Link layer shall notify the Transport layer of the condition and make a transition to the L1: L_IDLE state.

Transition LR1:4: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR2: L_RcvWaitFifo state: This state is entered when an X_RDY$_P$ has been received, and the FIFO is not ready to receive a FIS.

When in this state, the Link layer shall transmit SYNC$_P$.

Transition LR2:1: When the Link layer receives X_RDY_P from the Phy layer and the FIFO is ready to accept data, the Link layer shall make a transition to the LR1: L_RcvChkRdy state.

Transition LR2:2: When the Link layer receives X_RDY_P from the Phy layer and the FIFO is not ready to accept data, the Link layer shall make a transition to the LR2: L_RcvWaitFifo state.

Transition LR2:3: When the Link layer receives any Dword other than X_RDY_P from the Phy layer, the Link layer shall notify the Transport layer of the condition and make a transition to the L1: L_IDLE state.

Transition LR2:4: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR3: L_RcvData state: This state is entered when SOF_P has been received from the Phy layer.

When in this state, the Link layer receives an encoded character sequence from the Phy 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 R_IP_P to signal transmission to continue or transmits $DMAT_P$ to signal the transmitter to terminate the transmission.

Transition LR3:1: 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 LR3:2: When the Transport layer indicates that sufficient space is not available in its FIFO, the Link layer shall make a transition to the LR4: L_Hold state.

Transition LR3:3: When the Link layer receives $HOLD_P$ from the Phy layer, the Link layer shall make a transition to the LR5: L_RcvHold state.

Transition LR3:4: When the Link layer receives EOF_P from the Phy layer, the Link layer shall make a transition to the LR6: L_RcvEOF state.

Transition LR3:5: When the Link layer receives $WTRM_P$ from the Phy layer, the Link layer shall make a transition to the LR9: L_BadEnd state.

Transition LR3:6: When the Link layer receives $SYNC_P$ from the Phy layer, the Link layer shall notify the Transport layer that reception was aborted and shall make a transition to the L1: L_IDLE state.

Transition LR3:7: When the Link layer receives any Dword other than a $HOLD_P$, $HOLDA_P$, EOF_P, or $SYNC_P$ primitive from the Phy layer, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR3:8: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR3:9: When the Link layer receives notification from the Transport layer that the current frame should be escaped, a transition to the L_SyncEscape state shall be made.

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 $HOLD_P$ and may receive an encoded character from the Phy layer.

Transition LR4:1: When the Link layer receives any Dword other than a $HOLD_P$ primitive from the Phy 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 LR4:2: When the Link layer receives $HOLD_P$ from the Phy 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_RcvHold state.

Transition LR4:3: When the Link layer receives EOF_P from the Phy layer, the Link layer shall make a transition to the LR6: L_RcvEOF state. Note that due to pipeline latency, an EOF_P may be received when in the L_Hold state in which case the receiving Link shall use its FIFO headroom to receive the EOF_P and close the frame reception.

Transition LR4:4: When the Transport layer indicates that there is not sufficient space available in its FIFO and the Phy layer is ready, the Link layer shall make a transition to the LR4: L_Hold state.

Transition LR4:5: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR4:6: When the Link layer receives $SYNC_P$ from the Phy 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.

Transition LR4:7: When the Link layer receives notification from the Transport layer that the current frame should be escaped, a transition to the L_SyncEscape state shall be made.

LR5: L_RcvHold state: This state is entered when $HOLD_P$ has been received from the Phy layer.

When in this state, the Link layer shall either transmit $HOLDA_P$ to signal transmission to proceed when the transmitter becomes ready or transmit $DMAT_P$ to signal the transmitter to terminate the transmission.

Transition LR5:1: When the Link layer receives any Dword other than a $HOLD_P$ or $SYNC_P$ primitive from the Phy layer, the Link layer shall make a transition to the LR3: L_RcvData state.

Transition LR5:2: When the Link layer receives $HOLD_P$ from the Phy layer, the Link layer shall make a transition to the LR5: L_RcvHold state.

Transition LR5:3: When the Link layer receives EOF_P from the Phy layer, the Link layer shall make a transition to the LR6: L_RcvEOF state.

Transition LR5:4: When the Link layer receives $SYNC_P$ from the Phy 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.

Transition LR5:5: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR5:6: When the Link layer receives notification from the Transport layer that the current frame should be escaped, a transition to the L_SyncEscape state shall be made.

LR6: L_RcvEOF state: This state is entered when the Link layer has received EOF_P from the Phy layer.

When in this state, the Link layer shall check the calculated CRC for the frame and transmit one or more R_IP_P primitives.

Transition LR6:1: If the CRC calculation and check is not yet completed, the Link layer shall make a transition to the LR6: L_RcvEOF state.

Transition LR6:2: When the CRC indicates no error, the Link layer shall notify the Transport layer and make a transition to the LR7: L_GoodCRC state.

Transition LR6:3: When the CRC indicates an error has occurred, the Link layer shall notify the Transport layer and make a transition to the LR9: L_BadEnd state.

Transition LR6:4: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR7: L_GoodCRC state: This state is entered when the CRC for the frame has been checked and determined to be good.

When in this state, the Link layer shall wait for the Transport layer to check the frame and transmit one or more R_IP_P primitives.

Transition LR7:1: When the Transport layer indicates a good result, the Link layer shall transition to the LR8: L_GoodEnd state.

Transition LR7:2: When the Transport layer indicates an unrecognized FIS, the Link layer shall transition to the LR9: L_BadEnd state.

Transition LR7:3: If the Transport layer has not supplied status, then the Link layer shall transition to the LR7: L_GoodCRC state.

Transition LR7:4: When the Link layer detects that the Phy layer is not ready, the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

Transition LR7:5: When the Transport layer or Link layer indicates an error was encountered during the reception of the recognized FIS, the Link layer shall transition to the LR9: L_BadEnd state.

Transition LR7:6: When the Link layer receives $SYNC_P$ from the Phy 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 R_OK_P.

Transition LR8:1: When the Link layer receives $SYNC_P$ from the Phy layer, the Link layer shall make a transition to the L1: L_IDLE state.

Transition LR8:2: When the Link layer receives any Dword other than a $SYNC_P$ primitive from the Phy layer, the Link layer shall make a transition to the LR7: L_GoodEnd state.

Transition LR8:3: When the Link layer detects that the Phy layer is not ready, the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

LR9: L_BadEnd state: This state is entered when the CRC for the frame has been checked and determined to be bad or when the Transport layer has notified the Link layer that the received FIS is invalid.

When in this state, the Link layer shall transmit R_ERR_P.

Transition LR9:1: When the Link layer receives $SYNC_P$ from the Phy layer, the Link layer shall make a transition to the L1: L_IDLE state.

Transition LR9:2: When the Link layer receives any Dword other than $SYNC_P$ from the Phy layer, the Link layer shall make a transition to the LR9: BadEnd state.

Transition LR9:3: When the Link layer detects that the Phy layer is not ready the Link layer shall notify the Transport layer of the condition and make a transition to the LS1: L_NoCommErr state.

9.6.5 Link Power Mode State Diagram

Table 72 – State Diagram Link Power Mode

LPM1: L_TPMPartial	Transmit $PMREQ\_P_P$.	
1. $PMACK_P$ received from Phy layer.	→	L_ChkPhyRdy
2. X_RDY_P received from Phy layer.	→	L_RcvWaitFifo[1]
3. $SYNC_P$ or R_OK_P received from Phy layer.	→	L_TPMPartial
4. AnyDword other than ($PMACK_P$ or $PMNAK_P$ or X_RDY_P or $SYNC_P$ or R_OK_P or $PMREQ\_P_P$[3] or $PMREQ\_S_P$[3])[1] received from Phy layer.	→	L_IDLE
5. $PMREQ\_P_P$ or $PMREQ\_S_P$ received from Phy layer.	→	L_TPMPartial[3]
6. PHYRDYn	→	L_NoCommErr[2]
7. $PMNAK_P$ received from Phy layer.	→	L_NoPmnak

NOTE:
1. This transition aborts the request from the Transport layer to enter a power mode. A status indication to the Transport layer of this event is required.
2. This is an unexpected transition and constitutes an error condition. An error condition needs to be sent to the Transport layer as a result.
3. If $PMREQ\_P_P$ or $PMREQ\_S_P$ is received, the host shall make a transition to the L_IDLE state, but the device shall make a transition to the L_TPMPartial state.

LPM2: L_TPMSlumber	Transmit PMREQ_S_P.		
	1. $PMACK_P$ received from Phy layer.	→	L_ChkPhyRdy
	2. X_RDY_P received from Phy layer.	→	L_RcvWaitFifo[1]
	3. $SYNC_P$ or R_OK_P received from Phy layer.	→	L_TPMSlumber
	4. AnyDword other than ($PMACK_P$ or $PMNAK_P$ or X_RDY_P or $SYNC_P$ or R_OK_P or PMREQ_P_P[3] or PMREQ_S_P[3])[1] received from Phy layer.	→	L_IDLE
	5. PMREQ_P_P or PMREQ_S_P received from Phy layer.	→	L_TPMSlumber[3]
	6. PHYRDYn	→	L_NoCommErr[2]
	7. $PMNAK_P$ received from Phy layer.	→	L_NoPmnak
	NOTE: 1. This transition aborts the request from the Transport layer to enter a power mode. A status indication to the Transport layer of this event is required. 2. This is an unexpected transition and constitutes an error condition. An error condition needs to be sent to the Transport layer as a result. 3. If PMREQ_P_P or PMREQ_S_P is received, the host shall make a transition to the L_IDLE state, but the device shall make a transition to the L_TPMSlumber state.		

LPM3: L_PMOff	Transmit $PMACK_P$[1].		
	1. A total of 4<=n<=16 $PMACK_P$ primitives sent.	→	L_ChkPhyRdy
	2. Less than n $PMACK_P$ primitives sent.	→	L_PMOff
	NOTE: 1. A flag is set according to whether a PMREQ_P_P or PMREQ_S_P was received from the Phy layer.		

LPM4: L_PMDeny	Transmit $PMNAK_P$.		
	1. PMREQ_P_P or PMREQ_S_P received from Phy layer.	→	L_PMDeny
	2. AnyDword other than (PMREQ_P_P or PMREQ_S_P) received from Phy layer.	→	L_IDLE
	3. PHYRDYn	→	L_NoCommErr

LPM5: L_ChkPhyRdy	Assert Partial/Slumber to Phy layer (as appropriate).		
	1. PHYRDY	→	L_ChkPhyRdy
	2. PHYRDYn	→	L_NoCommPower

LPM6: L_NoCommPower	Maintain Partial/Slumber assertion (as appropriate).		
	1. Transport layer requests a wakeup or COMWAKE detected.	→	L_WakeUp1
	2. Transport layer not requesting wakeup and COMWAKE not detected.	→	L_NoCommPower

LPM7: L_WakeUp1	Negate both Partial and Slumber.		
	1. PHYRDY	→	L_WakeUp2
	2. PHYRDYn	→	L_WakeUp1

LPM8: L_WakeUp2	Transmit $ALIGN_P$.		
	1. PHYRDY	→	L_IDLE
	2. PHYRDYn	→	L_NoCommErr

LPM9: L_NoPmnak	Transmit $SYNC_P$.		
	1. $PMNAK_P$ received from Phy layer.	→	L_NoPmnak
	2. AnyDword other than ($PMNAK_P$) received from Phy layer.	→	L_IDLE

LPM1: L_TPMPartial state: This state is entered when the Transport layer has indicated that a transition to the Partial power state is desired.

Transition LPM1:1: When in this state $PMREQ\_P_P$ shall be transmitted. When the Link layer receives $PMACK_P$ a transition to the LPM5: L_ChkPhyRdy state shall be made.

Transition LPM1:2: If the Link layer receives X_RDY_P a transition shall be made to the LR2: L_RcvWaitFifo state, effectively aborting the request to a power mode state.

Transition LPM1:3: If the Link layer receives a $SYNC_P$ or R_OK_P primitive, then it is assumed that the opposite side has not yet processed $PMREQ\_P_P$ yet and time is needed. A transition to the LPM1: L_TPMPartial state shall be made.

Transition LPM1:4: If the host Link layer receives any Dword from the Phy layer other than a $PMACK_P$, $PMNAK_P$, X_RDY_P, $SYNC_P$ or R_OK_P primitive, then the request to enter the Partial state is aborted and a transition to L1: L_IDLE shall be made. If the device Link layer receives any Dword from the Phy layer other than a $PMACK_P$, $PMNAK_P$, X_RDY_P, $SYNC_P$, $PMREQ\_P_P$, $PMREQ\_S_P$, or R_OK_P primitive, then the request to enter the Partial state is aborted and a transition to L1: L_IDLE shall be made.

Transition LPM1:5: The host Link layer shall not make this transition as it applies only to the device Link layer. If the device Link layer receives $PMREQ\_P_P$ or $PMREQ\_S_P$ from the host, it shall remain in this state by transitioning back to LPM1: L_TPMPartial.

Transition LPM1:6: If the Link layer detects that the Phy layer has become not ready, this is interpreted as an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1: L_NoCommErr state.

Transition LPM1:7: If the Link layer receives a $PMNAK_P$, then the request to enter the Partial state is aborted and a transition to LPM9: L_NoPmnak shall be made.

LPM2: L_TPMSlumber state: This state is entered when the Transport layer has indicated that a transition to the Slumber power state is desired.

Transition LPM2:1: When in this state $PMREQ\_S_P$ shall be transmitted. When the Link layer receives $PMACK_P$, a transition to the LPM5: L_ChkPhyRdy state shall be made.

Transition LPM2:2: If the Link layer receives X_RDY_P, a transition to the LR2: L_RcvWaitFifo state shall be made, effectively aborting the request to a power mode state.

Transition LPM2:3: If the Link layer receives $SYNC_P$ or R_OK_P, then it is assumed that the opposite side has not yet processed $PMREQ\_S_P$ yet and time is needed. The transition to the LPM2: L_TPMSlumber state shall be made.

Transition LPM2:4: If the host Link layer receives any Dword from the Phy layer other than a $PMACK_P$, $PMNAK_P$, X_RDY_P, $SYNC_P$, or R_OK_P primitive, then the request to enter the Slumber state is aborted and a transition to L1: L_IDLE shall be made. If the device Link layer receives any Dword from the Phy layer other than a $PMACK_P$, $PMNAK_P$, X_RDY_P, $SYNC_P$, $PMREQ\_P_P$, $PMREQ\_S_P$, or R_OK_P primitive, then the request to enter the Slumber state is aborted and a transition to L1: L_IDLE shall be made.

Transition LPM2:5: The host Link layer shall not make this transition as it applies only to the device Link layer. If the device Link layer receives $PMREQ\_P_P$ or $PMREQ\_S_P$ from the host, it shall remain in this state by transitioning back to LPM2: L_TPMSlumber.

Transition LPM2:6: If the Link layer detects that the Phy layer has become not ready, this is interpreted as an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the L_NoCommErr state.

Transition LPM2:7: If the Link layer receives a $PMNAK_P$, then the request to enter the Slumber state is aborted and a transition to LPM9: L_NoPmnak shall be made.

LPM3: L_PMOff state: This state is entered when either $PMREQ\_S_P$ or $PMREQ\_P_P$ was received by the Link layer. The Link layer transmits $PMACK_P$ for each execution of this state.

Transition LPM3:1: If 4<=n<=16 $PMACK_P$ primitives have been transmitted, a transition shall be made to the L_ChkPhyRdy state.

Transition LPM3:2: If less than n $PMACK_P$ primitives have been transmitted, a transition shall be made to 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 $PMNAK_P$ to inform the opposite end that a power mode is not allowed.

Transition LPM4:1: 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 LPM4:2: 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 LPM4:3: If the Link layer detects that the Phy layer has become not ready, this is interpreted as an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1: L_NoCommErr state.

LPM5: L_ChkPhyRdy state: This state is entered whenever it is desired for the Phy layer to enter a low power condition. For each execution in this state a request is made to the Phy layer to enter the state and deactivate the PHYRDY signal. Partial or Slumber is asserted to the Phy layer as appropriate.

Transition LPM6:1: If the Phy layer has not yet processed the request to enter the power saving state and not deactivated the PHYRDY signal, then the Link layer shall remain in the LPM5: L_ChkPhyRdy state and continue to request the Phy layer to enter the power mode state.

Transition LPM6:2: When the Phy layer has processed the power mode request and has deactivated the PHYRDY signal, then a transition shall be made to the LPM6: L_NoCommPower state.

LPM6: L_NoCommPower state: This state is entered when the Phy layer has negated its PHYRDY signal indicating that it is in either Partial or Slumber state. In this state, the Link layer waits for the OOB detector to signal reception of the COMWAKE signal (for a wakeup initiated by the other device), or for the Transport layer to request a wakeup.

Transition LPM6:1: If the Transport layer requests a wakeup or the OOB signal detector indicates reception of the COMWAKE signal, then a transition shall be made to LPM7: L_WakeUp1

Transition LPM6:2: If the Transport layer does not request a wakeup and the OOB detector does not indicate reception of the COMWAKE signal, then a transition shall be made to LPM6: L_NoCommPower.

LPM7: L_WakeUp1 state: This state is entered when the Transport layer has initiated a wakeup. In this state, the Link layer shall negate both Partial and Slumber to the Phy layer, and wait for the PHYRDY signal from the Phy layer to be asserted. While in this state the Phy layer is performing the wakeup sequence.

Transition LPM7:1 When the Phy layer asserts its PHYRDY signal, a transition shall be made to LPM8: L_WakeUp2.

Transition LPM7:2: When the Phy layer remains not ready, a transition shall be made to LPM7: L_WakeUp1.

LPM8: L_WakeUp2 state: This state is entered when the Phy layer 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 LPM8:1 If the Phy layer keeps PHYRDY asserted, a transition shall be made to the L1: L_IDLE state.

Transition LPM8:2 If the Phy layer negates PHYRDY, this is an error condition. The Transport layer shall be notified of the condition and a transition shall be made to the LS1: L_NoCommErr state.

LPM9: L_NoPmnak state: This state is entered when the Link layer has indicated that a request to enter the Slumber or Partial state has been denied. The Link layer transmits $SYNC_P$ for each execution of this state. In this state, the Link layer waits for receipt of any Dword that is not $PMNAK_P$ from the Phy layer.

Transition LPM9:1: If the Link layer receives $PMNAK_P$, then the Link layer shall remain in the LPM9: L_NoPmnak state and continue to wait for receipt of a primitive that is not $PMNAK_P$ from the Phy layer.

Transition LPM9:2: If the Link layer receives any Dword from the Phy layer other than $PMNAK_P$, then the request to enter the power management state is aborted and a transition to L1: L_IDLE shall be made.

10 Transport Layer

10.1 Overview

The Transport layer need not be cognizant of how frames are transmitted and received. The Transport layer simply constructs Frame Information Structures (FISes) for transmission and decomposes received Frame Information Structures. Host and device Transport layer state differ in that the source of the FIS content differs. The Transport layer maintains no context in terms of ATA commands or previous FIS content.

10.1.1 FIS construction

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

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

10.1.2 FIS decomposition

When a FIS is received from the Link layer, the Transport layer provides the following services:

- Receives the FIS from the Link layer.
- Determines FIS type.
- Distributes the FIS content to the locations indicated by the FIS type.
- For the host Transport layer, receipt of a FIS may also cause the construction of a FIS to be returned to the device.
- Reports good reception or errors to higher layer

10.2 Frame Information Structure (FIS)

10.2.1 Overview

A FIS is a group of Dwords that convey information between host and device as described previously. Primitives are used to define the boundaries of the FIS and may be inserted to control the rate of the information flow. This section describes the information content of the FIS - referred to as payload - and assumes the reader is aware of the primitives that are needed to support the information content.

The contents of the info field is divided into three categories: (1) register type, (2) setup type, and (3) data type. For each category the organization of each frame is defined in the following section.

10.2.2 Payload content

The type and layout of the payload is indicated by the Frame Information Type field located in byte 0 of the first Dword of the payload. See Figure 194 as an example. This example type is used primarily to transfer the contents of the Shadow Register Block Registers from the host to the device. Table 73 may be referenced to refresh the reader's memory of a simplified version of the Shadow Register Block organization of an ATA adapter.

Table 73 – Simplified Shadow Register Block register numbering

				Register access operation	
	A2	**A1**	**A0**	**Read**	**Write**
CS 0 Active	0	0	0	Data Port	
	0	0	1	Error	Features
	0	1	0	Sector Count [15:8], [7:0]	
	0	1	1	LBA Low [31:24], [7:0]	
	1	0	0	LBA Mid [39:32], [15:8]	
	1	0	1	LBA High [47:40], [23:16]	
	1	1	0	Device / Head	
	1	1	1	Status	Command
CS 1 active	1	1	0	Alternate Status	Device Control

The following sections detail the types of payloads that are possible. The SOF_P, EOF_P and $HOLD_P$ primitives have been removed for clarity.

10.3 FIS Types

The following sections define the structure of each individual FIS.

10.3.1 FIS Type values

The value for the FIS Type fields of all FISes has been selected 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 FIS Type value may be acted upon prior to the integrity of the complete FIS being checked against its ending CRC, the FIS Type field values have been selected to maximize the Hamming distance between them.

Figure 193 enumerates the FIS Type values and their assignments.

Type field value	Description
27h	Register FIS – Host to Device
34h	Register FIS – Device to Host
39h	DMA Activate FIS – Device to Host
41h	DMA Setup FIS – Bi-directional
46h	Data FIS – Bi-directional
58h	BIST Activate FIS – Bi-directional
5Fh	PIO Setup FIS – Device to Host
A1h	Set Device Bits FIS – Device to Host
A6h	Reserved for future Serial ATA definition
B8h	Reserved for future Serial ATA definition
BFh	Reserved for future Serial ATA definition
C7h	Vendor specific
D4h	Vendor specific
D9h	Reserved for future Serial ATA definition

Figure 193 – FIS type value assignments

10.3.1.1 Unrecognized FIS Types

A device or host may receive a FIS type that is not defined or vendor specific in Figure 193. The receiver of a FIS determines whether to handle a FIS type that is not defined or reserved as "unrecognized". There may be cases where a receiver accepts undefined or vendor specific FISes. The host and device should negotiate any undefined or vendor specific FIS types that may be transmitted prior to their use.

If the receiver decides to treat a FIS type as unrecognized, it shall follow the Link layer state machine definitions in section 9.6 upon receipt of that FIS type.

10.3.2 CRC Errors on Data FISes

Following a Serial ATA CRC error on a Data FIS, if the device transmits a Device-to-Host FIS it shall set the ERR bit to one and both the BSY bit and DRQ bit cleared to zero in the Status field, and the ABRT bit set to one in the Error field. It is recommended for the device to also set the bit 7 (i.e. ICRC bit) to one in the Error field. See ATA8-ACS.

There is no Device-to-Host FIS transmitted after a Serial ATA CRC error on the last Data FIS of a PIO-in command nor following a Serial ATA CRC error on the ATAPI command packet transfer. Thus, there is no mechanism for the device to indicate a Serial ATA CRC error to the host in either of these cases. The host should check the SError register to determine if a Link layer error has occurred in both of these cases.

10.3.3 All FIS types

In all of the following FIS structures the following rules shall apply:

1. All reserved fields shall be written or transmitted as all zeroes
2. All reserved fields shall be ignored during the reading or reception process.

10.3.4 Register - Host to Device

0	Features(7:0)		Command		C	R	R	R	PM Port	FIS Type (27h)
1	Device		LBA(23:16)		LBA(15:8)					LBA(7:0)
2	Features(15:8)		LBA(47:40)		LBA(39:32)					LBA(31:24)
3	Control		ICC 7 6 5 4 3 2 1 0		Count(15:8)					Count(7:0)
4	Reserved (0)		Reserved (0)		Reserved (0)					Reserved (0)

Figure 194 – Register - Host to Device FIS layout

If a field in this FIS is not defined by a command, it shall be Reserved for that command.

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 Device Control register. Setting C bit to one and SRST bit to one in the Device Control Field is invalid and results in indeterminate behavior.

Command - Contains the contents of the Command register of the Shadow Register Block.

LBA(7:0) - Contains the contents of the LBA Low register of the Shadow Register Block.

Control - Contains the contents of the Device Control register of the Shadow Register Block.

LBA(15:8) - Contains the contents of the LBA Mid register of the Shadow Register Block.

LBA(39:32) – Contains the contents of the expanded address field of the Shadow Register Block

LBA(23:16) - Contains the contents of the LBA High register of the Shadow Register Block.

LBA(47:40) – Contains the contents of the expanded address field of the Shadow Register Block

Device - Contains the contents of the Device register of the Shadow Register Block.

Features(7:0) - Contains the contents of the Features register of the Shadow Register Block.

Features(15:8) – Contains the contents of the expanded address field of the Shadow Register Block

PM Port – When an endpoint device is attached via a Port Multiplier, specifies the device port address that the FIS should be delivered to. This field is set by the host.

R – Reserved – shall be cleared to zero.

Count(7:0) - Contains the contents of the Sector Count register of the Shadow Register Block.

Count(15:8) – Contains the contents of the expanded address field of the Shadow Register Block

LBA(7:0) - Contains the contents of the LBA Low register of the Shadow Register Block.

LBA(31:24) – Contains the contents of the expanded address field of the Shadow Register Block

ICC - Isochronous Command Completion (ICC) contains a value is set by the host to inform device of a time limit. If a command does not define the use of this field, it shall be reserved.

10.3.4.1 Description

The Register – Host to Device FIS is used to transfer the contents of the Shadow Register Block from the host to the device. This is the mechanism for issuing ATA commands to the device.

10.3.4.2 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 Device Control register with a value different than is currently in the Device Control register in the host adapter's Shadow Register Block. Upon initiating transmission, the current contents of the Shadow Register Block are transmitted and the C bit in the FIS is set according to whether the transmission was a result of the Command register being written or the Device Control register being written. The host adapter shall set the BSY bit in the shadow Status register to one within 400 ns of the write operation to the Command register that initiated the transmission. The host adapter shall set the BSY bit in the shadow Status register to one within 400 ns of a write operation to the Device Control register if the write to the Device Control register changes the state of the SRST bit from zero to one. The host adapter shall not set the BSY bit in the shadow Status register for writes to the Device Control register that do not change the state of the SRST bit from zero to one.

It is important to note that Serial ATA host adapters enforce the same access control to the Shadow Register Block as parallel ATA devices enforce to the Command Block Registers. Specifically, the host is prohibited from writing the Features(7:0), Count(7:0), LBA(7:0), LBA(15:8), LBA(23:16), or Device registers when either BSY bit or DRQ bit is set in the Status Register. Any write to the Command Register when BSY bit or DRQ bit is set is ignored unless the write is to issue a Device Reset command.

10.3.4.3 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. Then the device either initiates execution of the command indicated in the Command register or initiates execution of the control request indicated in the Device Control register, depending on the state of the C bit in the FIS.

There are legacy BIOS and drivers that write the Device Control register to enable the interrupt just prior to issuing a command. To avoid unnecessary overhead, this FIS is transmitted to the device only upon a change of state from the previous value.

10.3.5 Register - Device to Host

0	Error	Status	R	I	R	R	PM Port	FIS Type (34h)
1	Device	LBA(23:16)	LBA(15:8)					LBA(7:0)
2	Reserved (0)	LBA(47:40)	LBA(39:32)					LBA(31:24) (0)
3	Reserved (0)	Reserved (0)	Count(15:8)					Count(7:0)
4	Reserved (0)	Reserved (0)	Reserved (0)					Reserved (0)

Figure 195 – 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(15:8) - Contains the new value of the LBA Mid register of the Shadow Register Block.

LBA(39:32) – Contains the contents of the expanded address field of the Shadow Register Block

LBA(23:16) - Contains the new value of the LBA High register of the Shadow Register Block.

LBA(47:40) – Contains the contents of the expanded address field of the Shadow Register Block

Device - Contains the new value of the Device register of the Shadow Register Block.

Error - Contains the new value of the Error register of the Shadow Register Block.

I - Interrupt bit. This bit reflects the interrupt bit line 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.

PM Port – When an endpoint device is attached via a Port Multiplier, specifies the device port address that the FIS is received from. This field is set by the Port Multiplier. Endpoint devices shall set this field to 0h.

R - Reserved, – shall be cleared to zero.

Count(7:0) - Contains the new value of the Sector Count register of the Shadow Register Block.

Count(15:8) – Contains the contents of the expanded address field of the Shadow Register Block

LBA(7:0) - Contains the new value of the LBA Low register of the Shadow Register Block.

LBA(31:24) – Contains the contents of the expanded address field of the Shadow Register Block

Status - Contains the new value of the Status (and Alternate status) register of the Shadow Register Block.

10.3.5.1 Description

The Register – Device to Host FIS is used to by the device to update the contents of the host adapter's Shadow Register Block. This is the mechanism by which devices indicate command completion status or otherwise change the contents of the host adapter's Shadow Register Block.

10.3.5.2 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 Shadow Register 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 FIS shall be used to set the SERV bit when the BSY bit and DRQ bit are both cleared to zero in the Status Register. The SERV bit transmitted with the Register - Device to Host FIS is written to the shadow Status Register and so the bit should accurately reflect the state of pending service requests when the FIS is transmitted as a result of a command completion by the device.

10.3.5.3 Reception

Upon reception of a valid Register - Device to Host FIS the received register contents are transferred to the host adapter's Shadow Register Block.

If the BSY bit and DRQ bit in the shadow Status Register are both cleared when a Register - Device to Host FIS is received by the host adapter, then the host adapter shall discard the contents of the received FIS and not update the contents of any shadow register.

10.3.6 Set Device Bits - Device to Host

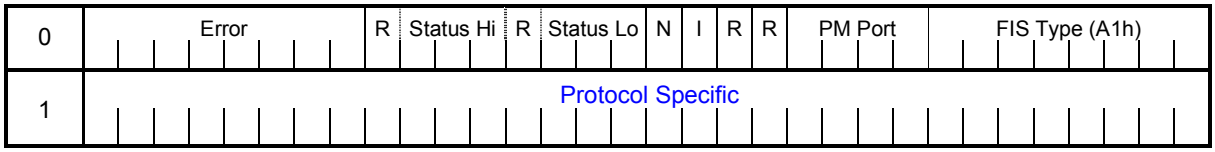


Figure 196 – Set Device Bits - Device to Host FIS layout

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 the host is executing tagged queued commands (READ FPDMA QUEUED, WRITE FPDMA QUEUED or NCQ QUEUE MANAGEMENT) with the device, the host should only enter the interrupt pending state if both the BSY bit and the DRQ bit in the shadow Status register are zero when the frame is received. If the host is executing native queued commands (READ FPDMA QUEUED, WRITE FPDMA QUEUED or NCQ QUEUE MANAGEMENT) with the device, the interrupt pending state is entered regardless of the current state of the BSY bit or the DRQ bit in the shadow Status register. Devices shall not modify the behavior of this bit based on the state of the nIEN bit received in Register Host to Device FISes.

N – Notification Bit. This bit signals the host that the device needs attention. If the bit is set to one, the host should interrogate the device and determine what type of action is needed. If the bit is cleared to zero, the device is not requesting attention from the host. Refer to section 13.8.

Error – Contains the new value of the Error register of the Shadow Register Block.
PM Port – When an endpoint device is attached via a Port Multiplier, specifies the device port address that the FIS is received from. This field is set by the Port Multiplier. Endpoint devices shall set this field to 0h.
Status-Hi – Contains the new value of bits 6, 5, and 4 of the Status register of the Shadow Register Block.
Status-Lo– Contains the new value of bits 2,1, and 0 of the Status register of the Shadow Register Block.
Protocol Specific – The value of this field is only defined for use with the Native Command Queuing Protocol. Refer to 13.6 for details. This field shall be cleared to zero for any uses other than Native Command Queuing (e.g. Asynchronous Notification).
R – Reserved – shall be cleared to zero.

10.3.6.1 Description

The Set Device Bits FIS is used by the device to load Shadow Register 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 the BSY bit or the DRQ bit of the Status register.

The FIS includes a bit to signal the host adapter to generate an interrupt if the BSY bit and the DRQ bit in the shadow Status Register are both cleared to zero when this FIS is received.

Some Serial ATA to parallel ATA bridge solutions may elect to not support this FIS based on the requirements of their target markets.

10.3.6.2 Transmission

The device transmits a Set Device Bits FIS to alter one or more bits in the Error register or in the Status register in the Shadow Register 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.

10.3.6.3 Reception

Upon receiving a Set Device Bits FIS, the host adapter shall load the data from the Error field into the shadow Error register, the data from the Status-Hi field into bits 6, 5, and 4, of the shadow Status register, and the data from the Status-Lo field into bits 2, 1, and 0 of the shadow Status register. The BSY bit and the DRQ bit of the shadow Status register shall not be changed. If the Interrupt bit in the FIS is set to a one, and if both the BSY bit and the DRQ bit in the Shadow status register are cleared to zero when this FIS is received, then the host adapter shall enter an interrupt pending state.

10.3.7 DMA Activate - Device to Host

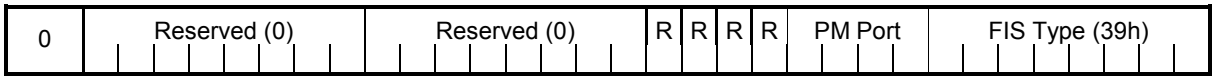


Figure 197 – DMA Activate - Device to Host FIS layout

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.

PM Port – When an endpoint device is attached via a Port Multiplier, specifies the device port address that the FIS is received from. This field is set by the Port Multiplier. Endpoint devices shall set this field to 0h.

R - Reserved – shall be cleared to zero.

10.3.7.1 Description

The DMA Activate 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.

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.

10.3.7.2 Transmission

The device transmits a DMA Activate 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. When transmitting this FIS, the device shall be prepared to subsequently receive a Data - Host to Device FIS from the host with the DMA data for the corresponding command.

10.3.7.3 Reception

Upon receiving a DMA Activate, 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.

10.3.8 DMA Setup – Device to Host or Host to Device (Bidirectional)

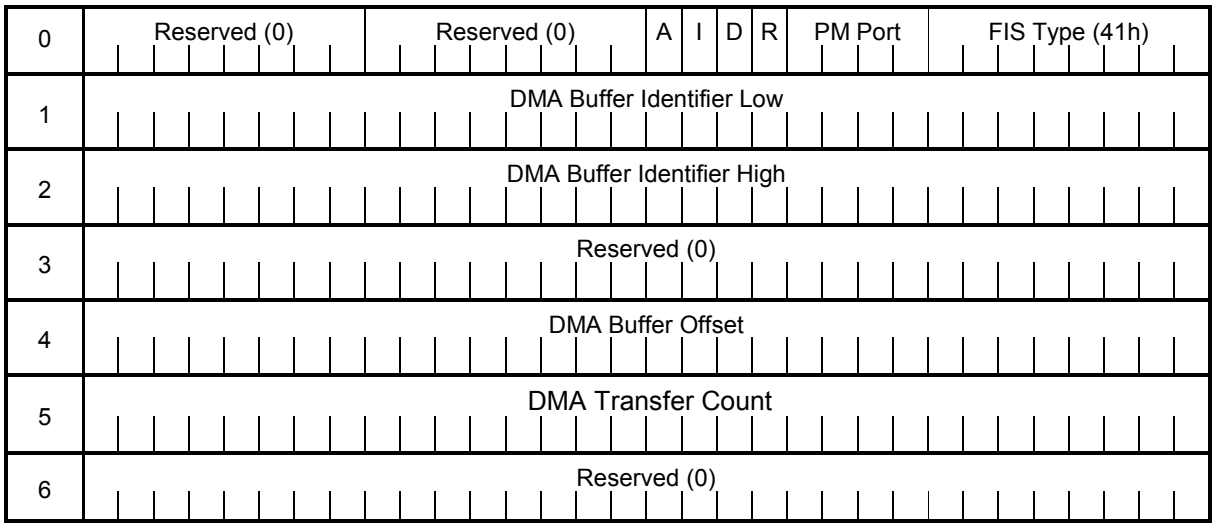


Figure 198 – DMA Setup – Device to Host or Host to Device FIS layout

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 Direction - Specifies whether subsequent data transferred after this FIS is from transmitter to receiver or from receiver to transmitter If set to one the direction is transmitter to receiver.If cleared to zero, the direction is receiver to transmitter.

A Auto-Activate - If set to one, in response to a DMA Setup FIS with data transfer direction of Host-to-Device, causes the host to initiate transfer of the first Data FIS to the device after the DMA context for the transfer has been established. The device shall not transmit a DMA Activate FIS to trigger the transmission of the first Data FIS from the host. If cleared to zero, a DMA Activate FIS is required to trigger the transmission of the first Data FIS from the host when the data transfer direction is Host-to-Device.

DMA Buffer Identifier Low/High - This field is used to identify a DMA buffer region in host memory. The contents are not described in this specification and are host dependent. The buffer identifier is supplied by the host to the device and the device echoes it back to the host. 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 may identify a DMA "channel".

DMA Buffer Offset - This is the byte offset into the buffer. Bits [1:0] shall be zero.

DMA Transfer Count - This is the number of bytes to be read or written. Bit zero shall be zero.

I Interrupt - If the Interrupt bit is set to one an interrupt pending shall be generated when the DMA transfer count is exhausted. Devices shall not modify the behavior of this bit based on the state of the nIEN bit received in Register Host to Device FISes.

PM Port – When an endpoint device is attached via a Port Multiplier, specifies the device port address that the FIS should be delivered to or is received from. This field is set by the host for Host to Device transmission and this field is set by the Port Multiplier for Device to Host transmission. Endpoint devices shall set this field to 0h for Device to Host transmissions.

R - Reserved – shall be cleared to zero.

10.3.8.1 Description

The DMA Setup – Device to Host or Host to Device FIS is the mechanism by which first-party DMA access to host 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 actual host memory regions to be abstracted (depending on implementation) by having memory regions referenced via a base memory descriptor representing a memory region that the host has granted the device access to. 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 an implementation specific mechanism such as a command issued to or as defined in a specification. Random access within a buffer is accomplished by using the buffer offset.

First party DMA is a superset capability not necessarily supported by legacy mode devices or legacy mode device drivers but essential for accommodating future capabilities.

10.3.8.2 Transmission

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

A DMA Setup – Device to Host or Host to Device is required only when the DMA context is to be changed. Multiple Data – Host to Device or Device to Host FISes may 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 DMA Setup – Device to Host or Host to Device FIS, the device or host shall place the data contained in the FIS in sequential addresses; that is, if the last Dword of a 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 DMA Setup – Device to Host or Host to Device FIS is used to alter the DMA context. This mechanism allows for the efficient streaming of data into a buffer.

10.3.8.3 Reception

Upon receiving a DMA Setup – Device to Host or Host to Device FIS, the receiver of the FIS shall validate the received 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 DMA Setup – Device to Host or Host to Device FIS with the D bit cleared to zero, the receiver of the DMA Setup – Device to Host or Host to Device FIS responds with one or more Data – Host to Device or Device to Host FISes until the DMA count is exhausted. After a valid DMA Setup – Device to Host or Host to Device FIS with the D bit set to one, the receiver 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 Interrupt bit is set to one. The definition of DMA transfer completion is system dependent but typically includes the exhaustion of the transfer count or the detection of an error by the DMA controller.

NOTE – First-party DMA accesses are categorized in two groups: command/status transfers and user-data transfers. Interrupts would not typically be generated on user-data transfers. The optimal interrupt scheme for command/status transfers is not defined in this specification.

10.3.8.3.1 Auto-Activate

First Party DMA transfers from the host to the device require transmission of both the DMA Setup FIS and a subsequent DMA Activate FIS in order to trigger the host transfer of data to the device. Because the device may elect to submit the DMA Setup FIS only when it is already prepared to receive the subsequent Data FIS from the host, the extra transaction for the DMA Activate FIS may be eliminated by merely having the DMA Setup FIS automatically activate the DMA controller by setting the Auto-Activate 'A' bit to one in the DMA Setup FIS.

Devices shall not attempt to utilize this capability prior to the optimization having been explicitly enabled by the host as defined in section 13.3.2. The host response to a DMA Setup FIS with the Auto-Activate bit set to one when the host has not enabled Auto-Activate is not defined.

10.3.8.4 HBA Enforcement of First-party DMA Data Phase Atomicity

The host bus adapter shall ensure the First-party DMA Data Phase is uninterrupted. Unless the ERR bit in the shadow Status register is set, the host shall ensure no FIS other than requested data payload or a FIS for a software reset is transmitted from the host to device between the reception of a DMA Setup FIS and the exhaustion of the associated transfer count.

10.3.9 BIST Activate - Bidirectional

0	Reserved (0)								Pattern Definition								R	R	R	R	PM Port				FIS Type (58h)							
									T	A	S	L	F	P	R	V															
1	Data1 [31:24]								Data1 [23:16]								Data1 [15:8]								Data1 [7:0]							
2	Data2 [31:24]								Data2 [23:16]								Data2 [15:8]								Data2 [7:0]							

Figure 199 – BIST Activate - Bidirectional

Field Definitions

FIS Type - Set to a value of 58h. Defines the rest of the FIS fields.

PM Port – When an endpoint device is attached via a Port Multiplier, specifies the device port address that the FIS should be delivered to or is received from. This field is set by the host for Host to Device transmission and this field is set by the Port Multiplier for Device to Host transmission. Endpoint devices shall set this field to 0h for Device to Host transmissions.

R - Reserved – shall be cleared to zero.

Pattern Definition

F – Far End Analog (AFE) Loopback (Optional)

L - Far End Retimed Loopback* Transmitter shall insert additional $ALIGN_P$ primitives

T - Far end transmit only mode

A - $ALIGN_P$ Bypass (Do not Transmit $ALIGN_P$ primitives) (valid only in combination with T Bit)

S - Bypass Scrambling (valid only in combination with T Bit)

P - Primitive bit. (valid only in combination with the T Bit) (Optional)

V - Vendor Specific Test Mode. Causes all other bits to be ignored

Data1 – Dword #1 of data information used to determine what pattern is transmitted as a result of the BIST Activate FIS. Applicable only when the T bit is set to one.

Data2 - Dword #2 of data information used to determine what pattern is transmitted as a result of the BIST Activate FIS. Applicable only when the T bit is set to one.

10.3.9.1 Description

The BIST Activate FIS shall be used to place the receiver in one of several loopback modes.

The BIST Activate FIS is a bi-directional FIS that may be sent by either the host or the device. The sender and receiver have distinct responsibilities in order to insure proper cooperation between the two parties. The state machines for transmission and reception of the FIS are symmetrical. The method of causing a BIST Activate FIS transmission is not defined in this specification.

The state machines for the transmission of the FIS do not attempt to specify the actions the sender takes once successful transmission of the request has been performed. After the Application layer is notified of the successful transmission of the FIS the sender's Application layer prepares its own Application, Transport and Physical layers into the appropriate states that support the transmission of a stream of data. The FIS shall not be considered successfully transmitted until the receiver has acknowledged reception of the FIS as per normal FIS transfers documented in various sections of this specification. The transmitter of the BIST Activate FIS should transmit continuous $SYNC_P$ primitives after reception of R_OK_P until such a time that it is ready to interact with the receiver in the BIST exchange.

Similarly, the state machines for the reception of the FIS do not specify the actions of the receiver's Application layer. Once the FIS has been received, the receiver's Application layer places its own Application, Transport and Physical layers into states that perform the appropriate retransmission of the sender's data. The receiver shall not enter the BIST state until after it has properly received a good BIST Activate FIS (good CRC), indicated a successful transfer of the FIS to the transmitting side via R_OK_P and has received at least one good $SYNC_P$. Once in the self-test mode, a receiver shall continue to allow processing of the COMINIT or COMRESET signals in order to exit from the self-test mode.

Note, that BIST mode is intended for Inspection/Observation Testing, as well as support for conventional laboratory equipment, rather than for in-system automated testing.

The setting of the F, L, and T bits is mutually exclusive. It is the responsibility of the sender of the BIST Activate FIS to ensure that only one of these bits is set. Refer to Table 74 for valid bit settings within a BIST Activate FIS.

Table 74 – BIST Activate FIS Modes and Bit Settings

BIST Test Mode	F	L	T	P	A	S	V
Far End Analog Loopback	1	0	0	0	0	0	0
Far End Retimed Loopback	0	1	0	0	0	0	0
Far End Transmit with ALIGNs, scrambled data	0	0	1	0	0	0	0
Far End Transmit with ALIGNs, unscrambled data	0	0	1	0	0	1	0
Far End Transmit without ALIGNs, scrambled data	0	0	1	0	1	0	0
Far End Transmit without ALIGNs, unscrambled data	0	0	1	0	1	1	0
Far End Transmit primitives with ALIGNs	0	0	1	1	0	na	0
Far End Transmit primitives without ALIGNs	0	0	1	1	1	na	0
Vendor Specific	na	na	na	na	na	na	1
Key: 0 – bit shall be cleared to zero 1 – bit shall be set to one							

F: The Far End Analog (Analog Front End - AFE) Loopback Mode is defined as a vendor optional mode where the raw data is received, and retransmitted, without any retiming or re-synchronization, etc. 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.

L: The Far End Retimed LoopbackMode is defined as a mode where the receiver retimes the data, and retransmits the retimed data. The initiator of the retimed loopback mode shall account for the loopback device consuming up to two $ALIGN_P$ primitives (one ALIGN sequence) every 256 Dwords transmitted and, if it requires any $ALIGN_P$ primitives to be present in the returned data stream, it should insert additional $ALIGN_P$ primitives in the transmitted stream. The initiator shall transmit additional ALIGN sequences in a single burst at the normal interval of every 256 Dwords transmitted (as opposed to inserting ALIGN sequences at half the interval).

The loopback device may remove zero, one, or two $ALIGN_P$ primitives from the received data. It may insert one or more $ALIGN_P$ primitives if they are directly preceded or followed by the initiator inserted $ALIGN_P$ primitives (resulting in ALIGN sequences consisting of at least two $ALIGN_P$

primitives) or it may insert two or more $ALIGN_P$ primitives if not preceded or followed by the initiator's $ALIGN_P$ primitives. One side effect of the loopback retiming is that the returned data stream may have instances of an odd number of $ALIGN_P$ primitives, however, returned $ALIGN_P$ primitives are always in bursts and if the initiator transmitted dual ALIGN sequences (four consecutive $ALIGN_P$ primitives), then the returned data stream shall include $ALIGN_P$ bursts that are no shorter than two $ALIGN_P$ primitives long (although the length of the $ALIGN_P$ burst may be odd). The initiator of the retimed loopback mode shall not assume any relationship between the relative position of the $ALIGN_P$ primitives returned by the loopback device and the relative position of the $ALIGN_P$ primitives sent by the initiator.

In retimed loopback mode, the initiator shall transmit only valid 8b/10b characters so the loopback device may 10b/8b decode it and re-encode it before retransmission. If the loopback 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 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 device shall return all received data unaltered and shall disregard protocol processing of primitives. Only the OOB signals and $ALIGN_P$ processing is acted on by the loopback device, while all other data is retransmitted without interpretation.

T: The Far-End Transmit Mode is defined as a mode that may be used to invoke the Far-End Interface to send data patterns, upon receipt of the BIST Activate FIS, as defined by the content located in Data1 and Data2. Note that Data1 and Data2 shall be applicable only when the T bit is active, indicating "Far-End Transmit Mode". It is not required that the values within Data1 and Data2 are equal. These two Dwords are programmable to any value.

This data is modified by the following bits.

P: The transmit primitives bit. When this bit is set in far end transmit mode, the lowest order byte of the two following Dwords are treated as K Characters in order to identify the appropriate primitive(s) for transmission. The encoding for primitives is defined in section 9.4. It is the responsibility of the sender of the BIST Activate FIS to ensure that the values contained within Data1 and Data2 are valid D character versions of the K character (i.e. BCh for K28.5). The setting of this bit is applicable only when the T bit is set.

A: $ALIGN_P$ sequence bypass mode. When set to one, no $ALIGN_P$ primitives are sent. When the A-bit is not asserted, $ALIGN_P$ primitives are sent normally as defined in this document. The setting of this bit is applicable only when the T bit is set.

S: The Bypass Scrambling Mode is defined as a mode that may be 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. The setting of this bit is applicable only when the T bit is set. The S bit is ignored when the P bit is set to one.

V: The vendor unique mode is implementation specific and shall be reserved for individual vendor use. All other bits are ignored in this mode.

10.3.9.2 Transmission

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

10.3.9.3 Reception

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

10.3.10 PIO Setup – Device to Host

0	Error		Status		R	I	D	R	PM Port	FIS Type (5Fh)
1	Device		LBA(23:16)		LBA(15:8)				LBA(7:0)
2	Reserved (0)		LBA(47:40)		LBA(39:32)				LBA(31:24) (0)
3	E_Status		Reserved (0)		Count(15:8)				Count(7:0)
4	Reserved (0)				Transfer Count				

Figure 200 – 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(15:8) - Holds the contents of the LBA(15:8) register of the Command Block.

LBA(39:32) – Contains the contents of the LBA(39:32) field of the Shadow Register Block

LBA(23:16) - Holds the contents of the LBA(23:16) register of the Command Block.

LBA(47:40) – Contains the contents of the LBA(47:40) field of the Shadow Register Block

D - Specifies the data transfer direction. When set to one the transfer is from device to host, when cleared to zero the transfer is from host to device.

Device - Holds the contents of the Device register of the Command Block.

Status - Contains the new value of the Status register of the Command Block for initiation of host data transfer.

Error - Contains the new 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 bit line 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.

PM Port – When an endpoint device is attached via a Port Multiplier, specifies the device port address that the FIS is received from. This field is set by the Port Multiplier. Endpoint devices shall set this field to 0h.

R – Reserved – shall be cleared to zero.

Count(7:0) - Holds the contents of the Count(7:0) register of the Command Block.

Count(15:8) – Contains the contents of the Count(15:8) field of the Shadow Register Block

LBA(7:0) - Holds the contents of the LBA(7:0) register of the Command Block.

LBA(31:24) – Contains the contents of the LBA(31:24) field of the Shadow Register Block

E_Status - Contains the new value of the Status register of the Command Block at the conclusion of the subsequent Data FIS.

Transfer Count – Holds the number of bytes to be transferred in the subsequent Data FIS. The Transfer Count value shall be nonzero and the low order bit shall be zero (even number of bytes transferred).

10.3.10.1 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 each and 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 ATA standard, the PIO Setup FIS includes both the starting and ending status values. These are used by the host adapter to first signal to host software readiness for PIO write data (BSY bit is cleared to zero and DRQ bit is set to one), and following the PIO write burst to properly signal host software by clearing the DRQ bit to zero and possibly setting the BSY bit to one.

10.3.10.2 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 PIO data payload transfer just before each and every PIO data payload transfer required to complete the total data transfer for a command. The device includes in the FIS the values to be placed in the Shadow Status register at the beginning of the PIO data payload transfer and the value to be placed in the Shadow Status register at the end of the data payload transfer. The device shall be prepared to receive a Data FIS in response to transmitting a PIO Setup FIS.

10.3.10.3 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 update all Shadow registers and shall hold the E_Status value in a temporary register. The Transfer Length value shall be loaded into a countdown register. Upon detecting the change in the Shadow Status register, host software proceeds to perform a series of write operations to the Data shadow register, which the host adapter shall collect to produce a Data FIS to the device. Each write of the Data shadow register results in another word of data being concatenated into the Data FIS, and the countdown register being decremented accordingly. The E_Status value shall be transferred to the Shadow Status register within 400 ns of 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 each and every data FIS needed to complete the overall data transfer of a command.

10.3.10.4 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 payload transfer just before each and every PIO data payload transfer required to complete the total data transfer for a command. The device includes in the FIS the values to be placed in the Shadow Status register at the beginning of the PIO data payload transfer and the value to be placed in the Shadow Status register at the end of the data payload transfer. The device shall be prepared to transmit a Data FIS following the transmittal of a PIO Setup FIS.

10.3.10.5 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 host shall update all Shadow registers and host software proceeds to perform a series of read operations from the Data shadow register. Each read of the Data shadow register results in a countdown register being decremented accordingly. The E_Status value shall be transferred to the Shadow Status register within 400 ns of the countdown register reaching terminal count. This process is repeated for each and every data FIS needed to complete the overall data transfer of a command.

10.3.11 Data - Host to Device or Device to Host (Bidirectional)

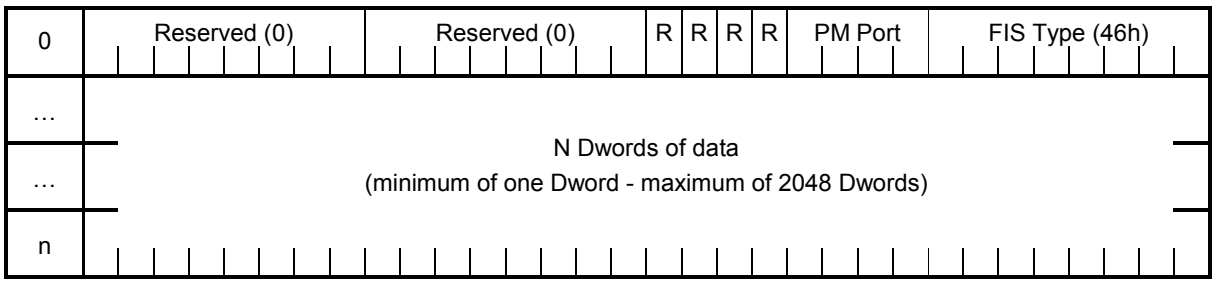


Figure 201 – Data – Host to Device or Device to Host FIS layout

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 may be sent in a single Data – Host to Device or Data – Device to Host FIS is limited. See description.

PM Port – When an endpoint device is attached via a Port Multiplier, specifies the device port address that the FIS should be delivered to or is received from. This field is set by the host for Host to Device transmission and this field is set by the Port Multiplier for Device to Host transmission. Endpoint devices shall set this field to 0h for Device to Host transmissions.

R – Reserved – shall be cleared to zero.

10.3.11.1 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 generally 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 SOF_P and EOF_P, and discounting the FIS type and CRC Dwords. The payload size shall be no more than 2048 Dwords (8192 bytes). Non-packet devices, with or without bridges, should 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.

10.3.11.2 Transmission

The device transmits a Data – Device to Host FIS to the host during the data transfer phase of legacy mode 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 shall be preceded by a 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 legacy mode 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 may follow in either direction. Segmentation may 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, the Data FIS shall contain the number of bytes indicated in the Transfer Count field of the preceding PIO Setup 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.

The Serial ATA protocol does not permit for the transfer of an odd number of bytes.

10.3.11.3 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 shall be reflected in the overall command completion status.

10.4 Host transport states

FIS reception is asynchronous in nature. In the case of a non-Data FIS transmission, the host may be pre-empted by a non-Data FIS reception from the device. The host shall hold off on the pending transmission and process the incoming FIS from the device before attempting to retransmit the pending FIS.

10.4.1 Host transport idle state diagram

HTI1: HT_HostIdle	Host adapter waits for frame or frame request.	
1. Command Register FIS transmission pending	→	HT_CmdFIS
2. Control Register FIS transmission pending	→	HT_CntrlFIS [1]
3. Frame receipt indicated by Link layer	→	HT_ChkTyp [3]
4. DMA Setup FIS transmission pending	→	HT_DMASTUPFIS
5. BIST Activate FIS transmission pending	→	HT_XmitBIST
6. Previous FIS was PIO Setup and Application layer indicates data direction is host to device	→	HT_PIOOTrans2 [2]

NOTE:

1. Transmission of a Control Register FIS is mandatory if the state of the SRST bit in the Device Control Shadow Register is changed, and is optional if the state of the SRST bit is not changed. If a Control Register FIS transmission with a modified value for the SRST bit is triggered while another non-Control Register FIS transmission is already pending, all pending non-Control Register FIS transmissions shall be aborted. If the state of the SRST bit is not modified from its previous value in a Control Register FIS transmission, then pending non-Control Register FIS transmissions shall not be aborted.
2. The PIO Setup FIS shall set an indication that PIO Setup was the last FIS received. Indication from the Application layer that it is transmitting data to the device may be determined from the Application layer performing write operations to the Data register in the Shadow Register Block.
3. FIS reception shall have priority over all other transitions in this state.

HTI2: HT_ChkTyp	Received FIS type checked.		
	1. Register FIS type detected	→	HT_RegFIS
	2. Set Device Bits FIS type detected	→	HT_DB_FIS
	3. DMA Activate FIS type detected	→	HT_DMA_FIS
	4. PIO Setup FIS type detected	→	HT_PS_FIS
	5. DMA Setup FIS type detected	→	HT_DS_FIS
	6. BIST FIS type detected	→	HT RcvBIST
	7. Data FIS type detected and previous FIS was not PIO Setup	→	HT_DMAITrans [1]
	8. Data FIS type detected and previous FIS was PIO Setup	→	HT_PIOITrans1 [1]
	9. Unrecognized FIS received	→	HT_HostIdle
	10. Notification of illegal transition error received from Link layer	→	HT_HostIdle
	NOTE: 1. The PIO Setup FIS shall set an indication that PIO Setup was the last FIS sent, so that this state may determine whether to transition to DMA data transfer, or PIO data transfer.		

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 the shadow Command register to be written, the shadow Device Control register to be written, or the Link layer to indicate that a FIS is being received.

TransitionHTI1:1: When a Command Register FIS transmission is pending, the Transport layer shall make a transition to the HTCM1: HT_CmdFIS state. A Command Register FIS becomes pending upon a write operation to the Command shadow register, and ceases pending at successful transmission of the FIS as indicated by the Link layer.

Transition HTI1:2: When a Control Register FIS transmission is pending, the Transport layer shall make a transition to the HTCR1: HT_CntrlFIS state. A Control Register FIS becomes pending upon a write operation to the Device Control shadow register that changes the state of the SRST bit from the previous value, or optionally upon a write operation to the Device Control shadow register that does not change the state of the SRST bit. A Control Register FIS ceases pending at successful transmission of the FIS as indicated by the Link layer.

Transition HTI1:3: When the Link layer indicates that a FIS is being received, the Transport layer shall make a transition to the HTI2: HT_ChkTyp state.

Transition HTI1:4: When the Application layer indicates that a DMA Setup FIS is to be sent, the Transport layer shall make a transition to the HT_DMASTUP0:HT_DMASTUPFIS state.

Transition HTI1:5: When the Application layer requests the transmission of a BIST request to the device the Transport layer shall make a transition to the HTXBIST1:HT state.

Transition HTI1:6: When the Application layer requests the transmission of data to the device and the previous FIS was a PIOSetup type, the Transport layer shall make a transition to the HTPS3:HT_PIOOTrans2 state. The Application layer signals transmission of PIO data to the device by performing writes to the Data register in the Shadow Register Block.

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

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

Transition HTI2:1: When the incoming FIS is a register type, the Transport layer shall notify the Link layer that it has received a valid FIS, and make a transition to the HTR1: HT_RegFIS state.

Transition HTI2:2: 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 and make a transition to the HTDB0:HT_DB_FIS state.

Transition HTI2:3: When the incoming FIS is a DMA Activate type, the Transport layer shall notify the Link layer that it has received a valid FIS, and make a transition to the HTDA1: HT_DMA_FIS state.

Transition HTI2:4: When the incoming FIS is a PIO Setup type, the Transport layer shall notify the Link layer that it has received a valid FIS, and make a transition to the HTPS1: HT_PS_FIS state.

Transition HTI2:5: When the incoming FIS is a DMA Setup type, the Transport layer shall notify the Link layer that it has received a valid FIS, and make a transition to the HTDS1: HT_DS_FIS state.

Transition HTI2:6: When the incoming FIS is a BIST Activate type, the Transport layer shall notify the Link layer that it has received a valid FIS, and make a transition to the HTRBIST1:HT_RcvBIST state.

Transition HTI2:7: 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, and make a transition to the HTDA5:HT_DMAITrans state.

Transition HTI2:8: 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, and make a transition to the HTPS5:HT_PIOITrans1 state.

Transition HTI2:9: 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.

Transition HTI2:10: 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.

10.4.2 Host Transport transmit command FIS diagram

This protocol builds a FIS that contains the host adapter shadow register content and sends it to the device when the software driver or BIOS writes the host adapter shadow Command register.

HTCM1: HT_CmdFIS	Construct Register – Host to Device FIS with C bit set to one from the content of the shadow registers and notify Link to transfer.		
	1. FIS transfer complete	→	HT_CmdTransStatus
	2. Notification of illegal transition error received from Link layer	→	HT_HostIdle
	3. Frame receipt indicated by Link layer	→	HT_HostIdle

HTCM2: HT_CmdTransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
	1. Status checked and no error detected	→	HT_HostIdle
	2. Status checked and error detected[1]	→	HT_HostIdle
	NOTE: 1. Upon return to the HT_HostIdle state in response to a detected error, the associated FIS remains pending for transmission		

HTCM1: HT_CmdFIS state: This state is entered when the shadow Command register is written.

When in this state, the Transport layer shall construct a Register – Host to Device FIS with C bit set to one, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer.

Transition HTCM1:1: 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_CmdTransStatus state.

Transition HTCM1:2: 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 HTCM1:3: When the Link layer indicates that a FIS is being received, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTCM2: HT_CmdTransStatus 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 layer ending status for the FIS and take appropriate error handling action if required.

Transition HTCM2:1: When the FIS status has been handled and no errors detected, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTCM2:2: When the FIS status has been handled and an error has been detected, the Transport layer shall transition to the HTI1: HT_HostIdle state. The associated FIS remains pending for transmission.

10.4.3 Host Transport transmit control FIS diagram

This protocol builds a FIS that contains the host adapter shadow register content and sends it to the device when the software driver or BIOS writes the host adapter shadow Device Control register.

HTCR1: HT_CntrlFIS	Construct Register – Host to Device FIS with C bit cleared to zero from the content of the shadow registers and notify Link to transfer.	
1. FIS transfer complete	→	HT_CtrlTransStatus
2. Notification of illegal transition error received from Link layer	→	HT_HostIdle
3. Frame receipt indicated by Link layer	→	HT_HostIdle

HTCR2: HT_CtrlTransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked and no errors detected	→	HT_HostIdle
2. Status checked and error detected[1]	→	HT_HostIdle
NOTE: 1. Upon return to the HT_HostIdle state in response to a detected error, the associated FIS remains pending for transmission		

HTCR1: HT_Cntrl_FIS state: This state is entered when the shadow Device Control register is written.

When in this state, the Transport layer shall construct a Register – Host to Device FIS with C bit cleared to zero, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer.

Transition HTCR1:1: 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_CtrlTransStatus state.

Transition HTCR1:2: 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 HTCR1:3: When the Link layer indicates that a FIS is being received, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTCR2: HT_CtrlTransStatus 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 HTCR2:1: When the FIS status has been handled and no errors have been detected, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTCR2:2: When the FIS status has been handled and an error has been detected, the Transport layer shall transition to the HTI1: HT_HostIdle state. The associated FIS remains pending for transmission.

10.4.4 Host Transport transmit DMA Setup – Device to Host or Host to Device FIS state diagram

This protocol transmits a DMA Setup – Device to Host or Host to Device FIS to a receiver. This FIS is a request by a transmitter for the receiver to program its DMA controller for a First-party DMA transfer and is followed by one or more Data FISes that transfer data. The DMA Setup – Device to Host or Host to Device FIS request includes the transfer direction indicator, the host buffer identifier, the host buffer offset, the byte count, and the interrupt flag.

HTDMASTUP0: HT_DMASTUPFIS	Construct the DMA Setup – Host to Device or Device to Host FIS from the content provided by the Application layer and notifies Link to transfer.	
1. FIS transfer complete	→	HT_DMASTUPTrans Status
2. Notification of illegal transition error received from Link layer	→	HT_HostIdle
3. Frame receipt indicated by Link layer	→	HT_HostIdle

HTPDMASTUP1: HT_DMASTUPTransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked and no error detected	→	HT_HostIdle
2. Status checked and error detected[1]	→	HT_HostIdle
NOTE: 1. Upon return to the HT_HostIdle state in response to a detected error, the associated FIS remains pending for transmission		

HTDMASTUP0: HT_DMASTUPFIS state: This state is entered when the Application requests the transmission of a DMA Setup – Host to Device or Device to Host FIS.

When in this state, the Transport layer shall construct a 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 HTDMASTUP0:1: 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_DMASTUPTransStatus state.

Transition HTDMASTUP0:2: 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 HTDMASTUP0:3: When the Link layer indicates that a FIS is being received, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTPDMASTUP1: HT_DMASTUPTransStatus 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 HTDMASTUP1:1: When the FIS status has been handled, and no error detected, the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTDMASTUP1:2: When the FIS status has been handled, and an error detected, the Transport layer shall transition to the HTI1: HT_HostIdle state. The associated FIS remains pending for transmission.

10.4.5 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 test data pattern 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 specification.

HTXBIST1: HT_XmitBIST	Construct the BIST Activate FIS from the content provided by the Application layer and notify Link to transfer.	
1. FIS transfer complete	→	HT_TransBISTStatus
2. Notification of illegal transition error received from Link layer	→	HT_HostIdle
3. Frame receipt indicated by Link layer	→	HT_HostIdle

HTXBIST2: HT_TransBISTStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status check completed	→	HT_HostIdle
2. Status check and at least one error detected	→	HT_HostIdle[1]
NOTE: 1. Re-transmission of the BIST Activate FIS due to errors is not required but allowed.		

HTXBIST1: HT_XmitBIST state: This state is entered to send a BIST FIS to the device.

Transition HTXBIST1:1: 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 HTXBIST2:HT_TransBISTStatus state.

Transition HTXBIST1:2: 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 HTXBIST1:3: When the Link layer indicates that a FIS is being received, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTXBIST2: HT_TransBISTStatus state: This state is entered when the entire FIS has been passed to the Link layer.

Transition HTXBIST2:1: When the FIS transmission is completed the Transport layer shall transition to the HTI1:HT_HostIdle state.

Transition HTXBIST2:2: When the FIS transmission is completed and at least one error is detected the Transport layer shall transition to the HTI1:HT_HostIdle state. The associated FIS may remain pending for transmission.

10.4.6 Host Transport decomposes Register FIS diagram

This protocol receives a Register – Device to Host FIS from the device containing new shadow register content and places that content into the shadow registers.

HTR1: HT_RegFIS	Place FIS contents from device into appropriate holding registers.		
	1. FIS transfer complete	→	HT_RegTransStatus
	2. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTR2: HT_RegTransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
	1. Status checked	→	HT_HostIdle

HTR1: HT_RegFIS state: This state is entered the Link layer has indicated that a FIS is being received and that the FIS is of the register type.

When in this state, the Transport layer shall decompose the Register FIS and place the contents into the appropriate holding registers.

Transition HTR1:1: When the entire Register – Device to Host FIS has been placed into the holding registers, the Transport layer shall make a transition to the HTR2: HT_RegTransStatus state.

Transition HTR1:2: 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_RegTransStatus state: This state is 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 layer ending status for the FIS and take appropriate error handling action if required.

Transition HTR2:1: When the FIS status has been handled and no errors detected, the contents of the holding registers shall be placed in the shadow 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 handled and at least one error detected, the contents of the holding registers shall not be transferred to the shadow registers, error status shall be returned to the device, and the Transport layer shall transition to the HTI1: HT_HostIdle state.

10.4.7 Host Transport decomposes a Set Device Bits FIS state diagram

This protocol receives a Set Device Bits FIS from the device containing new Error and Status Shadow register content and places that content into the Error and Status Shadow registers. The Set Device Bits FIS may also contain SActive register content and asynchronous notification content.

HTDB0:HT_DB_FIS	Receive Set Device Bits FIS		
	1. FIS status checked and no error detected	→	HT_Dev_Bits
	2. FIS status checked and error detected.	→	HT_HostIdle

HTDB1:HT_Dev_Bits	Load Error register and bits of the Status register		
	1. Register bits loaded	→	HT_HostIdle

HTDB0:HT_DB_FIS state: This state is entered when the Link layer has indicated that a 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 layer ending status to be posted.

Transition HTDB0:1: When the FIS reception is complete with no errors detected, the Transport layer shall transition to the HTDB1:HT_Dev_Bits state.

Transition HTDB0:2: When the FIS reception is complete with errors detected, the Transport layer shall return error status to the device and transition to the HTI1:HT_HostIdle state.

HTDB1:HT_Dev_Bits state: This state is entered when a Set Device Bits FIS has been received with no errors.

When in this state, the data in the Error field of the received FIS shall be loaded into the host adapter's shadow Error register. The data in the Status-Hi field of the received FIS shall be loaded into bits 6, 5, and 4 of the shadow Status register. The data in the Status-Lo field of the received FIS shall be loaded into bits 2, 1, and 0 of the shadow Status register. The BSY bit and the DRQ bit in the shadow Status register shall not be changed. If the Interrupt bit in the FIS is set to one and if both the BSY bit and the DRQ bit in the shadow Status register are cleared to zero, then the host adapter shall enter an interrupt pending state.

Transition HTDB1:1: The Transport layer shall transition to the HTI1:HT_HostIdle state.

10.4.8 Host Transport decomposes a DMA Activate FIS diagram

This protocol receives a DMA Activate FIS that requests a DMA data out transfer. The data transfer is from the host to the device and the DMA Activate FIS causes the host adapter to transmit the data in a subsequent Data FIS.

HTDA1: HT_DMA_FIS		
1. Status checked and no error detected.	→	HT_DMAOTrans1
2. Status checked and error detected	→	HT_HostIdle
3. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTDA2: HT_DMAOTrans1	DMA controller initialized?	
1. DMA controller not initialized.	→	HT_DMAOTrans1
2. DMA controller initialized.	→	HT_DMAOTrans2
3. SRST asserted, or DEVICE RESET command requested	→	HT_HostIdle

HTDA3: HT_DMAOTrans2	Activate DMA controller	
1. Transfer not complete and < 2048 Dwords transmitted	→	HT_DMAOTrans2
2. Transfer not complete and 2048 Dwords transmitted	→	HT_DMAEnd
3. Abort notification from Link layer	→	HT_DMAEnd
4. Transfer complete	→	HT_DMAEnd
5. Notification of illegal transition error received from Link layer	→	HT_HostIdle
6. SRST asserted, or DEVICE RESET command requested	→	HT_HostIdle

HTDA4: HT_DMAEnd	Check DMA Controller completion	
1. DMA controller actions completed, no error detected	→	HT_HostIdle
2. DMA controller actions completed, and error detected.	→	HT_HostIdle
3. Abort notification from Link layer, no error detected	→	HT_HostIdle
4. Abort notification from Link layer, error detected.	→	HT_HostIdle

HTDA5: HT_DMAITrans	Activate DMA controller if initialized, receive Data FIS.	
1. Transfer not complete	→	HT_DMAITrans
2. SRST asserted, or device reset command issued	→	HT_HostIdle
3. Transfer complete	→	HT_DMAEnd
4. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTDA1: HT_DMA_FIS state: This state is entered when the Link layer has indicated that a FIS is being received and the Transport layer has determined that a DMA Activate FIS is being received.

When in this state, the Transport layer shall determine the direction of the DMA transfer being activated.

Transition HTDA1:1: The Transport layer shall make a transition to the HTDA2: HT_DMAOTrans1 state. This transition occurs if no error is detected.

Transition HTDA1:2: 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.

Transition HTDA1:3: 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 it is determined that the DMA transfer that is being activated is a transfer from host to device..

When in this state, the Transport layer shall determine if the DMA controller has been initialized.

Transition HTDA2:1: When the DMA controller has not yet been initialized, the Transport layer shall transition to the HTDA2: HT_DMAOTrans1 state.

Transition HTDA2:2: When the DMA controller has been initialized, the Transport layer shall transition to the HTDA3: HT_DMAOTrans2 state.

Transition HTDA2:3: When the host has asserted the SRST bit by writing to the Device Control register, or the DEVICE RESET command is requested, the Transport layer shall inform the Link layer to send a SYNC Escape, 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 HTDA3:1: 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 HTDA3:2: 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.

Transition HTDA3:3: When notified by the Link layer that the DMA Abort primitive was received, the Transport layer shall transition to the HTDA4: HT_DMAEnd state.

Transition HTDA3:4: When the requested DMA transfer is complete, the Transport layer shall transition to the HTDA4: HT_DMAEnd state.

Transition HTDA3:5: 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 HTDA3:6: When the host has asserted the SRST bit by writing to the Device Control register, or the DEVICE RESET command is requested, the Transport layer shall inform the Link layer to send a SYNC Escape, 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 HTDA4:1: 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 Dword data payload limit, the Transport layer shall transition to the HTI1: HT_HostIdle state. This transition occurs if no error is detected.

NOTE : The host should not assume received data is valid (even with a valid CRC receipt for the FIS) until command completion status is returned by the device.

Transition HTDA4:2: 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.

Transition HTDA4:3: When notified by the Link layer that a DMA Abort primitive was received, the transfer shall be truncated, and the Link layer notified to append CRC and end the frame. 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.

Transition HTDA4:4: When notified by the Link layer that a DMA Abort primitive was received, the transfer shall be truncated, and the Link layer notified to append CRC and end the frame. When it is determined that the transfer is completed with an error, the Transport layer shall report status to the Application layer and 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 is 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 HTDA5:1: When 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 HTDA5:2: When the SRST bit is asserted by the host writing the Device Control register, or a device reset command has been written to an ATAPI device, the Link layer shall be informed to send $SYNC_P$, and the Transport layer shall transition to the HTI1:HT_HostIdle state.

Transition HTDA5:3: When the requested DMA transfer is complete, the Transport layer shall transition to the HTDA4: HT_DMAEnd state.

Transition HTDA5:4: 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.

10.4.9 Host Transport decomposes 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. The PIO data shall be sent in a single Data FIS.

HTPS1: HT_PS_FIS	Determine the direction of the requested PIO transfer.	
1. Transfer host to device, no error detected. D bit cleared to zero.	→	HT_PIOOTrans1
2. Transfer device to host, no error detected. D bit set to one.	→	HT_HostIdle
3. Error detected.	→	HT_HostIdle
4. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTPS2: HT_PIOOTrans1	Place initial register content received from FIS into shadow registers.	
1. Unconditional	→	HT_ HostIdle

HTPS3: HT_PIOOTrans2	Wait for Link layer to indicate data transfer complete	
1. Transfer not complete	→	HT_PIOOTrans2
2. Transfer complete	→	HT_PIOEnd
3. Abort notification from Link layer	→	HT_PIOEnd
4. Notification of illegal transition error received from Link layer	→	HT_HostIdle
5. SRST asserted, or DEVICE RESET command requested	→	HT_HostIdle

HTPS4: HT_PIOEnd	Place ending register content from PIO REQ FIS into shadow registers.	
1. No error detected.	→	HT_HostIdle
2. Error detected	→	HT_HostIdle

HTPS5: HT_PIOITrans1	Wait until initial PIO data received in data frame.	
1. Received PIO data available.	→	HT_PIOITrans2[1]
2. SRST asserted, or DEVICE RESET command requested	→	HT_HostIdle
3. Notification of illegal transition error received from Link layer	→	HT_HostIdle
NOTE: 1. When transitioning to the HT_PIOITrans2 state, the starting status and Interrupt bit value from the PIO Setup FIS shall be transferred to the Shadow Status register and interrupt signal shall then reflect the value of the Interrupt bit.		

HTPS6: HT_PIOITrans2	Wait for Link layer to indicate data transfer complete.	
1. Transfer not complete	→	HT_PIOITrans2
2. Transfer complete	→	HT_PIOEnd
3. Abort notification from Link layer	→	HT_PIOEnd
4. Notification of illegal transition error received from Link layer	→	HT_HostIdle
5. SRST asserted, or DEVICE RESET command requested	→	HT_HostIdle

HTPS1: HT_PS_FIS state: This state is entered when the Link layer has indicated that a 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 HTPS1:1: When the direction of transfer requested is from host to device (D bit cleared to zero), the Transport layer shall make a transition to the HTPS2: HT_PIOOTrans1 state. This transition occurs if no error is detected.

Transition HTPS1:2: When the direction of transfer requested is from device to host (D bit set to one), the Transport layer shall make a transition to the HTI1:HT_HostIdle state. This transition occurs if no error is detected.

Transition HTPS1:3: 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.

Transition HTPS1:4: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTPS2: HT_PIOOTrans1 state: This state is entered when the direction of the requested PIO data transfer is from host to device.

When in this state, the Transport layer shall place the FIS initial register content into the shadow registers, the FIS byte count, and set the interrupt pending flag if the FIS indicates to do so.

Transition HTPS2:1: When the FIS initial register content has been placed into the shadow 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 HTI1:HT_HostIdle state.

HTPS3: HT_PIOOTrans2 state: This state is entered when PIO data is available in the PIO FIFO to be passed the Link layer.

When in this state, the Transport layer shall wait for the Link layer to indicate that all data has been transferred.

NOTE – Since the software driver or BIOS sees the DRQ bit set to one and the BSY bit cleared to zero, it continues writing the Data register filling the PIO FIFO.

Transition HTPS3:1: When the transfer is not complete, the Transport layer shall transition to the HTPS3: HT_PIOOTrans2 state.

Transition HTPS3:2: When the byte count for this DRQ data block is reached, the Transport layer shall transition to the HTPS4: HT_PIOEnd state.

Transition HTPS3:3: When notified by the Link layer that the DMA Abort primitive was received, the Transport layer shall transition to the HTPS4: HT_PIOEnd state.

Transition HTPS3:4: 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 HTPS3:5: When the host has asserted the SRST bit by writing to the Device Control register, or the DEVICE RESET command is requested, the Transport layer shall inform the Link layer to send a SYNC Escape, and the Transport layer shall transition to the HTI1:HT_HostIdle state.

HTPS4: HT_PIOEnd state: This state is entered when the PIO data transfer is complete.

When in this state, the Transport layer shall place the ending register content from the received PIO request FIS into the shadow registers.

Transition HTPS4:1: When the ending register content for the PIO request FIS has been placed into the shadow registers and there were no errors detected with the transfer, the Transport layer shall transition to the HTI1: HT_HostIdle state.

NOTE : The host should not assume received data is valid (even with a valid CRC receipt for the FIS) until command completion status is returned by the device.

Transition HTPS4:2: When the ending register content from the previous PIO Setup FIS has been placed into the shadow 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 Register FIS to the host. If the transfer was the final transfer for the associated PIO data in command, the error condition is not detectable. For data out transfers, errors detected by the device shall be reflected in the end of frame handshake. The device shall reflect the error status by transmitting an appropriate Register FIS to the host.

HTPS5: HT_PIOITrans1 state: This state is entered when the direction of the PIO data transfer is device to host.

When in this state, the Transport layer shall wait until the Link layer has begun to receive the PIO data frame and data is available to be read by the host.

Transition HTPS5:1: When data is available for the host to read in the shadow Data register, the Transport layer shall place the initial register content received in the PIO Setup frame into the shadow registers and transition to the HTPS6: HT_PIOITrans2 state.

Transition HTPS5:2: When the host has asserted the SRST bit by writing to the Device Control register, or the DEVICE RESET command is requested, the Transport layer shall inform the Link layer to send a SYNC Escape, and the Transport layer shall transition to the HTI1: HT_HostIdle state.

Transition HTPS5:3: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

HTPS6: HT_PIOITrans2 state: This state is entered when PIO data is available in the PIO FIFO to be read by the host and the initial shadow register content has been set.

When in this state, the Transport layer shall wait for the Link layer to indicate that the data transfer is complete

Transition HTPS6:1: When the transfer is not complete, the Transport layer shall transition to the HTPS6: HT_PIOITrans2 state.

Transition HTPS6:2: When the byte count for this DRQ data block is reached, the Transport layer shall transition to the HTPS4: HT_PIOEnd state.

Transition HTPS6:3: When notified by the Link layer that the DMA Abort primitive was received, the Transport layer shall transition to the HTPS4: HT_PIOEnd state.

Transition HTPS6:4: 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 HTPS6:5: When the host has asserted the SRST bit by writing to the Device Control register, or the DEVICE RESET command is requested, the Transport layer shall inform the Link layer to send a SYNC Escape, and the Transport layer shall transition to the HTI1: HT_HostIdle state.

10.4.10 Host Transport decomposes a DMA Setup FIS state diagram

This protocol receives a FIS that sets up the host adapter DMA controller to allow the transfer of subsequent Data FISes according to the First-party DMA protocol. For a a First-party DMA write request when Auto-Activate is not used, a separate DMA Activate FIS is issued by the device to trigger the start of the data transfer from the host.

HTDS1: HT_DS_FIS	Initialize the DMA controller for a First-party DMA transfer with the content of the DMA Setup FIS.		
	1. No error detected and (D bit set to one in DMA Setup FIS) or (Dbit cleared to zero and Auto-Activate bit cleared to zero in DMA Setup FIS).	→	HT_HostIdle
	2. Error detected.	→	HT_HostIdle
	3. Notification of illegal transition error received from Link layer	→	HT_HostIdle
	4. No error detected and (D bit cleared to zero and Auto-Activate bit set to one in DMA Setup FIS).	→	HT_DMAOTrans2

HTDS1: HT_DS_FIS state: This state is entered when the Link layer has indicated that a FIS is being received and that the Transport layer has determined the FIS is of the DMA Setup type.

When in this state, the Transport layer shall initialize the DMA controller with content from the FIS.

Transition HTDS1:1: When the DMA controller has been initialized and the request is a read (Direction is set to one) or the request is a write (Direction is cleared to zero) and Auto-Activate is zero in the DMA Setup FIS, the Transport layer shall transition to the HTI1: HT_HostIdle state. This transition is made if no error is detected.

Transition HTDS1:2: When an error is detected, status shall be reported to the Link layer. The Transport layer shall transition to the HTI1:HI_HostIdle state.

Transition HTDS1:3: 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 HTDS1:4: When the DMA controller has been initialized and the request is a write (Direction is cleared to zero) and Auto-Activate is one in the DMA Setup FIS, the Transport layer shall transition to the HT_DMAOTrans2 state. This transition is made if no error is detected.

10.4.11 Host transport decomposes a BIST Activate FIS state diagram

This protocol receives a BIST Activate 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 places both the transmit and receive portions of the Transport, Link and/or Physical layers into appropriate state to perform the loopback operation.

HTRBIST1: HT_RcvBIST	Determine validity of loopback mode requested.	
1. Status checked, no error detected and Loopback mode valid	→	HT_BISTTrans1
2. Status checked, no error detected and Loopback mode is invalid or not supported.	→	HT_HostIdle
3. Status checked and error detected	→	HT_HostIdle
4. Notification of illegal transition error received from Link layer	→	HT_HostIdle

HTRBIST2: HT_BISTTrans1	Notify Application layer of desired BIST modes	
1. Unconditional	→	HT_HostIdle

HTRBIST1: HT_RcvBIST state: This state is entered when the Link layer has indicated that a 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 HTRBIST1:1: 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 HTRBIST1:2: 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 HTRBIST1:3: If a reception error is indicated the Transport layer shall make a transition to the HTI1:HT_HostIdle state.

Transition HTRBIST1:4: 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.

HTRBIST2: 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 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 not defined by this specification.

Transition HTRBIST2:1: When the Application layer has been notified the Transport layer shall transition to the HTI1:HostIdle state.

10.5 Device transport states

10.5.1 Device transport idle state diagram

DTI0: DT_DeviceIdle	Device waits for FIS or FIS request.		
	1. Transmission of Register – Device to Host FIS requested by Application layer	→	DT_RegDHFIS
	2. Transmission of Set Device Bits FIS requested by Application layer	→	DT_DB_FIS
	3. Transmission of PIO Setup FIS requested by Application layer	→	DT_PIOSTUPFIS
	4. Transmission of DMA Activate FIS requested by Application layer	→	DT_DMAACTFIS
	5. Transmission of DMA Setup FIS requested by Application layer	→	DT_DMASTUPDHFIS
	6. Transmission of Data FIS requested by Application layer	→	DT_DATAIFIS
	7. Transmission of BIST Activate FIS requested by Application layer	→	DT_XmitBIST
	8. Frame receipt indicated by Link layer	→	DT_ChkTyp

DTI1: DT_ChkTyp	Received FIS type checked.		
	1. Register – Host to Device FIS type detected	→	DT_RegHDFIS
	2. Data FIS type detected	→	DT_DATAOFIS
	3. DMA Setup FIS type detected	→	DT_DMASTUPHDFIS
	4. BIST Activate FIS type detected	→	DT_RcvBIST
	5. Notification of illegal transition error received from Link layer or unrecognized FIS type	→	DT_DeviceIdle

DTI0: DT_DeviceIdle 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 Application layer to indicate that a FIS is to be transmitted or the Link layer to indicate that a FIS is being received.

Transition DTI0:1: When the Application layer indicates that a Register – Device to Host FIS is to be transmitted, the Transport layer shall make a transition to the DTR0: DT_RegDHFIS state.

Transition DTI0:2: When the 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 DTI0:3: When the 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 DTI0:4: When the 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 DTI0:5: When the Application layer indicates that a DMA Setup FIS is to be transmitted, the Transport layer shall make a transition to the DTDMASTUP0: DT_DMASTUPDHFIS state.

Transition DTI0:6: When the 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 DTI0:7: When the 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 DTI0:8: When the Link layer indicates that a 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 a FIS is being received.

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

Transition DTI1:1: When the incoming FIS is a Register – Host to Device FIS type, the Transport layer shall make a transition to the DTCMD0: DT_RegHDFIS state.

Transition DTI1:2: 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 DTI1:3: When the incoming FIS is a DMA Setup FIS type, the Transport layer shall make a transition to the DTSTP0: DT_ DMASTUPHDFIS state.

Transition DTI1:4: When the incoming FIS is a BIST Activate FIS type, the Transport layer shall make a transition to the DTRBIST1:DT_RcvBIST state.

Transition DTI1:5: 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.

10.5.2 Device Transport sends Register – Device to Host state diagram

This protocol builds a Register – Device to Host FIS that contains the register content and sends it to the host when the Application layer requests the transmission.

DTR0: DT_RegDHFIS	Construct Register – Device to Host FIS from the content of the registers and notify Link to transfer.		
	1. FIS transfer complete	→	DT_RegTransStatus
	2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTR1: DT_RegTransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
	1. Status checked, and no error detected.	→	DT_DeviceIdle
	2. Status checked, and error detected.	→	DT_RegDHFIS

DTR0: DT_RegDHFIS state: This state is entered the Application requests the transmission of a Register – Device to Host FIS.

When in this state, the Transport layer shall construct a Register – Device to Host FIS, notify the Link layer that the FIS is to be transmitted, and pass the FIS to the Link layer.

Transition DTR0:1: 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_RegTransStatus state.

Transition DTR0:2: 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_RegTransStatus 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 layer ending status for the FIS and take appropriate error handling action if required.

Transition DTR1:1: When the FIS status has been handled, and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTR1:2: When the FIS status has been handled, and an error detected, the Transport layer shall report status to the Link layer, and retry this transfer by transitioning to the DT_RegDHFISstate.

10.5.3 Device Transport sends Set Device Bits FIS state diagram

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

DTDB0: DT_DB_FIS	Inform Link to transmit Set Device Bits FIS		
	1. FIS transfer complete	→	DT_SDBTransStatus
	2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDB1: DT_SDBTransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
	1. Status checked and no error detected	→	DT_DeviceIdle
	2. Status checked and error detected.	→	DT_DB_FIS

DTDB0:DT_DB_FIS state: This state is entered when the 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 DTDB0:1: 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_SDBTransStatus state.

Transition DTDB0:2: 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.

DTDB1:DT_SDBTransStatus 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 layer ending status for the FIS and take appropriate error handling action if required.

Transition DTDB1:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0:DT_DeviceIdle state.

Transition DTDB1:2: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DTDB0:DT_DB_FIS state.

10.5.4 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 frame, a single data frame containing PIO data shall be transmitted or received depending on the state of the D bit in the PIO Setup frame.

DTPIOSTUP0: DT_PIOSTUPFIS	Construct PIO Setup – Device to Host FIS from the content provided by the Application layer and notify Link to transfer. This FIS shall include the beginning and ending register content, the byte count, and the interrupt flag.	
1. FIS transfer complete	→	DT_PIOSTUPTransStatus
2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTPIOSTUP1: DT_PIOSTUPTransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked, and no error detected.	→	DT_Device Idle
2. Status checked, and error detected.	→	DT_PIOSTUPFIS

DTPIOSTUP0: DT_PIOSTUPFIS state: This state is entered the 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 DTPIOSTUP0:1: 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_PIOSTUPTransStatus state.

Transition DTPIOSTUP0:2: 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_PIOSTUPTransStatus 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 DTPIOSTUP1:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTPIOSTUP1:2: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DT_PIOSTUPFIS state.

10.5.5 Device Transport transmit DMA Activate FIS state diagram

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

DTDMAACT0: DT_DMAACTFIS	Construct DMA Activate FIS from the content provided by the Application layer and notify Link to transfer.	
1. FIS transfer complete	→	DT_DMAACTTransStatus
2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDMAACT1: DT_DMAACTTransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked, and no error detected.	→	DT_DeviceIdle
2. Status checked, and error detected.	→	DT_DMAACTFIS

DTDMAACT0: DT_DMAACTFIS state: This state is entered when the Application layer 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 DTDMAACT0:1: 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_DMAACTTransStatus state.

Transition DTDMAACT0:2: 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_DMAACTTransStatus state: This state 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 DTDMAACT1:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDMAACT1:2: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DT_DMAACTFIS state.

10.5.6 Device Transport transmit DMA Setup – Device to Host FIS state diagram

This protocol transmits a 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 DMA Setup – Device to Host request includes the transfer direction indicator, the host buffer identifier, the host buffer offset, the byte count, and the interrupt flag.

DTDMASTUP0: DT_DMASTUPDHFIS	Construct the DMA Setup – Device to Host FIS from the content provided by the Application layer and notify Link to transfer.		
	1. FIS transfer complete	→	DT_DMASTUPTrans Status
	2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDMASTUP1: DT_DMASTUPTransStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.		
	1. Status checked, and no error detected.	→	DT_DeviceIdle
	2. Status checked, and error detected.	→	DT_DMASTUPDHFIS

DTDMASTUP0: DT_DMASTUPDHFIS state: This state is entered when the Application layer requests the transmission of a DMA Setup – Device to Host FIS.

When in this state, the Transport layer shall construct a 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 DTDMASTUP0:1: 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_DMASTUPTransStatus state.

Transition DTIDMASTUP0:2: 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_DMASTUPTransStatus 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 DTDMASTUP1:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDMASTUP1:2: When the FIS status has been handled and an error detected, the Transport layer shall report status to the Link layer and retry this transfer by transitioning to the DT_DMASTUPDHFIS state.

10.5.7 Device Transport transmit Data – Device to Host FIS diagram

This protocol builds a Data – Device to Host FIS.

DTDATAI0: DT_DATAIFIS	Construct Data – Device to Host FIS content from the content provided by the Application layer and notify Link to transfer.	
1. Unconditional	→	DT_DATAITrans

DTDATAI1: DT_DATAITrans	Pass data Dwords from data FIFO to Link layer	
1. Transfer not complete	→	DT_DATAITrans
2. Transfer complete	→	DT_DATAIEnd
3. Application layer requests termination of DMA in transfer.	→	DT_DATAIEnd
4. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDATAI2: DT_DATAIEnd	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status checked, and no error detected.	→	DT_DeviceIdle
2. Status checked, and error detected.	→	DT_DeviceIdle
3. Application layer requests termination of DMA in transfer, no error detected.	→	DT_DeviceIdle
4. Application layer requests termination of DMA in transfer, error detected.	→	DT_DeviceIdle
5. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDATAI0: DT_DATAFIS state: This state is entered when the Application layer has requested the transmission of a Data – Device to Host FIS.

When in this state the Transport layer shall pass a portion of the DMA data to the Link layer.

Transition DTDATAI0:1: When ready and there is data in the FIFO 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 in the FIFO to be passed the Link layer.

When in this state, the Transport layer shall pass a Dword of data from the FIFO to the Link layer.

Transition DTDATAI1:1: When the transfer is not complete, the Transport layer shall transition to the DTDATAI1: DT_DATAITrans state.

Transition DTDATAI1:2: When the transfer is complete, the Transport layer shall transition to the DTDATAI2: DT_DATAIEnd state.

Transition DTDATAI1:3: When the Application layer requests that a DMA operation is to be aborted, the Transport layer shall transition to the DTDATAI2:DT_DATAIEnd state.

Transition DTDATAI1:4: 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 layer and Phy layer ending status for the FIS and take appropriate error handling action if required.

Transition DTDATAI2:1: When the FIS status has been handled and no error detected, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDATAI2:2: When the FIS status has been handled and an error detected, status shall be reported to the Link layer. The Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTDATAI2:3: When the Application layer requests the termination of a DMA data in transaction, it reports the abort condition to the Link layer, waits for an EOF_P and, if no error is detected, shall transition to the DTI0:DT_DeviceIdle state.

Transition DTDATAI2:4: When the Application layer requests the termination of a DMA data in transaction, it reports the abort condition to the Link layer, waits for an EOF_P, and if an error is detected, reports the error to the Link layer, and shall transition to the DTI0:DT_DeviceIdle state.

Transition DTDATAI2:5: When the Transport layer receives notification from the Link layer of an illegal state transition, the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

10.5.8 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 specification.

DTXBIST1: DT_XmitBIST	Construct the BIST Activate FIS from the content provided by the Application layer and notifies Link to transfer.	
1. FIS transfer complete	→	DT_TransBISTStatus
2. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTXBIST2: DT_TransBISTStatus	Check Link and Phy transmission results and if an error occurred take appropriate action.	
1. Status check completed	→	DT_DeviceIdle
2. Status check and at least one error detected	→	DT_XmitBIST[1]
NOTE: 1. Re-transmission of the BIST Activate FIS due to errors is not required but allowed.		

DTXBIST1: DT_XmitBIST state: This state is entered to send a BIST FIS to the host.

Transition DTXBIST1:1: 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 DTXBIST2:DT_TransBISTSTatus state.

Transition DTXBIST1:2: 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.

DTXBIST2: DT_TransBISTStatus state: This state is entered when the entire FIS has been passed to the Link layer.

Transition DTXBIST2:1: When the FIS transmission is completed the Transport layer shall transition to the DTI0: DT_DeviceIdle state.

Transition DTXBIST2:2: When the FIS transmission is completed and at least one error is detected, the Transport layer may transition to the DTXBIST1: DT_XmitBIST state.

10.5.9 Device Transport decomposes Register – Host to Device state diagram

This protocol receives a Register – Host to Device FIS, places received register content into the device registers, and notifies the Application layer of the FIS receipt.

DTCMD0: DT_RegHDFIS	Receive a Register – Host to Device FIS		
	1. FIS transfer complete, and no error detected.	→	DT_DeviceIdle
	2. FIS transfer complete, and error detected	→	DT_DeviceIdle
	3. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTCMD0: DT_RegHDFIS state: This state is entered when the receipt of a DT_RegHDFISFIS is recognized.

When in this state, the Transport layer shall receive the FIS and place the contents of the FIS into the device registers when it is determined that the FIS was received without error.

Transition DTCMD0:1: When the entire FIS has been received from the Link layer without error, the Transport layer shall indicate to the Application layer that a command FIS was received and make a transition to the DTI0: DT_DeviceIdle state.

Transition DTCMD0:2: 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 DTCMD0:3: 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.

10.5.10 Device Transport decomposes Data (Host to Device) FIS state diagram

This protocol receives a Data - Host to Device FIS.

DTDATAO0: DT_DATAOFIS	Prepare to receive data		
	Unconditional	→	DT_DATAOREC

DTDATAO1: DT_DATAOREC	Place received data Dword into data FIFO and signal Link to continue transfer.		
	1. Transfer not complete	→	DT_DATAOREC
	2. Transfer complete	→	DT_DeviceIdle
	3. Abort Transfer from Application layer	→	DT_DeviceAbort
	4. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTDATAO2: DT_DeviceAbort	Signal Link to abort transfer.		
	1. Transfer not complete	→	DT_DATAOREC
	2. Transfer complete	→	DT_ DeviceIdle

DTDATAO0: DT_DATAOFIS state: This state is entered when the Link layer has indicated that a 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 DTDATAO0:1: 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 DTDATAO1:1: 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 DTDATAO1:2: 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.

Transition DTDATAO1:3: When the Application layer indicates that the FIS is to be aborted, the Transport layer shall transition to the DTDATAO2: DT_DeviceAbort state.

Transition DTDATAO1:4: 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.

DTDATAO2: DT_DeviceAbort state: This state is entered when the Application layer indicates that the current transfer is to be aborted.

When in this state, the Transport layer shall signal the Link layer to Abort the incoming transmission and return to either DT_DATAOREC or DT_DeviceIdle, depending upon the current state of the FIFO. If the abort occurs coincident with an end of transfer indication from the Link, then the transition to DTI0: DT_DeviceIdle is also accommodated. After issuing an Abort, the Transport returns to normal data transfer, and awaits the end of transfer indication from the Link.

Transition DTDATAO2:1: Inform Link layer to issue an abort. When Transfer is not complete, the Transport layer shall transition to the DTDATAO1: DT_DATAOREC state.

Transition DTDATAO2:2: 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.

10.5.11 Device Transport decomposes DMA Setup – Host to Device state diagram

This protocol receives a DMA Setup – Host to Device FIS, passes received DMA Setup content, and notification of FIS receipt to the Application layer.

DTSTP0: DT_DMASTUPHDFIS	Receive a DMA Setup – Host to Device FIS		
	1. FIS transfer complete, and no error detected.	→	DT_DeviceIdle
	2. FIS transfer complete, and error detected	→	DT_DeviceIdle
	3. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTSTP0: DT_DMASTUPHDFIS state: This state is entered when the receipt of a DT_DMASTUP FIS is recognized.

Transition DTSTP0:1: When the entire FIS has been received from the Link layer without error, the Transport layer shall indicate to the Application layer that a DMA Setup – Host to Device FIS was received and make a transition to the DTI0: DT_DeviceIdle state.

Transition DTSTP0:2: 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 DTSTP0:3: 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.

10.5.12 Device Transport decomposes a BIST Activate FIS state diagram

This protocol receives a 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 places both the transmit and receive portions of the Transport, Link and/or Physical layers into appropriate state to perform the loopback operation.

DTRBIST1: DT_RcvBIST	Determine validity of loopback mode requested.	
1. Status checked, no error detected and Loopback mode valid	→	DT_BISTTrans1
2. Status checked, no error detected and Loopback mode is invalid or not supported.	→	DT_DeviceIdle
3. Status checked and error detected	→	DT_DeviceIdle
4. Notification of illegal transition error received from Link layer	→	DT_DeviceIdle

DTRBIST2: DT_BISTTrans1	Notify Application layer of desired BIST modes	
1. Unconditional	→	DT_DeviceIdle

DTRBIST1: DT_RcvBIST state: This state is entered when the Link layer has indicated that a 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 DTRBIST1:1: If no reception error is detected and the FIS contents indicate a form of loopback request that is supported by the device the Transport layer shall make a transition to the DTRBIST2: DT_BISTTrans1 state.

Transition DTRBIST1:2: If no reception error is detected and the FIS contents indicate a form of loopback request that is not supported by the device the Transport layer shall make a transition to the DTI0:DT_DeviceIdle state.

Transition DTRBIST1:3: If a reception error is indicated the Transport layer shall make a transition to the DTI0:DT_DeviceIdle state.

Transition DTRBIST1:4: 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.

DTRBIST2: 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 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 not defined by this specification.

Transition DTRBIST2:1: When the Application layer has been notified the Transport layer shall transition to the DTI0:DT_DeviceIdle state.

11 Device Command Layer protocol

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 start executing the COMRESET protocol from the beginning. If the device receives a Register – Host to Device FIS with C bit cleared to zero and the SRST bit set to one before the device has completed executing a command layer protocol, then the device shall start executing its software reset protocol from the beginning.

SYNC Escape is used by a host or device to bring the link back to a known state, and may be used for vendor specific recovery of error or hang conditions. After a SYNC Escape is performed, a software reset may be necessary prior to issuing the next command to the device.

11.1 Power-on and COMRESET protocol

If the host sends a hardware reset (power-on reset or COMRESET) regardless of the power management mode or the current device command layer state, the device shall execute the hardware reset protocol. Assertion of hardware reset is associated with entry into state DP1:DR_Reset within the Phy state machine.

DHR0: Hardware_reset_asserted	Wait.	
1. Hardware reset negated.	→	DHR1: Execute_diagnostics

DHR1: Execute_diagnostics	Initialize hardware and execute diagnostics.	
1. Initialization and diagnostics completed successfully.	→	DHR2: Send_good_status
2. Initialization completed and diagnostics failed.	→	DHR3: Send_bad_status

DHR2: Send_good_status	Request transmission of Register FIS with good status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DHR3: Send_bad_status	Request transmission of Register FIS with bad status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DHR0: Hardware_reset_asserted: This state is entered when the Transport layer indicates that a hardware reset (power-on reset or COMRESET) is asserted.

When in this state, the device awaits the negation of a hardware reset which is associated with exit from state DP1:DR_Reset within the Phy state machine.

Transition DHR0:1: When the Transport layer indicates that hardware reset 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 DHR1:1: 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 DHR1:2: 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 Register FIS to the host. If the device does not implement the PACKET command feature set the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	00h
LBA(23:16)	00h
Device	na
Error	01h
Status	00h-70h (see note)
NOTE – Setting of bits [6:4] in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	14h
LBA(23:16	EBh
Device	na
Error	01h
Status	00h

Transition DHR2:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

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

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. If the device does not implement the PACKET command feature set the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	00h
LBA(23:16	00h
Device	na
Error	00h, 02h-7Fh
Status	00h-70h (see note)
NOTE – Setting of bits [6:4] in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	14h
LBA(23:16	EBh
Device	na
Error	00h, 02h-7Fh
Status	00h

Transition DHR3:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

11.2 Device Idle protocol

The state diagram below describes the idle protocol for a device. States and transitions preceded by an * are utilized when Native Command Queuing or ATA Tagged Command Queuing commands are implemented.

DI0: Device_idle	Wait.	
1. FIS receipt	→	DI1: Check_FIS
2. * Ready to complete released command.	→	DI4: Set_service
3. * Ready to receive data for WRITE FPDMA QUEUED command and FIS receipt not indicated and no error encountered	→	DFPDMAQ4: DataPhase_ PreWriteSetup
4. * Ready to transmit data for READ FPDMA QUEUED command and FIS receipt not indicated and no error encountered	→	DFPDMAQ3: DataPhase_ ReadSetup
5. * One or more FPDMA QUEUED or NCQ QUEUE MANAGEMENT commands completed successfully and FIS receipt not indicated and no error encountered	→	DFPDMAQ10: SendStatus[1]
6. * FPDMA QUEUED or NCQ QUEUE MANAGEMENT command terminated with failure and FIS receipt not indicated	→	DFPDMAQ11: ERROR
7. Asynchronous Notification is enabled and event has occurred that requires notification and NotifyPending = 0 and FIS receipt not indicated.	→	AN0: Notify_host
NOTE: 1. This condition may be true simultaneously with condition 3 or 4. Devices implementing status aggregation may select any of the transitions 3, 4, or 5 if their conditions evaluate to true. Devices not implementing status aggregation shall prioritize transition 5 over transitions 3 and 4.		

DI1: Check_FIS	Check_FIS type and C bit.	
1. Register type, C bit cleared to zero, and SRST set to one.	→	DSR0: Software_reset_ asserted
2. Register type, C bit cleared to zero, and SRST cleared to zero.	→	DI0: Device_idle[1]
3. Register type and C bit set to one.	→	DI2: Check_command
4. DMA Setup FIS Received	→	DI0 : Device_idle
5. Unexpected FIS type.	→	DI0: Device_idle
NOTE: 1. A Device to Host Register FIS shall not be sent in response to the received Register FIS.		

DI2: Check_command[1]	Check the command to determine required command protocol. If asynchronous notification is supported then NotifyPending is cleared to zero.

1. Non-data command protocol and no native queued command outstanding.	→	DND0: Non-data
2. PIO data-in command protocol and no native queued command outstanding.	→	DPIOI0: PIO_in
3. PIO data-out command protocol and no native queued command outstanding.	→	DPIOO0: PIO_out
4. READ DMA command protocol and no native queued command outstanding.	→	DDMAI0: DMA_in
5. WRITE DMA command protocol and no native queued command outstanding.	→	DDMAO0: DMA_out
6. PACKET command protocol and no native queued command outstanding.	→	DPKT0: PACKET
7. * READ DMA QUEUED command protocol and no native queued command outstanding.	→	DDMAQI0: DMA_queued_in
8. * WRITE DMA QUEUED command protocol and no native queued command outstanding.	→	DDMAQO0: DMA_queued_out
9. EXECUTE DEVICE DIAGNOSTIC command protocol and no native queued command outstanding.	→	DEDD0: Execute_device_diag
10. DEVICE RESET command protocol.	→	DDR0: Device_reset
11. Command not implemented and no native queued command outstanding.	→	DI3: No_command
12. * SERVICE command protocol and no native queued command outstanding.	→	DI5: Service_test
13. * READ FPDMA QUEUED command protocol.	→	DFPDMAQ1: AddCommand_ ToQueue
14. * WRITE FPDMA QUEUED command protocol.	→	DFPDMAQ1: AddCommand_ ToQueue
15. * NCQ QUEUE MANAGEMENT command protocol.	→	DFPDMAQ1: AddCommand_ ToQueue
16. * Not READ FPDMA QUEUED and not WRITE FPDMA QUEUED and not NCQ QUEUE MANAGEMENT and not DEVICE RESET and native queued command(s) outstanding	→	DFPDMAQ12: BrokenHost_ ClearBusy
NOTE: 1. This state shows transitions for all commands. If a device does not implement any particular command, then that transition should not be processed.		

DI3: No_command	Request transmission of Register FIS with ABRT bit set to one.

1. FIS transmission complete.	→	DI0: Device_idle

* DI4: Set_service	Request transmission of Set Device Bits FIS with SERV set.

1. FIS transmission complete.	→	DI0: Device_idle

* DI5: Service_test	Test command to see if Register FIS is needed to set DRQ bit and set Tag.	
1. PACKET PIO data-in or PACKET PIO data-out.	→	DI7: Service_decode
2. Other	→	DI6: Service_send_tag

* DI6: Service_send_tag	Request transmission of Register FIS with BSYbit cleared to zero, DRQ bit set to one, and appropriate Tag	
1. FIS transmission complete	→	DI7: Service_decode

* DI7: Service_decode	Check command type to be serviced.	
1. PACKET PIO data-in.	→	DPKT4: PACKET_PIO_in
2. PACKET PIO data-out.	→	DPKT6: PACKET_PIO_out
3. PACKET DMA data-in.	→	DPKT9: PACKET_DMA_in
4. PACKET DMA data-out.	→	DPKT11: PACKET_DMA_out
5. READ DMA QUEUED.	→	DDMAQI1: Send_data
6. WRITE DMA QUEUED.	→	DDMAQO1: Send_DMA_activate

DI0: Device_Idle: This state is entered when the device has completed the execution of a command protocol, a COMRESET protocol, a software reset protocol, or a queued command has been released.

When in this state, the device is awaiting a command. If queuing is supported, the device may be waiting to acquire data or establish buffer space to complete a queued command

Transition DI0:1: When the device receives a FIS from the Transport layer, the device shall transition to the DI1: Check_FIS state.

*** Transition DI0:2:** When the device is ready to complete the data transfer for a queued command, the device shall transition to the DI4: Set_service state.

*** Transition DI0:3:** When the device is ready to receive the data for a WRITE FPDMA QUEUED command, the device shall transition to the DFPDMAQ4: DataPhasePreWriteSetup state. This condition also applies for the case where non-zero buffer offsets are used to complete a previous partial data transfer.

*** Transition DI0:4:** When the device is ready to transmit the data for a READ FPDMA QUEUED command, the device shall transition to the DFPDMAQ3: DataPhaseReadSetup state. This condition also applies for the case where non-zero buffer offsets are used to complete a previous partial data transfer.

*** Transition DI0:5:** When the device has successfully completed a FPDMA QUEUED or a NCQ QUEUE MANAGEMENT command, the device shall transition to the DFPDMAQ10: SendStatus state.

* **Transition DI0:6:** When the device has encountered an error in a FPDMA QUEUED or a NCQ QUEUE MANAGEMENT command, the device shall transition to the DFPDMAQ11: ERROR state.

Transition DI0:7: If Asynchronous Notification is enabled and an event requiring notification of the host has occurred and the NotifyPending variable is cleared to zero and FIS receipt not indicated, the device shall transition to the AN0: Notify_host state.

DI1: Check_FIS state: This state is entered when the device receives a FIS from the Transport layer.

When in this state, the device shall check the FIS type.

Transition DI1:1: If the FIS type is a Register FIS, the C bit in the FIS is cleared to zero, and the SRST bit in the FIS is set to one, the device shall transition to the DSR0: Software_reset_asserted state.

Transition DI1:2: If the FIS type is a Register FIS, the C bit in the FIS is cleared to zero, and the SRST bit in the FIS is cleared to zero, the device shall transition to the DI0: Device_idle state.

Transition DI1:3: If the FIS type is a Register FIS and the C bit in the FIS is set to one, the device shall transition to the DI2: Check_command state.

Transition DI1:4: If the FIS type is a First Party DMA Setup FIS, the device shall inform the Transport layer of the reception of the First Party DMA Setup FIS, and transition to the DI0: Device_idle state.

Transition DI1:5: For any other FIS, the device shall transition to the DI0: Device_idle state.

DI2: Check_command state: This state is entered when the device recognizes that the received Register FIS contains a new command. NOTE: This state shows transitions for all commands. If a device does not implement any particular command, then transition DI2:11 to state DI3:No_command shall be made.

When in this state, the device shall check the command protocol required by the received command and clears NotifyPending to zero if asynchronous notification is supported. Clearing NotifyPending to zero allows future asynchronous notification messages to be sent to the host.

Transition DI2:1: When the received command is a non-data transfer command, the device shall transition to the DND0: Non-data state.

Transition DI2:2: When the received command is a PIO data-in command, the device shall transition to the DPIOI0: PIO_in state.

Transition DI2:3: When the received command is a PIO data-out command, the device shall transition to the DPIOO0: PIO_out state.

Transition DI2:4: When the received command is a READ DMA command, the device shall transition to the DDMAI0: DMA_in state.

Transition DI2:5: When the received command is a WRITE DMA command, the device shall transition to the DDMAO0: DMA_out state.

Transition DI2:6: When the received command is a PACKET command, the device shall transition to the DPKT0: PACKET state.

* **Transition DI2:7:** When the received command is a READ DMA QUEUED command, the device shall transition to the DDMAQI0: DMA_queued_in state.

* **Transition DI2:8:** When the received command is a WRITE DMA QUEUED command, the device shall transition to the DDMAQO0: DMA_queued_out state.

Transition DI2:9: When the received command is an EXECUTE DEVICE DIAGNOSTICS command, the device shall transition to the DEDD0: Execute_device_diag state.

Transition DI2:10: When the received command is an RESET DEVICE command, the device shall transition to the DDR0: Device_reset state.

Transition DI2:11: When the received command is not implemented by the device, the device shall transition to the DI3: No_command state.

* **Transition DI2:12:** When the received command is a SERVICE command, the device shall transition to the DI5: Service_test state.

* **Transition DI2:13:** When the received command is a READ FPDMA QUEUED command protocol, the device shall transition to the DFPDMAQ1: AddCommandToQueue state.

* **Transition DI2:14:** When the received command is a WRITE FPDMA QUEUED command protocol, the device shall transition to the DFPDMAQ1: AddCommandToQueue state.

* **Transition DI2:15:** When the received command is a NCQ QUEUE MANAGEMENT command protocol, the device shall transition to the DFPDMAQ1: AddCommandToQueue state.

* **Transition DI2:16:** When the received command is a not a READ FPDMA QUEUED; and not a WRITE FPDMA QUEUED; and not a NCQ QUEUE MANAGEMENT; and not a DEVICE RESET; and there are native queued command(s) outstanding, an error has occurred and the device shall transition to the DFPDMAQ12: BrokenHost_ClearBusy state.

DI3: No_command state: This state is entered when the device recognizes that the received command is not implemented by the device.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one.

Transition DI2:1: When the Transport layer has transmitted the Register FIS, the device shall transition to the DI0: Device_idle state.

*** DI4: Set_service state:** This state is entered when the ready to complete the data transfer for a queued command.

When in this state, the device shall request that the Transport layer transmit a Set Device Bits FIS with the SERV bit set in the Status register and with all other bits in the Error and Status fields the same as the current contents of the respective registers, and the Interrupt bit set to one.

Transition DI4:1: 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 command using the PIO protocol provides its own register update to set DRQ and send the command tag, but other queued commands require a Register FIS.

Transition DI5:1: 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 DI5:2: When the command to be serviced is neither a PIO data-in nor a PIO data-out command, the device shall transition to the DI6: Service_send_tag state.

*** DI6: Service_send_tag state:** This state is entered when the SERVICE command has been received and sending a Register Device to Host FIS is necessary for the command being serviced.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register contents, including the desired command tag, as described in the command description of the ATA8-ACS standard for the command being serviced.

Transition DI6:1: When the Transport layer has transmitted the Register FIS, the device shall transition to theDI7: Service_decode state.

*** DI7: Service_decode state:** This state is entered when a Register FIS has been transmitted, if necessary to send the register contents, including the desired command tag, in response to a SERVICE command.

When in this state, the device shall again determine the type of command that the device has requested service to complete, and branch to that command's data transfer and completion.

Transition DI7:1: When the command to be serviced is a PIO data-in command, the device shall transition to the DPKT4: PACKET_PIO_in state.

Transition DI7:2: When the command to be serviced is a PIO data-out command, the device shall transition to the DPKT6: PACKET_PIO_out state.

Transition DI7:3: When the command to be serviced is a DMA data-in command, the device shall transition to the DPKT9: PACKET_DMA_in state.

Transition DI7:4: When the command to be serviced is a DMA data-out command, the device shall transition to the DPKT11: PACKET_DMA_out state.

Transition DI7:5: When the command to be serviced is a READ DMA QUEUED command, the device shall transition to the DDMAQI1: Send_data state.

Transition DI7:6: When the command to be serviced is a WRITE DMA QUEUED command, the device shall transition to the DDMAQO1: Send_DMA_activate state.

11.3 Software reset protocol

When the host sends a Register FIS with a one in the SRST bit position of the Device Control register byte, regardless of the device power management mode (e.g. SLEEP, STANDBY), the device shall execute the software reset protocol.

DSR0: Software_reset_asserted	Begin initialization and diagnostic execution.	
1. Software reset negated.	→	DSR1: Execute_diagnostics
2. Software reset asserted.	→	DSR0: Software_reset_ asserted

DSR1: Execute_diagnostics	Complete Initialization and diagnostics execution.	
1. Initialization and diagnostics completed successfully.	→	DSR2: Send_good_status
2. Initialization completed and diagnostics failed.	→	DSR3: Send_bad_status

DSR2: Send_good_status	Request transmission of Register FIS with good status.	
1. Register FIS transmitted.	→	DI0: Device_idle

DSR3: Send_bad_status	Request transmission of Register FIS with bad status.	
1. Register FIS transmitted.	→	DI0: Device_idle

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

When in this state, the device begins its initialization and diagnostics execution and awaits the clearing of the SRST bit.

Transition DSR0:1: When a Register FIS is received with the C bit in the FIS cleared to zero and the SRST bit cleared to zero in the Device Control register, the device shall transition to the DSR1: Execute_diagnostics state.

Transition DSR0:2: If a Register FIS is received with the C bit in the FIS set to one, or the SRST bit set to one in the Device Control register, the device shall transition to the DSR0: Software_reset_asserted state.

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

When in this state, the device completes initialization and execution of its diagnostics.

Transition DSR1:1: When the device has been initialized and the diagnostics successfully completed, the device shall transition to the DSR2: Send_good_status state.

Transition DSR1:2: When the device has been initialized and the diagnostics failed, the device shall transition to the DSR3: Send_bad_status state.

DSR2: Send_good_status: This state is entered when the device has been initialized and the diagnostics successfully completed.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. If the device does not implement the PACKET command feature set the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	00h
LBA(23:16	00h
Device	na
Error	01h
Status	00h-70h (see note)
NOTE – Setting of bits [6:4] in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	14h
LBA(23:16	EBh
Device	na
Error	01h
Status	00h

Transition DSR2:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

DSR3: Send_bad_status: This state is entered when the device has been initialized and the diagnostics failed.

When in this state, the device requests that the Transport layer transmit a Register FIS to the host. If the device does not implement the PACKET command feature set the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	00h
LBA(23:16	00h
Device	na
Error	00h, 02h-7Fh
Status	00h-70h (see note)
NOTE – Setting of bits [6:4] in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	14h
LBA(23:16	EBh
Device	na
Error	00h, 02h-7Fh
Status	00h

Transition DSR3:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

11.4 EXECUTE DEVICE DIAGNOSTIC command protocol

If the host sends COMRESET before the device has completed executing the EXECUTE DEVICE DIAGNOSTIC protocol, then the device shall immediately start executing the COMRESET protocol form the beginning. If the host asserts SRST in the Device Control register before the device has completed executing the EXECUTE DEVICE DIAGNOSTIC protocol, then the device shall immediately start executing its software reset protocol from the beginning.

DEDD0: Execute_device_diag	Execute diagnostics.		
	1. Diagnostics completed successfully.	→	DEDD1: Send_good_status
	2. Diagnostics failed.	→	DEDD2: Send_bad_status

DEDD1: Send_good_status	Request transmission of Register FIS with good status.		
	1. Register FIS transmitted.	→	DI0: Device_idle

DEDD2: Send_bad_status	Request transmission of Register FIS with bad status.		
	1. Register FIS transmitted.	→	DI0: Device_idle

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 DEDD0:1: When the device successfully completed the diagnostics, the device shall transition to the DEDD1: Send_good_status state.

Transition DEDD1:2: When the device has failed the diagnostics, the device shall transition to the DEDD2: Send_bad_status state.

DEDD1: Send_good_status: This state is entered when the device has successfully completed the diagnostics.

When in this state, the device shall request that the Transport layer transmit a Register FIS to the host, with the Interrupt bit set to one. If the device does not implement the PACKET command feature set the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	00h
LBA(23:16	00h
Device	na
Error	01h
Status	00h-70h (see note)
NOTE – Setting of bits [6:4] in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	14h
LBA(23:16	EBh
Device	na
Error	01h
Status	00h

Transition DEDD1:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

DEDD2: Send_bad_status: This state is entered when the device has been initialized and the diagnostics failed.

When in this state, the device shall request that the Transport layer transmit a Register FIS to the host, with the Interrupt bit set to one. If the device does not implement the PACKET command feature set the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	00h
LBA(23:16	00h
Device	na
Error	00h, 02h-7Fh
Status	00h-70h (see note)
NOTE – Setting of bits [6:4] in the Status register are device specific.	

If the device implements the PACKET command feature set, the register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	14h
LBA(23:16	EBh
Device	na
Error	00h, 02h-7Fh
Status	00h

Transition DEDD2:1: When the Transport layer indicates that the Register FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

11.5 DEVICE RESET command protocol

If the host sends COMRESET before the device has completed executing the DEVICE RESET protocol, then the device shall immediately start executing the COMRESET protocol from the beginning. If the host asserts SRST in the Device Control register before the device has completed executing the DEVICE RESET protocol, then the device shall immediately start executing its software reset protocol from the beginning.

DDR0: Device_reset	Stop execution and background activity.	
1. Activity ceased.	→	DDR1: Send_good_status

DDR1: Send_good_status	Request transmission of Register – Device to Host FIS with good status.	
1. Register FIS transmitted.	→	DI0: Device_idle

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 DDR0:1: 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 Register – Device to Host FIS to the host.

The register content shall be:

Count(7:0)	01h
LBA(7:0)	01h
LBA(15:8)	14h
LBA(23:16	EBh
Device	na
Error	01h
Status	00h

Transition DDR1:1: When the Transport layer indicates that the Register – Device to Host FIS has been transmitted, the device shall transition to the DI0: Device_Idle state.

11.6 Non-data command protocol

Execution of this class of command involves no data transfer. See the NOP command description and the SLEEP command description for additional protocol requirements.

DND0: Non-data	Execute Non-data command.		
	1. Command execution complete.	→	DND1: Send_status

DND1: Send_status	Request transmission of a Register FIS.		
	1. FIS transmission complete.	→	DI0: Device_idle

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: 1: 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 Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one.

Transition DND1:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

11.7 PIO data-in command protocol

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

DPIOI0: PIO_in	Prepare a DRQ data block for transfer to the host.		
	1. DRQ data block ready to transfer and no error encountered.	→	DPIOI1: Send_PIO_setup
	2. Error encountered during command execution.	→	DPIOI3: Error_status

DPIOI1: Send_PIO_setup	Request transmission of a PIO Setup FIS to host.		
	1. PIO Setup FIS transmitted.	→	DPIOI2: Transmit_data

DPIOI2: Transmit_data	Request transmission of a Data FIS to host.		
	1. Data FIS transmitted, no more data transfer required for this command.	→	DI0: Device_idle
	2. Data FIS transmitted, more data transfer required for this command, or 2048 Dwords transmitted.	→	DPIOI0: PIO_in

DPIOI3: Error_status	Request transmission of a Register - Device to Host FIS.		
	1. FIS transmission complete.	→	DI0: Device_idle

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 DPIOI0:1: When the device has a DRQ data block ready to transfer and no error was encountered, the device shall transition to the DPIOI1: Send_PIO_setup state.

Transition DPIOI0:2: When the device has encountered an error during command execution, 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 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 bit cleared to zero and DRQ bit set to one and with register content as described in the command description in the ATA8-ACS standard. The Interrupt bit shall be set. If this is the last DRQ data block requested by the command, the ending status shall have BSY bit cleared to zero and DRQ bit cleared to zero. If this is not the last data block requested by the command, the ending status shall have BSY bit set to one and DRQ bit cleared to zero.

Transition DPIOI1:1: 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 DPIOI2:1: When the Data FIS has been transferred and all data requested by this command have been transferred, the device shall transition to the DI0: Device_idle.

Transition DPIOI2:2: When the Data FIS has been transferred but all data requested by this command has not been transferred, or the 2048 Dword 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 Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one. In addition to the ATA8-ACS requirements, the device may set bit 7 of the error field in the FIS to one if a CRC error was encountered in transmission of a previous FIS for this command.

Transition DPIOI3:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

11.8 PIO data-out command protocol

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

DPIOO0: PIO_out	Prepare to receive DRQ data block transfer from the host.	
1. Ready to receive DRQ data block transfer.	→	DPIOO1: Send_PIO_setup
2. All DRQ data blocks received or command aborted due to error.	→	DPIOO3: Send_status

DPIOO1: Send_PIO_setup	Request transmission of a PIO Setup FIS to host.	
1. PIO Setup FIS transmitted.	→	DPIOO2: Receive_data

DPIOO2: Receive_data	Receive Data FIS from the Transport layer.	
1. Data FIS received.	→	DPIOO0: PIO_out

DPIOO3: Send_status	Request transmission of a Register - Device to Host FIS.	
1. FIS transmission complete.	→	DI0: Device_idle

DPIOO0: 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 DPIOO0:1: When the device is ready to receive a DRQ data block, the device shall transition to the DPIOO1: Send_PIO_setup state.

Transition DPIOO0:2: When the device has received all DRQ data blocks requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data, then the device shall transition to the DPIOO3: Send_status state.

DPIOO1: Send_PIO_setup: This state is entered when the device is ready to receive a DRQ data block from the host.

When in this state, the device shall request that the Transport layer transmit a PIO Setup FIS. The initial status shall have BSY bit cleared to zero and DRQ bit set to one. If this is the first DRQ data block for this command, the Interrupt bit shall be cleared to zero. If this is not the first DRQ data block for this command, the Interrupt bit shall be set to one. The ending status shall have BSY bit set to one and DRQ bit cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DPIOO1:1: When the PIO Setup FIS has been transferred, the device shall transition to the DPIOO2: Receive_data state.

DPIOO2:Receive_data: This state is entered when the device has transmitted a PIO Setup FIS to the host.

When in this state, the device shall receive the requested Data FIS from the Transport layer.

Transition DPIOO2:1: When the Data FIS has been received, the device shall transition to the DPIOO0: PIO_out state.

DPIOO3: Send_status: This state is entered when the device has received all DRQ data blocks requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one. In addition to the ATA8-ACS requirements, the device may set bit 7 of the error field in the FIS to one if a CRC error was encountered in transmission of a previous FIS for this command.

Transition DPIOO3:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

11.9 DMA data in command protocol

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.

DDMAI0: DMA_in	Prepare data for the transfer of a Data FIS.		
	1. Data for Data FIS ready to transfer.	→	DDMAI1: Send_data
	2. Command completed or aborted due to error.	→	DDMAI2: Send_status

DDMAI1: Send_data	Request transmission of a Data FIS to host.		
	1. Data FIS transmitted. No more data transfer required for this command, or 2048 Dwords transmitted.	→	DDMAI0: DMA_in

DDMAI2: Send_status	Request transmission of a Register FIS to host.		
	1. Register FIS transmitted.	→	DI0: Device_idle

DDMAI0: DMA_in State: This state is entered when the device receives a DMA data-in command or the transmission of one or more data FIS is required to complete the command.

When in this state, device shall prepare the data for transfer of a data FIS to the host.

Transition DDMAI0:1: When the device has the data ready to transfer a data FIS, the device shall transition to the DDMAI1: Send_data state.

Transition DDMAI0:2: 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 DDMAI1:1: 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 Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one.

Transition DDMAI2:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

11.10 DMA data out command protocol

Execution of this class of command includes the transfer of one or more blocks of data from the host to the device using DMA transfer. A single interrupt is issued at the completion of the successful transfer of all data required by the command.

DDMAO0: DMA_out	Prepare to receive a Data FIS from the host.	
1. Ready to receive Data FIS.	→	DDMAO1: Send_DMA_activate
2. All data requested for this command received or command aborted due to error.	→	DDMAO3: Send_status

DDMAO1: Send_DMA_activate	Request transmission of a DMA Activate FIS to host.	
1. DMA Activate FIS transmitted.	→	DDMAO2: Receive_data

DDMAO2: Receive_data	Receive Data FIS from the Transport layer.	
1. Data FIS received.	→	DDMAO0: DMA_out

DDMAO3: Send_status	Request transmission of a Register FIS to host.	
1. Register FIS transmitted.	→	DI0: Device_idle

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 DDMAO0:1: When the device is ready to receive a Data FIS, the device shall transition to the DDMAO1: Send_DMA_activate state.

Transition DDMAO0:2: 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 DDMAO1:1: 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 DDMAO2:1: 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 Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one.

Transition DDMAO3:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

11.11 PACKET protocol

States marked with an * are only utilized when queuing is implemented

DPKT0: PACKET	Request transmission of a PIO Setup FIS.		
	1. FIS transmission complete.	→	DPKT1: Receive_command

DPKT1: Receive_command	Receive Data FIS containing command packet.		
	1. FIS reception complete.	→	DPKT2: Check_command

DPKT2: Check_command	Determine the protocol required for the received command.	
1. Non-data command.	→	DPKT3: PACKET_non-data
2. PIO data-in command.	→	DPKT4: PACKET_PIO_in
3. PIO data-out command.	→	DPKT6: PACKET_PIO_out
4. DMA data-in command	→	DPKT9: PACKET_DMA_in
5. DMA data-out command	→	DPKT11: PACKET_DMA_out

DPKT3: PACKET_non-data	Execute Non-data command.	
1. Command execution complete.	→	DPKT14: Send_status

DPKT4: PACKET_PIO_in	Prepare a DRQ data block for transfer to the host.	
1. DRQ data block ready to transfer.	→	DPKT4a: PIO_in_setup
2. Transfer complete or command aborted due to error.	→	DPKT14: Send_status
3. * DRQ block is not ready for immediate transfer	→	DPKT15: Release

DPKT4a: PIO_in_setup	Request transmission of a PIO Setup FIS to host.	
1. PIO Setup FIS transmitted.	→	DPKT5: Send_PIO_data

DPKT5: Send_PIO_data	Request transmission of a Data FIS to host.	
1. Data FIS transmitted.	→	DPKT4: PACKET_PIO_in

DPKT6: PACKET_PIO_out	Prepare to receive DRQ data block from the host.	
1. Ready to receive DRQ data block transfer.	→	DPKT7: PIO_out_setup
2. All DRQ data blocks received or command aborted due to error.	→	DPKT14: Send_status
3. * Not ready to accept DRQ block immediately.	→	DPKT15: Release

DPKT7: PIO_out_setup	Request transmission of a PIO Setup FIS to host.	
1. PIO Setup FIS transmitted.	→	DPKT8: Receive_PIO_data

DPKT8: Receive_PIO_data	Receive Data FIS from the Transport layer.	
1. Data FIS received.	→	DPKT6: PACKET_PIO_out

DPKT9: PACKET_DMA_in	Prepare data for the transfer of a Data FIS.		
	1. Data for Data FIS ready to transfer.	→	DPKT10: Send_DMA_data
	2. Command completed or aborted due to error.	→	DPKT14: Send_status
	3. * Data is not ready for immediate transfer.	→	DPKT15: Release

DPKT10: Send_DMA_data	Request transmission of a Data FIS to host.		
	1. Data FIS transmitted. No more data transfer required for this command, or 2048 Dwords transmitted.	→	DPKT9: PACKET_DMA_IN

DPKT11: PACKET_DMA_out	Prepare to receive a Data FIS from the host.		
	1. Ready to receive Data FIS.	→	DPKT12: Send_DMA_activate
	2. All data requested for this command received or command aborted due to error.	→	DPKT14: Send_status
	3. * Not ready for immediate transfer.	→	DPKT15: Release

DPKT12: Send_DMA_activate	Request transmission of a DMA Activate FIS to host.		
	1. DMA Activate FIS transmitted.	→	DPKT13: Receive_DMA_data

DPKT13: Receive_DMA_data	Receive Data FIS from the Transport layer.		
	1. Data FIS received.	→	DPKT11: PACKET_DMA_out

DPKT14: Send_status	Request transmission of a Register FIS.		
	1. FIS transmission complete.	→	DI0: Device_idle

* DPKT15: Release	Request transmission of a Register FIS.		
	1. FIS transmission complete.	→	DI0: Device_idle

DPKT0: 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 bit cleared to zero and DRQ bit set to one. The Interrupt bit shall be cleared to zero. The ending status shall have BSY bit set to one and DRQ bit cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DPKT0:1: When the PIO Setup FIS has been transferred, the device shall transition to the DPKT1: Receive_command state.

DPKT1: 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 DPKT1:1: When the Data FIS has been received, the device shall transition to the DPKT2: Check_command state.

DPKT2: 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 DPKT2:1: When the command is a non-data transfer command, the device shall transition to the DPKT3: PACKET_non-data state.

Transition DPKT2:2: When the command is a PIO data-in transfer command, the device shall transition to the DPKT4: PACKET_PIO_in state.

Transition DPKT2:3: When the command is a PIO data-out transfer command, the device shall transition to the DPKT6: PACKET_PIO_out state.

Transition DPKT2:4: When the command is a DMA data-in transfer command, the device shall transition to the DP9: PACKET_DMA_in state.

Transition DPKT2:5: When the command is a DMA data-out transfer command, the device shall transition to the DPKT11: PACKET_DMA_out state.

DPKT3: 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 DPKT3:1: When command execution completes, the device shall transition to the DPKT14: Send_status state.

DPKT4: 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 DPKT4:1: When the device has a DRQ data block ready to transfer, the device shall transition to the DPKT4a: PIO_in_setup.

Transition DPKT4:2: 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 DPKT14: Send_status state.

*** Transition DPKT4:3:** 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 DPKT15: Release state.

DPKT4a: 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 bit cleared to zero and DRQ bit set to one. The Interrupt bit shall

be set to one. The ending status shall have BSY bit set to one and DRQ bit cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DPKT4a:1: When the PIO Setup FIS has been transferred, the device shall transition to the DPKT5:Send_PIO_data state.

DPKT5: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 DPKT5:1: When the Data FIS has been transferred, the device shall transition to the DPKT4: PACKET_PIO_in state.

DPKT6: 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 DPKT6:1: When the device is ready to receive a DRQ data block transfer, the device shall transition to the DPKT7: PIO_out_setup state.

Transition DPKT6:2: 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 DPKT14: Send_status state.

*** Transition DPKT6:3:** When the device supports overlap and queuing and is not in a state in which it can accept a DRQ data block immediately, the device shall transition to the DPKT15: Release state.

DPKT7: 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 bit cleared to zero and DRQ bit set to one. The Interrupt bit shall be set to one. The ending status shall have BSY bit set to one and DRQ bit cleared to zero. The byte count for the DRQ data block shall be indicated.

Transition DPKT7:1: When the PIO Setup FIS has been transferred, the device shall transition to the DPKT8: Receive_PIO_data state.

DPKT8: 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 DPKT8:1: When the Data FIS has been received, the device shall transition to the DPKT6: PACKET_PIO_out state.

DPKT9: 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 DPKT9:1: When the device has the data ready to transfer a Data FIS, the device shall transition to the DPKT10: Send_DMA_data state.

Transition DPKT9:2: 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 DPKT14: Send_status state.

*** Transition DPKT9:3:** When the device supports overlap and queuing and does not have data ready to transfer immediately, the device shall transition to the DPKT15: Release state.

DPKT10: 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 DPKT10:1: When the Data FIS has been transferred, the device shall transition to the DPKT9: PACKET_DMA_in state. The device command layer shall request a data FIS size of no more than 2048 Dwords.

DPKT11: 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 DPKT11:1: When the device is ready to receive a Data FIS, the device shall transition to the DPKT12: Send_DMA_activate state.

Transition DPKT11:2: 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 DPKT14: Send_status state.

*** Transition DPKT11:3:** When the device supports overlap and queuing and is not in a state which it can accept a Data FIS immediately, the device shall transition to the DPKT15: Release state.

DPKT12: 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 DPKT12:1: When the DMA Activate FIS has been transferred, the device shall transition to the DPKT13: Receive_DMA_data state.

DPKT13: 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 DPKT13:1: When the Data FIS has been received, the device shall transition to the DPKT11: PACKET_DMA_out state.

DPKT14: Send_status: This state is entered when the device has received all the data requested by this command or the device has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one.

Transition DPKT14:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

*** DPKT15: Release:** This state is entered when the device is not able to do a data transfer immediately.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in the ATA8-ACS standard, with the REL bit set to one, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the Interrupt bit set to one.

Transition DPKT15:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

11.12 READ DMA QUEUED command protocol

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.

DDMAQI0: DMA_queued_in	Determine whether to transfer or release.		
	1. Data for Data FIS ready to transfer.	→	DDMAQI1: Send_data
	2. Command aborted due to error.	→	DDMAQI3: Send_status
	3. Bus release	→	DDMAQI4: Release

DDMAQI1: Send_data	Request transmission of a Data FIS to host.		
	1. Data FIS transmitted. No more data transfer required for this command, or 2048 Dwords transmitted.	→	DDMAQI2: Prepare_data

DDMAQI2: Prepare_data	Prepare data for the next Data FIS.		
	1. Data ready.	→	DDMAQI1: Send _data
	2. Command complete or aborted due to error.	→	DDMAQI3: Send_status

DDMAQI3: Send_status	Request transmission of a Register FIS to host.		
	1. Register FIS transmitted.	→	DI0: Device_idle

DDMAQI4: Release	Request transmission of a Register FIS to host.		
1. Register FIS transmitted.		→	DI0: Device_idle

DDMAQI0: DMA_queued_in State: This state is entered when the device receives a READ DMA QUEUED command.

When in this state, device shall determine if the requested data is ready to transfer to the host.

Transition DDMAQI0:1: When the device has the requested data ready to transfer a Data FIS immediately, the device shall transition to the DDMAQI1: Send_data state.

Transition DDMAQI0:2: 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 DDMAQI3: Send_status state.

Transition DDMAQI0:3: When the device does not have the requested data ready to transfer a Data FIS immediately, the device shall transition to the DDMAQI4: Release state.

DDMAQI1: Send_data: This state is entered when the device has the data ready to transfer a Data FIS to the host.

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

Transition DDMAQI1:1: When the Data FIS has been transferred, the device shall transition to the DDMAQI2: Prepare_data state. The device command layer shall request a Data FIS size of no more than 2048 Dwords .

DDMAQI2: Prepare_data: This state is entered when the device has completed the transfer a Data FIS to the host.

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

Transition DDMAQI2:1: When data is ready for the Data FIS, the device shall transition to the DDMAQI1: Send_data state.

Transition DDMAQI2:2: When all data requested for the command has been transmitted or an error has been encountered that causes the command to abort before completing the transfer of the requested data, the device shall transition to the DDMAQI3: Send_status state.

DDMAQI3: Send_status: This state is entered when the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one.

Transition DDMAQI3:1: When the FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

DDMAQI4: Release: This state is entered when the device does not have the requested data available for immediate transfer.

When in this state, the device shall request that the Transport layer transmit a Register FIS with the REL bit set to one, with register content as described in the command description in the ATA8-ACS standard, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the Interrupt bit set to one.

Transition DDMAQI4:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

11.13 WRITE DMA QUEUED command protocol

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.

DDMAQO0: DMA-queued_out	Determine whether to transfer or release.		
	1. Ready to accept Data FIS	→	DDMAQO1: Send_DMA_activate
	2. Command due to error.	→	DDMAQO4: Send_status
	3. Bus release	→	DDMAQO5: Release

DDMAQO1: Send_DMA_activate	Request transmission of a DMA Activate FIS to host.		
	1. DMA Activate FIS transmitted.	→	DDMAQO2: Receive_data

DDMAQO2: Receive_data	Receive Data FIS from the Transport layer.		
	1. Data FIS received.	→	DDMAQO3: Prepare_data_buffer

DDMAQO3: Prepare_data_buffer	Prepare to receive the next Data FIS.		
	1. Ready to receive.	→	DDMAQO1: Send DMA_activate
	2. Command complete or aborted due to error.	→	DDMAQO4: Send_status

DDMAQO4: Send_status	Request transmission of a Register FIS to host.		
	1. Register FIS transmitted.	→	DI0: Device_idle

DDMAQO5: Release	Request transmission of a Register FIS to host.		
	1. Register FIS transmitted.	→	DI0: Device_idle

DDMAQO0: DMA_queued_out State: 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: 1: When the device is ready to receive a Data FIS immediately, the device shall transition to the DDMAQO1: Send_DMA_activate state.

Transition DDMAQO0:2: 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 DDMAQO0:3: When the device is not ready to receive a Data FIS immediately, the device shall transition to the DDMAQO5: Release 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 DDMAQO1:1: 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 DDMAQO2:1: 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 DDMAQO3:1: When ready to receive the Data FIS, the device shall transition to the DDMAQO1: Send_DMA_activate state.

Transition DDMAQO3:2: 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 Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one.

Transition DDMAQO4:1: When the FIS has been transmitted, then the device shall transition to the DI0: 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 Register FIS with REL set to one, with register content as described in the command description in the ATA8-ACS standard, and, if the bus release interrupt has been enabled by a previous Set Features Command, with the Interrupt bit set to one.

Transition DDMAQO5:1: When the FIS has been transmitted, then the device shall transition to the DI0: Device_idle state.

11.14 FPDMA QUEUED command protocol

This class includes:

- READ FPDMA QUEUED
- WRITE FPDMA QUEUED
- NCQ QUEUE MANAGEMENT

DFPDMAQ1: AddCommandToQueue	Append command to internal device command queue and store TAG value.	
1. Device successfully en-queued the command	→	DFPDMAQ2: ClearInterfaceBsy
2. Command malformed	→	DFPDMAQ12: BrokenHost_ ClearBusy

DFPDMAQ2: ClearInterfaceBsy	Transmit Register FIS with BSY bit cleared to zero and DRQ bit cleared to zero and Interrupt bit cleared to zero to mark interface ready for the next command.	
1. Register FIS transmission complete	→	DI0: Device_idle

DFPDMAQ3: DataPhaseReadSetup	Transmit a DMA Setup FIS to the host with the DMA Buffer Identifier = TAG and D bit set to one (direction is device to host) and Interrupt bit cleared to zero	
1. First Party DMA Setup FIS transmission complete	→	DFPDMAQ8: DataXmitRead

DFPDMAQ4: DataPhasePreWriteSetup		
1. DMA Setup FIS Auto-Activate option supported and enabled	→	DFPDMAQ5: DataPhase_ WriteSetup
2. DMA Setup FIS Auto-Activate option not supported or not enabled	→	DFPDMAQ6: DataPhase_ OldWriteSetup

DFPDMAQ5: DataPhase_WriteSetup	Transmit a DMA Setup FIS to the host with the DMA Buffer Identifier = TAG and D bit cleared to zero (direction is host to device) and Auto-Activate bit set to one and Interrupt bit cleared to zero	
1. DMA Setup FIS transmission complete	→	DFPDMAQ9: DataXmitWrite

DFPDMAQ6: DataPhase_OldWriteSetup	Transmit a DMA Setup FIS to the host with the DMA Buffer Identifier = TAG and D bit cleared to zero (direction is host to device) and Auto-Activate bit cleared to zero and Interrupt bit I cleared to zero	
1. DMA Setup FIS transmission complete	→	DFPDMAQ7: DataPhase_ XmitActivate

DFPDMAQ7: DataPhase_XmitActivate	Transmit a DMA Activate FIS to the host	
1. DMA Activate FIS transmission complete	→	DFPDMAQ9: DataXmitWrite

DFPDMAQ8: DataXmitRead	Transmit Data FIS to the host	
1. Transfer count for previous DMA Setup FIS not exhausted and no error encountered	→	DFPDMAQ8: DataXmitRead
2. Transfer count for previous DMA Setup FIS exhausted and data transfer for this command not complete and no error encountered[1]	→	DI0: Device_idle
3. Finished with data transfer for this command and no error encountered	→	DI0: Device_idle
4. Unrecoverable error has occurred	→	DI0: Device_idle
NOTE: 1. This condition requires that non-zero buffer offsets be supported and enabled. The transition also applies if a device switches between multiple active commands and is performing partial data transfers for the multiple outstanding commands.		

DFPDMAQ9: DataXmitWrite	Receive Data FIS from host	
1. Transfer count for previous DMA Setup FIS not exhausted and no error encountered	→	DFPDMAQ7: DataPhase_ XmitActivate
2. Transfer count for previous DMA Setup FIS exhausted and data transfer for this command not complete and no error encountered[1]	→	DI0: Device_idle
3. Finished with data transfer for this command and no error encountered	→	DI0: Device_idle
4. Unrecoverable error has occurred	→	DI0: Device_idle
NOTE: 1. This condition requires that non-zero buffer offsets be supported and enabled. The transition also applies if a device switches between multiple active commands and is performing partial data transfers for the multiple outstanding commands.		

DFPDMAQ10: SendStatus	Transmit Set Device Bits FIS with ERR bit cleared to zero, Interrupt bit set to one, and bit ***n*** in ACT field set to one where ***n*** = TAG for each command TAG value that has completed since the last status return		
	1. Set Device Bits FIS transmission complete	→	DI0: Device_idle

DFPDMAQ11: ERROR	Halt command processing and transmit Set Device Bits FIS to host with ERR bit in Status field set to one, Interrupt bit set to one, ATA error code set in Error field, and bits in ACT field cleared to zero for any outstanding queued commands and bits set to one for any successfully completed queued commands for which completion notification not yet delivered.		
	1. Set Device Bits FIS transmission complete	→	DFPDMAQ13: WaitforClear

DFPDMAQ12: BrokenHost_ClearBusy	Halt command processing and transmit Register FIS to host with ERR bit in Status field set to one, Interrupt bit set to one, BSY bit cleared to zero, DRQ bit cleared to zero, and Error field = 04h. If error condition was due to reception of an Unload request and the device supports Unload when NCQ commands are outstanding, the device shall unload/park the heads.		
	1. Register FIS transmission complete	→	DFPDMAQ13: WaitforClear

DFPDMAQ13: WaitforClear	Wait for host to either issue a command to read the Queued Error Log or issue SRST		
	1. READ LOG EXT command with Queued Error Log received	→	DFPDMAQ14: SendQueue_ CleanACK
	2. READ LOG DMA EXT command with Queued Error Log received[1].	→	DFPDMAQ15: SendQueue_ CleanACKDMA
	3. SRST received	→	DSR0: Software_reset_ asserted
	4. Any other command received	→	DFPDMAQ12: BrokenHost_ ClearBusy
	NOTE: 1. See 13.7		

DFPDMAQ14: SendQueue_CleanACK	Discard all commands in the pending device queue. Transmit Set Device Bits FIS with ERR in Status field cleared to zero, Error field set to 00h, ACT field = FFFFFFFFh, and Interrupt bit cleared to zero.		
	1. Set Device Bits FIS transmission complete	→	DPIOI0: PIO_in

DFPDMAQ15: SendQueue_ CleanACKDMA	Discard all commands in the pending device queue. Transmit Set Device Bits FIS with ERR in Status field cleared to zero, Error field set to 00h, ACT field = FFFFFFFFh, and Interrupt bit cleared to zero.		
	1. Set Device Bits FIS transmission complete	→	DDMAI0: DMA_in

DFPDMAQ1: AddCommandToQueue: This state is entered when the device has checked the command and determined it to be a native queued type command, and Native Command Queuing is supported and enabled.

When in this state, the device shall check the TAG validity and verify that it is not already assigned to an outstanding command. If valid, the device shall append the command to its internal command queue and store the new TAG value.

Transition DFPDMAQ1:1: When the device determines the TAG is valid, and has added the command to its internal command queue, the device shall transition to the DFPDMAQ2: ClearInterfaceBusy state.

Transition DFPDMAQ1:2: When the device determines that the received command is malformed, an error has occurred and the device shall transition to the DFPDMAQ12: BrokenHost_ClearBusy state. A command may be considered malformed as a result of any of its parameters being invalid, including the use of a TAG value that corresponds to an existing TAG value for a pending command.

DFPDMAQ2: ClearInterfaceBusy: This state is entered when the device has appended the command to its internal queue and is ready transmit a Register FIS with BSY bit cleared to zero and DRQ bit cleared to zero to indicate that the interface is ready to receive the next command

Transition DFPDMAQ2:1: When the Register FIS has been transmitted, the device shall transition to the DI0: Device_idle state.

DFPDMAQ3: DataPhaseReadSetup: This state is entered when the device has determined that it is ready to transmit data for a previously queued READ FPDMA QUEUED command.

When in this state, the device shall transmit a DMA Setup FIS to the host with the DMA buffer identifier set to the queued TAG value and the Direction bit set to one (host memory write).

Transition DFPDMAQ3:1: When the device completes the transmission of the DMA Setup FIS, the device shall transition to the DFPDMAQ8: DataXmitRead state.

DFPDMAQ4: DataPhasePreWriteSetup: This state is entered when the device has determined that it is ready to receive data for a previously queued WRITE FPDMA QUEUED command.

When in this state, the device shall determine if the DMA Setup Auto-Activate option is supported and enabled, and then make the appropriate state transition.

Transition DFPDMAQ4:1: If the DMA Setup FIS Auto-Activate option is enabled, the device shall transition to the DFPDMAQ5: DataPhase_WriteSetup state.

Transition DFPDMAQ4:2: If the DMA Setup FIS Auto-Activate option is not supported or not enabled, the device shall transition to the DFPDMAQ6: DataPhase_OldWriteSetup state.

DFPDMAQ5: DataPhase_WriteSetup: This state is entered when the device is ready to Auto Activate and receive data for a previously queued WRITE FPDMA QUEUED command.

When in this state, the device transmits a DMA Setup FIS to the host with the DMA buffer identifier set to the queued TAG value and the Direction bit cleared to zero (host memory read), and Auto-Activate bit set to one.

Transition DFPDMAQ5:1: When the device completes the transmission of the DMA Setup FIS, the device shall transition to the DFPDMAQ9: DataXmitWrite state.

DFPDMAQ6: DataPhase_OldWriteSetup: This state is entered when the device is ready to receive data for a previously queued WRITE FPDMA QUEUED command, and the device does not support Auto-Activate, or it is not enabled.

When in this state, the device transmits a DMA setup FIS to the host with the DMA buffer identifier set to the queued TAG value and the Direction bit cleared to zero (host memory read), and Auto-Activate bit cleared to zero.

Transition DFPDMAQ6:1: When the device completes the transmission of the DMA Setup FIS, the device shall transition to the DFPDMAQ7: DataPhase_XmitActivate state.

DFPDMAQ7: DataPhaseXmit_Activate: This state is entered after the device has completed transmission of a DMA Setup FIS for a WRITE FPDMA QUEUED command or the device has finished receiving a Data FIS for a WRITE FPDMA QUEUED command, and the transfer count is not exhausted.

When in this state, the device transmits a DMA Activate FIS to the host indicating readiness to receive Data FISes from the host.

Transition DFPDMAQ7:1: When the device completes the transmission of the DMA Activate FIS, the device shall transition to the DFPDMAQ9: DataXmitWrite state.

DFPDMAQ8: DataXmitRead: This state is entered after the device has completed transmission of a DMA Setup FIS for a READ FPDMA QUEUED command.

When in this state, the device transmits a Data FIS to the host.

Transition DFPDMAQ8:1: If the transfer count for the previous DMA Setup FIS is not exhausted and no error is encountered, the device remains in the DFPDMAQ8: DataXmitRead state.

Transition DFPDMAQ8:2: If the transfer count for the previous DMA Setup FIS is exhausted, and the data transfer for this command is not complete, and no error is encountered, the device shall transition to the DI0: Device_idle state.

This condition requires that non-zero buffer offsets be supported and enabled. The transition also applies if a device switches between multiple active commands and is performing partial data transfers for the multiple outstanding commands

Transition DFPDMAQ8:3: If the device has completed the data transfer for this command, and no error is encountered, the device shall transition to the DI0: Device_idle state.

Transition DFPDMAQ8:4: If the device determines that an unrecoverable error has occurred, the device shall transition to the DI0: Device_idle state.

DFPDMAQ9: DataXmitWrite: This state is entered after the device has completed transmission of a DMA Setup FIS for a WRITE FPDMA QUEUED command.

When in this state, the device receives a Data FIS from the host.

Transition DFPDMAQ9:1: After the data FIS reception is complete, and if the transfer count for the previous DMA Setup FIS is not exhausted and no error is encountered, the device shall transition to the DFPDMAQ7: DataPhase_XmitActivate state.

Transition DFPDMAQ9:2: If the transfer count for the previous DMA Setup FIS is exhausted, and the data transfer for this command is not complete, and no error is encountered, the device shall transition to the DI0: Device_idle state.

This condition requires that non-zero buffer offsets be supported and enabled. The transition also applies if a device switches between multiple active commands and is performing partial data transfers for the multiple outstanding commands

Transition DFPDMAQ9:3: If the device has completed the data transfer for this command, and no error is encountered, the device shall transition to the DI0: Device_idle state.

Transition DFPDMAQ9:4: If the device determines that an unrecoverable error has occurred, the device shall transition to the DI0: Device_idle state.

DFPDMAQ10: SendStatus: This state is entered when the data transfer for this command, or aggregated commands, is completed and the device is ready to send status.

When in this state, the device transmits a Set Device Bits FIS to the host with ERR bit cleared to zero, Interrupt bit set to one, and bit ***n*** in ACT field set to one where ***n*** = TAG for each command TAG value that has completed since the last status return.

Transition DFPDMAQ10:1: When the device completes the transmission of the Set Device Bits FIS, the device shall transition to the DI0: Device_idle state.

DFPDMAQ11: ERROR: This state is entered when the device has encountered an unrecoverable error.

When in this state, the device halts command processing and transmits a Set Device Bits FIS to the host with ERR bit set to one, Interrupt bit set to one, ATA error code set in the Error field, and bits in ACT field cleared to zero for any outstanding queued commands (including the erring command) and bits set to one for any successfully completed queued command for which a completion notification has not yet been provided to the host.

Transition DFPDMAQ11:1: When the device completes the transmission of the Set Device Bits FIS, the device shall transition to the DFPDMAQ13: WaitforClear state.

DFPDMAQ12: BrokenHost_ClearBusy: This state is entered when the device has received a READ FPDMA QUEUED or WRITE FPDMA QUEUED command with a TAG that already exists in its command queue, or when the received command is a not a READ FPDMA QUEUED; and not a WRITE FPDMA QUEUED; and not a DEVICE RESET; and there are native queued command(s) outstanding.

When in this state, the device halts command processing and transmits a Register FIS to the host with ERR set to one in the Status field, Interrupt bit set to one, BSY bit cleared to zero, DRQ bit cleared to zero, and ATA error code set in the Error field. If error condition was due to reception of an Unload request and the device supports Unload when NCQ commands are outstanding, the device shall unload/park the heads.

Transition DFPDMAQ12:1: When the device completes the transmission of the Register FIS, the device shall transition to the DFPDMAQ13: WaitforClear state.

DFPDMAQ13: WaitforClear: This state is entered when the device has transmitted an error FIS to the host and is awaiting a command to read the Queued Error Log (see 13.7) or a soft reset. Any other commands return Register – Device to Host FIS with the ERR bit set to one in the Status field.

Transition DFPDMAQ13:1: If the device receives a READ LOG EXT command to read the Queued Error Log, the device shall transition to the DFPDMAQ14: SendQueue_CleanACK state.

Transition DFPDMAQ13:2: If the device receives a READ LOG DMA EXT command to read the Queued Error Log, and IDENTIFY DEVICE word 76 bit 15 is set to one, the device shall transition to the DFPDMAQ15: SendQueue_CleanACKDMA state.

Transition DFPDMAQ13:3: If the device receives a SRST, the device shall transition to the DSR0: Software_reset_asserted state.

Transition DFPDMAQ13:4: If the device receives any other command, the device shall transition to the DFPDMAQ12: BrokenHost_ClearBusy state.

DFPDMAQ14: SendQueue_CleanACK: This state is entered when the host has responded to an error FIS to a command to read the Queued Error Log.

The device shall discard all commands in the pending queue and transmit a Set Device Bits FIS with ERR bit in the Status field cleared to zero, Error field set to 00h, ACT field = FFFFFFFFh, and Interrupt bit cleared to zero.

Transition DFPDMAQ14:1: when the Set Device Bits FIS transmission is complete, the device shall transition to the DPIOI0: PIO_in state.

DFPDMAQ15: SendQueue_CleanACKDMA: This state is entered when the host has responded to an error FIS to a command to read the Queued Error Log .

The device shall discard all commands in the pending queue and transmit a Set Device Bits FIS with ERR bit in the Status field cleared to zero, Error field set to 00h, ACT field = FFFFFFFFh, and Interrupt bit cleared to zero.

Transition DFPDMAQ14:1: when the Set Device Bits FIS transmission is complete, the device shall transition to the DDMAI0: DMA_in state.

12 Host Command Layer protocol

12.1 FPDMA QUEUED command protocol

This high-level state machine describes the behavior of the host for the Native Command Queuing command protocol. The host behavior described by the state machine may be provided by host software and/or host hardware and the intent of the state machines is not to indicate any particular implementation.

This class includes:

- READ FPDMA QUEUED
- WRITE FPDMA QUEUED
- NCQ QUEUE MANAGEMENT

HFPI0: Idle		
1. Free TAG location and command waiting to have TAG assigned	→	HFPDMAQ2: PresetACTBit
2. Command with assigned TAG awaiting issue and BSY bit cleared to zero and not First-party DMA Data Phase	→	HFPDMAQ3: IssueCommand
3. Interrupt received from device.	→	HFPDMAQ4: DeviceINT
4. Default	→	HFPI0: Idle
NOTE: 1. If more than one condition is true, the host may apply a vendor specific priority		

HFPDMAQ1: AddCommandToQueue	Append command to internal host command queue	
1. Unconditional	→	HFPI0: Idle

HFPDMAQ2: PresetACTBit	Assign free TAG value to command. Write SActive register with value that has bit set in bit position corresponding to assigned TAG value	
1. TAG value assigned to command and SActive register written with new TAG bitmask	→	HFPI0: Idle

HFPDMAQ3: IssueCommand	If not First-party DMA Data Phase, transmit Register FIS to device with new command and assigned TAG value	
1. Register FIS transmission complete (command issued)	→	HFPI0: Idle
2. Register FIS transmission deferred (command not issued)	→	HFPI0: Idle

HFPDMAQ4: DeviceINT	Read Status register to clear pending interrupt flag and save value as SavedStatus	
1. Unconditional	→	HFPDMAQ5: CompleteRequests1

HFPDMAQ5: CompleteRequests1	Compare SActive register with stored SActive register from last interrupt to identify completed commands	
1. SActive comparison indicates one or more commands are completed	→	HFPDMAQ6: CompleteRequests2
2. SActive comparison indicates no commands are completed	→	HFPDMAQ7: CompleteRequests3

HFPDMAQ6: CompleteRequests2	Retire host requests associated with TAG values corresponding to newly cleared bits in the SActive register and update stored SActive with new value	
1. Unconditional	→	HFPDMAQ7: CompleteRequests3

HFPDMAQ7: CompleteRequests3	Test ERR bit in SavedStatus value	
1. ERR bit cleared to zero	→	HFPI0: Idle
2. ERR set to one	→	HFPDMAQ8: ResetQueue

HFPDMAQ8: ResetQueue	Issue a command to read the Queued Error Log to device	
1. READ LOG EXT command was accepted	→	HFPDMAQ9: CleanupACK
2. READ LOG DMA EXT command was accepted	→	HFPDMAQ12: RetrieveRequest_ SenseDMA
3. Command was not accepted	→	HFPDMAQ8: ResetQueue

HFPDMAQ9: CleanupACK	Wait for DRQ bit set to one and BSY bit cleared to zero [1]	
1. DRQ bit set to one and BSY bit cleared to zero	→	HFPDMAQ10: RetrieveRequest_ Sense
2. DRQ cleared to zero or BSY set to one	→	HFPDMAQ9: CleanupACK
NOTE: 1. The host may wait for this condition using any means including awaiting an interrupt and checking the DRQ bit and BSY bit status, spinning, or periodic timer.		

HFPDMAQ10: RetrieveRequest_Sense	Receive PIO Data FIS with Queued Error Log contents	
1. PIO Data FIS reception complete	→	HFPDMAQ11: ErrorFlush

HFPDMAQ11: ErrorFlush	Retire failed queued command with status set to error condition reported by device. Flush all allocated native queued command tags. Flush pending native queued commands from host command queue with system-specific error condition or re-issue pending queued commands.		
	1. Unconditional	→	HFPI0: Idle

HFPDMAQ12: RetrieveRequest_ SenseDMA	Receive Data FIS with Queued Error Log contents		
	1. Data FIS reception complete	→	HFPDMAQ13: SendStatus

HFPDMAQ11: ErrorFlush	Request transmission of a Register FIS to the host		
	1. Register FIS transmitted	→	HFPDMAQ11: ErrorFlush

HFPI0: Idle: When in this state, if queuing is supported and enabled, the Command layer is awaiting a READ FPDMA QUEUED, WRITE FPDMA QUEUED or NCQ QUEUE MANAGEMENT command from the higher level protocol, or awaiting an interrupt from the Device indicating completion of previously queued commands, or waiting for a TAG location to become available for a command waiting in the command queue.

Transition HFPI0:1: If a READ FPDMA QUEUED, WRITE FPDMA QUEUED or NCQ QUEUE MANAGEMENT command is pending that has not had a TAG value assigned to it and there is a free TAG location available for assignment then a transition shall be made to the HFPDMAQ2: PresetACTBit state.

Transition HFPI0:2: If a command with assigned TAG value is awaiting issue to the device and BSY bit cleared to zero and the interface is not in the First-party DMA Data Phase, then a transition shall be made to the HFPDMAQ3: IssueCommand state.

Transition HFPI0:3: If an interrupt is received from the device, indicating status is available for a previously queued command, it shall transition to the HFPDMAQ4: DeviceINT state.

Transition HFPI0:4: If the queuing is supported and enabled, and the Command layer is awaiting a free TAG, a new command, or an interrupt for a previously queued command, it shall transition to the HFPI0: Idle state.

HFPDMAQ1: AddCommandToQueue: The Command layer enters this state when it has received a READ FPDMA QUEUED, WRITE FPDMA QUEUED or NCQ QUEUE MANAGEMENT command from the higher level protocol, and adds it to the internal host command queue.

Transition HFPDMAQ1:1: After the Command Layer has added the command to the internal host command queue, it shall transition to the HFPI0: Idle state.

HFPDMAQ2: PresetACTBit: When in this state, the Command layer assigns a free TAG value to the previously queued command and writes the SActive register with the bit position corresponding to the assigned TAG value.

Transition HFPDMAQ2:1: After the Command layer has assigned a TAG value and written the corresponding bit to the SActive register, it shall transition to the HFPI0: Idle state.

HFPDMAQ3: IssueCommand: When in this state, the Command layer attempts to issue a command with preassigned TAG to the device by transmitting a Register FIS to the device with the new command and assigned TAG value if the interface state permits it.

Transition HFPDMAQ3:1: After the Command layer has transmitted the Register FIS, it shall mark the corresponding command as issued and transition to the HFPI0: Idle state.

Transition HFPDMAQ3:2: After the Command layer has deferred transmission of the Register FIS due to the interface state not permitting it to be delivered, it shall transition to the HFPI0: Idle state. The corresponding command is still considered as not having been issued.

HFPDMAQ4: DeviceINT: When in this state, the Command layer reads the Device Status Register to reset the pending interrupt flag and save the value as SavedStatus.

Transition HFPDMAQ4:1: After the Command layer has read the Device Status Register, it shall transition to the HFPDMAQ5: CompleteRequests1 state.

HFPDMAQ5: CompleteRequests1: When in this state, the Command layer compares the SActive Register with the SavedStatus SActive Register value that resulted from the last interrupt to identify completed commands.

Transition HFPDMAQ5:1: If the SActive comparison indicates one or more commands have completed, it shall transition to the HFPDMAQ6: CompleteRequests2 state.

Transition HFPDMAQ5:2: If the SActive comparison indicates no commands have completed, it shall transition to the HFPDMAQ7: CompleteRequests3 state.

HFPDMAQ6: CompleteRequests2: When in this state, the Command layer retires commands in its internal host command queue that are associated with TAG values corresponding to newly cleared bits in the SActive Register and updates the stored SActive register with the new value.

Transition HFPDMAQ6:1: After updating the stored SActive value, it shall transition to the HFPDMAQ7: CompleteRequests3 state.

HFPDMAQ7: CompleteRequests3: When in this state, the Application layer tests the ERR bit in the SavedStatus value to determine whether the queue should be maintained or reset.

Transition HFPDMAQ7:1: If the SavedStatus value ERR bit cleared to zero, the queue is maintained and it shall transition to the HFPI0: Idle state.

Transition HFPDMAQ7:2: If the SavedStatus value ERR bit set to one, an error has been reported by the Device and the Command layer shall transition to the HFPDMAQ8: ResetQueue state.

HFPDMAQ8: ResetQueue: When in this state, the Application layer issues a command to read the Queued Error Log (see 13.7) to the device.

Transition HFPDMAQ8:1: After a READ LOG EXT command has been accepted, it shall transition to the HFPDMAQ9: CleanupACK state.

Transition HFPDMAQ8:2: After a READ LOG DMA EXT command has been accepted, it shall transition to the HFPDMAQ12: RetrieveRequest_SenseDMA .

Transition HFPDMAQ8:3: If the command was not accepted, the host shall transition to the HPFDMAQ8: ResetQueue state.

HFPDMAQ9: CleanupACK: When in this state, the Command layer tests the Device for DRQ bit set to one and BSY bit cleared to zero in preparation for a PIO data FIS transfer.

Transition HFPDMAQ9:1: If DRQ bit is set to one and BSY bit is cleared to zero, it shall transition to the HFPDMAQ10: ReceiveRequestSense state.

Transition HFPDMAQ9:2: If DRQ bit is cleared to zero or BSY bit is set to one, it shall transition to the HFPDMAQ9: CleanupACK state.

HFPDMAQ10: RetrieveRequest_Sense: When in this state, the Command layer completes the PIO Data FIS that retrieves the Queued Error Log contents.

Transition HFPDMAQ10:1: After the completion of the PIO Data FIS, it shall transition to the HFPDMAQ11: ErrorFlush state.

HFPDMAQ11: ErrorFlush: When in this state, the Command layer retires the failed queued command with the error status set to the error condition reported by the device. It flushes all allocated native queued command tags, and flushes pending native commands from the host command queue with system-specific error condition or re-issue pending queued commands

Transition HFPDMAQ11:1: After the error flush actions have been completed, it shall transition to the HFPI0: Idle state.

HFPDMAQ12: RetrieveRequest_SenseDMA: This state is entered when the device has the data ready to transfer a data FIS to the host containing the Queued Error Log contents.

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 HFPDMAQ13:1: When the FIS has been transmitted, the device shall transition to the HFPDMAQ13: SendStatus state.

HFPDMAQ13: SendStatus: This state is entered when the device has transferred all of the data requested by the command or has encountered an error that causes the command to abort before completing the transfer of the requested data.

When in this state, the device shall request that the Transport layer transmit a Register FIS with register content as described in the command description in the ATA8-ACS standard and the Interrupt bit set to one.

Transition DDMAI2:1: When the FIS has been transmitted, the device shall transition to the HFPDMAQ11: ErrorFlush state.

13 Application Layer

13.1 Parallel ATA Emulation

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

The host adapter contains a set of registers that shadow the contents of the traditional device registers, referred to as the Shadow Register Block. All Serial ATA devices behave like Device 0 devices. Devices shall ignore the DEV bit in the Device field of received Register – Host to Device FISes, and it is the responsibility of the host adapter to gate transmission of Register – Host to Device 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 7Fh in its Shadow Status register and shall place the value FFh in all the other Shadow Command Block registers (FFFFh in the Data register). In this state the host bus adapter shall not accept writes to the Shadow Command Block Registers. When the Phy detects presence of an attached device, the host bus adapter shall place the value FFh or 80h in the Shadow Status register, and the host bus adapter shall now allow writes to the Shadow Command Block Registers. 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 the BSY bit to one in the Shadow Status register. Placing the value 80h in the Shadow Status register is recommended as it provides the highest level of BIOS compatibility. Note that when the BSY bit is set to one in the Shadow Status register all other bits in that register have indeterminate values. When the attached device establishes communication with the host bus adapter, it shall send a Register – Device to Host FIS to the host, resulting in the Shadow Command Block Registers being updated with values appropriate for the attached device.

BIOS and system software writers should be aware of the 10 ms latency the interface may incur in determining device presence, and either ensure the Shadow Status register is read no sooner than 10 ms after initialization or ensure the Shadow Status register is re-read 10 ms after having read a value of 7Fh in order to positively determine presence of a device.

The host adapter may present a Master-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 Master/Slave emulation to host software, that is, two devices on two separate Serial ATA ports are represented to host software as a Device 0 and a Device 1 accessed at the same set of host bus addresses.

13.1.1 Software Reset

According to the ATA/ATAPI-6 standard, issuing a software reset is performed by toggling the SRST bit in the Device Control register. The toggle period is stipulated as being no shorter than 5 us. As a result of the SRST bit changing in the Device Control register, Serial ATA host adapters shall issue at least two Register – Host to Device FISes to the device (one with the SRST bit set to one and a subsequent one with the SRST bit cleared to zero). See section 10.3.4 for a detailed definition of Register – Host to Device FIS. Although host software is required to toggle the SRST bit no faster than 5 us, devices may not rely on the inter-arrival time of received Register – Host to Device FISes also meeting this timing. Because of flow control, frame handshaking, and other protocol interlocks, devices may effectively receive the resulting Register – Host to Device FISes back-to-back.

Due to flow control, protocol interlocks, power management state, or other transmission latencies, the subsequent Register – Host to Device FIS transmission clearing the SRST bit to zero during a software reset may be triggered prior to the previous Register – Host to Device FIS transmission having been completed. Host adapters are required to allow host software to toggle the SRST bit with the minimum 5 us timing specified in the ATA/ATAPI-6 standard even if frame transmission latencies result in the first Register FIS transmission taking longer than 5 us. Host adapters are required to ensure transmission of the two resulting Register – Host to Device FISes to the device regardless of the transmission latency of each individual FIS.

13.1.2 Master-only emulation

NOTE: Unlike the remainder of this specification, this section is based on the ATA/ATAPI-5 standard.

A native Serial ATA host adapter behaves the same as if a legacy mode master only device were attached with no slave 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 ATA devices, therefore, need not be aware of master / slave issues and ignore legacy mode task file information that deal 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 ATA/ATAPI-5 standard. This includes not setting the BSY bit to one in the Shadow Status register when Device 1 is selected, as described in the ATA/ATAPI-5 standard. When Device 0 is selected, the host adapter shall execute the Serial ATA protocols for managing the Shadow Status register contents as defined in later sections.

When Device 0 is selected and the Command register is written in the Shadow Register Block, the host adapter sets the BSY bit to one in its shadow Status register. The host adapter then transmits a Register – Host to Device FIS to the device containing the new register contents. When the Device Control register is written in the Shadow Register Block with a change of state of the SRST bit, the host adapter sets the BSY bit to one in its shadow Status register and transmit a Register – Host to Device FIS to the device containing the new register contents. Transmission of register contents when the Device Control register is written with any value that is not a change of state of the SRST bit shall not set the BSY bit to one in the shadow Status register, and transmission of a frame to the device containing new register contents is optional. Similarly, the host adapter sets the BSY bit in its shadow Status register to one when a COMRESET is requested or the SRST bit is set to one in the Device Control register. The expected timing for setting BSY bit to one is thereby preserved.

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

Finally the host adapter cooperates with the device by providing an interrupt pending flag in the host adapter. This flag is set by the host adapter when the device sends a Register – Device to Host FIS with the Interrupt bit set to one. The host adapter asserts the interrupt to the host processor any time the interrupt pending flag is set, the DEV bit is cleared to zero in the shadow Device register, and the nIEN bit in the shadow Device Control register is cleared to zero. The host adapter clears the interrupt pending flag any time a COMRESET is requested, the SRST bit in the shadow Device Control register is set to one, the shadow Command register is written with DEV bit cleared to zero, or the Shadow Status register is read with DEV bit cleared to zero. This allows the emulation of the host interrupt and its proper timing.

13.1.3 Master/Slave emulation (optional)

NOTE: Unlike the remainder of this specification, this section is based on the ATA/ATAPI-5 standard.

All devices behave as if they are Device 0 devices. However, the host adapter may optionally implement emulation of the behavior of a Master/Slave configuration by pairing two Serial ATA ports, and managing their associated shadow registers accordingly, as though they were a Device 0 and Device 1 at the same set of host bus addresses.

A host adapter that emulates Master/Slave behavior shall manage the two sets of shadow registers (one set for each of the two devices) based on the value of the DEV bit in the shadow Device register. Based on the value of the DEV bit, the host adapter shall direct accesses to the Shadow Register Block Registers, and accesses to the Control Block Registers, to the appropriate set of shadow registers 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 Master/Slave 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 ATA/ATAPI-5 standard.

13.1.3.1 Software reset

Host adapters that emulate Master/Slave behavior shall emulate proper 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 Master/Slave emulation configuration. Device Control register writes, that have the SRST bit set to one, shall result in the associated Shadow Device Control register being written for each port to which a device is attached. The frame transmission protocol for each associated port executed, results in a Register – Host to Device FIS being transmitted to each attached device. Similarly, the subsequent write to the Shadow Device Control register that clears the SRST bit to zero shall result in a Register – Host to Device FIS being sent to each attached device. The host adapter shall then await a response from each attached device (or timeout), and shall merge the contents of the Error and Status registers for the attached devices, in accordance with the ATA/ATAPI-5 standard, to produce the Error and Status register values visible to host software.

13.1.3.2 EXECUTE DEVICE DIAGNOSTICS

Host adapters that emulate Master/Slave behavior shall emulate proper behavior for EXECUTE DEVICE DIAGNOSTICS. The host adapter shall detect the EXECUTE DEVICE DIAGNOSTIC command being written to the Command register. Detecting the EXECUTE DEVICE DIAGNOSTICS command shall result in the associated shadow register being written for each port to which a device is attached. The frame transmission protocol for each associated port executed, results in a Register – Host to Device FIS being transmitted to each attached device. The host adapter shall then await response from each attached device (or timeout), and shall merge the contents of the Error and Status registers for the attached devices, in accordance with the ATA/ATAPI-5 standard to produce the Error and Status register values visible to host software.

13.1.3.3 Restrictions and limitations

Superset capabilities that are unique to Serial ATA and not supported by parallel ATA may not be supported in Master/Slave emulation mode. Such capabilities include but are not limited to support for First-party DMA, hot plug/unplug, interface power management, and superset Status

and Control registers. Master/Slave emulation is recommended only in configurations where software written for parallel ATA is used and the number of attached devices exceeds the number of Shadow Command Block register interfaces that software supports.

13.1.3.4 Shadow Command Block Register Access Restrictions

Host software should take measures to ensure that the access restrictions for the Command Block registers (see ATA/ATAPI-5) are observed when accessing the Shadow Command Block registers in order to avoid indeterminate behavior. Some of these measures include:

a) Prior to writing the Shadow Command Block registers to issue a new command, host software should check the Shadow Status register to verify that both BSY and DRQ bits are cleared to zero.

b) If DRQ bit is set to one when software expects it to be cleared to zero (e.g. when ERR bit is set to one), host software should perform an error recovery action (e.g., issue a software reset to the device) prior to writing the Shadow Command register to issue a new command.

   Note: Issuing a software reset by setting the SRST bit to one in the Device Control register to one while BSY bit and/or DRQ bit are set to one is legal behavior as described in the Serial ATA state machines, and completion of a software reset ensures that the device and interface are in a known state.

The ATA/ATAPI-5 standard defines restrictions for writing to the Command Block registers.

It is prohibited to write the Shadow Command register when BSY bit or DRQ bit is set to one except for the DEVICE RESET command. The Serial ATA access restrictions differ from parallel ATA where similar access to the Command register is indeterminate. However, the resultant indeterminate behavior may, in some Serial ATA implementations, not be as benign as in some parallel ATA implementations. This is because in Serial ATA the Shadow Command Block registers are cooperatively managed between the host and the device, and the defined coordination is based on the values of BSY and DRQ bits.

An example situation where parallel ATA and Serial ATA behavior may differ is when the device sets both DRQ and ERR bits to one in the (Shadow) Status register in a PIO read operation. In parallel ATA, many device implementations allow software to issue a new command without transferring the data from the device. In Serial ATA, issuing a new command without transferring the data may lead to a hang condition. In Serial ATA, when DRQ bit is set to one there is an associated Data FIS that is being transferred on the Serial ATA interface. Until the Data FIS transfer is completed by the host reading the associated data from the Shadow Data register, the Data FIS remains on the interface in a $HOLD_P$/ $HOLDA_P$ flow-controlled condition. While the Data FIS remains on the interface, a new command cannot be issued to the device.

13.1.3.5 Parallel ATA interoperability state diagrams

This state diagram defines the protocol of the host adapter to emulate Master only parallel 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.

HA0: HA_SEL/NOINTRQ	The interrupt signal is not asserted to the host.		
1.	COMRESET requested by the host.	→	HA_SEL/NOINTRQ
2.	Device Control register written by the host with SRST bit cleared to zero and (nIEN bit set to one or IPF bit set to one).	→	HA_SEL/NOINTRQ
3.	Device Control register written by the host with SRST bit cleared to zero and (nIEN bit cleared to zero and IPF bit set to one).	→	HA_SEL/INTRQ
4.	Device Control register written by the host with SRST bit set to one.	→	HA_SEL/NOINTRQ
5.	Command register written by the host.	→	HA_SEL/NOINTRQ
6.	Device register written by the host with DEV bit set to one.	→	HA_NOTSEL
7.	Any other register read or write by the host	→	HA_SEL/NOINTRQ
8.	Transport layer indicates new register content from the device with the IPF bit cleared to zero.	→	HA_SEL/NOINTRQ
9.	Transport layer indicates new register content from the device with the IPF bit set to one and nIEN bit to one.	→	HA_SEL/NOINTRQ
10.	Transport layer indicates new register content from the device with the IPF bit set to one and nIEN bit cleared to zero.	→	HA_SEL/INTRQ

HA1: HA_SEL/INTRQ	The interrupt signal is asserted to the host.		
1.	COMRESET requested by the host.	→	HA_SEL/NOINTRQ
2.	Device Control register written by the host with SRST bit cleared to zero and (nIEN bit set to one or IPF bit cleared to zero).	→	HA_SEL/NOINTRQ
3.	Device Control register written by the host with SRST bit cleared to zero and (nIEN bit cleared to zero and IPF bit set to one).	→	HA_SEL/INTRQ
4.	Device Control register written by the host with SRST bit set to one.	→	HA_SEL/NOINTRQ
5.	Command register written by the host.	→	HA_SEL/NOINTRQ
6.	Device register written by the host with DEV bit set to one.	→	HA_NOTSEL
7.	Status register read by the host	→	HA_SEL/NOINTRQ
8.	Any other register write by the host	→	HA_SEL/INTRQ
9.	Any other register read by the host	→	HA_SEL/INTRQ
10.	Transport layer indicates new register content from the device.	→	HA_SEL/INTRQ

HA2: HA_NOTSEL	The interrupt signal is not asserted to the host.		
	1. COMRESET requested by the host.	→	HA_SEL/NOINTRQ
	2. Device Control register written by the host with SRST bit cleared to zero.	→	HA_NOTSEL
	3. Device Control register written by the host with SRST bit set to one.	→	HA_SEL/NOINTRQ
	4. Command register written by the host with command other than EXECUTE DEVICE DIAGNOSTIC.	→	HA_NOTSEL
	5. Command register written by the host with command EXECUTE DEVICE DIAGNOSTIC.	→	HA_SEL/NOINTRQ
	6. Device register written by the host with DEV bit cleared to zero and (nIEN bit set to one or IPF bit cleared to zero).	→	HA_SEL/NOINTRQ
	7. Device register written by the host with DEV bit cleared to zero and (nIEN bit cleared to zero and IPF bit set to one).	→	HA_SEL/INTRQ
	8. Any other register write by the host	→	HA_NOTSEL
	9. Read of Status or Alternate Status register by the host.	→	HA_NOTSEL
	10. Read of any other register by the host.	→	HA_NOTSEL
	11. Transport layer indicates new register content from the device.	→	HA_NOTSEL

HA0: HA_SEL/NOINTRQ state: This state is entered when Device 0 is selected and either IPF bit is cleared to zero or nIEN bit is set to one.

When in this state, the interrupt signal to the host shall not be asserted.

Transition HA0:1: When a COMRESET is requested by the host, the host adapter shall set BSY bit to one in the shadow Status register, clear IPF bit to zero, notify the Link layer to have a bus COMRESET asserted, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:2: When the Device Control register is written by the host with SRST bit cleared to zero and either nIEN bit set to one or IPF bit is cleared to zero, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C bit cleared to zero with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:3: When the Device Control register is written by the host with SRST bit cleared to zero, nIEN bit cleared to zero, and IPF bit is set to one, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C bit cleared to zero with the current content of the shadow registers, and make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA0:4: When the Device Control register is written by the host with SRST bit set to one, the host adapter shall set BSY bit to one in the Shadow Status register, clear IPF bit to zero, notify the Transport layer to send a Register – Host to Device FIS with C bit cleared to zero with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:5: When the Command register is written by the host, the host adapter shall set BSY bit to one in the Shadow Status register, clear IPF bit to zero, notify the Transport layer to send a Register – Host to Device FIS with C bit set to one with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:6: When the Device register is written by the host with DEV bit set to one, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA0:7: When any register is read or written by the host other than those described above, the host adapter shall make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:8: When the Transport layer indicates new register content from the device with IPF bit cleared to zero, the host adapter shall place the new register content into the shadow registers and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:9: When the Transport layer indicates new register content from the device with IPF bit set to one and nIEN bit set to one, the host adapter shall set IPF bit to one, place the new register content into the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA0:10: When the Transport layer indicates new register content from the device with IPF bit set to one and nIEN bit is cleared to zero, the host adapter shall set IPF bit to one, place the new register content into the shadow registers, 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 bit is cleared to zero, and IPF bit is set to one.

When in this state, the interrupt signal to the host shall be asserted.

Transition HA1:1: When a COMRESET is requested by the host, the host adapter shall set BSY bit to one in the Shadow Status register, clear IPF bit to zero, notify the Link layer to have a bus COMRESET asserted, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:2: When the Device Control register is written by the host with SRST bit cleared to zero, with nIEN bit set to one or IPF bit cleared to zero, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C bit cleared to zero with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:3: When the Device Control register is written by the host with SRST bit cleared to zero with nIEN bit cleared to zero and IPF bit set to one, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C bit cleared to zero with the current content of the shadow registers, and make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA1:4: When the Device Control register is written by the host with SRST bit set to one, the host adapter shall set BSY bit to one in the Shadow Status register, clear IPF bit to zero, notify the Transport layer to send a Register – Host to Device FIS with C bit cleared to zero with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:5: When the Command register is written by the host, the host adapter shall set BSY bit to one in the Shadow Status register, clear IPF bit to zero, notify the Transport layer to send a Register – Host to Device FIS with C bit set to one with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:6: When the Device register is written by the host with DEV bit set to one, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA1:7: When the Status register is read by the host, the host adapter shall clear IPF bit to zero and make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA1:8: 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 HA1:9: 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 HA1:10: When the Transport layer indicates new register content from the device, the host adapter shall place the new register content into the shadow registers 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 HA2:1: When a COMRESET is requested by the host, the host adapter shall set BSY bit to one in the Shadow Status register, clear DEV bit to zero in the Shadow Device register, clear IPF bit to zero, notify the Link layer to have a bus COMRESET asserted, and make a transition to the HA0: SEL/NOINTRQ state.

Transition HA2:2: When the Device Control register is written by the host with SRST bit cleared to zero, the host adapter shall notify the Transport layer to send a Register – Host to Device FIS with C bit cleared to zero with the current content of the shadow registers, and make a transition to the HA2: HA_NOTSEL state.

Transition HA2:3: When the Device Control register is written by the host with SRST bit set to one, the host adapter shall set BSY bit to one in the Shadow Status register, clear DEV bit to zero in the Shadow Device register, clear IPF bit to zero, notify the Transport layer to send a Register – Host to Device FIS with C bit cleared to zero with the current content of the shadow registers, and make a transition to the HA0: HA_ SEL/NOINTRQ state.

Transition HA2:4: 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 bit in the Shadow Status register and shall make a transition to the HA2: HA_NOTSEL state.

Transition HA2:5: When the Command register is written by the host with the EXECUTE DEVICE DIAGNOSTIC command code, the host adapter shall set BSY bit to one in the Shadow Status register, clear DEV bit to zero in the Device register, clear IPF bit to zero, notify the Transport layer to send a Register – Host to Device FIS with C bit set to one with the current content of the shadow registers, and make a transition to the HA0: HA_SEL/INTRQ state.

Transition HA2:6: When the Device register is written by the host with DEV bit cleared to zero and either nIEN bit set to one or IPF bit cleared to zero, the host adapter shall make a transition to the HA0: HA_SEL/NOINTRQ state.

Transition HA2:7: When the Device register is written by the host with DEV bit cleared to zero, nIEN bit cleared to zero, and IPF bit set to one, the host adapter shall make a transition to the HA1: HA_SEL/INTRQ state.

Transition HA2:8: When any register is written by the host other than those described above, the host adapter shall make a transition to the HA2: HA_NOTSEL state.

Transition HA2:9: 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 HA2:10: When any register is read by the host other than that described above, the host adapter shall return the current shadow register content to the host and make a transition to the HA2: HA_NOTSEL state.

Transition HA2:11: When the Transport layer indicates new register content from the device, the host adapter shall place the new register content into the shadow registers and make a transition to the HA2: HA_NOTSEL state.

13.2 IDENTIFY (PACKET) DEVICE

In the IDENTIFY DEVICE command various parameters are communicated to the host from the device. The following sections define those words that are different from and additions to the ATA8-ACS standard definition of the data contents. Serial ATA features and capabilities include a means by which their presence and support may be determined, and a means for enabling them if optionally supported.

The IDENTIFY (PACKET) DEVICE settings requirements shall be implemented by native Serial ATA devices. The IDENTIFY (PACKET) DEVICE settings requirements are optional for parallel ATA devices with an external Serial ATA bridge attached.

13.2.1 IDENTIFY DEVICE

Table 75 – IDENTIFY DEVICE information

Word	O/M	F/V	
0-46			Set as indicated in ATA8-ACS
47	M		Multiple Count
		F	15-8 80h
		R	7-0 00h = Reserved 01h-10h = Maximum number of sectors that shall be transferred per interrupt on READ/WRITE MULTIPLE commands 11h-FFh = Reserved
48			Set as indicated in ATA8-ACS
49	M		Capabilities
			15-12 Set as indicated in ATA8-ACS
		F	11 Shall be set to one
		F	10 Shall be set to one
			9-0 Set as indicated in ATA8-ACS
50-52			Set as indicated in ATA8-ACS
53	M		Field validity
		R	15-3 Reserved
		F	2 1=the fields reported in word 88 are valid 0=the fields reported in word 88 are not valid
		F	1 1=the fields reported in words (70:64) are valid 0=the fields reported in words (70:64) are not valid
		F	0 Obsolete
54-62			Set as indicated in ATA8-ACS
63	M		Multiword DMA transfer
			15-3 Set as indicated in ATA8-ACS
		F	2 1= Multiword DMA mode 2 and below are supported
		F	1 1= Multiword DMA mode 1 and below are supported
		F	0 1= Multiword DMA mode 0 is supported

Word	O/M	F/V		
64	M		PIO transfer modes supported	
			15-2	Set as indicated in ATA8-ACS
		F	1-0	PIO modes 3 and 4 supported
65	M		Minimum Multiword DMA transfer cycle time per word	
		F	15-0	Cycle time in nanoseconds
66	M		Manufacturer's recommended Multiword DMA transfer cycle time	
		F	15-0	Cycle time in nanoseconds
67	M		Minimum PIO transfer cycle time without flow control	
		F	15-0	Cycle time in nanoseconds
68	M		Minimum PIO transfer cycle time with IORDY flow control	
		F	15-0	Cycle time in nanoseconds
69-74			Set as indicated in ATA8-ACS	
75	O		Queue depth	
		R	15-5	Reserved
		F	4-0	Maximum queue depth - 1
76	O		Serial ATA capabilities	
		F	15	Supports READ LOG DMA EXT as equivalent to READ LOG EXT
		F	14	Supports Device Automatic Partial to Slumber transitions
		F	13	Supports Host Automatic Partial to Slumber transitions
		F	12	Supports Native Command Queuing priority information
		F	11	Supports Unload while NCQ commands outstanding
		F	10	Supports Phy event counters
		F	9	Supports receipt of host-initiated interface power management requests
		F	8	Supports Native Command Queuing
		R	7-4	Reserved for future Serial ATA signaling speed grades
		F	3	1 = Supports Serial ATA Gen3 signaling speed (6.0 Gbps)
		F	2	1 = Supports Serial ATA Gen2 signaling speed (3.0 Gbps)
		F	1	1 = Supports Serial ATA Gen1 signaling speed (1.5 Gbps)
		F	0	Shall be cleared to zero
77	O			Serial ATA Additional capabilities
		R	15-6	Reserved
		F	5	Supports NCQ Queue Management Command
		F	4	Supports NCQ Streaming
		V	3-1	Coded value indicating current negotiated Serial ATA signal speed
		F	0	Shall be cleared to zero
78	O		Serial ATA features supported	
		R	15-7	Reserved
		F	6	1 = Supports software settings preservation
		R	5	Reserved
		F	4	1 = Supports in-order data delivery
		F	3	1 = Device supports initiating interface power management
		F	2	1 = Supports DMA Setup Auto-Activate optimization
		F	1	1 = Supports non-zero buffer offsets in DMA Setup FIS
		F	0	Shall be cleared to zero
79	O		Serial ATA features enabled	
		R	15-8	Reserved
		V	F	1 = Device Automatic Partial to Slumber transitions enabled
		V	6	1 = Software settings preservation enabled
		R	5	Reserved
		V	4	1 = In-order data delivery enabled

Word	O/M	F/V	
		V	3 1 = Device initiating interface power management enabled
		V	2 1 = DMA Setup Auto-Activate optimization enabled
		V	1 1 = Non-zero buffer offsets in DMA Setup FIS enabled
		F	0 Shall be cleared to zero
80-87			Set as indicated in ATA8-ACS
88			15-6 Set as indicated in ATA8-ACS
		F	5 1=Ultra DMA mode 5 and below are supported
		F	4 1=Ultra DMA mode 4 and below are supported
		F	3 1=Ultra DMA mode 3 and below are supported
		F	2 1=Ultra DMA mode 2 and below are supported
		F	1 1=Ultra DMA mode 1 and below are supported
		F	0 1=Ultra DMA mode 0 is supported
89-92			Set as indicated in ATA8-ACS
93		V	COMRESET result. The contents of this word shall be cleared to zero.
94-221			Set as indicated in ATA8-ACS
222			Transport Major Revision 0000h or FFFFh = device does not report version **Bits** **Description**
		F	15:12 Transport Type 0h = Parallel 1h = Serial 2h - Fh = Reserved
			11:6 Reserved
		F	5 SATA Rev 3.0
		F	4 SATA Rev 2.6
		F	3 SATA Rev 2.5
		F	2 SATA II: Extensions
		F	1 SATA 1.0a
		F	0 ATA8-AST
223		F	Transport Minor Revision
224-255			Set as indicated in ATA8-ACS

Key:
M = Support of the word is mandatory.
O = Support of the word is optional.
F = the content of the word is fixed and does not change. For removable media devices, these values may change when media is removed or changed.
V = the contents of the word is variable and may change depending on the state of the device or the commands executed by the device.
X = the content of the word is vendor specific and may be fixed or variable.
R = the content of the word is reserved and shall be zero.

13.2.1.1 Word 0 - 46: Set as indicated in ATA8-ACS

13.2.1.2 Word 47: Multiword PIO transfer

Bits 15 through 8 of word 47 shall be set as indicated in ATA8-ACS.

Bits 7 through 0 are used to indicate the maximum number of sectors that shall be transferred per interrupt on READ/WRITE MULTIPLE commands. This field shall be set to 16 or less. See section 10.3.10.1.

13.2.1.3 Word 48: Set as indicated in ATA8-ACS

13.2.1.4 Word 49: Capabilities

Bits 15 through 12 of word 49 shall be set as indicated in ATA8-ACS.

Bit 11 of word 49 is used to determine whether a device supports IORDY. This bit shall be set to one, indicating the device supports IORDY operation.

Bit 10 of word 49 is used to indicate a device's ability to enable or disable the use of IORDY. This bit shall be set to one, indicating the device supports the disabling of IORDY. Disabling and enabling of IORDY is accomplished using the SET FEATURES command.

Bits 9 - 0 of word 49 shall be set as indicated in ATA8-ACS.

13.2.1.5 Words 50 - 52: Set as indicated in ATA8-ACS

13.2.1.6 Word 53: Field validity

Bit 0 shall be set to one. Bit 1 of word 53 shall be set to one, the values reported in words 64 through 70 are valid. Any device that supports PIO mode 3 or above, or supports Multiword DMA mode 1 or above, shall set bit 1 of word 53 to one and support the fields contained in words 64 through 70. Bit 2 of word 53 shall be set to one indicating the device supports Ultra DMA and the values reported in word 88 are valid. Bits 15-3 are reserved.

13.2.1.7 Word 54 - 62: Set as indicated in ATA8-ACS

13.2.1.8 Word 63: Multiword DMA transfer

Bits 2 - 0 of word 63 shall be set to one indicating that the device supports Multiword DMA modes 0, 1, and 2. Bits 15 – 3 shall be set as indicated in ATA8-ACS.

13.2.1.9 Word 64: PIO transfer modes supported

Bits 1 - 0 of word 64 shall be set to one indicating that the device supports PIO modes 3 and 4. Bits 15 – 2 shall be set as indicated in ATA8-ACS.

13.2.1.10 Word 65: Minimum Multiword DMA transfer cycle time per word

Shall be set to indicate 120 ns.

13.2.1.11 Word 66: Device recommended Multiword DMA cycle time

Shall be set to indicate 120 ns.

13.2.1.12 Word 67: Minimum PIO transfer cycle time without flow control

Shall be set to indicate 120 ns.

13.2.1.13 Word 68: Minimum PIO transfer cycle time with IORDY

Shall be set to indicate 120 ns.

13.2.1.14 Words 69-74: Set as indicated in ATA8-ACS

13.2.1.15 Word 75: Queue depth

This word is as defined in the ATA8-ACS standard. The Native Command Queuing protocol supports at most 32 queued commands. With Native Command Queueing, the host shall issue only unique tag values for queued commands that have a value less than or equal to the value reflected in this field (e.g., for device reporting a value in this field of 15, corresponding to a maximum of 16 outstanding commands, the host shall never use a tag value greater than 15 when issuing Native Command Queueing commands).

13.2.1.16 Word 76: Serial ATA capabilities

If not 0000h or FFFFh, the device claims compliance with the Serial ATA specification and supports the signaling speed indicated in bits 1-3. Since Serial ATA supports generational compatibility, multiple bits may be set. Bit 0 is reserved and shall be cleared to zero (thus a Serial ATA device has at least one bit cleared in this field and at least one bit set providing clear differentiation). If this field is not 0000h or FFFFh, words 77 through 79 shall be valid. If this field is 0000h or FFFFh the device does not claim compliance with the Serial ATA specification and Words 76 through 79 are not valid and shall be ignored.

Bit 0 shall be cleared to zero.

Bit 1 when set to one indicates that the device is a Serial ATA device and supports the Gen1 signaling speed of 1.5 Gbps.

Bit 2 when set to one indicates that the device is a Serial ATA device and supports the Gen2 signaling speed of 3.0 Gbps.

Bit 3 when set to one indicates that the device is a Serial ATA device and supports the Gen3 signaling speed of 6.0 Gbps.

Bit 4-7 are reserved for future Serial ATA signaling speed grades and shall be cleared to zero.

Bit 8 when set to one indicates that the Serial ATA device supports the Native Command Queuing scheme defined in section 13.6.

Bit 9 when set to one indicates that the Serial ATA device supports the Partial and Slumber interface power management states when initiated by the host.

Bit 10 when set to one indicates that the Serial ATA device supports Phy event counters. If the device supports Phy event counters, it shall support the Phy Event Counter Log (see 13.9.3)

Bit 11 when set to one indicates that the device supports performing an unload/park of the heads upon reception of the IDLE IMMEDIATE command with the Unload Feature specified while NCQ commands are outstanding. This bit shall only be set to one if the device supports NCQ as shown in bit 8 of Word 76.

Bit 12 when set to one indicates that the device supports the Priority field in the READ FPDMA QUEUED and WRITE FPDMA QUEUED commands and optimization based on this information. This bit shall only be set to one if the device supports NCQ as shown in bit 8 of Word 76.

Bit 13 indicates that the device supports host Automatic Partial to Slumber transitions. The device shall tolerate a Partial exit latency up to the max Slumber exit latency. This allows the host to asynchronously transition from Partial to Slumber.
If Word 76, bit 9 (supports receipt of host-initiated interface power management requests) is cleared to zero, then bit 13 shall be cleared to zero.

Bit 14 indicates that the device supports Automatic Partial to Slumber transitions and may asynchronously transition from Partial to Slumber when enabled.
If Word 78, bit 3 (supports initiating interface power management) is cleared to zero, then Word 76 bit 14 shall be cleared to zero.

Bit 15 when set to one indicates that either the READ LOG DMA EXT and READ LOG EXT commands may be used in all cases with identical results (see 13.7). If IDENTIFY DEVICE word 119 bit 3 is cleared to zero, this bit shall be cleared to zero. If bit 15 is cleared to zero and the host issues the READ LOG DMA EXT command to read the Queued Error Log or the Phy Event Counters log, the device shall return command aborted.

13.2.1.17 Word 77: Serial ATA Additional capabilities

Word 77 reports additional optional capabilities supported by the device. Support for this word is optional and if not supported, the word shall be zero indicating the device has no support for additional Serial ATA capabilities.

Bit 0 shall be cleared to zero

Bits 1-3 are a coded value that indicates the Serial ATA Phy speed at which the device is currently communicating. Table 76 defines these values:

Coded Values			
Bit 3	Bit 2	Bit 1	Description
0	0	1	Gen1 signaling speed of 1.5 Gbps
0	1	0	Gen2 signaling speed of 3.0 Gbps
0	1	1	Gen3 signaling speed of 6.0 Gbps
All Non-Defined Values			Reserved for future Serial ATA signaling speeds

Table 76 - Coded Values for Negotiated Serial ATA Signaling Speed

Note: In the case of system configurations that have more than one Phy link in the data path (eg. port multiplier), the indicated speed is only relevant for the link between the device Phy and its immediate host Phy. It is possible for each link in the data path to negotiate a different Serial ATA signaling speed.

Bit 4, when set to one, indicates that the device supports NCQ Streaming. See the use of the ICC field by the READ FPDMA QUEUED and WRITE FPDMA QUEUED commands. This bit shall only be set to one if the device supports NCQ as shown in bit 8 of Word 76.

Bit 5, when set to one indicates that the device supports use of the NCQ QUEUE MANAGEMENT command by the host. This bit shall only be set to one if the device supports NCQ as shown in bit 8 of Word 76.

Bit 6-15 are reserved and shall be cleared to zero

13.2.1.18 Word 78: Serial ATA features supported

Word 78 reports the optional features supported by the device. Support for this word is optional and if not supported the word shall be zero indicating the device has no support for new Serial ATA capabilities.

Bit 0 shall be cleared to zero.

Bit 1 indicates whether the device supports the use of non-zero buffer offsets in the DMA Setup FIS. When set to one, the device supports transmission and reception of DMA Setup FISes with a non-zero value in the Buffer Offset field of the FIS. When cleared to zero, the device supports transmission and reception of the DMA Setup FIS only with the Buffer Offset field cleared to zero.

Bit 2 indicates whether the device supports the use of the DMA Setup FIS Auto-Activate optimization as described in section 10.3.8.3.1. When set to one the device supports use of the Auto-Activate optimization and when cleared to zero the device does not support the Auto-Activate optimization.

Bit 3 indicates whether the device supports initiating power management requests to the host. When set to one the device supports initiating interface power management requests and when cleared to zero the device does not support initiating power management requests. A device may support reception of power management requests initiated by the host as described in the definition of bit 9 of Word 76 without supporting initiating such power management requests as indicated by this bit.

Bit 4 indicates whether the device supports guaranteed in-order data delivery when non-zero buffer offsets are used in the DMA Setup FIS. When set to one, the device guarantees in-order data delivery for READ FPDMA QUEUED or WRITE FPDMA QUEUED commands when non-zero buffer offsets are used with multiple DMA Setup FIS. Target data is delivered in order, starting with the first LBA through command completion. When Bit 4 is cleared to zero, the device does not guarantee in-order data delivery when non-zero buffer offsets are enabled. In this case, data may be interleaved both within a command and across multiple commands. By default this field shall be zero.

Bit 5 is reserved and shall be cleared to zero.

Bit 6 indicates whether the device supports software settings preservation as defined in section 13.5. When set to one the device supports software settings preservation across COMRESET. When cleared to zero the device clears all software settings when a COMRESET occurs.

Bit 7-15 are reserved and shall be cleared to zero

13.2.1.19 Word 79: Serial ATA features enabled

Word 79 reports which optional features supported by the device are enabled. This word shall be supported if optional Word 78 is supported and shall not be supported if optional Word 78 is not supported.

Bit 0 shall be cleared to zero.

Bit 1 indicates whether device support for use of non-zero buffer offsets in the DMA Setup FIS is enabled. When set to one, device transmission of DMA Setup FISes with a non-zero value in the Buffer Offset field of the FIS is enabled. When cleared to zero, the device is permitted to transmit DMA Setup FIS only with the Buffer Offset field cleared to zero. By default this field shall be zero.

Bit 2 indicates whether device support for use of the DMA Setup FIS Auto-Activate optimization as described in section 10.3.8.3.1 is enabled. When set to one, the device may utilize the Auto-

Activate optimization. When cleared to zero the device shall not utilize the Auto-Activate optimization. By default, this field shall be zero.

Bit 3 indicates whether device support for initiating power management requests to the host is enabled. When set to one the device may initiate power management transition requests. When cleared to zero the device shall not initiate interface power management requests to the host. This field shall be zero by default.

Bit 4 indicates whether device support for guaranteed in-order data delivery when non-zero buffer offsets are used in the DMA Setup FIS is enabled. When set to one and non-zero buffer offset is enabled, the device may satisfy a READ FPDMA QUEUED or WRITE FPDMA QUEUED command by transmitting multiple DMA Setup FISes with non-zero buffer offset values where appropriate, provided that the target data is delivered in order, starting with the first LBA through command completion. When Bit 4 is cleared to zero, the device may interleave data both in a command and across multiple commands using non-zero buffer offsets if non-zero buffer offsets are enabled. By default this field shall be zero.

Bit 5 is reserved and shall be cleared to zero.

Bit 6 indicates whether device support for software settings preservation is enabled. When set to one the device shall preserve software settings across COMRESET. When cleared to zero the device shall clear software settings when COMRESET occurs. If the device supports software settings preservation this field shall be one by default. If the device does not support software settings preservation this field shall be zero by default.

Bit 7 indicates whether or not device Automatic Partial to Slumber transitions are enabled. When enabled the device may asynchronously transition from Partial to Slumber. If Word 76, bit 14 (Supports Device Automatic Partial to Slumber transitions) is cleared to zero, then bit 7 shall be cleared to zero. If Word 79, bit 3 (Device initiating interface power management enabled) is cleared to zero, then bit 7 shall be cleared to zero.

Bits 8-15 are reserved and shall be cleared to zero.

13.2.1.20 Words 80-87: Set as indicated in ATA8-ACS

13.2.1.21 Word 88: Ultra DMA modes

Bits 5 - 0 of Word 88 shall be set to one indicating that the device supports Ultra DMA modes 0, 1, 2, 3, 4, and 5. Bits 15 – 5 shall be set as indicated in ATA8-ACS.

13.2.1.22 Words 89 - 92: Set as indicated in ATA8-ACS

13.2.1.23 Word 93: Hardware configuration test results

Word 93 shall be set to 0000h indicating that the word is not supported.

13.2.1.24 Words 94-221: Set as indicated in ATA8-ACS.

13.2.1.25 Word 222: Transport Major Revision

Bits (15:12) shall be set to 1h.

13.2.1.26 Word 223: Transport Minor Revision

Set as indicated in ATA8-ACS.

13.2.1.27 Words 224-255: Set as indicated in ATA8-ACS.

13.2.2 IDENTIFY PACKET DEVICE

Table 77 – IDENTIFY PACKET DEVICE information

Word	O/M	F/V	
0-48			Set as indicated in ATA8-ACS
49	M		Capabilities
			15-12 Set as indicated in ATA8-ACS
		F	11 Shall be set to one
		F	10 Shall be set to one
			9-0 Set as indicated in ATA8-ACS
50-52			Set as indicated in ATA8-ACS
53	M		Field validity
		R	15-3 Reserved
		F	2 1=the fields reported in word 88 are valid
			0=the fields reported in word 88 are not valid
		F	1 1=the fields reported in words (70:64) are valid
			0=the fields reported in words (70:64) are not valid
		F	0 Obsolete
54-62			Set as indicated in ATA8-ACS
63	M		Multiword DMA transfer
			15-3 Set as indicated in ATA8-ACS
		F	2 1= Multiword DMA mode 2 and below are supported
		F	1 1= Multiword DMA mode 1 and below are supported
		F	0 1= Multiword DMA mode 0 is supported
64	M		PIO transfer modes supported
			15-2 Set as indicated in ATA8-ACS
		F	1-0 PIO modes 3 and 4 supported
65	M		Minimum Multiword DMA transfer cycle time per word
		F	15-0 Cycle time in nanoseconds
66	M		Manufacturer's recommended Multiword DMA transfer cycle time
		F	15-0 Cycle time in nanoseconds
67	M		Minimum PIO transfer cycle time without flow control
		F	15-0 Cycle time in nanoseconds
68	M		Minimum PIO transfer cycle time with IORDY flow control
		F	15-0 Cycle time in nanoseconds
69-75			Set as indicated in ATA8-ACS
76	O		Serial ATA capabilities
		F	15 Reserved
		F	14 Supports Device Automatic Partial to Slumber transitions
		F	13 Supports Host Automatic Partial to Slumber transitions
		R	12-11 Reserved
		F	10 Supports Phy event counters
		F	9 Supports receipt of host-initiated interface power management requests
		F	8 Reserved
		R	7-4 Reserved for future Serial ATA signaling speed grades
		F	3 1 = Supports Serial ATA Gen3 signaling speed (6.0 Gbps)
		F	2 1 = Supports Serial ATA Gen2 signaling speed (3.0 Gbps)
		F	1 1 = Supports Serial ATA Gen1 signaling speed (1.5 Gbps)

Word	O/M	F/V	
		F	0 Shall be cleared to zero
77	O		Serial ATA Additional capabilities
		R	15-4 Reserved
		V	3-1 Coded value indicating current negotiated Serial ATA signal speed
		F	0 Shall be cleared to zero
78	O		Serial ATA features supported
		R	15-7 Reserved
		F	6 1 = Supports software settings preservation
		F	5 1 = Supports asynchronous notification
		R	4 Reserved
		F	3 1 = Device supports initiating interface power management
		F	2-1 Reserved
		F	0 Shall be cleared to zero
79	O		Serial ATA features enabled
		R	15-7 Reserved
		V	6 1 = Software settings preservation enabled
		V	5 1 = Asynchronous notification enabled
		R	4 Reserved
		V	3 1 = Device initiating interface power management enabled
		R	2-1 Reserved
		F	0 Shall be cleared to zero
80-87			Set as indicated in ATA8-ACS
88			15-6 Set as indicated in ATA8-ACS
		F	5 1=Ultra DMA mode 5 and below are supported
		F	4 1=Ultra DMA mode 4 and below are supported
		F	3 1=Ultra DMA mode 3 and below are supported
		F	2 1=Ultra DMA mode 2 and below are supported
		F	1 1=Ultra DMA mode 1 and below are supported
		F	0 1=Ultra DMA mode 0 is supported
89-92			Set as indicated in ATA8-ACS
93		V	COMRESET result. The contents of this word shall be cleared to zero.
94-221			Set as indicated in ATA8-ACS
222			Transport Major Revision 0000h or FFFFh = device does not report version **Bits** **Description**
		F	15:12 Transport Type 0h = Parallel 1h = Serial 2h - Fh = Reserved
			11:6 Reserved
		F	5 SATA Rev 3.0
		F	4 SATA Rev 2.6
		F	3 SATA Rev 2.5
		F	2 SATA II: Extensions
		F	1 SATA 1.0a
		F	0 ATA8-AST
223		F	Transport Minor Revision
224-255			Set as indicated in ATA8-ACS

Key:
M = Support of the word is mandatory.
O = Support of the word is optional.
F = the content of the word is fixed and does not change. For removable media devices, these values may change when media is removed or changed.
V = the contents of the word is variable and may change depending on the state of the device or the commands executed by the device.
X = the content of the word is vendor specific and may be fixed or variable.
R = the content of the word is reserved and shall be zero.

13.2.2.1 Word 0 - 48: Set as indicated in ATA8-ACS

13.2.2.2 Word 49: Capabilities

Bits 15 through 12 of word 49 shall be set as indicated in ATA8-ACS.

Bit 11 of word 49 is used to determine whether a device supports IORDY. This bit shall be set to one, indicating the device supports IORDY operation.

Bit 10 of word 49 is used to indicate a device's ability to enable or disable the use of IORDY. This bit shall be set to one, indicating the device supports the disabling of IORDY. Disabling and enabling of IORDY is accomplished using the SET FEATURES command.

Bits 9 - 0 of word 49 shall be set as indicated in ATA8-ACS.

13.2.2.3 Words 50 - 52: Set as indicated in ATA8-ACS

13.2.2.4 Word 53: Field validity

Bit 0 shall be set to one. Bit 1 of word 53 shall be set to one, the values reported in words 64 through 70 are valid. Any device that supports PIO mode 3 or above, or supports Multiword DMA mode 1 or above, shall set bit 1 of word 53 to one and support the fields contained in words 64 through 70. Bit 2 of word 53 shall be set to one indicating the device supports Ultra DMA and the values reported in word 88 are valid. Bits 15-3 are reserved.

13.2.2.5 Word 54 - 62: Set as indicated in ATA8-ACS

13.2.2.6 Word 63: Multiword DMA transfer

Bits 2 - 0 of word 63 shall be set to one indicating that the device supports Multiword DMA modes 0, 1, and 2. Bits 15 – 3 shall be set as indicated in ATA8-ACS.

13.2.2.7 Word 64: PIO transfer modes supported

Bits 1 - 0 of word 64 shall be set to one indicating that the device supports PIO modes 3 and 4. Bits 15 – 2 shall be set as indicated in ATA8-ACS.

13.2.2.8 Word 65: Minimum Multiword DMA transfer cycle time per word

Shall be set to indicate 120 ns.

13.2.2.9 Word 66: Device recommended Multiword DMA cycle time

Shall be set to indicate 120 ns.

13.2.2.10 Word 67: Minimum PIO transfer cycle time without flow control

Shall be set to indicate 120 ns.

13.2.2.11 Word 68: Minimum PIO transfer cycle time with IORDY

Shall be set to indicate 120 ns.

13.2.2.12 Words 69-75: Set as indicated in ATA8-ACS

13.2.2.13 Word 76: Serial ATA capabilities

Word 76 shall have the content described for IDENTIFY DEVICE data word 76

13.2.2.14 Word 77: Serial ATA Additional capabilities

Support for this word is optional and if not supported, the word shall be zero indicating the device has no support for additional Serial ATA
capabilities.

Bit 0 shall be cleared to zero

Bit 1-3 are a coded value to indicate the current Serial ATA phy speed that device is communicating at. Table 76 defines these values.

> Note: In the case of system configurations that have more than one Phy link in the data path (eg. port multiplier), the indicated speed is only relevant for the link between the device Phy and its immediate host Phy. It is possible for each link in the data path to negotiate a different Serial ATA signaling speed.

Bit 4-15 are reserved and shall be cleared to zero

13.2.2.15 Word 78: Serial ATA features supported

Word 78 reports the optional features supported by the device. Support for this word is optional and if not supported the word shall be zero indicating the device has no support for new Serial ATA capabilities.

Bit 0 shall be cleared to zero.

Bit 1-2 are reserved and shall be cleared to zero.

Bit 3 indicates whether the device supports initiating power management requests to the host. When set to one the device supports initiating interface power management requests and when cleared to zero the device does not support initiating power management requests. A device may support reception of power management requests initiated by the host as described in the definition of bit 9 of Word 76 without supporting initiating such power management requests as indicated by this bit.

Bit 4 is reserved and shall be cleared to zero.

Bit 5 indicates whether the device supports asynchronous notification to indicate to the host that attention is required. When set to one the device supports initiating notification events and when cleared to zero the device does not support initiating notification events. An example of an event

that the device may need attention for includes a media change. Asynchronous device notification is described in section 13.8.

Bit 6 indicates whether the device supports software settings preservation as defined in section 13.5. When set to one the device supports software settings preservation across COMRESET. When cleared to zero the device clears all software settings when a COMRESET occurs.

Bit 7-15 are reserved and shall be cleared to zero.

13.2.2.16 Word 79: Serial ATA features enabled

Word 79 shall have the content described for IDENTIFY DEVICE data word 79.

13.2.2.17 Words 80-87: Set as indicated in ATA8-ACS

13.2.2.18 Word 88: Ultra DMA modes

Bits 5 - 0 of Word 88 shall be set to one indicating that the device supports Ultra DMA modes 0, 1, 2, 3, 4, and 5. Bits 15 – 5 shall be set as indicated in ATA8-ACS.

13.2.2.19 Word 89 - 92: Set as indicated in ATA8-ACS

13.2.2.20 Word 93: Hardware configuration test results

Word 93 shall be set to 0000h indicating that the word is not supported.

13.2.2.21 Words 94-221: Set as indicated in ATA8-ACS.

13.2.2.22 Word 222: Transport Major Revision

Bits (15:12) shall be set to 1h.

13.2.2.23 Word 223: Transport Minor Revision

Set as indicated in ATA8-ACS.

13.2.2.24 Words 224-255: Set as indicated in ATA8-ACS.

13.2.3 Determining Support for Serial ATA Features

Software should verify a device's Serial ATA capabilities by reading the relevant bits in Words 76-79 of the IDENTIFY (PACKET) DEVICE data. A device claims compliance with the Serial ATA specification by setting IDENTIFY (PACKET) DEVICE Word 76 appropriately.

Although Serial ATA was first introduced in the ATA/ATAPI specification material in the ATA/ATAPI-7 revision, it cannot be assumed that when Word 80 (Major version number) of the IDENTIFY (PACKET) DEVICE data is read with support of ATA/ATAPI-7 or later that the device supports Serial ATA or any specific Serial ATA features.

13.3 SET FEATURES

Devices are informed of host capabilities and have optional features enabled/disabled through the SET FEATURES command defined in the ATA8-ACS standard. Serial ATA features are controlled using a features value as defined in Table 78.

Table 78 - Features enable/disable values

Features(7:0) Value	Description
10h	Enable use of Serial ATA feature
90h	Disable use of Serial ATA feature

Count(7:0) contains the specific Serial ATA feature to enable or disable. The specific Serial ATA features in which SET FEATURES is applicable are defined in Table 79.

Table 79 - Feature identification values

Count(7:0) Value	Description
00h	Reserved
01h	Non-zero buffer offset in DMA Setup FIS
02h	DMA Setup FIS Auto-Activate optimization
03h	Device-initiated interface power state transitions
04h	Guaranteed In-Order Data Delivery
05h	Asynchronous Notification
06h	Software Settings Preservation
07h	Device Automatic Partial to Slumber transitions
08h - FFh	Reserved for future Serial ATA definition

13.3.1 Enable/Disable Non-Zero Offsets in DMA Setup

A Count(7:0) value of 01h is used by the host to enable or disable non-zero buffer offsets in the DMA Setup FIS when the device utilizes the First-party DMA mechanism (see section 13.6.1.2). By default, non-zero buffer offsets in the DMA Setup FIS are disabled. Enabling non-zero buffer offsets in the DMA Setup FIS is useful for performing out of order data delivery within commands, e.g. delivering the last half of the data before the first half of the data, or to support segmentation of large First-party DMA operations into multiple data phases, i.e. to effectively allow sharing of the SATA link during Data I/O, e.g. to provide a mechanism for the host to transmit additional READ FPDMA QUEUED or WRITE FPDMA QUEUED commands to the device for re-order consideration. The enable/disable state for non-zero offsets in DMA Setup FISes shall be preserved across software reset. The enable/disable state for non-zero offsets in DMA Setup FISes shall be reset to its default state upon COMRESET.

13.3.2 Enable/Disable DMA Setup FIS Auto-Activate Optimization

A Count(7:0) value of 02h is used by the host to enable or disable the DMA Setup FIS optimization for automatically activating transfer of the first host-to-device Data FIS following a DMA Setup FIS with a host-to-device transfer direction. For transfers from the host to the device, First-party DMA transfers require a sequence of DMA Setup FIS followed by a DMA Activate FIS to initiate the transfer. The Auto-Activate optimization allows the DMA Setup FIS operation to imply immediate activation thereby eliminating the need for the additional separate DMA Activate FIS to start the transfer. Enabling the optimization notifies the device that the host bus adapter implementation allows the DMA Setup FIS to include the Auto-Activate bit to trigger immediate transfer following receipt and processing of the DMA Setup FIS. By default, the optimization is disabled. See section 10.3.8.3.1 for additional details. The enable/disable state for the auto-

activate optimization shall be preserved across software reset. The enable/disable state for the auto-activate optimization shall be reset to its default state upon COMRESET.

13.3.3 Enable/Disable Device-Initiated Interface Power State Transitions

A Count(7:0) value of 03h is used by the host to enable or disable device initiation of interface power state transitions. By default, the device is not permitted to attempt interface power state transitions by issuing PMREQ_P_P or PMREQ_S_P to the host. The host may enable device initiation of such interface power state transitions for such cases where it may be desirable for the device to attempt initiating such transitions. The enable/disable state for device initiated power management shall persist across software reset. The enable/disable state shall be reset to its default disabled state upon COMRESET.

If device initiated interface power management is enabled, the device shall not attempt to initiate an interface power state transition between reset and the delivery of the device reset signature.

13.3.4 Enable/Disable Guaranteed in-Order Data Delivery

A Count(7:0) value of 04h is used by the host to enable or disable guaranteed in-order data delivery when the device utilizes the First-party DMA mechanism and non-zero buffer offsets in the DMA Setup FIS. By default, guaranteed in-order data delivery is disabled. Enabling guaranteed in-order data delivery is useful for segmenting large I/O processes into multiple atomic data phases using non-zero buffer offsets in the DMA Setup FIS, while minimizing the complexity that may be imposed on the host with out-of-order data delivery. The enable/disable state for guaranteed in-order data delivery shall be preserved across software reset. The enable/disable state for guaranteed in-order data delivery shall be reset to its default state upon COMRESET.

13.3.5 Enable/Disable Asynchronous Notification

A Count(7:0) value of 05h is used by the host to enable or disable asynchronous notification. By default, asynchronous notification is disabled. The host may enable asynchronous notification in order to allow the device to request attention without the host polling. As an example, this may be useful to avoid polling for media change events in ATAPI devices. The enable/disable state for asynchronous notification shall be preserved across software reset. The enable/disable state for asynchronous notification shall be reset to its default state upon COMRESET.

13.3.6 Enable/Disable Software Settings Preservation

A Count(7:0) value of 06h is used by the host to enable or disable software settings preservation, as defined in section 13.5. By default, if the device supports software settings preservation the feature is enabled on power-up. The enable/disable state for software settings preservation shall persist across software reset. The enable/disable state for software settings preservation shall be reset to its default state upon COMRESET. The host may disable software settings preservation in order to not preserve software settings across COMRESET (and make COMRESET equivalent to hardware reset in Parallel ATA).

13.3.7 Enable/Disable Device Automatic Partial to Slumber Transitions

A Count(7:0) value of 07h is used by the host to enable or disable Device Automatic Partial to Slumber transitions. By default, if the device supports Device Automatic Partial to Slumber transitions the feature is disabled on power-up. The enable/disable state for Device Automatic Partial to Slumber transitions shall persist across software reset. The enable/disable state for Automatic Partial to Slumber transitions shall be reset to its default state upon COMRESET.

Device Automatic Partial to Slumber transitions shall not be enabled if Device-Initiated Interface Power State transitions are disabled. Attempting to enable Automatic Partial to Slumber

transitions while Device-Initiated Interface Power State transitions are disabled will result in the device aborting the Set Features command. Attempting to disable Device Automatic Partial to Slumber transitions when it is already disabled will have no effect and the device shall return successful completion of the Set Features command.

13.4 Device Configuration Overlay

13.4.1 Device Configuration Overlay Identify

Figure 202 defines additional features and capabilities that support may be controlled for using the DEVICE CONFIGURATION IDENTIFY command in the ATA8-ACS standard. The device is only required to support setting these features if the device reports support for Device Configuration Overlay in either IDENTIFY DEVICE or IDENTIFY PACKET DEVICE, respectively.

Word	Description
0-7	As defined in the ATA8-ACS standard
8	Serial ATA command / feature sets supported 15-7 Reserved (0) 6 1 = Reporting support for NCQ QUEUE MANAGEMENT[1] is allowed is changeable 5 1 = Reporting support for Automatic Partial to Slumber transitions is changeable 4 1 = Reporting support for software settings preservation is changeable 3 1 = Reporting support for asynchronous notification is changeable 2 1 = Reporting support for interface power management is changeable 1 1 = Reporting support for non-zero buffer offsets in DMA Setup FIS[1] is changeable 0 1 = Reporting support for Native Command Queuing[1] is changeable
9	Reserved for Serial ATA
10-255	As defined in the ATA8-ACS standard
NOTE: 1. Applicable to non-PACKET devices only – i.e. IDENTIFY DEVICE.	

Figure 202 –DEVICE CONFIGURATION IDENTIFY data structure

WORD 8: Serial ATA command / feature sets supported

This word describes which features for which support is changeable. A feature may be supported but not be changeable. If bit 0 of word 8 is set to one, then support for Native Command Queuing is changeable. The setting of this bit is applicable to non-PACKET devices only.

If bit 1 of word 8 is set to one, then support for non-zero buffer offsets in the DMA Setup FIS is changeable. The setting of this bit is applicable to non-PACKET devices only.

If bit 2 of word 8 is set to one, then suport receiving host initiated power management requests and/or sending device initiated power management requests is changeable.

If bit 3 of word 8 is set to one, then support for asynchronous notification is changeable.

If bit 4 of word 8 is set to one, then support for software settings preservation is changeable.

If bit 5 of word 8 is set to one, then support for Automatic Partial to Slumber transitions is changeable.

If bit 6 of word 8 is set to one, then support for the NCQ QUEUE MANAGEMENT command is changeable.

Bits 7-15 are reserved and shall be cleared to zero.

WORD 9: Reserved for Serial ATA

This word is reserved for Serial ATA and all bits shall be cleared to zero.

13.4.2 Device Configuration Overlay Set

Figure 203 defines additional features and capabilities that support may be controlled for using the DEVICE CONFIGURATION SET command in the ATA8-ACS standard. The device is only required to support setting these features if the device reports support for Device Configuration Overlay in either IDENTIFY DEVICE or IDENTIFY PACKET DEVICE, respectively.

Word	Description
0-7	As defined in the ATA8-ACS standard
8	Serial ATA command / feature sets supported 15-7 Reserved 6 1 = Reporting support for the NCQ QUEUE MANAGEMENT command[1] is allowed 5 1 = Reporting support for Automatic Partial to Slumber transitions is allowed 4 1 = Reporting support for software settings preservation is allowed 3 1 = Reporting support for asynchronous notification is allowed 2 1 = Reporting support for interface power management is allowed 1 1 = Reporting support for non-zero buffer offsets in DMA Setup FIS[1] is allowed 0 1 = Reporting support for Native Command Queuing[1] is allowed
9	Reserved for Serial ATA
10-255	As defined in the ATA8-ACS standard
NOTE: 1. Applicable to non-PACKET devices only – i.e. IDENTIFY DEVICE.	

Figure 203 - DEVICE CONFIGURATION SET data structure

WORD 8: Serial ATA command / feature sets supported

This word enables configuration of command sets and feature sets.

If bit 0 of word 8 is cleared to zero, then the device shall:

a) disable support for Native Command Queuing;
b) clear word 76 bits 8,11,and 12 in the IDENTIFY DEVICE data to zero;
c) clear word 78 bits 1, 2, and 4 in the IDENTIFY DEVICE data to zero;
d) clear word 79 bits 1, 2 and 4 in the IDENTIFY DEVICE data to zero; and
e) if NCQ is disabled and READ FPDMA QUEUED or WRITE FPDMA QUEUED is issued to the device, the device shall abort the command with the ERR bit set to one in the Status field and the ABRT bit set to one in the Error field.

The setting of this bit is applicable to non-PACKET devices only.

If bit 1 of word 8 is cleared to zero, then the device shall

a) disable support for non-zero buffer offsets in the DMA Setup FIS;
b) clear word 78 bits 1 and 4 in the IDENTIFY DEVICE data to zero;
c) clear word 79 bits 1 and 4in the in the IDENTIFY DEVICE data to zero; and
d) if non-zero buffer offsets in the DMA Setup FIS are disabled, the device shall only issue a DMA Setup FIS that has the DMA Buffer Offset field cleared to zero.

The setting of this bit is applicable to non-PACKET devices only.

If bit 2 of word 8 is cleared to zero, then the device shall:

a) disable support for receiving host initiated power management requests and shall not support device initiated power management requests;
b) clear word 76 bit 9 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data to zero;
c) clear word 78 bit 3 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data to zero;
d) clear word 79 bit 3 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data to zero; and
e) if interface power management requests are disabled, the device shall respond with $PMNAK_P$ to any interface power management requests and the device shall not issue $PMREQ\_P_P$ or $PMREQ\_S_P$ to the host.

If bit 3 of word 8 is cleared to zero, then the device shall:

a) disable support for asynchronous notification;
b) clear word 78 bit 5 of IDENTIFY PACKET DEVICE data to zero;
c) clear word 79 bit 5 of IDENTIFY PACKET DEVICE data to zero; and
d) when asynchronous notification is disabled, the device shall not initiate a Set Device Bits FIS with the Notification bit set to one.

If bit 4 of word 8 is cleared to zero, then the device shall:

a) disable support for software settings preservation;
b) clear word 78 bit 6 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data to zero;
c) clear word 79 bit 6 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data to zero; and
d) when software settings preservation is disabled, the device shall not preserve any software settings that are normally cleared following a COMRESET.

If bit 5 of word 8 is cleared to zero, then the device shall:

a) disable support for Automatic Partial to Slumber transitions;
b) clear word 76 bits 13 and 14 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data to zero;
c) clear word 79 bit 7 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE data to zero; and
d) when Automatic Partial to Slumber transitions are disabled neither the device nor host may transition to Slumber from Partial without first entering Active.

If bit 5 of word 8 is set to one and bit 2 or word 8 is cleared to zero, then the device shall return command aborted.

If bit 6 of word 8 is cleared to zero, then the device shall:

a) disable support for the NCQ QUEUE MANAGEMENT command;
b) clear word 77 bit 5 of IDENTIFY DEVICE data to zero; and
c) if an NCQ QUEUE MANAGEMENT command is issued to the device, the device shall return command aborted.

Bits (15:7) of word 8 are reserved and shall be cleared to zero.

WORD 9: Reserved for Serial ATA

This word is reserved for Serial ATA and all bits shall be cleared to zero.

13.5 Software Settings Preservation (Optional)

When a device is enumerated, software configures the device using SET FEATURES and other commands. These software settings are often preserved across software reset but not necessarily across COMRESET. In Parallel ATA, only commanded hardware resets may occur, thus legacy mode software only reprograms settings that are cleared for the particular type of reset it has issued. In Serial ATA, COMRESET is equivalent to hardware reset and a non-commanded COMRESET may occur if there is an asynchronous loss of signal. Since COMRESET is equivalent to hardware reset, in the case of an asynchronous loss of signal some software settings may be lost without legacy mode software knowledge. In order to avoid losing important software settings without legacy mode driver knowledge, the software settings preservation ensures that the value of important software settings is maintained across a COMRESET. Software settings preservation may be enabled or disabled using SET FEATURES with a subcommand code of 06h (refer to section 13.3.6). If a device supports software settings preservation, the feature shall be enabled by default.

The software settings that shall be preserved across COMRESET are listed below. The device is only required to preserve the indicated software setting if it supports the particular feature/command the setting is associated with.

INITIALIZE DEVICE PARAMETERS: Device settings established with the INITIALIZE DEVICE PARAMETERS command. This command is obsolete in the ATA8-ACS standard, and was last defined in the ATA/ATAPI-5 standard.

Power Management Feature Set Standby Timer: The Standby timer used in the Power Management feature set.

Read/Write Stream Error Log: The Read Stream Error Log and Write Stream Error Logs (accessed using the GPL feature set commands).

Security mode state: The security mode state established by Security Mode feature set commands (refer to the ATA8-ACS standard). The device shall not transition to a different security mode state based on a COMRESET. For example, the device shall not transition from the SEC5: Unlocked / not Frozen state to state SEC4: Security enabled / Locked when a COMRESET occurs, instead the device shall remain in the SEC5: Unlocked / not Frozen state.

SECURITY FREEZE LOCK: The Frozen mode setting established by the SECURITY FREEZE LOCK command.

SECURITY UNLOCK: The unlock counter that is decremented as part of a failed SECURITY UNLOCK command attempt.

SET MAX ADDRESS (EXT): The maximum LBA specified in SET MAX ADDRESS or SET MAX ADDRESS EXT.

SET FEATURES (Write Cache Enable/Disable): The write cache enable/disable setting established by the SET FEATURES command with subcommand code of 02h or 82h.

SET FEATURES (Set Transfer Mode): PIO, Multiword, and UDMA transfer mode settings established by the SET FEATURES command with subcommand code of 03h.

SET FEATURES (Advanced Power Management Enable/Disable): The advanced power management enable/disable setting established by the SET FEATURES command with subcommand code of 05h or 85h. The advanced power management level established in the Count(7:0) register when advanced power management is enabled (SET FEATURES subcommand code 05h) shall also be preserved.

SET FEATURES (Read Look-Ahead): The read look-ahead enable/disable setting established by the SET FEATURES command with subcommand code of 55h or AAh.

SET FEATURES (Release Interrupt): The release interrupt enable/disable setting established by the SET FEATURES command with a subcommand code of 5Dh or DDh.

SET FEATURES (SERVICE Interrupt): The SERVICE interrupt enable/disable setting established by the SET FEATURES command with a subcommand code of 5Eh or DEh.

SET FEATURES (Reverting to Defaults): The reverting to power-on defaults enable/disable setting established by the SET FEATURES command with a subcommand code of CCh or 66h.

SET MULTIPLE MODE: The block size established with the SET MULTIPLE MODE command.

NCQ QUEUE MANAGEMENT (Deadline Handling): The state of WDNC and RDNC.

Write-Read-Verify feature set: The contents of IDENTIFY DEVICE data word 120 bit 1, words 210-211, and word 220 bits (7:0). The device shall not return to its Write-Read-Verify factory default setting after processing a COMRESET.

13.5.1 Warm Reboot Considerations (Informative)

During a system reboot, the security settings maintained by software settings preservation may cause an error condition. Some system implementations choose to reboot the system by sending a COMRESET to the device. When the device has software settings preservation enabled, the security settings remain in the Unlock / Frozen State (SEC6). When in the Unlock / Frozen State (SEC6), system software sending a SECURITY UNLOCK command with the password is aborted by the device. System software should implement recommendations in this section to avoid the password entered by the user being aborted.

It is recommended that system software not prompt for the user password during a warm reboot. If system software detects that the device is in the SEC5 state during the warm reboot, the SECURITY FREEZE command may be issued by system software to enter the Unlock / Frozen State (SEC6).

The SEC states and SECURITY commands are described in ATA8-ACS. Note that these recommendations do not apply for external SATA devices.

13.6 Native Command Queuing (Optional)

This section defines a simple and streamlined command queuing model for Serial ATA.

The native queuing definition utilizes the reserved 32-bit field in the Set Device Bits FIS to convey the pending status for each of up to 32 outstanding commands. The BSY bit in the Status register conveys only the device's readiness to receive another command, and does not convey the completion status of queued commands. Upon receipt of a new command, the device clears the BSY bit to zero before proceeding to execute received commands. The 32 protocol specific bits in

the Set Device Bits FIS are handled as a 32-element array of active command bits (referred to as ACT bits), one for each possible outstanding command, and the array is bit significant such that bit “n” in the array corresponds to the pending status of the command with tag “n.”

Data returned by the device (or transferred to the device) for queued commands use the First Party DMA mechanism to cause the host controller to select the appropriate destination/source memory buffer for the transfer. The memory handle used for the buffer selection is the same as the tag that is associated with the command. For traditional desktop host controllers, the handle may be used to index into a vector of pointers to pre-constructed scatter/gather lists (often referred to as physical region descriptor tables or simply PRD tables) in order to establish the proper context in the host’s DMA engine. The First-party DMA Data Phase is defined as the period from reception of a DMA Setup FIS until either the associated transfer count is exhausted or the ERR bit in the shadow Status register is set. During this period the host may not issue new commands to the device nor may the device signal new command completions to the host.

Status is returned by updating the 32-element bit array in the Set Device Bits FIS for successful completions. For failed commands, the device halts processing commands allowing host software or controller firmware to intervene and resolve the source of the failure, by using the general purpose logging feature set, before processing is again explicitly restarted.

Devices supporting Native Command Queuing shall implement, and report support for, the general purpose logging feature set as defined in ATA8-ACS. In addition, the device shall implement the Queued Error Log.

13.6.1 Definition

13.6.1.1 Command Issue Mechanism

The Serial ATA transmission protocol is sensitive to the state of the BSY bit in the Shadow Status register that provides write protection to the shared Shadow Command Block registers. Since the Shadow Command Block registers may be safely written only when the BSY bit is cleared to zero, the BSY bit conventions defined in the Transport layer shall be adhered to, and issuing a new command shall only be attempted when the BSY bit is cleared to zero. When the BSY bit in the Shadow Status register is cleared to zero, another command may be issued to the device.

The state of the BSY bit in the Shadow Status register shall be checked prior to attempting to issue a new queued command. If the BSY bit is set to one, issuing the next command shall be deferred until the BSY bit is cleared to zero. It is desirable to minimize such command issue deferrals, so devices should clear the BSY bit to zero in a timely manner. Host controllers may have internal designs that mitigate the need for host software to block on the state of the BSY bit.

The native queuing commands include a tag value that identifies the command. The tag value is in the range 0 through 31 inclusive, and is conveyed in the Register – Host to Device FIS when the command is issued. For devices that report a value less than 31 in their IDENTIFY DEVICE word 75, the host shall issue only unique tag values that are less than or equal to the value reported.

Upon issuing a new native queued command, the bit in the SActive register corresponding to the tag value of the command being issued shall be set to one by the HBA prior to the command being transmitted to the device. Section 14.1.4 describes the SActive register and the access conventions for it.

Upon accepting the command, the device shall clear the BSY bit to zero when it is prepared to receive another command by transmitting a Register – Device to Host FIS to the host with the BSY bit cleared in the Status field of the FIS, and the Interrupt bit cleared to zero.

13.6.1.2 Data Delivery Mechanism

The First-party DMA mechanism is used by the device to transmit (or receive) data for an arbitrary queued command. The command's tag value shall also be the DMA Buffer Identifier used to uniquely identify the source/destination memory buffer for the transfer.

The DMA Setup FIS is used by the device to select the proper transfer buffer prior to each data transfer. Only a single DMA Setup FIS is required at the beginning of each transfer and if the transfer spans multiple Data FISes a new DMA Setup FIS is not required before each Data FIS. Serial ATA host controller hardware shall account for the DMA Setup FIS buffer identifier being a value between 0 and 31 and the host controller shall select the proper transfer buffer based on such an index.

For data transfers from the host to the device, an optimization to the First-party DMA mechanism is included to eliminate one transaction by allowing the requested data to immediately be transmitted to the device following such a request without the need for a subsequent DMA Activate FIS for starting the flow of data. This optimization to the First-party DMA mechanism is defined in section 10.3.8.3.1.

If non-zero buffer offsets in the DMA Setup FIS are not enabled (see section 13.3.1) or not supported (see section 13.2.1), the data transfer for a command shall be satisfied to completion following a DMA Setup FIS before data transfer for a different command may be started. Host controllers are not required to preserve DMA engine context upon receipt of a new DMA Setup FIS, and if non-zero buffer offsets are not enabled or not supported, a device cannot resume data transfer for a previously abandoned context at the point where it left off.

If the host controller hardware supports non-zero buffer offsets in the DMA Setup FIS and use of non-zero offsets is enabled, and if guaranteed in-order data delivery is either not supported by the device (see section 13.2.1) or is disabled (see section 13.3.4), the device may return (or receive) data for a given command out of order (i.e. returning data for the last half of the command first). In this case the device may also interleave partial data delivery for multiple commands provided the device keeps track of the appropriate buffer offsets. For example, data for the first half of command 0 may be delivered followed by data for the first half of command 1 followed by the remaining data for command 0. By default use of non-zero buffer offsets is disabled. See section 13.3.1 for information on enabling non-zero buffer offsets for the DMA Setup FIS.

If the host controller hardware supports non-zero buffer offsets in the DMA Setup FIS and use of non-zero offsets is enabled, and if the device supports guaranteed in-order data delivery and guaranteed in-order data delivery is enabled, the device may use multiple DMA Setup FISes to satisfy a particular I/O process, but if multiple DMA Setup FISes are used, the data shall be delivered in-order, starting at the first LBA. In this case the device may not interleave partial data delivery for either individual or multiple commands. For example, data for the first half of a command may be delivered using one DMA Setup FIS and one or more subsequent Data FISes, followed by the remaining data for that command, delivered using a second DMA Setup FIS and one or more subsequent Data FISes. Non-zero buffer offsets are used as in the more general out-of-order data delivery case described above. By default use of guaranteed in-order data delivery is disabled.

For selecting the memory buffer for data transfers, the DMA Setup FIS is issued by the device. The DMA Setup FIS fields are defined in Figure 204 (see also section 10.3.8).

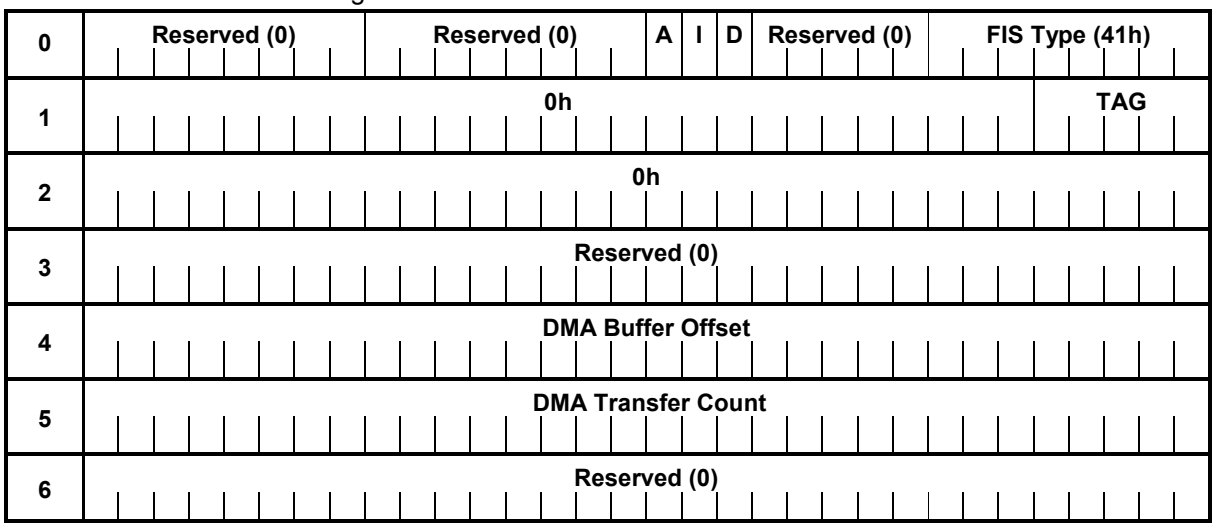


Figure 204 – DMA Setup FIS definition for memory buffer selection

Field Definitions

FIS Type — As defined in section 10.3.8.

D — As defined in section 10.3.8. Since the DMA Setup FIS is only issued by the device for the queuing model defined here, the value in the field is defined as 1 = device to host transfer (write to host memory), 0 = host to device transfer (read from host memory).

A — As defined in section 10.3.8, including additional details in section 10.3.8.3.1. For DMA Setup with transfer direction from device to host, this bit shall be zero.

TAG — This field is used to identify the DMA buffer region in host memory to select for the data transfer. The low order 5 bits of the DMA Buffer Identifier Low field shall be set to the TAG value corresponding to the command TAG for which data is being transferred. The remaining bits of the DMA Buffer Identifier Low/High shall be cleared to zero. The 64-bit Buffer Identifier field defined in the DMA Setup FIS in section 10.3.8 is used to convey a TAG value that occupies the five least-significant bits of the field.

DMA Buffer Offset
As defined in section 10.3.8. The device may specify a non-zero value in this field only if the host indicates support for it through the SET FEATURES mechanism defined in section 13.3. Data is transferred to/from sequentially increasing logical addresses starting at the specified offset in the specified buffer.

DMA Transfer Count
As defined in section 10.3.8. The value shall accurately reflect the length of the data transfer to follow. Refer to section 10.3.11.2 for special considerations if the transfer count is for an odd number of words. Devices shall not set this field to 0h; a value of 0h for this field is illegal and results in indeterminate behavior.

I Interrupt — Native Command Queuing does not make use of an interrupt following the data transfer phase (after the transfer count is exhausted). The Interrupt bit shall be cleared to zero.

R/Reserved — All reserved fields shall be cleared to zero.

13.6.1.3 Status Return Mechanism

For maximum efficiency, the status return mechanism is not interlocked (does not include a handshake) while at the same time ensuring no status notifications are lost or overwritten (i.e. status notifications are race-free). The status return mechanism relies on an array of ACT bits – one ACT bit to convey the active status for each of the 32 possible outstanding commands, resulting in a 32-bit ACT status field. The 32-bit reserved field in the Set Device Bits FIS as defined in section 10.3.6 is defined as the SActive field and is used to convey command completion information for updating the ACT bit array. The zero bit position in the 32-bit field corresponds to the ACT bit for the command with tag value of zero. Host software shall check the SActive register (containing the ACT bit array) when checking status in order to determine which command(s) have completed since the last time the host processed a command completion. It is possible for multiple commands to indicate completion by the time the host checks the status due to the software latencies in the host (i.e. by the time the host responds to one completion notification, another command may have completed). Only successfully completed commands indicate their status using this mechanism – failed commands use an additional mechanism described in 13.6.3.3.1.1 and 13.6.3.7.1.2 to convey error information as well as the affected command tag. The Queued Error Log is used to convey additional queued command error information as outlined in 13.7.

13.6.1.4 Priority

Host knowledge of I/O priority may be transmitted to the device as part of the command. There are two priority values for NCQ commands, normal and high. When the host marks an NCQ command as high priority, the host is requesting a better quality of service for that command than commands issued with normal priority.

The classes are forms of soft priority. The device may choose to complete a normal priority command before an outstanding high priority command, although preference should be given to the high priority commands. One example where a normal priority command may be completed before a high priority command is when the normal priority command is a cache hit, whereas the high priority command requires access of the device media.

The priority class is specified in the PRIO bit for NCQ commands (READ FPDMA QUEUED and WRITE FPDMA QUEUED). This bit may specify either the normal priority or high priority value. If a command is marked by the host as high priority, the device should attempt to provide better quality of service for the command. It is not required that devices process all high priority requests before satisfying normal priority requests.

The device should complete high priority requests in a more timely fashion than normal and isochronous requests. The device should complete isochronous requests prior to its associated deadline.

13.6.1.5 Unload

When using Native Command Queuing in a laptop environment, the host needs to be able to park the head due to excessive movement (e.g. the laptop being dropped). This section defines a mechanism that the host may use to park the heads when NCQ commands are outstanding in the device. The typical time for completion of the unload operation is defined in ATA/ATAPI-7 clause 6.20.10.

When NCQ commands are outstanding, the device is able to accept the IDLE IMMEDIATE command with the Unload Feature defined in ATA/ATAPI-7 clause 6.20. Upon reception of this command with the Unload Feature specified, the device shall:

1. Unload/park the heads immediately.
2. Respond to the host with a Register – Device to Host FIS with the ERR bit set to one in the Status register since this is a non-queued command.

When the host receives the error indication, it should proceed to read the Queued Error Log (see 13.7). In the Queued Error Log, the device shall indicate whether the error was due to receiving an UNLOAD and whether the UNLOAD was executed. The device shall not load the heads to the media when reading the Queued Error Log.

The Queued Error Log indicates whether the device has accepted the Unload and is in the process of executing the command. To get a definitive indication of Unload completion (and success), the IDLE IMMEDIATE command with the Unload Feature needs to be issued again after the Queued Error Log has been read. After the Queued Error Log has been read, there are no NCQ commands outstanding and the NCQ error is cleared such that the IDLE IMMEDIATE command with the Unload Feature will be processed normally and a successful status will be returned when the unload process completes successfully.

There may be a delay in issuing the IDLE IMMEDIATE command with the Unload feature to the device if the device is currently performing a data transfer for a previously issued NCQ command. If the device happens to be executing extensive data error recovery procedures, this delay could be longer than acceptable. However, this same issue may occur when a non-queued data command is outstanding and the device is performing error recovery procedures.

13.6.2 Intermixing Non-Native Queued Commands and Native Queued Commands

The host shall not issue a non-native queued command while a native queued command is outstanding. Upon receiving a non-native queued command while a native queued command is outstanding, the device shall signal the error condition to the host by transmitting a Register FIS to the host with the ERR and ABRT bits set to one and the BSY bit cleared to zero in the Status field of the FIS and halt command processing as described in section 13.6.3.4 except as noted below. Non-native queued commands include all commands other than the READ FPDMA QUEUED, WRITE FPDMA QUEUED, and NCQ QUEUE MANAGEMENT commands defined in 13.6.3.1, 13.6.3.5 and 13.6.3.8.

Reception of a command to read the Queued Error Log (see 13.7) after an error has occurred shall cause any outstanding Serial ATA native queued commands to be aborted, and the device shall perform necessary state cleanup to return to a state with no commands pending. The device shall clear all bits in the SActive register by transmitting a Set Device Bits FIS to the host with all the bits in the SActive field set to one (i.e. FFFFFFFFh). After reading the Queued Error Log, the device shall be prepared to execute subsequently issued queued commands regardless of any previous errors on a queued command.

In the case that a command to read the Queued Error Log is issued while a native queued command is outstanding AND no error was previously reported by the device, the device shall signal an error condition. The receipt of this command when no error is outstanding shall be handled as any other non-native queued command when a native queued command is outstanding. In this case, a subsequent command to read the Queued Error Log is required to recover from the error.

13.6.3 Command Definitions

13.6.3.1 READ FPDMA QUEUED

Queued native read commands make use of a new command. The new command supports LBA mode only and uses 48-bit addressing only. The format of the new command is defined in Figure 205.

Register	7	6	5	4	3	2	1	0
Features(7:0)	Sector Count 7:0							
Features(15:8)	Sector Count 15:8							
Count(7:0)	TAG					Reserved		
Count(15:8)	PRIO(1:0)		Reserved					
LBA(7:0)	LBA 7:0							
LBA(31:24)	LBA 31:24							
LBA(15:8)	LBA 15:8							
LBA(39:32)	LBA 39:32							
LBA(23:16	LBA 23:16							
LBA(47:40)	LBA 47:40							
ICC	ICC(7:0)							
Device	FUA	1	Res	0	Reserved			
Command	60h							

Figure 205 – READ FPDMA QUEUED command definition

TAG — The TAG value shall be assigned by host software to be different from all other TAG values corresponding to outstanding commands. The assigned TAG value shall not exceed the value specified in IDENTIFY DEVICE word 75.

PRIO — The Priority (PRIO) value is assigned by the host based on the priority of the command issued. The device should complete high priority requests in a more timely fashion than normal and isochronous requests. The device should complete isochronous requests prior to its associated deadline.

- 00b Normal Priority
- 01b Isochronous – deadline dependent priority
- 10b High priority
- 11b Reserved

ICC — The Isochronous Command Completion (ICC) field is valid when PRIO is set to a value of 01b. It is assigned by the host based on the intended deadline associated with the command issued. When a deadline has expired, the device shall continue to complete the command as soon as possible. This behavior may be modified by the host if the device supports the NCQ QUEUE MANAGEMENT command (see 13.6.3.8) and supports the Deadline Handling subcommand (see 13.6.3.8.2). This subcommand allows the host to set whether the device shall abort (or continue processing) commands that have exceeded the time set in ICC.

There are several parameters encoded in the ICC field: Fine or Coarse timing, Interval and the Max Time. The Interval indicates the time units of the Time Limit parameter.

If ICC Bit 7 is cleared to zero, then

- The time interval is fine-grained.
- Interval = 10 msec
- Time Limit = (ICC[6:0] + 1) * 10 msec
- Max Fine Time = 128 * 10 msec = 1.28 sec

If ICC Bit 7 is set to one (coarse encoding), then

- The time interval is coarse-grained
- Interval = 0.5 sec;
- Time Limit = (ICC[6:0] + 1) * 0.5 sec
- Max Coarse Time = 128 * 0.5 sec = 64 sec

FUA — When set to one forces the data to be retrieved from the storage media regardless of whether the storage device holds the requested information in its buffers or cache. If the device holds a modified copy of the requested data as a result of having cached writes, the modified data is first written to the media before being retrieved from the storage media as part of this operation. When cleared to zero the data may be retrieved either from the device's storage media or from buffers/cache that the device may include.

Others — All other registers have contents consistent with the READ DMA QUEUED EXT command defined in the ATA8-ACS standard, including the Sector Count 15:0 convention where a value of zero specifies that 65,536 sectors are to be transferred.

13.6.3.2 Success Outputs

Upon successful completion of one or more outstanding commands, the device shall transmit a Set Device Bits FIS with the Interrupt bit set to one and one or more bits set to one in the ACT field corresponding to the bit position for each command TAG that has completed since the last status notification was transmitted. The ERR bit in the Status register shall be cleared to zero and the value in the Error register shall be zero.

The ACT field occupies the last 32 bits of the Set Device Bits FIS as defined in Figure 190 below.

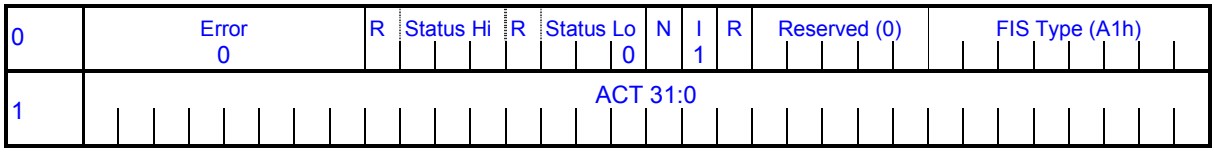


Figure 190 – Set Device Bits FIS for successful READ FPDMA QUEUED command completion

ACT — The ACT field of the Set Device Bits FIS communicates successful completion notification for each of up to 32 queued commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating successful completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns.

Error — The Error register shall be cleared to zero.

Status — As defined in section 10.3.6. The ERR bit shall be cleared to zero indicating successful command completion.

I Interrupt bit. The interrupt bit shall be set to one.

All other fields as defined in section 10.3.6.

NOTE: Devices should be aware that if choosing to aggregate status to the point where many of the outstanding commands have actually completed successfully without notification to the host, that an error may cause the final completion status of those commands to be failure. A device should be selective when using status aggregation for outstanding queued commands to ensure the host is made aware of successful completion for outstanding commands in a way that an error would not force a high number of unnecessary command retries.

13.6.3.3 Error Outputs

13.6.3.3.1.1 Upon receipt of a command

If the device has received a command that has not yet been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Register FIS (see Figure 206) to the host with the ERR bit set to one and the BSY bit cleared to zero in the Status field, the ATA error code in the Error field.

Register	7	6	5	4	3	2	1	0
Error	ERROR							
Count(7:0)	na							
Count(15:8)	na							
LBA(7:0)	na							
LBA(31:24)	na							
LBA(15:8)	na							
LBA(39:32)	na							
LBA(23:16	na							
LBA(47:40)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Figure 206 – READ FPDMA QUEUED error on command receipt

ERROR	ATA error code for the failure condition of the failed command
BSY	0
DRDY	1
DF	0
DRQ	0
ERR	1

Following transmission of the Register FIS, the device shall stop processing any outstanding or new commands until the Queued Error Log (see 13.7) has been read before continuing to abort all outstanding commands.

13.6.3.3.1.2 During execution of a command

If all commands have been acknowledged by clearing the BSY bit to zero and an error condition is detected, the device shall transmit a Set Device Bits FIS (see Figure 191 below) to the host with the ERR bit set to one in the Status field, the ATA error code in the Error field, and the Interrupt bit set to one. All outstanding commands at the time of an error shall be aborted as part of the error response and may be re-issued as appropriate by the host. For any commands that

have not completed successfully or have resulted in error, the device shall clear the corresponding ACT bits to zero in the Set Device Bits FIS.

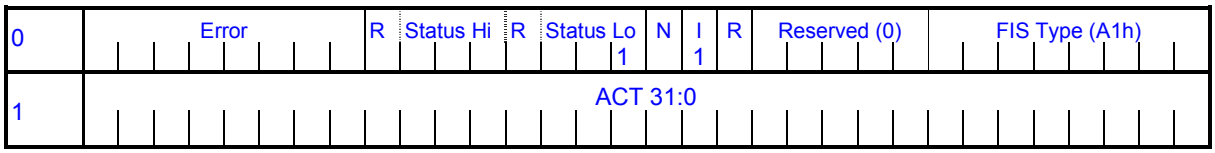


Figure 207 - Set Device Bits FIS with error notification, and command completions

SActive — The ACT field of the Set Device Bits FIS communicates successful completion notification for each of up to 32 queued commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating successful completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns.

Error — The Error register shall contain the ATA error code.

Status — As defined in section 10.3.6. The ERR bit shall be set to one indicating an NCQ error has occurred.

I — Interrupt bit. The interrupt bit shall be set to one.

All other fields as defined in section 10.3.6.

Only the registers that are updated as part of the Set Device Bits FIS are modified if the device signals an error condition when the BSY bit in the Shadow Status register is cleared to zero, leaving the other Shadow Register Block Registers unchanged. If the device signals an error condition when the BSY bit in the shadow Status register is set to one, the device clears the BSY bit to zero with a Register FIS which updates all registers in the Shadow Register Block, but the corresponding error information for the command is still retrieved by reading the Queued Error Log.

Following transmission of the Set Device Bits FIS, the device shall stop processing any outstanding or new commands until the Queued Error Log has been read before continuing to abort all outstanding commands. See 13.6.3.4 for more details.

13.6.3.4 Queue abort

Following transmission of the Register FIS or Set Device Bits FIS in response to an NCQ error condition, the device shall stop processing any outstanding or new commands until the Queued Error Log (see 13.7) is read using the General Purpose Logging (GPL) feature set.

When a command to read the Queued Error Log is received, the device shall perform any necessary cleanup before returning detailed error information for the last failed command including the tag value for the failed command as described in 13.7.

In response to a received command to read the Queued Error Log the device shall transmit a Set Device Bits FIS (see Figure 208 below) to the host with all the bits in the ACT field set to one. This policy avoids the host inadvertently completing a failed command with successful status. The exception to this policy is if the host reads the Queued Error Log for information which is not directly tied to a specific error reported by the device. In the case where a device receives a command to read the Queued Error Log which is not in direct response to an error reported by the device as well as no queued commands being outstanding, it is not required that a Set Device Bits FIS is delivered in response as it is not a necessity to abort any commands at that time.

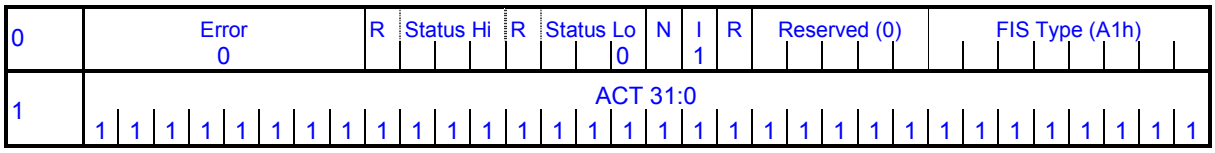


Figure 208 - Set Device Bits FIS aborting all outstanding command

ACT — The entire ACT field shall be set to one as an indication that all outstanding commands are being aborted.

Error — The Error register shall be cleared to zero.

Status — As defined in 10.3.6. The ERR bit shall be cleared to zero indicating clean up of all previously outstanding commands.

I — Interrupt bit. The interrupt bit shall be set to one.

All other fields as defined in 10.3.6.

When an error is indicated, the host shall treat any outstanding commands that do not have their corresponding SActive register bit cleared as failed.

13.6.3.5 WRITE FPDMA QUEUED

Queued native write commands make use of a new command. The format of the new command is defined in Figure 209.

Register	7	6	5	4	3	2	1	0
Features(7:0)	Sector Count 7:0							
Features(15:8)	Sector Count 15:8							
Count(7:0)	TAG					Reserved		
Count(15:8)	PRIO(1:0)		Reserved					
LBA(7:0)	LBA 7:0							
LBA(31:24)	LBA 31:24							
LBA(15:8)	LBA 15:8							
LBA(39:32)	LBA 39:32							
LBA(23:16	LBA 23:16							
LBA(47:40)	LBA 47:40							
ICC	ICC(7:0)							
Device	FUA	1	0	0	Reserved			
Command	61h							

Figure 209 – WRITE FPDMA QUEUED command definition

TAG — The TAG value shall be assigned by host software to be different from all other TAG values corresponding to outstanding commands. The assigned TAG value shall not exceed the value specified in IDENTIFY DEVICE word 75.

PRIO — The Priority (PRIO) value is assigned by the host based on the priority of the command issued. The device should complete high priority requests in a more timely fashion than normal and isochronous requests. The device should complete isochronous requests prior to its associated deadline.

00b	Normal Priority
01b	Isochronous – deadline dependent priority
10b	High priority
11b	Reserved

ICC — The Isochronous Command Completion (ICC) field is valid when PRIO is set to a value of 01b. It is assigned by the host based on the intended deadline associated with the command issued. When a deadline has expired, the device shall continue to complete the command as soon as possible. This behavior may be modified by the host if the device supports the NCQ QUEUE MANAGEMENT command (see 13.6.3.8) and supports the Deadline Handling subcommand (see 13.6.3.8.2). This subcommand allows the host to set whether the device shall abort (or continue processing) commands that have exceeded the time set in ICC.

There are several parameters encoded in the ICC field: Fine or Coarse timing, Interval and the Max Time. The Interval indicates the time units of the Time Limit parameter.

If ICC Bit 7 is cleared to zero, then

- The time interval is fine-grained.
- Interval = 10 msec

- Time Limit = (ICC[6:0] + 1) * 10 msec
- Max Fine Time = 128 * 10 msec = 1.28 sec

If ICC Bit 7 is set to one (coarse encoding), then

- The time interval is coarse-grained
- Interval = 0.5 sec;
- Time Limit = (ICC[6:0] + 1) * 0.5 sec
  Max Coarse Time = 128 * 0.5 sec = 64 sec

FUA — When set to one forces the data to be written to the storage media before completion status is indicated. When cleared to zero the device may indicate completion status before the data is committed to the media.

Others — All other registers as specified for the WRITE DMA QUEUED EXT command defined in the ATA8-ACS standard, including the Sector Count 15:0 convention where a value of zero specifies that 65,536 sectors are to be transferred.

13.6.3.6 Success Outputs

Upon successful completion of one or more outstanding commands, the device shall transmit a Set Device Bits FIS with the Interrupt bit set to one and one or more bits set to one in the ACT field corresponding to the bit position for each command TAG that has completed since the last status notification was transmitted. The ERR bit in the Status register shall be cleared to zero and the value in the Error register shall be zero.

The ACT field occupies the last 32 bits of the Set Device Bits FIS as defined in Figure 210 below.

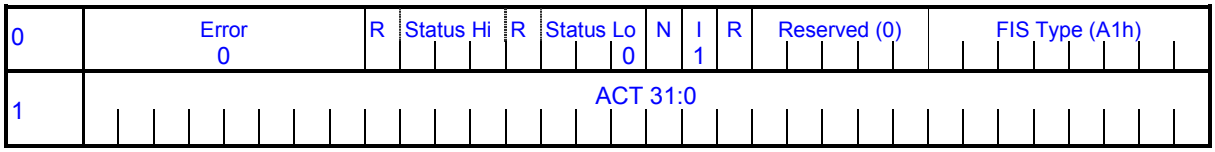


Figure 210 - Set Device Bits FIS for successful WRITE FPDMA QUEUED command completion

ACT — The ACT field of the Set Device Bits FIS communicates successful completion notification for each of up to 32 queued commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating successful completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns.

Error — The Error register shall be cleared to zero.

Status — As defined in 10.3.6. The ERR bit shall be cleared to zero indicating successful command completion.

I — Interrupt bit. The interrupt bit shall be set to one.

All other fields as defined in 10.3.6.

NOTE: Devices should be aware that if choosing to aggregate status to the point where many of the outstanding commands have actually completed successfully without notification to the host, that an error may cause the final completion status of those commands to be failure. A device should be selective when using status aggregation for outstanding queued commands to ensure the host is made aware of successful completion for outstanding commands in a way that an error would not force a high number of unnecessary command retries.

13.6.3.7 Error Outputs

13.6.3.7.1.1 Upon receipt of a command

If the device has received a command that has not yet been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Register FIS (see Figure 211 below) to the host with the ERR bit set to one and the BSY bit cleared to zero in the Status field, the ATA error code in the Error field.

Register	7	6	5	4	3	2	1	0
Error	ERROR							
Count(7:0)	na							
Count(15:8)	na							
LBA(7:0)	na							
LBA(31:24)	na							
LBA(15:8)	na							
LBA(39:32)	na							
LBA(23:16	na							
LBA(47:40)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Figure 211 – WRITE FPDMA QUEUED error on command receipt

ERROR	ATA error code for the failure condition of the failed command
BSY	0
DRDY	1
DF	0
DRQ	0
ERR	1

Following transmission of the Register FIS, the device shall stop processing any outstanding or new commands until the Queued Error Log (see 13.7) has been read before continuing to abort all outstanding commands. See 13.6.3.4 for more details.

13.6.3.7.1.2 During execution of a command

If all commands have been acknowledged by clearing the BSY bit to zero and an error condition is detected, the device shall transmit a Set Device Bits FIS (see Figure 212 below) to the host with the ERR bit set to one in the Status field, the ATA error code in the Error field, and the Interrupt bit set to one. All outstanding commands at the time of an error shall be aborted as part of the error response and may be re-issued as appropriate by the host. For any commands that have not completed successfully or have resulted in error, the device shall clear the corresponding ACT bits to zero in the Set Device Bits FIS.

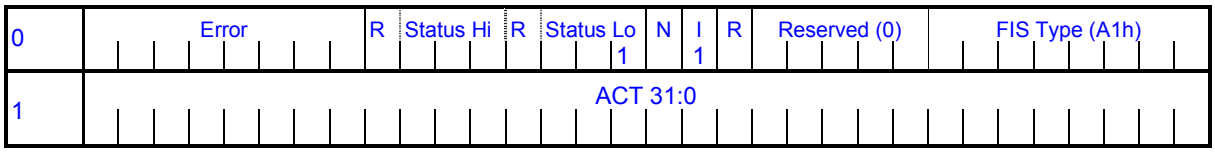


Figure 212 - Set Device Bits FIS with error notification, and command completions

ACT — The ACT field of the Set Device Bits FIS communicates successful completion notification for each of up to 32 queued commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating successful completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns.

Error — The Error register shall contain the ATA error code.

Status — As defined in 10.3.6. The ERR bit shall be set to one indicating an NCQ error has occurred.

I — Interrupt bit. The interrupt bit shall be set to one.

All other fields as defined in 10.3.6.

Only the registers that are updated as part of the Set Device Bits FIS are modified if the device signals an error condition when the BSY bit in the shadow Status register is cleared to zero, leaving the other Shadow Register Block Registers unchanged. If the device signals an error condition when the BSY bit in the shadow Status register is set to one, the device clears the BSY bit to zero with a Register FIS which updates all registers in the Shadow Register Block.

Following transmission of the Set Device Bits FIS, the device shall stop processing any outstanding or new commands until the Queued Error Log (see 13.7) has been read before continuing to abort all outstanding commands. See 13.6.3.4 for more details.

13.6.3.8 NCQ QUEUE MANAGEMENT

The NCQ Queue Management feature allows the host to manage the outstanding NCQ commands and/or affect the processing of NCQ commands.

The NCQ QUEUE MANAGEMENT command is a non-data NCQ command. Only specified NCQ QUEUE MANAGEMENT subcommands are executed as Immediate NCQ commands.

If NCQ is disabled and an NCQ QUEUE MANAGEMENT command is issued to the device, then the device shall abort the command with the ERR bit set to one in the Status register and the ABRT bit set to one in the Error register. This command is prohibited for devices that implement the PACKET feature set. The queueing behavior of the device depends on which subcommand is specified.

Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Features(7:0)	Subcommand Specific				Subcommand			
Features(15:8)	Reserved							
Count(7:0)	TAG					Reserved		
Count(15:8)	Reserved							
LBA(7:0)	Subcommand Specific (TTAG)					Reserved		
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	Res	1	Res	0	Reserved			
Command	63h							

Figure 213 - NCQ QUEUE MANAGEMENT - Command definition

Table 80 defines the Subcommand values. If an invalid subcommand is specified, then the device shall abort the command with the ERR bit set to one in the Status register, the ABRT bit set to one in the Error register, and shall cause all outstanding commands to be aborted.

Table 80 - Subcommand Field

Subcommand	Description	Reference
0h	Abort NCQ queue	13.6.3.8.1
1h	Deadline Handling	13.6.3.8.2
2h - Fh	Reserved	

TAG The TAG value shall be assigned by host software to be different from all other TAG values corresponding to outstanding commands. The assigned TAG value shall not exceed the value specified in IDENTIFY DEVICE word 75.

Subcommand Specific (TTAG) is the selected queue TAG. This allows the host to select the specific outstanding queued command to be managed.

The error and normal returrns for this command are subcommand specific.

13.6.3.8.1 ABORT NCQ QUEUE Subcommand (0h)

A Subcommand set to 0h specifies the ABORT NCQ QUEUE subcommand. The ABORT NCQ QUEUE subcommand is an immediate NCQ command. Support for this subcommand is indicated in the NCQ Queue Management Log (see 13.7.5).

The ABORT NCQ QUEUE subcommand shall affect only those NCQ commands for which the device has indicated command acceptance before accepting this NCQ QUEUE MANAGEMENT command

The format of the command is defined in Figure 181.

Register	7	6	5	4	3	2	1	0
Features(7:0)	Abort Type				0h			
Features(15:8)	Reserved							
Count(7:0)	TAG					Reserved		
Count(15:8)	Reserved							
LBA(7:0)	TTAG					Reserved		
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	Res	1	Res	0	Reserved			
Command	63h							

Figure 181 – NCQ QUEUE MANAGEMENT, Abort NCQ Queue - Command definition

Abort Type describes the action requested. Table 81 shows the defined abort types. The NCQ Queue Management Log (see 13.7.5) provides a list of abort types supported by the device.

TAG The TAG value shall be assigned by host software to be different from all other TAG values corresponding to outstanding commands. TAG shall not exceed the value specified in IDENTIFY DEVICE word 75.

TTAG The TTAG field contains the value of the TAG of the outstanding command that is requested to be aborted.The TTAG value is only valid when the Abort Type field is set to 3h (Abort Selected). TTAG shall not exceed the value specified in IDENTIFY DEVICE word 75.

Table 81 - Abort Type

Abort Type	Abort Type	Description
0h	Abort All	The device shall attempt to abort all outstanding NCQ commands.
1h	Abort Streaming	The device shall attempt to abort all outstanding NCQ Streaming commands. All non-streaming NCQ commands shall be unaffected.
2h	Abort Non-Streaming	The device shall attempt to abort all outstanding NCQ Non-Streaming commands. All NCQ Streaming commands shall be unaffected.
3h	Abort Selected	The device shall attempt to abort the outstanding NCQ command associated with the tag represented in TTAG field.
4h - Fh		Reserved

13.6.3.8.1.1 Success Outputs

If a supported Abort Type parameter is specified, then the device shall indicate success, even if the command results in no commands being aborted.

When an Abort NCQ Queue command completes successfully, a Set Device Bits FIS shall be sent to the host to complete the Abort subcommand and commands that were aborted as a consequence of the Abort subcommand by setting the ACT bits for those commands to one. This SDB FIS may also indicate other completed commands.

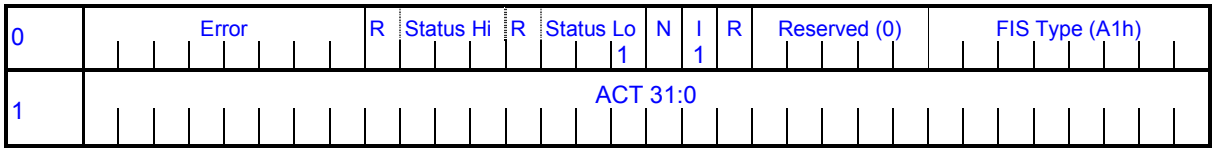


Figure 214 – NCQ QUEUE MANAGEMENT, Abort NCQ Queue - Successful completion

ACT — The ACT field of the Set Device Bits FIS communicates completion notification for each of up to 32 commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns. The device shall set the appropriate bit to one for each queued command that has been aborted, and shall set to one the bit associated with the TAG value for the Abort NCQ Queue command.

Error — The Error register shall contain 00h.

Status — As defined in 10.3.6. The ERR bit shall be cleared to zero.

I — Interrupt bit. The interrupt bit shall be set to one.

All other fields as defined in 10.3.6.

13.6.3.8.1.2 Error Outputs

13.6.3.8.1.2.1 Upon receipt of a command

If the value of the TTAG field equals the value of the TAG field, or if an unsupported Abort type parameter is specified, the device shall return command aborted.

If the device has received a command that has not yet been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Register FIS (see Figure 215 below) to the host with the ERR bit set to one and the BSY bit cleared to zero in the Status field, the ATA error code in the Error field.

Register	7	6	5	4	3	2	1	0
Error	ERROR							
Count(7:0)	na							
Count(15:8)	na							
LBA(7:0)	na							
LBA(31:24)	na							
LBA(15:8)	na							
LBA(39:32)	na							
LBA(23:16)	na							
LBA(47:40)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Figure 215 – NCQ QUEUE MANAGEMENT, Abort NCQ Queue - error on command receipt

ERROR	ATA error code for the failure condition of the failed command
BSY	0
DRDY	1
DF	0
DRQ	0
ERR	1

Following transmission of the Register FIS, the device shall stop processing any outstanding or new commands until the Queued Error Log (see 13.7) has been read before continuing to abort all outstanding commands. See 13.6.3.4 for more details.

13.6.3.8.1.2.2 During execution of a command

If all commands have been acknowledged by clearing the BSY bit to zero and an error condition is detected, the device shall transmit a Set Device Bits FIS (see Figure 212 below) to the host with the ERR bit set to one in the Status field, the ATA error code in the Error field, and the Interrupt bit set to one. All outstanding commands at the time of an error are aborted as part of the error response and may be re-issued as appropriate by the host. For any commands that have not completed successfully or have resulted in error, the device shall clear the corresponding ACT bits to zero in the Set Device Bits FIS.

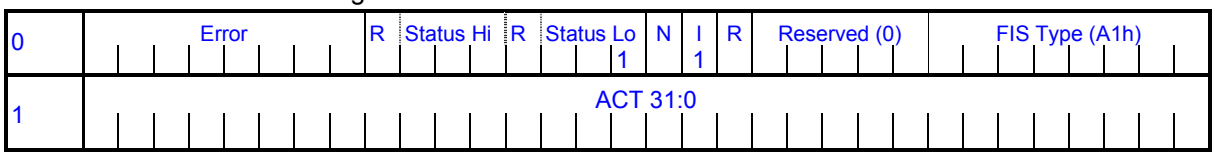


Figure 216 - NCQ QUEUE MANAGEMENT, Abort NCQ Queue – error during execution

ACT	The ACT field of the Set Device Bits FIS communicates successful completion notification for each of up to 32 queued commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating successful completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns.
Error	The Error register shall contain the ATA error code.
Status	As defined in 10.3.6. The ERR bit shall be set to one indicating an NCQ error has occurred.
I	Interrupt bit. The interrupt bit shall be set to one.

All other fields as defined in 10.3.6.

Only the registers that are updated as part of the Set Device Bits FIS are modified if the device signals an error condition when the BSY bit in the shadow Status register is cleared to zero, leaving the other Shadow Register Block Registers unchanged. If the device signals an error condition when the BSY bit in the shadow Status register is set to one, the device clears the BSY bit to zero with a Register FIS which updates all registers in the Shadow Register Block.

Following transmission of the Set Device Bits FIS, the device shall stop processing any outstanding or new commands until the Queued Error Log (see 13.7) has been read before continuing to abort all outstanding commands. See 13.6.3.4 for more details.

13.6.3.8.2 Deadline Handling Subcommand (1h)

A Subcommand set to 1h specifies the Deadline Handling Subcommand This subcommand controls how NCQ Streaming commands are processed by the device. Support for this subcommand is indicated in the NCQ Queue Management Log (see 13.7.5). The format of the command is defined in Figure 184.

Register	7	6	5	4	3	2	1	0
Features(7:0)	Reserved		RDNC	WDNC	1h			
Features(15:8)	Reserved							
Count(7:0)	TAG					Reserved		
Count(15:8)	Reserved							
LBA(7:0)	Reserved							
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	Res	1	Res	0	Reserved			
Command	63h							

Figure 184 – NCQ QUEUE MANAGEMENT, Deadline Handling - Command definition

WDNC — If the WDNC (Write Data Not Contiunue) bit is cleared to zero, then the device may allow WRITE FPDMA QUEUED command completion times to exceed what the ICC parameter specified. If the WDNC bit is set to one, then the all WRITE FPDMA QUEUED commands shall be completed by the the time specified by the ICC timer value, othewise the device shall return command aborted for all outstanding commands. WDNC is only applicable to WRITE FPDMA QUEUED commands with PRIO is set to 01b (Isochronous – deadline dependent priority). (See 13.6.3.5).

RDNC — If the RDNC (Read Data Not Continue) bit is cleared to zero, then the device may allow READ FPDMA QUEUED command completion times to exceed what the ICC parameter specified. If the RDNC bit is set to one, then all READ FPDMA QUEUED commands shall be completed by the time specified by the ICC timer value, otherwise the device shall return command aborted for all outstanding commands. RDNC is only applicable to READ FPDMA QUEUED commands with PRIO is set to 01b (Isochronous – deadline dependent priority). (See 13.6.3.1).

The state of the WDNC and RDNC bits shall be preserved across software resets and COMRESETs (via Software Setting Preservations), and shall not be preserved across power cycles.

TAG — The TAG value shall be assigned by host software to be different from all other TAG values corresponding to outstanding commands. TAG shall not exceed the value specified in IDENTIFY DEVICE word 75.

13.6.3.8.2.1 Success Outputs

If this Deadline Handling Subcommand command is supported, the device shall return command completed with no error.

When an Deadline Handling Subcommand command completes successfully, a Set Device Bits FIS shall be sent to the host to complete the Deadline Handling subcommand. This SDB FIS may also indicate other completed commands.

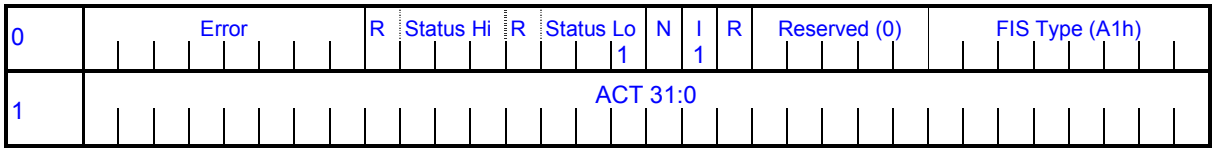


Figure 217 – NCQ QUEUE MANAGEMENT, Deadline Handling - Successful completion

ACT The ACT field of the Set Device Bits FIS communicates completion notification for each of up to 32 commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns. The device shall set to one the bit associated with the TAG value for the Deadline Handling command.

Error The Error register shall contain 00h.

Status As defined in 10.3.6. The ERR bit shall be cleared to zero.

I Interrupt bit. The interrupt bit shall be set to one.

All other fields as defined in 10.3.6.

13.6.3.8.2.2 Error Outputs

13.6.3.8.2.2.1 Upon receipt of a command

If the device has received a command that has not yet been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Register FIS (see Figure 218 below) to the host with the ERR bit set to one and the BSY bit cleared to zero in the Status field, the ATA error code in the Error field.

Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Error	ERROR							
Count(7:0)	na							
Count(15:8)	na							
LBA(7:0)	na							
LBA(31:24)	na							
LBA(15:8)	na							
LBA(39:32)	na							
LBA(23:16	na							
LBA(47:40)	na							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Figure 218 – NCQ QUEUE MANAGEMENT, Deadline Handling - error on command receipt

ERROR ATA error code for the failure condition of the failed command

BSY	0
DRDY	1
DF	0
DRQ	0
ERR	1

Following transmission of the Register FIS, the device shall stop processing any outstanding or new commands until the Queued Error Log (see 13.7) has been read before continuing to abort all outstanding commands. See 13.6.3.4 for more details.

13.6.3.8.2.2.2 During execution of a command

If all commands have been acknowledged by clearing the BSY bit to zero and an error condition is detected, the device shall transmit a Set Device Bits FIS (see Figure 212 below) to the host with the ERR bit set to one in the Status field, the ATA error code in the Error field, and the Interrupt bit set to one. All outstanding commands at the time of an error are aborted as part of the error response and may be re-issued as appropriate by the host. For any commands that have not completed successfully or have resulted in error, the device shall clear the corresponding ACT bits to zero in the Set Device Bits FIS.

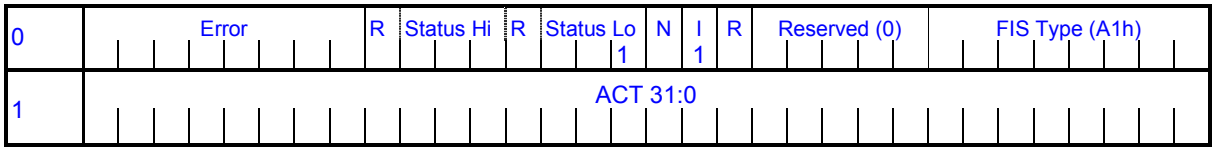


Figure 219 - NCQ QUEUE MANAGEMENT, Deadline Handling – error during execution

ACT	The ACT field of the Set Device Bits FIS communicates successful completion notification for each of up to 32 queued commands. The field is bit-significant and the device sets bit positions to one for each command tag it is indicating successful completion notification for. The device may set more than one bit to one if it is explicitly aggregating successful status returns.
Error	The Error register shall contain the ATA error code.
Status	As defined in 10.3.6. The ERR bit shall be set to one indicating an NCQ error has occurred.
I	Interrupt bit. The interrupt bit shall be set to one.

All other fields as defined in 10.3.6.

Only the registers that are updated as part of the Set Device Bits FIS are modified if the device signals an error condition when the BSY bit in the shadow Status register is cleared to zero, leaving the other Shadow Register Block Registers unchanged. If the device signals an error condition when the BSY bit in the shadow Status register is set to one, the device clears the BSY bit to zero with a Register FIS which updates all registers in the Shadow Register Block.

Following transmission of the Set Device Bits FIS, the device shall stop processing any outstanding or new commands until the Queued Error Log (see 13.7) has been read before continuing to abort all outstanding commands. See 13.6.3.4 for more details.

13.6.4 First-party DMA HBA Support (Informative)

The Serial ATA native queuing model utilizes the First-party DMA mechanism to allow the device to select the appropriate host memory buffer to transfer data to or from. The First-party DMA mechanism ensures memory protection in order to avoid a rogue or errant device indiscriminately accessing host memory. This is accomplished in Serial ATA by having the device only refer to memory buffers by a DMA Buffer Identifier, rather than through the use of physical memory addresses.

For the Native Command Queuing protocol, the buffer identifier used for selecting memory buffers is the same as the unique tag value used to identify the corresponding command. The tags have a value in the range 0 to 31 inclusive and correspond to the tag values assigned by the host at the time commands are issued to the device.

For mainstream desktop host controllers, upon receipt of a DMA Setup FIS the buffer identifier may be used by the host as an index into a vector of pointers to pre-constructed PRD tables (physical region descriptor tables, also commonly referred to as scatter/gather lists) that correspond to the memory buffers for the various outstanding queued commands. The pointer in the vector table at the appropriate index may be transferred into the DMA engine as the base pointer for the active PRD table, effectively causing the DMA engine to select the corresponding memory buffer for subsequent data transfers. This allows minimal change to the existing host DMA architecture and provides a streamlined and efficient buffer selection mechanism.

For such an implementation, host software would be responsible for pre-constructing corresponding PRD tables and updating the vector table entry prior to issuing a new native queued command. Figure 220 illustrates these concepts (the figure is intended as illustrative and does not exclude other possible host controller implementations).

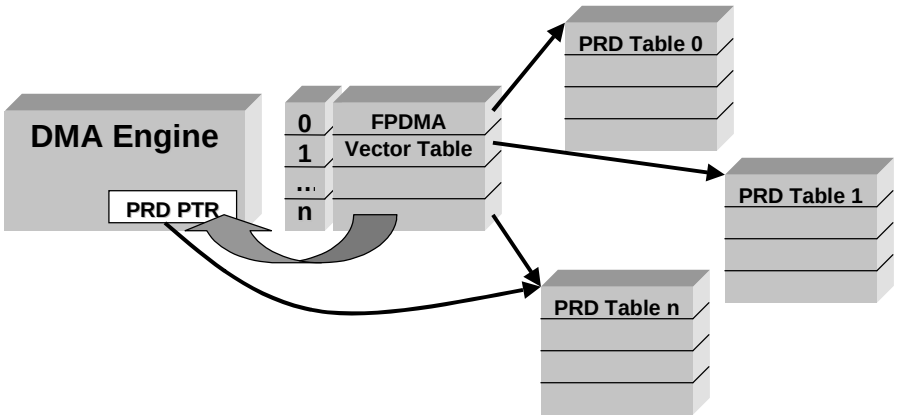


Figure 220 – Example DMA engine indirection for First-party DMA support

This illustrative host controller implementation for supporting First-party DMA has a known shortcoming in that handling non-zero buffer offsets for First-party DMA accesses is cumbersome since the entries in the pre-constructed PRD tables do not necessarily have uniform lengths. For the Native Command Queuing model, there is no requirement for non-zero buffer offset support, however, if out of order data delivery within commands is desired (for example, data for a given command is return by delivering the last half of the data first followed by the first half of the data), support for non-zero buffer offsets is required. See section 13.3 for information on non-zero buffer offsets.

13.7 SATA Logs

There are several log 'files' available in a SATA device. They are all read-only. The READ LOG EXT command in the General Purpose Logging feature set (see ATA8-ACS) are used to read the SATA logs. In some cases the READ LOG DMA EXT command may also be used.

Each log has an 'address' by which it is referenced. Each log contains zero of more 'pages' of data. Each 'page' contains 512 bytes of data.

13.7.1 Log Address Definitions

The log addresses assigned for Serial ATA are defined in **Error! Reference source not found.**.

Table 82- Log Addresses for Serial ATA

Log Address	Description
00h - 0Fh	As defined in the ATA8-ACS standard
10h	NCQ Queued Error log
11h	Phy Event Counters log
12h	NCQ Queue Management log
13h - 17h	Reserved for future Serial ATA definition
18h - FFh	As defined in the ATA8-ACS standard

13.7.2 General Purpose Log Directory (00h)

Devices supporting the Queued Error Log (see 13.7) reflect this support in the General Purpose Log Directory log (00h) by having the value 1 at offset 20h and the value 0 at offset 21h of that log to indicate existence of a log at address 10h of 1 page in length.

Devices supporting the Phy Event Counters Log reflect this support in the General Purpose Log Directory (00h) by having the value 1 at offset 22h and the value 0 at offset 23h of that log to indicate existence of a log at address 11h of 1 page in length.

Devices supporting the NCQ Queue Management Log reflect this support in the General Purpose Log Directory (log 00h) by having the value 1 at offset 24h and the value 0 at offset 25h of that log to indicate existence of a log at address 12h of 1 page in length.

Table 83 - General Purpose Log directory values for Serial ATA

Byte	Value
0 - 1Fh	As defined in the ATA8-ACS standard
20h	1 if Native Command Queuing is supported, 0 if Native Command Queuing is not supported
21h	0
22h	1 if Phy Event Counters are supported 0 if Phy Event Counters are not supported
23h	0
24h	1 if NCQ Queue Management is supported 0 if NCQ Queue Management is not supported
25h	0
26h - 2Fh	Reserved for future Serial ATA definition
30h - 1FFh	As defined in the ATA8-ACS standard

13.7.3 Queued Error Log (10h)

The error-handling scheme for native queued commands halts processing of commands after the host is notified of an error on a native queued command. This allows host software to intervene and take appropriate action to resolve the error and avoids the potential for inconsistency due to data dependencies in the outstanding commands. The host explicitly restarts command processing by issuing a specific command to the device that results in the device aborting all remaining outstanding commands. Because the shadow Status and Error registers are not sufficiently large to contain both information about the error condition and the tag identifying the erring queued command, an additional log has been added in order for the host to be able to retrieve additional information for erring queued commands.

The General Purpose Logging (GPL) feature set is defined in the ATA8-ACS standard.

If IDENTIFY DEVICE word 76 bit 15 is set to one, the Queued Error Log may be read using either of the READ LOG EXT or READ LOG DMA EXT commands.

If IDENTIFY DEVICE word 76 bit 15 is cleared to zero, the Queued Error Log shall be read using the READ LOG EXT command. An attempt to read the Queued Error Log using the READ LOG DMA EXT command shall be aborted and the state of the device shall not change.

Reading the Queued Error Log (10h) has the additional side effect defined in section 13.6.2 of aborting any outstanding queued commands and returns a device that has halted due to a queued command error to a state where it has no commands outstanding and is again ready to accept commands (e.g., after completion of a command to read the log the device returns to state DI0:Device_idle state as defined in section 11.2). The Queued Error Log contains extended command error information.

The Queued Error Log reflects the error information for the first recorded NCQ command with error until such time as another NCQ error is encountered after reading the Queued Error Log. The contents of the Queued Error Log are indeterminate after a software reset or a COMRESET.

Devices supporting the native queued capability shall support the Queued Error Log. The Queued Error Log is one page in length and is defined in Figure 221.

Byte	7	6	5	4	3	2	1	0
0	NQ	UNL	R	TAG				
1	Reserved							
2	Status							
3	Error							
4	LBA(7:0)							
5	LBA(15:8)							
6	LBA(23:16							
7	Device							
8	LBA(31:24)							
9	LBA(39:24)							
10	LBA(47:40)							
11	Reserved							
12	Count(7:0)							
13	Count(15:8)							
14	Reserved							
15	Reserved							
16	Reserved							
17	Reserved							
18	Reserved							
19	Reserved							
20	Reserved							
…								
255								
256	Vendor Specific							
…								
510								
511	Data Structure Checksum							

Figure 221 – Queued Error Log data structure definition

TAG If the NQ bit is cleared to zero, the TAG field contains the TAG corresponding to the queued command that failed.

UNL If set to one indicates that the error condition was a result of receiving an IDLE IMMEDIATE command with the Unload Feature specified. If cleared to zero, the reason for the error was not due to reception of an IDLE IMMEDIATE command with the Unload Feature specified. If the last command received was an Unload Immediate, the device shall not load the heads to the media when reading the Queued Error Log.

If set to one, the NQ bit shall also be set to one to indicate the failure was due to reception of a non-queued command. When set to one, the value of the Status, Error, and LBA(7:0) fields (bytes 3-5) in the log shall be set as follows:

Status: BSY bit shall be cleared to zero and ERR bit shall be set to one

Error: ABRT bit shall be set to one

LBA(7:0): Shall be set to C4h if the unload is being executed or has completed successfully. Shall be set to 4Ch if the unload was not accepted or has failed.

NQ If set to one indicates that the error condition was a result of a non-queued command having been issued and that the TAG field is therefore not valid. If

cleared to zero indicates that the TAG field is valid and that the error condition applies to a queued command.

BYTE1-19 An image of a device to host Register FIS is embedded in the data structure. The fields correspond to the Shadow Register Block Registers and are encoded with error information consistent with the READ DMA QUEUED EXT or WRITE DMA QUEUED EXT command defined in the ATA8-ACS standard.

ERROR The value corresponding to the ATA ERROR register value for the command that failed. The command-specific error condition of invalid tag value shall be handled as an invalid command parameter and shall be reported as such (i.e. ABRT bit set to one in the error register and all other bits cleared to zero).

Note that the value returned in the ERROR field of the data structure is separate from the value returned in the Error shadow register when the initial error condition is signaled. The Error shadow register value is used for the purpose of signaling a queued command error, while the value in the ERROR field of the data structure provides specific information about the error condition that the specific queued command encountered.

Vendor Specific
Allocated for vendor specific use.

Data Structure Checksum
The data structure checksum is the 2's complement of the sum of the first 511 bytes in the data structure. Each byte shall be added with unsigned arithmetic and overflow shall be ignored. The sum of all 512 bytes of the data structure is zero when the checksum is correct.

Reserved/R
All reserved fields shall be cleared to zero.

13.7.4 Phy Event Counters Log (11h)

See 13.9 for details.

13.7.5 NCQ Queue Management Log (12h)

To determine the supported NCQ QUEUE MANAGEMENT subcommands and their respective features, host software shall read log 12h.

This log shall be supported if the NCQ QUEUE MANAGEMENT command is supported (IDENTIFY DEVICE word 77 bit 5 is set to one.)

Dword	Bits	Description
0	31-5	Reserved for NCQ Abort features
0	4	Supports 4h Abort Type (Abort Selected TTag)
0	3	Supports 3h Abort Type (Abort Non-Streaming)
0	2	Supports 2h Abort Type (Abort Streaming)
0	1	Supports 1h Abort Type (Abort All)
0	0	Supports Abort NCQ (subcommand 0h)
1	31-3	Reserved for NCQ Deadline features
1	2	Supports Read Data Not Continue
1	1	Supports Write Data Not Continue
1	0	Supports Deadline Handling (subcommand 1h)
2	31-1	Reserved for subcommand 2h features

2	0	Supports subcommand 2h
…	…	…
15	31-1	Reserved for subcommand Fh features
15	0	Supports subcommand Fh
128-16	31-0	Reserved

Figure 222 – NCQ Queue Management Log (12h) data structure definition

13.8 Asynchronous Notification (Optional)

Asynchronous notification is a mechanism for a device to send a notification to the host that the device requires attention. A few examples of how this mechanism could be used include indicating media has been inserted in an ATAPI device or indicating that a hot plug event has occurred on a Port Multiplier port.

This definition does not list all events that cause a device to generate an asynchronous notification. The mechanism that the host uses to determine the event and the action that is required is outside the scope of this specification, refer to the command set specification for the specific device for more information.

13.8.1 Set Device Bits FIS Notification bit

The Set Device Bits FIS Notification 'N' bit is used by devices to notify the host that attention is needed. The N bit is set to one by a device when the device needs attention, otherwise it is cleared to zero.

By default, a device shall not set the N bit to one in the Set Device Bits FIS. The device shall have the Asynchronous Notification feature enabled by the host before the device may set the N bit to one in the Set Device Bits FIS. After receiving a Set Device Bits with the N bit set, it is the responsibility of the host to interrogate the device and determine what type of action is needed.

13.8.2 Notification Mechanism

To indicate that the device needs attention, the device shall issue a Set Devices Bits FIS to the host with the Interrupt 'I' bit set to one and the Notification 'N' bit set to one. The Error, Status Hi, and Status Lo fields shall accurately reflect the current values of the corresponding register fields in the device.

Reception of a Set Device Bits FIS with the Notification bit set to one may be reflected to the host using the SNotification register, defined in section 14.1.5. A host may support Asynchronous Notification without supporting the SNotification register. The requirement for a host to support Asynchronous Notification is that it may generate an interrupt when the Set Device Bits FIS is received; the SNotification register enables software to be sure that the interrupt was due to a notification event.

13.8.3 State Diagram for Asynchronous Notification

The following state diagram defines the required behavior if the device supports Asynchronous Notification. A device shall only send a Set Device Bits FIS with the Notification bit set when it is in a state that may explicitly send a Set Device Bits FIS to the host as described by the command layer state machines. The state machine is entered from the Device_idle state defined in section 11.2. A device that supports asynchronous notification is also required to support the enhancements for Asynchronous Notification support in the Device_idle and Check_command states in section 11.2.

The state diagram utilizes an internal variable called NotifyPending. The NotifyPending variable indicates that an asynchronous notification has been sent to the host. When NotifyPending is cleared to zero and an event occurs that requires attention, an asynchronous notification may be sent to the host. When NotifyPending is set to one and an event occurs that requires attention, an asynchronous notification shall not be sent to the host since the host has not yet acknowledged reception of the last asynchronous notification received. The host acknowledges reception of an asynchronous notification by sending a Register FIS to the device.

AN0: Notify_host	Transmit Set Device Bits FIS to host that has 'I' bit set to one, 'N' bit set to one, current Status and Error values, and all Reserved fields cleared to zero. Set NotifyPending to 1.	
1. Unconditional	→	DI0: Device_idle

AN0: Notify_host: This state is entered when the asynchronous notification feature is enabled and an asynchronous notification needs to be sent to the host.

When in this state, the device shall issue a Set Device Bits FIS to the host that has the Interrupt 'I' bit set to one, the Notification 'N' bit set to one, current Status and Error field values, and all Reserved fields cleared to zero. The internal variable NotifyPending shall be set to one to indicate that a notification has been sent to the host that has not yet been acknowledged by the host through the reception of a Register FIS from the host.

Transition AN0:1: The device shall unconditionally transition to the state DI0: Device_idle state.

13.8.4 ATAPI Notification

An ATAPI device shall indicate whether it supports asynchronous notification in Word 78 of IDENTIFY PACKET DEVICE, refer to section 13.2.2. The feature is enabled by using SET FEATURES, as defined in section 13.3.5.

13.8.4.1 Event Example (Informative)

An example of an event that may cause the device to generate an asynchronous notification to the host to request attention is the Media Change Event. The Media Change Event occurs when an ATAPI device has detected a change in device state – either media has been inserted or removed.

13.9 Phy Event Counters (Optional)

Phy event counters are an optional feature to obtain more information about Phy level events that occur on the interface. This information may aid designers and integrators in testing and evaluating the quality of the interface. A device indicates whether it supports the Phy event counter feature in IDENTIFY (PACKET) DEVICE Word 76, bit 10. The host determines the current values of Phy event counters by reading the Phy Event Counters Log (see 13.9.3). The counter values shall not be retained across power cycles. The counter values shall be preserved across COMRESET and software resets.

The counters defined are grouped into three basic categories: those that count events that occur during Data FIS transfers, those that count events that occur during non-Data FIS transfers, and events that are unrelated to FIS transfers. Counters related to events that occur during FIS transfers may count events related to host-to-device FIS transfers, device-to-host FIS transfers, or bi-directional FIS transfers. A counter that records bi-directional events is not required to be the sum of the counters that record the same events that occur on device-to-host FIS transfers and host-to-device FIS transfers.

Implementations that support Phy event counters shall implement all mandatory counters, and may support any of the optional counters as shown in Figure 223. Note that some counters may increment differently based on the speed at which non-Data FIS retries are performed by the host and device. Implementations may record CRC and non-CRC error events differently. For example, there is a strong likelihood that a disparity error may cause a CRC error. Thus, the disparity error may cause both the event counter that records non-CRC events and the event counter that records CRC events to be incremented for the same event. Another example implementation difference is how a missing EOF_P event is recorded; a missing EOF_P may imply a bad CRC even though the CRC on the FIS may be correct. These examples illustrate that some Phy event counters are sensitive to the implementation of the counters themselves, and thus these implementation sensitive counters cannot be used as an absolute measure of interface quality between different implementations.

Devices supporting Phy Event Counters shall implement, and report support for, the general purpose logging feature set as defined in the ATA8-ACS standard. In addition, the device shall implement the Phy Event Counters Log.

13.9.1 Counter Reset Mechanisms

There are two mechanisms by which the host may explicitly cause the Phy counters to be reset. The first mechanism is to issue a BIST Activate FIS to the device. Upon reception of a BIST Activate FIS the device shall reset all Phy event counters to their reset value. In addition, when the host reads the Phy Event Counters Log and bit 0 in Features(7:0) is set to one, the device shall return the current counter values for the command and then reset all Phy event counter values.

13.9.2 Counter Identifiers

Each counter begins with a 16-bit identifier. Figure 223 defines the counter value for each identifier. Any unused counter slots in the log should have a counter identifier value of 0h. Optional counters that are not implemented shall not be returned in the Phy Event Counter Log. A value of ‘0’ returned for a counter means that there have been no instances of that particular event. There is no required ordering for event counters within the log; the order is arbitrary and selected by the device vendor.

Bits 14:12 of the counter identifier convey the number of significant bits that counter uses. All counter values consume a multiple of 16-bits. The valid values for bits 14:12 and the corresponding counter sizes are:

1h	16-bit counter
2h	32-bit counter
3h	48-bit counter
4h	64-bit counter

Any counter that has an identifier with bit 15 set to one is vendor specific. This creates a vendor specific range of counter identifiers from 8000h to FFFFh. Vendor specific counters shall observe the number of significant bits 14:12 as defined above.

Identifier (Bits 11:0)	Mandatory/ Optional	Description
000h	Mandatory	No counter value; marks end of counters in the log
001h	Mandatory	Command failed and ICRC error bit set to one in Error register
002h	Optional	R_ERR$_P$ response for Data FIS
003h	Optional	R_ERR$_P$ response for Device-to-Host Data FIS
004h	Optional	R_ERR$_P$ response for Host-to-Device Data FIS
005h	Optional	R_ERR$_P$ response for non-Data FIS
006h	Optional	R_ERR$_P$ response for Device-to-Host non-Data FIS
007h	Optional	R_ERR$_P$ response for Host-to-Device non-Data FIS
008h	Optional	Device-to-Host non-Data FIS retries
009h	Optional	Transitions from drive PHYRDY to drive PHYRDYn
00Ah	Mandatory	Signature Device-to-Host Register FISes sent due to a COMRESET
00Bh	Optional	CRC errors within a Host-to-Device FIS
00Dh	Optional	Non-CRC errors within a Host-to-Device FIS
00Fh	Optional	R_ERR$_P$ response for Host-to-Device Data FIS due to CRC errors
010h	Optional	R_ERR$_P$ response for Host-to-Device Data FIS due to non-CRC errors
012h	Optional	R_ERR$_P$ response for Host-to-Device non-Data FIS due to CRC errors
013h	Optional	R_ERR$_P$ response for Host-to-Device non-Data FIS due to non-CRC errors
C00h	Optional	(Port Multiplier) Host-to-Device non-Data FIS R_ERR$_P$ ending status due to collision
C01h	Optional	(Port Multiplier) Signature Register – Device-to-Host FISes
C02h	Optional	(Port Multiplier) Corrupt CRC propagation of Device-to-Host FISes

Figure 223 – Phy Event Counter Identifiers

FISes that are terminated due to reception of SYNC$_P$ primitives before the end of the FIS (a SYNC Escape) are not counted in the R_ERR$_P$ ending status counters.

13.9.2.1 Counter Definitions

The counter definitions in this section specify the events that a particular counter identifier represents.

13.9.2.1.1 Identifier 000h

There is no counter associated with identifier 000h. A counter identifier of 000h indicates that there are no additional counters in the log.

13.9.2.1.2 Identifier 001h

The counter with identifier 001h returns the number of commands that returned an ending status with the ERR bit set to one in the Status register and the ICRC bit set to one in the Error register.

13.9.2.1.3 Identifier 002h

The counter with identifier 002h returns the sum of (the number of transmitted Device-to-Host Data FISes to which the host responded with R_ERR$_P$) and (the number of received Host-to-Device Data FISes to which the device responded with R_ERR$_P$). The count returned for identifier 002h is not required to be equal to the sum of the counters with identifiers 003h and 004h.

13.9.2.1.4 Identifier 003h

The counter with identifier 003h returns the number of transmitted Device-to-Host Data FISes to which the host responded with R_ERR$_P$.

13.9.2.1.5 Identifier 004h

The counter with identifier 004h returns the number of received Host-to-Device Data FISes to which the device responded with R_ERR$_P$. The count returned for identifier 004h is not required to be equal to the sum of the counters with identifiers 00Fh and 010h.

13.9.2.1.6 Identifier 005h

The counter with identifier 005h returns the sum of (the number of transmitted Device-to-Host non-Data FISes to which the host responded with R_ERR$_P$) and (the number of received Host-to-Device non-Data FISes to which the device responded with R_ERR$_P$). Retries of non-Data FISes are included in this count.

13.9.2.1.7 Identifier 006h

The counter with identifier 006h returns the number of transmitted Device-to-Host non-Data FISes to which the host responded with R_ERR$_P$. Retries of non-Data FISes are included in this count.

13.9.2.1.8 Identifier 007h

The counter with identifier 007h returns the number of received Host-to-Device non-Data FISes to which the device responded with R_ERR$_P$. Retries of non-Data FISes are included in this count.

13.9.2.1.9 Identifier 008h

The counter with identifier 008h returns the number of transmitted Device-to-Host non-Data FISes which were retried after which the host responded with R_ERR$_P$.

13.9.2.1.10 Identifier 009h

The counter with identifier 009h returns the number of times the device transitioned into the PHYRDYn state from the PHYRDY state, including but not limited to asynchronous signal events, power management events, and COMRESET events. If interface power management is enabled, then this counter may be incremented due to interface power management transitions.

13.9.2.1.11 Identifier 00Ah

The counter with identifier 00Ah returns the number of transmitted Device-to-Host Register FISes with the device reset signature in response to a COMRESET, which were successfully followed by an R_OK$_P$ from the host.

13.9.2.1.12 Identifier 00Bh

The counter with identifier 00Bh returns the number of received Host-to-Device FISes of all types (Data and non-Data) to which the device responded with R_ERR$_P$ due to CRC error. The count returned for identifier 00Bh is not required to be equal to the sum of the counters with identifiers 00Fh and 012h.

13.9.2.1.13 Identifier 00Dh

The counter with identifier 00Dh returns the number of received Host-to-Device FISes of all types (Data and non-Data) to which the devices responded with R_ERR$_P$ for reasons other than CRC error. The count returned for identifier 00Dh is not required to be equal to the sum of the counters with identifiers 010h and 013h.

13.9.2.1.14 Identifier 00Fh

The counter with identifier 00Fh returns the number of received Host-to-Device Data FISes to which the device responded with R_ERR$_P$ due to CRC error.

13.9.2.1.15 Identifier 010h

The counter with identifier 010h returns the number of received Host-to-Device Data FISes to which the device responded with R_ERR$_P$ for reasons other than CRC error.

13.9.2.1.16 Identifier 012h

The counter with identifier 012h returns the number of received Host-to-Device non-Data FISes to which the device responded with R_ERR$_P$ due to CRC error.

13.9.2.1.17 Identifier 013h

The counter with identifier 013h returns the number of received Host-to-Device non-Data FISes to which the device responded with R_ERR_P for reasons other than CRC error.

13.9.3 Phy Event Counters Log (11h)

The Phy Event Counters Log is one page (512 bytes) in length. The first Dword of the log contains information that applies to the rest of the log. Software should continue to process counters until a counter identifier with value 0h is found or the entire log has been read. A counter identifier with value 0h indicates that the log contains no more counter values past that point. Log 11h is defined in Figure 224.

If IDENTIFY DEVICE word 76 bit 15 is set to one, the Phy Event Counters may be read using either of the READ LOG EXT or READ LOG DMA EXT commands.

If IDENTIFY DEVICE word 76 bit 15 is cleared to zero, the Queued Error Log shall be read using the READ LOG EXT command. An attempt to read the Phy Event Counters Log using the READ LOG DMA EXT command shall be aborted and the state of the device shall not change.

Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved							
3	Reserved							
…	…							
n	Counter n Identifier							
n+1								
n+2	Counter n Value							
n+ Counter n Length								
…	…							
508	Reserved							
509								
510								
511	Data Structure Checksum							

Figure 224 – Phy Event Counters Log data structure definition

Counter n Identifier

Phy event counter identifier that corresponds to Counter n Value. Specifies the particular event counter that is being reported. The Identifier is 16 bits in length. Valid identifiers are listed in Figure 223.

Counter n Value

Value of the Phy event counter that corresponds to Counter n Identifier. The number of significant bits is determined by Counter n Identifier bits 14:12 (as defined in section 13.9.2). The length of Counter n Value shall always be a multiple of 16-bits. All counters are one-extended. For example, if a counter is only physically implemented as 8-bits when it reaches the maximum value of FFh, it shall be one-extended to FFFFh. The counter shall stop (and not wrap to zero) after reaching its maximum value.

Counter n Length

Size of the Phy event counter as defined by bits 14:12 of Counter n Identifier. The size of the Phy event counter shall be a multiple of 16-bits.

Data Structure Checksum

The data structure checksum is the 2's complement of the sum of the first 511 bytes in the data structure. Each byte shall be added with unsigned arithmetic and overflow shall be ignored. The sum of all 512 bytes of the data structure is zero when the checksum is correct.

Reserved All reserved fields shall be cleared to zero

13.10 Staggered Spin-up (Optional)

Storage subsystems that include numerous Serial ATA hard disk drives are presented with power system design issues related to the current load presented during system power-up. It is desirable to provide a simple mechanism by which the storage subsystem controller(s) may sequence disk device initialization and spin up. Note that Serial ATA disk drive vendors may not always provide the capability to parse or execute ATA commands prior to spinning up a device and completing device initialization, therefore this mechanism may not rely on the ATA protocol.

In order to accommodate staggered spin-up of an array of disk drives in an enclosure, disk drives shall not spin up until after successful Phy initialization, that is after the Phy enters the DP7:DR_Ready state. Any of a number of methods may be used by the disk drive to defer spin-up prior to Phy initialization and to maintain correct interface status during device initialization.

Storage subsystem controllers may employ a variety of methods to sequence Phy initialization across their plurality of Serial ATA ports, including but not limited to staged release of chip-level resets of host-side Serial ATA transceivers, or embedded advanced power management logic.

System implementations should comprehend the various scenarios that may require power management, and the corresponding Phy initialization sequences. For example, upon power up of a populated storage subsystem, Phy communication is initiated with a COMRESET signal generated by the host-side transceiver. The use of the term "host-side transceiver" here refers to the Serial ATA interface located on the storage subsystem controllers. This contrasts with sequences associated with hot-plugging of a Serial ATA disk drive into an operational storage subsystem, wherein COMINIT signals generated by disk-side transceivers initiate Phy communications. In both of these cases, COMRESET or COMINIT signals are followed by exchange of COMWAKE signals. It is the successful entry into the DP7:DR_Ready state that gates disk drive spin-up.

The device shall not use the staggered spin-up mechanism to cause the device to spin-up in cases where the device was spun down using an ATA command. See ATA8-ACS for more information.

13.11 Non-512 Byte Sector Size (Informative)

The Serial ATA interface has no inherent sector size dependency and there is nothing in the Serial ATA interface specification that precludes sector sizes other than 512 bytes.
Regardless of the physical sector size of storage devices, the maximum Data FIS payload length is 2048 Dword as defined in section 10.3.11. This may imply that either the number of physical sectors that are encompassed by a single Data FIS is reduced for devices with larger sector sizes or that Data FISes convey a non-integer number of sectors of data being transferred.

13.12 Defect Management (Informative)

13.12.1 Overview (Informative)

Storage subsystems that have been based on SCSI disk drives have evolved processes to address disk drive failure and mitigate effects of disk defects. For example, SCSI commands such as READ DEFECT DATA (37h) and REASSIGN BLOCKS (07h) have been used to allow storage administrators or applications to proactively address problematic sectors on a disk surface.

Serial ATA devices leverage the economies of the desktop device market, and as a consequence inherit both the design philosophy and implementation of desktop devices. From a design philosophy perspective, desktop devices have never yielded the low-level of control, such as reassignment of logical blocks that is commonplace in enterprise-class devices. Instead, desktop devices have been positioned as a "black-box" data repository. This approach has many benefits, such as elimination of defect management as a design task for the computer (O/S, file system, I/O, device driver) designer, at the expense in some cases of deterministic device performance and awareness of defect management activity in general.

As "black box" providers, Serial ATA device vendors assume the bulk of the responsibility for defect management and data availability. This section provides a high-level overview of approaches that Serial ATA device vendors employ in these areas, and of the tools that are available to system designers, storage management application designers and system administrators to maximize storage and data dependability.

In summary, Serial ATA devices provide similar information via SMART (Self-Monitoring, Analysis and Reporting Technology) commands as the information returned with SCSI READ DEFECT DATA commands. Subsystem designers make no provisions for reassignment of blocks; rather reassignment of defective blocks is performed automatically when indicated by desktop-class devices. Additionally, disk device manufacturers provide a spectrum of tools and embedded features that help prevent occurrence of non-recoverable read errors, that help detect disk device degradation that might cause catastrophic loss of data, and that help administrators diagnose and isolate the cause of error events.

13.12.2 Typical Serial ATA Reliability Metrics (Informative)

Various methods are employed by disk manufacturers to recover bad bits in a block of data. First and foremost, extensive error correcting codes (ECC) are used "on-the-fly" to detect and correct errors without impacting device performance. Second, various read-retry schemes may be used when ECC fails to correct an error on the fly. Only after retry processes are exhausted are errors posted to the host.

Design Recommendation: Accommodate inherent non-recoverable read error rate through RAID schemes appropriate for the target market's reliability needs.

13.12.3 An Overview of Serial ATA Defect Management (Informative)

Defects that cause non-recoverable read errors come in two distinct flavors, temporary and permanent. Various schemes are employed by device manufacturers to determine the nature of a defect. If a defect is determined to be of a permanent nature, device firmware re-maps the LBA to a predefined spare sector, and marks the defective sector as such. Subsequent accesses to the re-mapped LBA are directed to the spare sector in a fashion transparent to the host. As part of the re-mapping process, the device preserves the error state of the remapped block until such time as it is written with new data.

The number of spare sectors configured in a disk drive is generally unique across different device vendors and device models and reflects tradeoffs between device capacity and expected defect rates. Generally, spares are associated with defined allocation groups in a disk drive. If the

number of spare sectors becomes exhausted over time, subsequent permanent defects result in sectors that cannot be re-mapped. The affected Logical Block Addresses (LBAs) are then recognized as bad by the host operating system or disk utilities such as “scandisk”, and are subsequently not used.

Design Recommendation: If a host or a Serial ATA storage subsystem encounters a non-recoverable read error, that error is managed by the disk drive. The disk drive is responsible for performing extensive read retry processes prior to communicating an error condition, exercising internal device diagnostics to determine the nature of the error, and re-map the physical sector if required. Subsequent accesses to that logical address are directed to a known good region on the disk.

Care should be taken to assure that spares are not exhausted if write caching is enabled. If this care is not taken, there is a possibility that a write operation may not actually complete successfully and return the appropriate error condition to the host in a timely fashion. Refer to the information on SMART in section 13.12.5 for monitoring spares status.

In especially critical data applications, specific queries of disk drive logs may be performed to determine whether the error event is a random event, or if there exists some degrading condition that warrants additional system or administrator action.

In storage subsystems where known good data may be recovered from redundant sources (such as in a RAID subsystem), it is recommended that known good data be written back to any LBA for which a read error is reported. This behavior ensures that the disk drive has an opportunity to remedy the error condition and write known good data to known good blocks on disk, whether those be remapped blocks resulting from a permanent error condition, or the same blocks that may have been affected by an error of a temporary nature. Subsequent read accesses are assured of being satisfied without error.

13.12.4 Continuous Background Defect Scanning (Informative)

Serial ATA device vendors may employ schemes called continuous background defect scanning (CBDS), where during idle periods, firmware routines are used to scan sectors on the disk to look for and correct defects. CBDS therefore executes as a background task. The effect of CBDS is to reduce the occurrence of non-recoverable read errors by proactively “cleaning” defects from good sectors or copying data from suspect regions on disk to spare sectors. CBDS is described in more detail in the SMART specification which also provides a means for enabling/disabling the capability.

Design Recommendation: Consider CBDS requirements when evaluating system power-saving schemes that might power-off disk drives during apparent periods of inactivity.

13.12.5 Self-Monitoring, Analysis and Reporting Technology (Informative)

Some disk failures are predictable. Such failure mechanisms are characterized by degradation over time, and in some cases may be effectively monitored by the disk drive and logged for periodic reporting to storage managers or management utilities.

The ATA8-ACS standard describes SMART command support in detail. Individual device vendors provide unique SMART capabilities on their devices and documentation on those capabilities so that host or controller developers may effectively use predictive failure information

Design Recommendation: Use capabilities provided through ATA SMART commands to predict disk failure when possible. Predictive knowledge may be used to swap in spare disk drives in RAID configurations, to trigger data backup or protection routines, or to schedule storage subsystem maintenance.

13.13 Enclosure Services/Management (Optional)

13.13.1 Overview

A means for providing support for industry-standard SAF-TE (SCSI Accessed Fault-Tolerant Enclosures) and SES (SCSI Enclosure Services) enclosure services is provided in order to improve the functionality of Serial ATA storage subsystems. No modification to Serial ATA devices contained within an enclosure is required.

A storage enclosure processor (SEP) is a device which interfaces with various sensors and indicators within a storage enclosure subsystem. A Serial ATA Enclosure Management Bridge (SEMB) is a device that allows a Serial ATA host controller or Port Multiplier to communicate with a SEP, acting as a bridge between the host and the enclosure processor. This section defines a standard interface for software to communicate with SEMB devices via the ATA Command Block Registers. It also defines how SEMB devices communicate with SEP devices using the I^2C IPMI protocol. To allow software to communicate effectively with the SEMB, the SEMB is viewed as another Serial ATA device connected to the storage subsystem. The SEMB may be implemented within a Serial ATA host controller, within a Port Multiplier, or as a hardware bridge between a host and the SEP.

In this specification, the host (the Serial ATA RAID controller or HBA) may use either the SAF-TE or SES command protocol to communicate control/status with the SEP. These protocols provide the necessary features and have the advantage of being well known and widely implemented, and should therefore minimize the impact on RAID controller firmware and host management software. The implementation requires that Serial ATA host controllers or Port Multipliers that support enclosure management have an I^2C interface to communicate with the SEP device.

This specification also addresses the need to support a generic and standardized interface that allows application software from different vendors to communicate with the management device. This is implemented by having the appropriate commands be sent/received using the ATA Command Block Register set (or Serial ATA Register FIS) using the READ SEP/WRITE SEP commands associated with this interface. The ATA Command Block Register interface allows for consistency with the other devices in the subsystem as well as being simple to use and well understood.

13.13.2 Topology

The enclosure services support mechanisms should support the configurations and topologies expected for storage subsystems that require such enclosure services (such as external storage enclosures). Figure 225 illustrates a generic configuration.

The concentrator is controller logic that bridges a host interface to one or more Serial ATA devices. Typically a concentrator would be a RAID controller or a Port Multiplier, but it may be a simple HBA or part of an integrated multi-function chipset. Because a concentrator may be embodied differently depending on implementation, the generic term is used in order to avoid implying any particular implementation.

The host interface is the interface through which the host communicates with the concentrator. For RAID controllers plugged into a PCI slot, the host interface would be PCI, while for external RAID controllers in a storage subsystem, the host interface might be Fibre Channel, InfiniBand Architecture, Ethernet (iSCSI) or any of a number of different external subsystem interconnects.
The SEMB (Serial ATA enclosure management bridge) is logic that bridges enclosure management data from a host interface to an enclosure management bus (indicated as I^2C in the figure). For an intelligent PCI RAID controller, the SEMB may be firmware running on the RAID processor and associated design-specific I^2C controller interface logic, while for a Port Multiplier, the SEMB would be controller logic that bridges the I^2C interface (and transactions) to Serial ATA via a logical ATA Command Block Register interface.

The SEP (storage enclosure processor) interfaces with the various sensors and indicators in the enclosure such as temperature sensors, fan tachometers, and indicator LEDs.

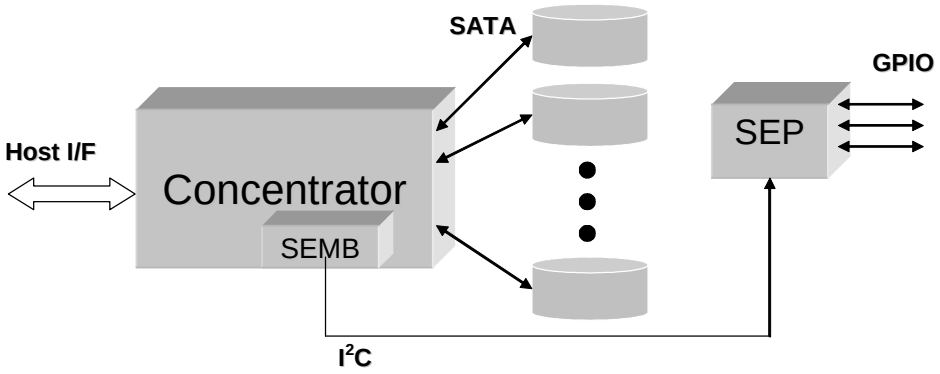


Figure 225 – Generic enclosure services topology

13.13.2.1 Definition Configuration

The definition configuration is distilled from the generic topology in Figure 225 where the extraneous elements have been removed for the sake of definition clarity. For the definition configuration, the host interface is selected as Serial ATA since this results in the maximum number of interfaces/elements being exposed for definition. In practice, the various elements may be integrated into another subsystem element (for instance, the SEMB might be integrated with the RAID controller or Port Multiplier). For the sake of definition, the elements are shown separately and the solution is presented as if it were a Serial ATA target device. In such a configuration, the SEP is being presented as merely another exposed Serial ATA device. This configuration is most analogous to existing SCSI enclosure services schemes where the enclosure services device is implemented using a SCSI target ID. For this configuration the enclosure services bridge and associated storage enclosure processor are exposed as a single-purpose ATA device.

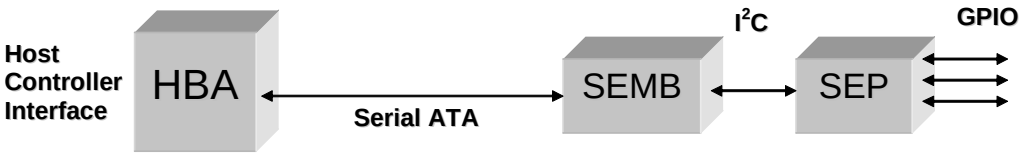


Figure 226 – Simplified view of generic topology

Since the HBA (host bus adapter) in the previous figure is a standard Serial ATA HBA and is therefore fixed, the elements being defined in this section may be further reduced to the definition configuration in Figure 227 that presents the enclosure services facility as an ATA device with a Command Block Register interface.

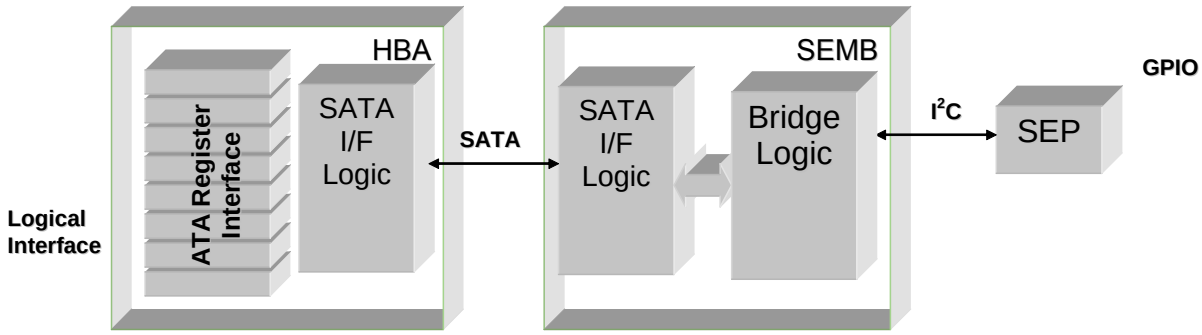


Figure 227 – Enclosure services definition configuration

For implementations where the SEP is not provided packaged with its front-end SEMB or equivalent (i.e. the SEP and SEMB are provided by different vendors), it is recommended that the interconnect between these two elements be I^2C and the command protocol for this interconnect is defined in section 13.13.4.2. For implementations where the SEP is provided by the same supplier as the SEMB, the interconnect between these elements may be vendor-specific (and may be embedded for those cases where the two elements are integrated).

Similarly, for implementations where the SEMB is packaged with its front-end HBA or equivalent, the interconnect between these elements may be vendor specific and is not required to be Serial ATA (in some configurations, the SEMB may be integrated into the HBA in which case the interconnect between these two logical elements is embedded).

13.13.3 Limitations

In order to accommodate a range of possible implementations, the interfaces and elements defined in this section represent more than would typically be utilized by any one design/implementation. Only those interfaces that are exposed and interconnect ingredients from different vendors are expected to utilize the corresponding definitions and specifications described in this section. For example, an intelligent RAID controller may have the SEMB functionality implemented as firmware running on the embedded RAID processor, and such a solution would not expose the SEMB front-end interface, which would be vendor-specific (i.e internal to the RAID card and managed by the vendor-supplied firmware). However, such a solution probably would expose the I^2C interface if it interconnects with another vendor's SEP, and would therefore need to comply with the specification for the communications between the SEMB and the SEP.

13.13.4 Definition

This specification supports the same enclosure management command/status as the SAF-TE protocol (plus addendums). Since SAF-TE is widely used in current SCSI applications, it is a natural path to try to reuse as much of the data format as possible. This specification also supports the SES enclosure management command/status as defined in the SCSI-3 Enclosure Services Command Set specification. See section 3 for specific references.

For the case of the Serial ATA Register set, a subset of the existing ATA task file is used for creating the commands to send/receive data from the SEP device. Where possible, the same command structure as the existing ATA protocol has been used. All SEP commands are issued to the SEMB with the SEP_ATTN opcode in the Command register and the actual SEP command is passed as a parameter in Features(7:0). Section 13.13.4.3 and 13.13.4.4 define the host-to-SEP and SEP-to-host command protocols.

13.13.4.1 Discovery

Following a COMRESET or software reset, the SEMB shall place the unique SEMB-specific signature as identified in Figure 228 into the logical Command Block Registers if the SEMB detects the presence of an attached SEP. The signature shall be available in the logical Command Block Registers no later than 3 seconds after a reset operation (whether power-on reset, COMRESET, or soft reset). For implementations where the SEMB is separate from the HBA, the Command Block Register signature shall be conveyed over the Serial ATA interconnect using a Register FIS device-to-host.

Register	7	6	5	4	3	2	1	0
Error	00h							
Count(7:0)	01h							
Count(15:8)	00h							
LBA(7:0)	01h							
LBA(31:24)	00h							
LBA(15:8)	3Ch							
LBA(39:32)	00h							
LBA(23:16	C3h							
LBA(47:40)	00h							
Device	na							
Status	BSY	DRDY	DF	DSC	DRQ	0	0	ERR

Status =50h

Figure 228 – Register Signature Indicating Presence of Enclosure Services Device

If the SEMB does not detect the presence of an attached SEP, then the SEMB shall place the signature as identified in Figure 229 into the logical Command Block Registers in order to convey to the host that there is no device present at the logical interface, and the SEMB shall not respond to any subsequent Command Block Register accesses or issued commands until the next COMRESET or software reset. For implementations where the SEMB is separate from the HBA, the Command Block Register signature is conveyed over the Serial ATA interconnect using a Register FIS device-to-host.

Register	7	6	5	4	3	2	1	0
Error	FFh							
Count(7:0)	FFh							
Count(15:8)	FFh							
LBA(7:0)	FFh							
LBA(31:24)	FFh							
LBA(15:8)	FFh							
LBA(39:32)	FFh							
LBA(23:16	FFh							
LBA(47:40)	FFh							
Device	FFh							
Status	0	1	1	1	1	1	1	1

Status=7Fh

Figure 229 – Register Signature for Absent Enclosure Processor

In addition to the device reset signature, SEPs shall support the IDENTIFY SEP command described in section 13.13.5.1 in order to allow host software to determine its capabilities and to identify whether it supports the SAF-TE or SES command set.

13.13.4.2 Logical Command Block Registers to I^2C Mapping

The SEMB provides the mapping and translation from transactions between the SEP and SEMB, and the transactions presented to the host via the logical Command Block Register interface.

13.13.4.2.1 Command Delivery

Commands are delivered to the SEP by the SEMB via the Command Block Register interface as a result of the Command or Device Control register being written (if the SEMB logical register interface is closely coupled to the host/HBA) or in response to receipt of a Register FIS (if the SEMB logical register interface is at the far end of a Serial ATA physical interconnect).

Commands are delivered to the SEP over I^2C by the SEMB which extracts the SEP command and command type fields from the Command Block Registers (or received Register FIS) and forwards the fields to the SEP. If the SEP is discrete and connected via an I^2C interconnect, the SEP command fields are packaged as an I^2C frame and forwarded to the SEP via the I^2C interconnect. The SEMB logical Command Block Registers observes the same conventions as Serial ATA for handling of the BSY bit. The BSY bit is set by the interface/SEMB in response to the Command register being written, and the SEP later clears this bit to indicate command completion/status.

For issuing commands, the SEMB shall generate and transmit an I^2C packet based on the contents of the Command Block Registers or Register FIS. The mapping of Command Block Registers to transmitted I^2C packet shall only be done for commands written using the SEP_ATTN opcode, and the SEMB need not respond to any other opcode in the Command register. The Command Block Registers mapping to I^2C packet shall be as indicated in Figure 230.

Register	7	6	5	4	3	2	1	0
Features(7:0)	SEP_CMD							
Features(15:8)	Reserved							
Count(7:0)	LEN							
Count(15:8)	Reserved							
LBA(7:0)	CMD_TYPE							
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	Reserved			0	Reserved			
Command	SEP_ATTN (67h)							

Figure 230 – Command Block Register Fields Used in Enclosure Processor Communications

SEP_CMD The SAF-TE or SES command code to be issued in conjunction with the command type specified in CMD_TYPE.

SAF-TE READ BUFFER usage: SEP_CMD is equivalent to the BUFFER ID field of the SCSI READ BUFFER command. See section 3.1 in the SAF-TE specification reference for the command codes and their functions.

SAF-TE WRITE BUFFER usage: SEP_CMD is equivalent OPERATION CODE field transferred in the parameter data of the SCSI WRITE BUFFER command. See section 3.2 in the SAF-TE specification reference for the command codes and their functions.

SES RECEIVE DIAGNOSTIC RESULTS usage: SEP_CMD is equivalent to the PAGE CODE field of the SCSI RECEIVE DIAGNOSTIC RESULTS command. See section 6.1 of the SES specification reference for the command codes and their functions.

SES SEND DIAGNOSTIC usage: SEP_CMD is equivalent to the PAGE CODE field transferred in the parameter data of the SCSI SEND DIAGNOSTIC command. See section 6.1 of the SES specification reference for the command codes and their functions.

IDENTIFY SEP usage: SEP_CMD is equal to ECh.

LEN The transfer length of the data transfer phase of the command in Dword units. Valid values are 1-255 (yielding a maximum transfer length of 1020 bytes). Data transfers that are not a multiple of 4 bytes shall be padded by the transmitter with zeros to the next 4-byte (Dword) granularity.

CMD_TYPE
Flag indicating whether the issued SEP command is a SAF-TE command code or a SES command code and whether the data transfer direction is from SEP-to-host or host-to-SEP. The encoding of the field is as follows:

00h SAF-TE command code with SEP-to-host data transfer, including IDENTIFY SEP

80h SAF-TE command code with host-to-SEP data transfer

02h SES command code with SEP-to-host data transfer, including IDENTIFY SEP

82h SES command code with host-to-SEP data transfer

All other values reserved

The resulting I^2C frame for delivering a command is as indicated in Figure 231.

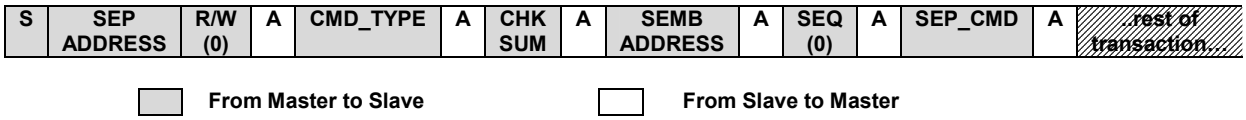


Figure 231 – I^2C Frame for Conveying an Enclosure Services Command

The SEMB need not support any Command register writes with values other than SEP_ATTN. In response to a Command register write of a value other than SEP_ATTN, the SEMB shall set the ERR bit and clear the BSY bit in the Status register. In response to such illegal host behavior, the SEMB shall not generate any I^2C traffic for that illegal command. The SEMB shall support Device

Control register writes where the state of the SRST bit changes, but shall take no action in response to Device Control register writes where SRST does not change state.

The SEP need not support both SAF-TE and SES command protocols. In response to a SEP command issues with a protocol not supported by the SEP, the SEP shall return with error status as defined in section 13.13.4.2.2.

13.13.4.2.2 Status Mechanism

The SEMB indicates command completion to the host by clearing the BSY bit to zero in the Status register and by triggering an interrupt. SEP status returns to the SEMB consist only of the Status byte and no other Command Block Registers are used to convey status. If the SEP has encountered some error condition or does not support the issued SEP command, then the ERR bit in the Status register is set to one.

If the SEP is communicating to the SEMB over an I^2C interconnect, then the status byte is included at the end of the transactions as indicated in the SEP read and write definitions that follow. Upon transferring the status value into the Status register, the SEMB shall clear the BSY bit to zero and signal an interrupt. If the SEMB is connected to the host via a Serial ATA interconnect, then the ending status shall be collected in a Register FIS and transmitted to the host. In response to the read and write SEP commands, the SEP_CMD and CMD_TYPE byte values are returned.

Upon successful completion of a command, the status value in the Status register shall be 50h. Upon an error condition, the status value shall be 51h.

13.13.4.3 Host-to-SEP Data Commands

All host-to-SEP data transfers transfer a data payload with length as indicated in the LEN field for the command. The SAF-TE and SES references define the commands and functions supported as well as the format of the transferred data structures.

All SEP commands are issued using the SEP_ATTN command (opcode 67h) in the Command register and the SEP command code in Features(7:0) as illustrated in the Command Block Registers image of Figure 232. The CMD_TYPE field identifies whether the issued SEP command is a read or a write and whether the command protocol is SAF-TE or SES.

Register	7	6	5	4	3	2	1	0
Features(7:0)	SEP_CMD							
Features(15:8)	Reserved							
Count(7:0)	LEN							
Count(15:8)	Reserved							
LBA(7:0)	CMD_TYPE (80h or 82h)							
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	Reserved			0	Reserved			
Command	SEP_ATTN (67h)							

Figure 232 – WRITE SEP Command Block Registers

Host-to-SEP data transfer commands shall be followed by a data transfer from the host to complete the SEP command delivery. If the command is delivered to the SEP over an I^2C interface, the transmitted I^2C packets shall be of the form indicated in Figure 233.

SEMB (master) to SEP (slave) transfer – transfers both the command and data

S	SEP ADDRESS	R/W (0)	A	CMD_TYPE	A	CHK SUM	A	SEMB ADDRESS	A	SEQ (0)	A	SEP_CMD	A	...

...	Data[0]	A	Data[1]	A	…	Data[LEN*4-1]	A	CHKSUM	A	P

SEP (master) to SEMB (slave) transfer – transfers status

S	SEMB ADDRESS	R/W (0)	A	CMD_ TYPE	A	CHK SUM	A	SEP ADDRESS	A	SEQ (0)	A	SEP_ CMD	A	SEP STATUS	A	CHK SUM	A	P

From Master to Slave From Slave to Master

Figure 233 – I^2C Transactions Corresponding to a WRITE SEP Command

The I^2C transport for Serial ATA enclosure service traffic shall be Master to Slave (IPMB) compliant. See the IPMB and I^2C references for details on Master to Slave transport conventions for IPMB.

The host shall use the DMA protocol for transferring data from the host to the SEMB, and if the interface between the SEMB and the host is a Serial ATA interface, the SEMB shall trigger the DMA data transfer by transmitting a DMA Activate FIS to the host.

13.13.4.4 SEP-to-Host Data Commands

All SEP-to-Host data transfers transfer a data payload. The LEN field for the command indicates the length of the data payload being transferred. The SAF-TE and SES references define the commands and functions supported as well as the format of the transferred data structures. Both SAF-TE and SES SEPs shall support the IDENTIFY SEP command, which has the same format for both command sets.

All SEP commands are issued using the SEP_ATTN command (opcode 67h) in the Command register and the SEP command code in Features(7:0) as illustrated in the Command Block Registers image of Figure 234. The CMD_TYPE field identifies whether the issued SEP command is a read or a write and whether the command protocol is SAF-TE or SES.

Register	7	6	5	4	3	2	1	0
Features(7:0)	SEP_CMD							
Features(15:8)	Reserved							
Count(7:0)	LEN							
Count(15:8)	Reserved							
LBA(7:0)	CMD_TYPE (00h or 02h)							
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	Reserved			0	Reserved			
Command	SEP_ATTN (67h)							

Figure 234 – READ SEP Command Block Registers

SEP-to-host data transfer commands shall be followed by a subsequent data transfer from the device to the host to complete the SEP command. If the command is delivered to the SEP over an I^2C interface, the transmitted I^2C packets shall be of the form indicated in Figure 235.

SEMB (master) to SEP (slave) transfer – transfer the command

SEP (master) to SEMB (slave) transfer – transfers both the data and status

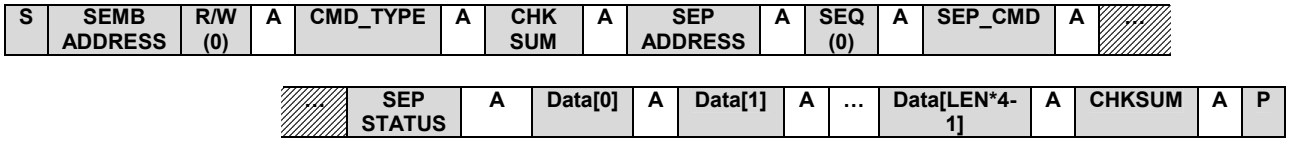


Figure 235 – I^2C Transactions Corresponding to READ SEP Command

The host shall use the DMA protocol for transferring data from the SEMB to the host.

13.13.5 SES and SAF-TE Extensions

13.13.5.1 IDENTIFY SEP Command (ECh)

Both SAF-TE and SES SEPs shall support the IDENTIFY SEP command as defined here. The IDENTIFY SEP command is a SEP-to-host command code used as the SEP command argument for the SEP_ATTN command with the CMD_TYPE value set to either 00h or 02h depending on the command protocol being used (see section 13.13.4.2.1). The command returns a data structure that describes the capabilities and attributes of the attached SEP. The IDENTIFY SEP command requests that the SEP return enclosure specific information (not device or environmental information). The command is roughly analogous to a SCSI INQUIRY command.

For parameters defined as a string of ASCII characters, the ASCII data fields shall contain only graphic codes (i.e., code values 20h through 7Eh) and all strings shall be padded with space characters to the full width of the field. For the string “Copyright”, the character “C” is the first byte, the character “o” is the second byte, etc.

13.13.5.1.1 IDENTIFY SEP Data Structure

Figure 236 describes the data structure that is returned by the SEP in response to the IDENTIFY_SEP command. All reserved fields shall be cleared to zero. By setting the LEN field of the issued Read SEP command, the amount of returned data may be controlled (i.e. the transfer time may be reduced by not transferring the reserved and vendor specific bytes at the end of the data structure).

The IDENTIFY SEP data structure is normally 64 bytes long and may be extended in order to support larger VENDOR SPECIFIC ENCLOSURE INFORMATION.

The data structure does not include a list of elements in an enclosure.

This data block provides enclosure descriptor information and parameters.

Bytes	Field name
0	ENCLOSURE DESCRIPTOR LENGTH
1	SUB-ENCLOSURE IDENTIFIER
2-9	ENCLOSURE LOGICAL IDENTIFIER
10-17	ENCLOSURE VENDOR IDENTIFICATION
18-33	PRODUCT IDENTIFICATION
34-37	PRODUCT REVISION LEVEL
38	CHANNEL IDENTIFIER
39-42	FIRMWARE REVISION LEVEL
43-48	INTERFACE IDENTIFICATION STRING
49-52	INTERFACE SPECIFICATION REVISION LEVEL
53-63	VENDOR SPECIFIC ENCLOSURE INFORMATION

Figure 236 – IDENTIFY SEP data structure definition

ENCLOSURE DESCRIPTOR LENGTH

The ENCLOSURE DESCRIPTOR LENGTH field specifies the number of valid bytes contained in the IDENTIFY SEP data structure.

SUB-ENCLOSURE IDENTIFIER

As defined in the SES reference. Unless sub-enclosures are defined this field shall be cleared to zero.

ENCLOSURE LOGICAL IDENTIFIER

The ENCLOSURE LOGICAL IDENTIFIER field contains a unique logical identifier for the subenclosure. It shall use an 8-byte NAA identifier, the format of which is defined in the SCSI Primary Commands reference. The ENCLOSURE LOGICAL IDENTIFIER is unique to the enclosure and may be different from the worldwide name of the device providing the enclosure services. The combination of this field, along with the ENCLOSURE VENDOR IDENTIFICATION and PRODUCT IDENTIFICATION fields, uniquely identifies any SEP unit from any manufacturer. The worldwide name should have an NAA field of 5h, indicating that IEEE is the naming authority.

ENCLOSURE VENDOR IDENTIFICATION

The ENCLOSURE VENDOR IDENTIFICATION field shall contain the identification string for the vendor of the enclosure in the same format as specified for the vendor identification field of the standard SCSI INQUIRY data (see SCSI Primary Commands reference). The ENCLOSURE VENDOR IDENTIFICATION may be different from the vendor identification of the device providing the enclosure services.

PRODUCT IDENTIFICATION

The PRODUCT IDENTIFICATION field shall contain the product identification string for the enclosure in the same format as specified for the product identification field of the standard SCSI INQUIRY data (see SCSI Primary Commands reference). The PRODUCT IDENTIFICATION field may be different from the product identification of the device providing the enclosure services.

PRODUCT REVISION LEVEL

The PRODUCT REVISION LEVEL field shall contain the product revision level string for the enclosure in the same format as specified for the product revision level field of the standard SCSI INQUIRY data (see SCSI Primary Commands reference). The PRODUCT REVISION LEVEL may be different from the product revision level of the device providing the enclosure services.

CHANNEL IDENTIFIER

The CHANNEL IDENTIFIER field is used to distinguish between separate HBA channels supported by a single enclosure (through multiple/redundant host connections for example). The value in this field is unique for each channel. This field is optional and if not used shall be cleared to zero.

FIRMWARE REVISION LEVEL

The FIRMWARE REVISION LEVEL field is a 4-byte ASCII string that identifies the current firmware revision of the SEP device.

INTERFACE IDENTIFICATION STRING

The INTERFACE IDENTIFICATION STRING field is a 6-byte field that holds the constant ASCII string “SAF-TE” (if the command is issued with the SAF-TE protocol bit set in the CMD_TYPE field and the SEP supports this protocol) or “S-E-S” (if the command is issued with the SES protocol bit set in the CMD_TYPE field and the SEP supports this protocol). This serves to identify that the enclosure is compliant with the command protocol indicated by the CMD_TYPE field used in issuing the IDENTIFY SEP command.

INTERFACE SPECIFICATION REVISION LEVEL

The INTERFACE SPECIFICATION REVISION LEVEL field is a 4-byte field that holds an ASCII string of the format “x.xx”, which identifies the revision of the Interface Specification to which this SEP device claims compliance. ASCII string data is stored with the most significant (leftmost) character stored at the lowest byte offset of the field.

VENDOR-SPECIFIC ENCLOSURE INFORMATION

The VENDOR-SPECIFIC ENCLOSURE INFORMATION field is available for vendor-specific definition and use.

13.13.5.2 Activity LED Control

The SES and SAF-TE protocols provide commands that are used to inform the SEP device of the state of each of its associated slots and the devices potentially inserted. This information is used to drive the enclosure status signals (LEDs, LCD, audible alarm, etc.) to some meaningful state,

or to force the SEP to respond with a preprogrammed response as required; depending on the vendor's implementation.

Since Serial ATA devices do not necessarily provide an activity indication, extensions to the SAF-TE and SES facilities are defined in section 13.13.5.2.1 and 13.13.5.2.2 to accommodate a means by which an enclosure processor may be used to provide operator activity indication.

13.13.5.2.1 SAF-TE - Write Device Slot Status Modification

The length of the valid data for the SAF-TE Write Device Slot Status depends on the number of device slots (d) on this channel. There are three bytes of data for each device slot on the channel and the associated data structure is defined in Figure 237.

Bit/Byte	7	6	5	4	3	2	1	0
0	Slot 0 Byte 0							
1	Slot 0 Byte 1							
2	Slot 0 Byte 2							
…								
d*3-3	Slot d-1 Byte 0							
d*3-2	Slot d-1 Byte 1							
d*3-1	Slot d-1 Byte 2							
d*3	Vendor Specific							
…63								

Figure 237 – SAF-TE Write Device Slot Status data structure

The Serial ATA modification to the definition of the fields is as follows:

Slot *d* Byte 0 — As defined in SAF-TE

Slot *d* Byte 1 — Modified from definition in SAF-TE

- Bit 0-1 — As defined in SAF-TE
- Bit 2 — DR_ACT: Set to one by the host to indicate device activity when issuing commands to the device associated with this slot. (Defined as Reserved in the SAF-TE reference)
- Bit 3-7 — As defined in SAF-TE

Slot *d* Byte 2 — As defined in SAF-TE

If no flags are set in any byte for a device slot this is a NO CHANGE FROM CURRENT STATE indication. This allows a Host to change the state of one particular device slot without having to be aware of the current state of all device slots. Setting one or more flags requires that all flags be written to the correct binary value. Thus, changes should be preceded by a read of the corresponding device slot status and the Write Device Slot Status implemented as a "read-modify-write" operation.

13.13.5.2.2 SES- Device Element Definition Modification

The format of the CONTROL INFORMATION field for a device element type in the enclosure control page is defined in Figure 238. The data structure is modified with the addition of the device activity control bit (DR_ACT) to bit 7 of byte 2 (this bit is defined as Reserved in the SES reference).

Bits Bytes	7	6	5	4	3	2	1	0
0	COMMON CONTROL							
1	Reserved							
2	DR_ACT	DO NOT REMOVE	Reserved		RQST INSERT	RQST REMOVE	RQST IDENT	Rsrvd
3	Reserved		RQST FAULT	DEVICE OFF	ENABLE BYP A	ENABLE BYP B	Reserved	

Figure 238 – SES Device Element data structure

13.13.5.2.3 Activity Indication Behavior and Operation

The host sets the DR_ACT bit to one to set the external DRIVE ACTIVITY LED to ‘on’. In response, the SEP shall blink the associated LED at a vendor-specific rate. The LED shall blink for a duration of approximately 0.5 seconds after which this bit shall be automatically cleared to zero by the SEP (and the LED stops blinking).

13.13.5.3 Slot – to – Port Correspondence

For storage subsystems that do not have a direct one-to-one correspondence between host connection and device slot, a correspondence convention is required in order to ensure the host has a means for accurately controlling the proper enclosure slot for a particular storage device. Configurations that do not have direct correspondence include those that have intervening elements between the host and the device that compromise direct correspondence such as would be the case if Serial ATA Port Multipliers were used.

Figure 239 illustrates a configuration where there is no direct correspondence between host connection and enclosure device slot.

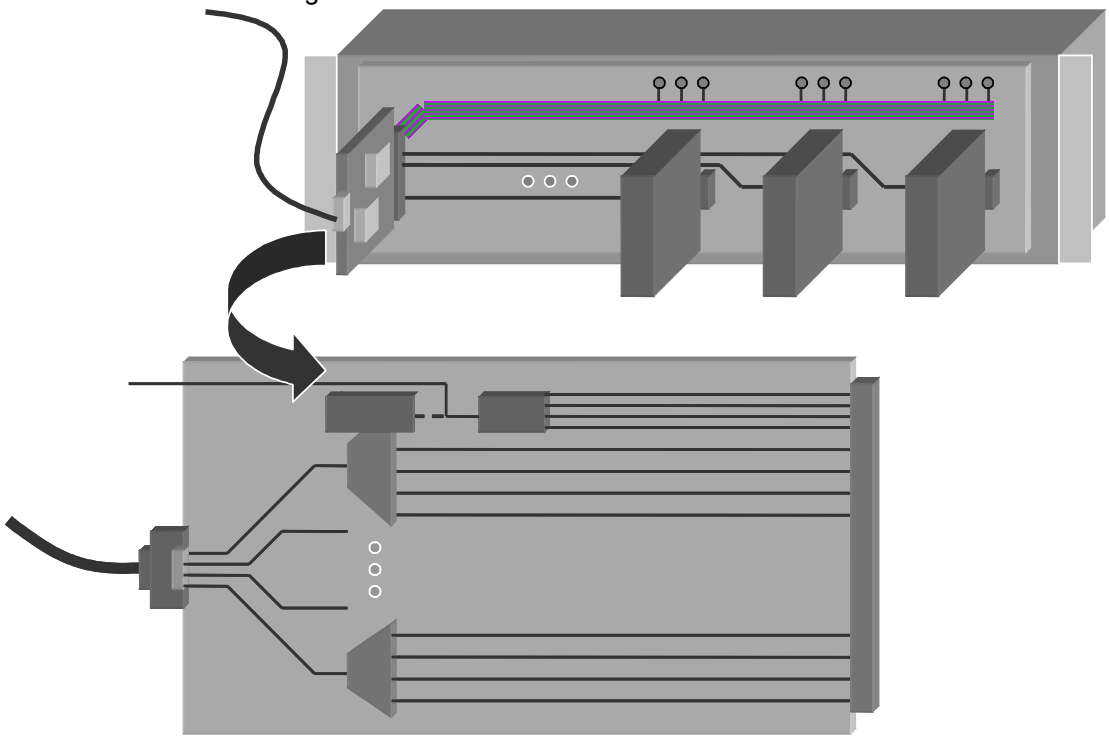


Figure 239 – Example Subsystem

13.13.5.3.1 SAF-TE Correspondence Definition (Optional)

Figure 240 defines the SAF-TE SCSI ID field convention for establishing correspondence between host connections and enclosure slots. SEPs that support SAF-TE may optionally adopt this convention. If an enclosure complies with this correspondence definition, it shall set bit 0 of byte 7 in the data returned for the Read Enclosure Configuration command to '1'. If an enclosure does not comply with this correspondence definition, it shall clear bit 0 of byte 7 in the data returned for the Read Enclosure Configuration command to '0'.

This correspondence convention, if implemented, shall be used for the SCSI ID in the Read Enclosure Status command where the SCSI ID for each device slot is returned to the host. The correspondence shall also be used by the host for the SCSI ID specified in the Set SCSI ID command. An empty or unused slot shall have a SCSI ID of FFh; a SCSI ID of FFh shall not be used to identify an active device slot.

Bits Field	7	6	5	4	3	2	1	0
SCSI ID	Port				Channel			

Figure 240 – SAF-TE Device ID field convention

Channel The Channel field corresponds to a host connection. For a given enclosure slot, it indicates which host connection the slot is associated with. In the instance where there is an intervening Port Multiplier, the Channel field indicates which host channel the upstream port of the associated Port Multiplier is connected to. The Channel value is assigned serially starting at zero and is indicated on the host connections packaging accordingly.

Port The Port field corresponds to a device connection. For a given enclosure slot, it indicates which port of an intervening multi-port controller the slot is associated with. In the instance where the intervening controller is a Port Multiplier, the Port field indicates which port of the Port Multiplier the slot is connected to and corresponds to the PM Port field used in the FIS to address the device. In the absence of an intervening multi-port controller, this field is zero.

For the configuration in Figure 239, the SCSI ID value corresponding to Slot 0 in the enclosure would be 33h, while the SCSI ID value corresponding to Slot 3 in the enclosure would be 03h.

13.13.5.3.2 SES Correspondence Definition (Optional)

Figure 241 defines the SES Slot Address field convention for establishing correspondence between host connections and enclosure slots. SEPs that support SES may optionally adopt this convention. The method for determining whether an enclosure complies with this correspondence definition is vendor specific or as defined by the SES-2 specification.

The SEP shall use this convention, if implemented, in the Status Information field for a device element type (element type 01h) in the enclosure status page. The first byte in this field for an element of device type is the Slot Address and shall be set as specified in Figure 241. The enclosure status page is read with the Receive Diagnostic Results command.

Bits Field	7	6	5	4	3	2	1	0
Slot Address	**Port**				**Channel**			

Figure 241 – SES Slot Address field convention

Channel The Channel field corresponds to a host connection. For a given enclosure slot, it indicates which host connection the slot is associated with. In the instance where there is an intervening Port Multiplier, the Channel field indicates which host channel the upstream port of the associated Port Multiplier is connected to. The Channel value is assigned serially starting at zero and is indicated on the host connections packaging accordingly.

Port The Port field corresponds to a device connection. For a given enclosure slot, it indicates which port of an intervening multi-port controller the slot is associated with. In the instance where the intervening controller is a Port Multiplier, the Port field indicates which port of the Port Multiplier the slot is connected to and corresponds to the PM Port field used in the FIS to address the device. In the absence of an intervening multi-port controller, this field is zero.

For the configuration in Figure 239, the Slot Address value corresponding to Slot 0 in the enclosure would be 33h, while the Slot Address value corresponding to Slot 3 in the enclosure would be 03h.

13.13.6 Enclosure Services Hardware Interface

For implementations where the enclosure/backplane is not bundled with the storage subsystem controller, it is recommended that the out-of-band enclosure services interface used by both the enclosure and the controller be I^2C. Section 13.13.6.1 defines the connector and cable interconnect between the enclosure/backplane and the storage subsystem controller.

13.13.6.1 I^2C Cable/Connector definition

Implementations where the storage subsystem controller is not directly connected to the enclosure/backplane require an interconnect between the backplane and the controller for the I^2C enclosure services bus. For example, a self-contained system that uses a PCI-based Serial ATA RAID controller and houses a backplane with front-panel accessible devices, requires a means by which the I^2C interface originating on the PCI controller is connected to the backplane that interfaces with the various storage devices and operator indicators (LEDs).

Products that utilize I^2C for enclosure services and desire interoperability/interchangeability with others' products shall use Molex Part # 22-43-6030 or equivalent as the bus connector. This connector has the same footprint/dimensions as the IPMB connector but its color is white instead of yellow as used by the actual IPMB connector.

13.13.6.2 SEP Discovery and Enumeration

For implementations where the SEP is an I^2C device and is not provided as part of a complete storage solution (i.e. is intended to communicate with a SEMB provided by another vendor), the SEP should have a dedicated I^2C interface with the SEMB and should not share the I^2C interface with enclosure sensors and indicators.

In order to enable the SEMB to efficiently discover and enumerate an attached SEP, the first SEP on an I^2C bus attached to a SEMB in a storage subsystem shall be assigned I^2C address C0h (note that the LSB of the I^2C address field is the R/W bit effectively resulting in the I^2C address field being 7 bits in length). A SEMB directly supports only a single SEP, so use of more than one SEP attached to the same SEMB in a storage subsystem is vendor specific. Any additional SEP(s) on the same I^2C interface in a storage subsystem shall be assigned consecutively higher I^2C addresses.

13.14 HDD Activity Indication (Optional)

Operator notification/indication of storage device status and activity may be driven by the host through the enclosure services facilities defined in Section 13.13 or through other host-driven means.
Operator notification/indication of storage device status and activity may also be driven through an intelligent processor such as an IO Processor. Reference the SAF-TE Write Device Slot Status command in the SAF-TE reference and section 13.13.5.2.1 for methods to support a SEP controlling HDD Activity Indication. As this reference suggests, these implementations may be vendor specific.

13.14.1 HDD Activity Emulation of Desktop Behavior

If the host controller optionally implements the desktop activity LED functionality, with the desired behavior to be compatible with current parallel ATA solutions, the host controller shall generate such a signal with the behavior defined in Figure 242.

```
//          POR - Power On Reset
//          HRESET - Hardware RESET
RESET = POR  || HRESET;


if ((BSY || SActive) && DEVICE_TYPE != ATAPI)       // How !ATAPI determined is implementation specific
          ACTIVITY_LED = ON;
else
          ACTIVITY_LED = OFF;


if (MASTER_SLAVE_EMULATION_ENABLED && SLAVE_PRESENT)
{
          if (RESET)
                    SLAVE_LED = ON;
}
else
          SLAVE_LED = OFF;


If (REGISTER_FIS_TRANSMITTED_ON_SLAVE_CHANNEL)
          SLAVE_LED = OFF;


LED = ACTIVITY_LED || SLAVE_LED;
```

Figure 242 – Activity LED definition for desktop behavior emulation

In the LED behavioral logic, the means by which an implementation may determine that the attached device type is ATAPI (see logic expression DEVICE_TYPE != ATAPI in logic behavioral definition) is implementation specific, and may include detection of device type based on reset signature, detection of the command opcodes issued to the device (i.e. ATAPI devices have command issued using the ATAPI command codes of A0h and A1h), or through other means. The required behavior is that activity to ATAPI devices not generate LED activity indication.

For implementations that have multiple Serial ATA channels, the controller should provide an aggregate activity signal that is the wired-OR of the individual activity signals from each channel.

13.14.1.1 Desktop HDD Activity Signal Electrical Requirements

For implementations that provide an activity signal in accordance with section 13.14.1, the signal shall be active low and shall be of an open collector/drain design. The voltage and current requirements/capabilities of the signal is vendor-specific.

13.14.2 Activity/Status Indication Reference (Informative)

Serial ATA controller devices may include discrete physical pins for the purpose of connecting device activity LEDs. Such controllers may provide one device activity pin per port and/or an aggregate activity signal. They may be included so that the host (or IO Processor) software need not supply a device activity status, or so that a SEP need not be burdened with blinking a LED when writes/reads to/from a device are occurring. These signals allow device activity LEDs without need for changes to the Serial ATA specification. It is likely that the methods outlined in this section may become obsolete as enclosure management facilities for Serial ATA mature and become readily available.

Figure 243 shows an example configuration for implementing device activity LEDs within an enclosure. In this example, it would be likely that the solution would use standard 0.1" headers that are common and widely available. Depending upon the number of devices, the standard header would vary in size, but would generally require two pins per device activity LED in the configuration. Thus, a 4-port solution with 4-device status LEDs would require a 2x4 pin standard header.

Figure 244 is identical in configuration to that shown in Figure 243, except that it shows the use of a ribbon cable for ease of assembly.

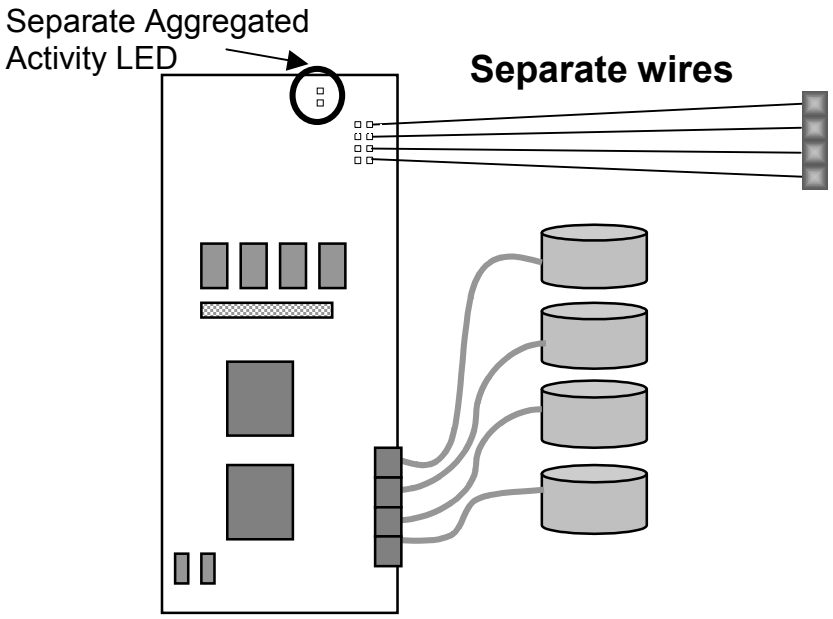


Figure 243 – Device Activity LEDs with Separate Wires

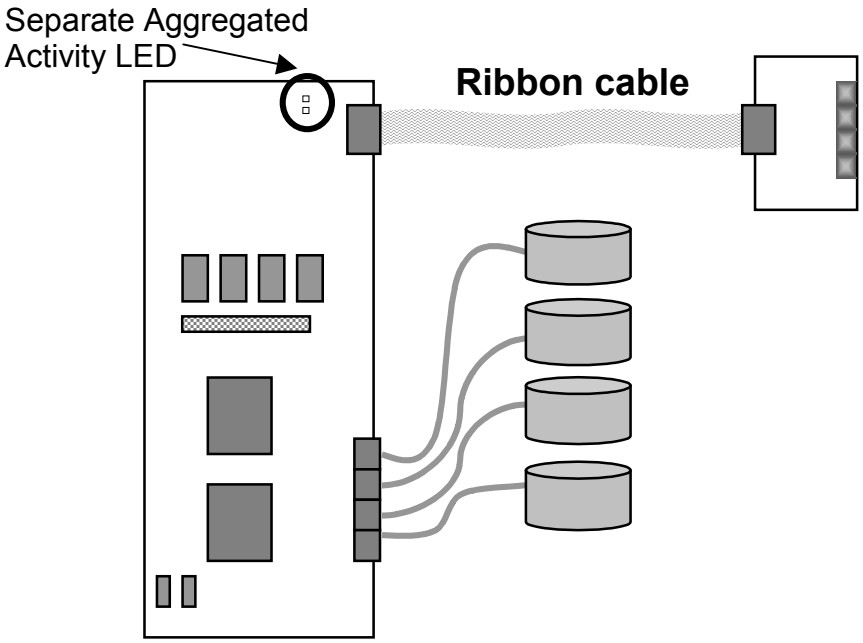


Figure 244 – Device Activity LEDs with Ribbon Cable

Note also the activity outputs have been designed to support wired-OR configurations. They may then support a configuration where all the outputs are connected together for the purpose of generating an aggregate activity indication.

Taking advantage of the wired-OR functionality, Figure 245 shows a more complex configuration that would support additional needs that may be required by an integrated system. In this example, the devices plug straight into a backplane. The device activity LEDs are placed on an separate, small mezzanine card that could wire-OR the LED activity signals from both the Serial ATA controller on the I/O Controller and the SEP, which in this example is located on the backplane. This would then allow normal device activity to be indicated as described in the preceding paragraphs, but would also support the concept of the SEP generating distinguishable visual patterns to the LEDs for other purposes. For example, an LED could be placed into a steady state ON by the SEP to indicate a failed card or a card that is targeted for hot-plug.

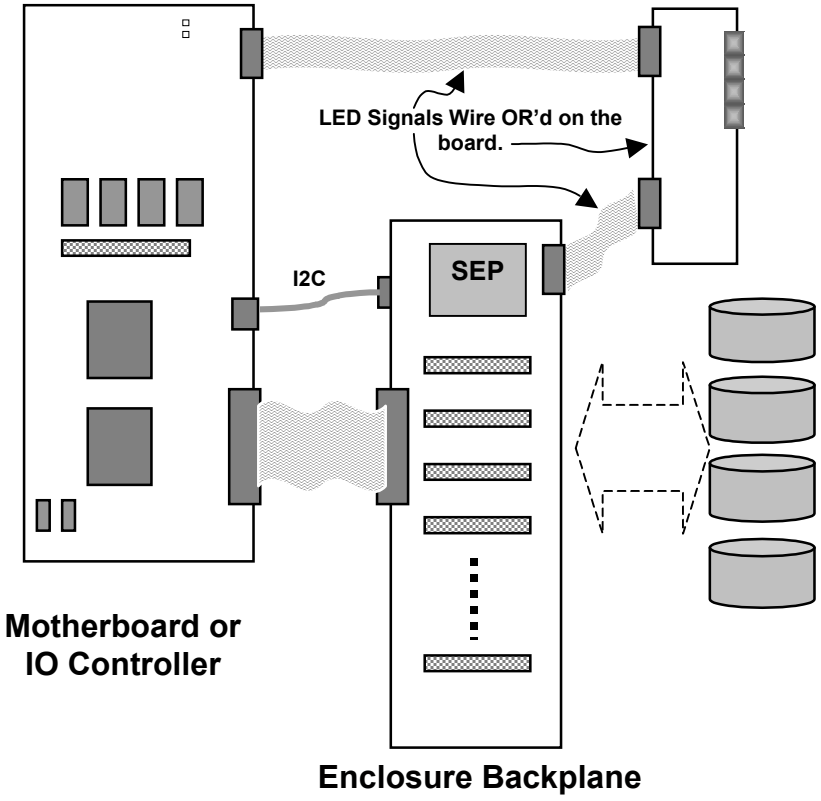


Figure 245 – Device Activity LEDs in a Storage Subsystem

Designers should be aware of some of the limitations of the solutions described in this section. If a solution is using a Port Multiplier, for example, then the LED might only be useful for general front end port activity, and might not identify a specific device on a specific Port Multiplier port for which an operator may be needing to take an action upon (such as device failed). Another consideration is that the signal generated by the Serial ATA controller may change states based upon certain states, and these state transitions may be too rapid to be readily detected by the operator. The LED pin may be negated if the BSY bit is set to one or if the transport state machine is active for example. Thus, in certain configurations where many short packets are being transmitted/received, the device may be active when the LED appears to be off if no additional methods are implemented to prevent this case from occurring.

13.15 Port Multiplier Discovery and Enumeration

13.15.1 Power-up

On power-up, the Port Multiplier shall enter state HPHP1:NoComm in the hot plug state machine for the host port, specified in section 16.3.3.5.1. Upon entering this state, the Port Multiplier shall:

1. Clear any internal state and reset all parts of the Port Multiplier hardware.
2. Place the reset values in all Port Multiplier registers, including port specific registers. The reset values shall disable all device ports.

After performing this sequence, the Port Multiplier shall proceed with the actions described in the hot plug state machine for the host port.

If the Port Multiplier receives a COMRESET from the host before issuing the initial COMINIT signal to the host, the Port Multiplier shall immediately perform the actions detailed in section 13.15.2.1 and cease performing the power-up sequence listed above.

13.15.2 Resets

There are three mechanisms to reset a device: COMRESET, software reset, and the DEVICE RESET command for PACKET devices.

To reset a Port Multiplier, the host shall issue a COMRESET. A Port Multiplier does not reset in response to a software reset or a DEVICE RESET command. The specific actions that a Port Multiplier takes in response to each reset type is detailed in the following sections.

13.15.2.1 COMRESET

When the Port Multiplier receives a COMRESET over the host port, the Port Multiplier shall enter state HPHP1:NoComm in the hot plug state machine for the host port, specified in section 16.3.3.5.1. Upon entering this state, the Port Multiplier shall:

1. Clear any internal state and reset all parts of the Port Multiplier hardware.
2. Place the reset values in all Port Multiplier registers, including port specific registers. The reset values shall disable all device ports.

After performing this sequence, the Port Multiplier shall proceed with the actions described in the hot plug state machine for the host port.

13.15.2.2 Software Reset

When the host issues a software reset to the control port, two Registers FISes are sent to the control port as a result. In the first Register FIS, the SRST bit in the Device Control register is set to one. In the second Register FIS, the SRST bit in the Device Control register is cleared to zero.

Upon receiving the Register FIS with the SRST bit asserted, the Port Multiplier shall wait for the Register FIS that has the SRST bit cleared to zero before issuing a Register FIS with the Port Multiplier signature to the host. The Port Multiplier's behavior shall be consistent with the Software reset protocol described in section 11.3. The values to be placed in the Register FIS are listed in Figure 246.

Register	7	6	5	4	3	2	1	0
Error	00h							
Count(7:0)	01h							
Count(15:8)	00h							
LBA(7:0)	01h							
LBA(31:24)	00h							
LBA(15:8)	69h							
LBA(39:32)	00h							
LBA(23:16	96h							
LBA(47:40)	00h							
Device	na							
Status	BSY	DRDY	DF	na	DRQ	0	0	ERR

Figure 246 – Software reset to control port result values

BSY = 0
DRDY = 1
DF = 0
DRQ = 0
ERR = 0

The Port Multiplier shall take no reset actions based on the reception of a software reset. The only action that a Port Multiplier shall take is to respond with a Register FIS that includes the Port Multiplier signature. To cause a general Port Multiplier reset, the COMRESET mechanism is used.

13.15.2.3 Device Reset

A device reset command issued to the control port shall be treated as an unsupported command by the Port Multiplier. Refer to section 16.3.3.8.6.

13.15.3 Software Initialization Sequences (Informative)

This section details the sequences that host software should take to initialize a Port Multiplier device.

13.15.3.1 Port Multiplier Aware Software

Port Multiplier aware software checks the host's SStatus register to determine if a device is connected to the port. If a device is connected to the port, the host then issues a software reset to the control port. If the Port Multiplier signature is returned, then a Port Multiplier is attached to the port. Then the host proceeds with enumeration of devices on Port Multiplier ports as detailed in section 13.15.4.2.

Port Multiplier aware software shall not require a device to be present on device port 0h in order to determine if a Port Multiplier is present.

13.15.3.2 Non-Port Multiplier Aware Software

Non-Port Multiplier aware software waits for the signature of the attached device to be returned to the host. If a device is present on device port 0h, the device connected to device port 0h returns a Register FIS to the host that contains its signature. If a device is not present on device port 0h, non-Port Multiplier aware software times out waiting for the signature to be returned and assumes that a device failure has occurred. If fast boot is a requirement, the system should have a Port Multiplier aware BIOS and Port Multiplier aware OS driver.

When non-Port Multiplier aware software is loaded, all device ports other than device port 0h are disabled. The host will only receive FISes from the device attached to device port 0h.

13.15.3.3 Boot Devices Connected to Port Multiplier

System designers should only connect multiple boot devices to a Port Multiplier if the BIOS is Port Multiplier aware. For example, in a system that contains three bootable devices (hard drive, CD-ROM, and DVD) these devices should only be attached to the Port Multiplier if the BIOS is Port Multiplier aware or the user cannot boot off of the devices that are not connected to device port 0h.

13.15.4 Port Multiplier Discovery and Device Enumeration (Informative)

13.15.4.1 Port Multiplier Discovery

To determine if a Port Multiplier is present, the host performs the following procedure. The host determines if communication is established on the host's Serial ATA port by checking the host's SStatus register. If a device is present, the host issues a software reset with the PM Port field set to the control port. The host checks the signature value returned and if it corresponds to the Port Multiplier Signature, the host knows that a Port Multiplier is present. If the signature value does not correspond to a Port Multiplier, the host may proceed with the normal initialization sequence for that device type. The host shall not rely on a device being attached to device port 0h to determine that a Port Multiplier is present.

When a Port Multiplier receives a software reset to the control port, the Port Multiplier shall issue a Register FIS to the host according to the procedure detailed in section 13.15.2.2. The signature value contained in the Register FIS is shown in Figure 247 .

Register	7	6	5	4	3	2	1	0
Error	na							
Count(7:0)	01h							
Count(15:8)	na							
LBA(7:0)	01h							
LBA(31:24)	na							
LBA(15:8)	69h							
LBA(39:32)	na							
LBA(23:16	96h							
LBA(47:40)	na							
Device	na							
Status	na							

Figure 247 – Port Multiplier Signature

13.15.4.1.1 Considerations when Port Multiplier Not Present

Directly attached devices may not be prepared to receive a software reset from the host prior to transmission of the Signature FIS.

It is recommended that host software not issue software reset prior to successful reception of the Signature FIS by the host, unless the host is Port Multiplier aware.

Devices may receive a software reset prior to transmission of the Signature FIS if the host is Port Multiplier aware. In the case that a device has not sent a Signature FIS and receives X_RDY_P from the host, a device may hold off responding with R_RDY_P until its application layer is in a state that is able to process a new command/control FIS from the HBA, including software reset. In some cases, the device may hold off transmission of R_RDY_P until it is prepared to transmit the Signature FIS.

13.15.4.2 Device Enumeration

After discovering a Port Multiplier, the host enumerates all devices connected to the Port Multiplier. The host reads GSCR[2], defined in section 16.4.1.1, to determine the number of device ports on the Port Multiplier.

For each device port on the Port Multiplier, the host performs the following procedure to enumerate a device connected to that port:

1. The host enables the device port. The host enables a device port by setting the DET field appropriately in the device port's PSCR[2] (SControl) register, as specified in section 14.1.3. The host uses the WRITE PORT MULTIPLIER command to write PSCR[2] (SControl) for the device port to be enabled.

2. The host should allow for communication to be established and device presence to be detected after enabling a device port. Refer to section 8.4.1 that describes the host PHY initialization sequence.

3. The host reads PSCR[0] (SStatus) for the device port using the READ PORT MULTIPLIER command. If PSCR[0] (SStatus) indicates that a device is present, the host queries PSCR[1] (SError) for the device port and clears the X bit indicating device presence has changed.

4. The signature generated by the device as a consequence of the initial COMRESET to the device port may be discarded if the host does not support context switching because the BSY

bit may be clear when the Register FIS is received by the host. Therefore, to determine the signature of the attached device, the host should issue a software reset to the device port. If a valid signature is returned for a recognized device, the host may then proceed with normal initialization for that device type.

Cascading Port Multipliers shall not be supported.

13.16 Automatic Partial to Slumber Transitions

It is possible that the host and device may be independently aware of conditions where they may transition to the Slumber state to save power without impacting performance (e.g., idle, seeking). To allow for increased power savings, Automatic Partial to Slumber Transitions defines a capability which allows the Phy to transition to Slumber from Partial without first entering Active.

The following are requirements of the Automatic Partial to Slumber Transitions capability:

1) Host Automatic Partial to Slumber shall only be used if the device reports host Automatic Partial to Slumber support (Word 76, Bit 13 of Identify Device).
2) Device Automatic Partial to Slumber shall only be used if the host supports an increased Partial exit latency (up to the max Slumber latency) and has enabled device Automatic Partial to Slumber transitions (via Set Features) on a device that has claimed device support (Word 76, Bit 14 of Identify Device)

Transitioning to Slumber from Partial does not require a COMWAKE followed by an Active state transition, and the respective attached Phy will not be aware of the transition. Since the respective Phy is in Partial it shall tolerate an increased Partial exit latency because the other Phy may be in Slumber.

14 Host adapter register interface

Serial ATA host adapters include an additional block of registers mapped separately and independently from the ATA Command Block Registers for reporting additional status and error information and to allow control of capabilities unique to Serial ATA. These additional registers, referred to as the Serial ATA Status and Control Registers (SCR's) 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 SCR's using the PCI mapping capabilities. Table 84 illustrates the overall organization of the Serial ATA register interface including both the ATA Command Block Registers and the Status and Control registers. Legacy mode software does not make use of the Serial ATA Status and Control registers. The Serial ATA Status and Control register are associated with the serial interface and are independent of any master/slave emulation the host adapter may implement.

Table 84 – SCR definition

ATA Command Block and Control Block registers

	A2	A1	A0	Read	Write
CS 0	0	0	0	Data	Data
	0	0	1	Error	Features
	0	1	0	Count(7:0) [15:8] [7:0]	Count(7:0) [15:8] [7:0]
	0	1	1	LBA Low [31:24] [7:0]	LBA Low [31:24] [7:0]
	1	0	0	LBA Mid [39:32] [15:8]	LBA Mid [39:32] [15:8]
	1	0	1	LBA High [47:40] [23:16]	LBA High [47:40] [23:16]
	1	1	0	Device	Device
	1	1	1	Status	Command
CS 1	1	1	0	Alternate Status	Device Control

Serial ATA Status and Control 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

14.1 Status and Control Registers

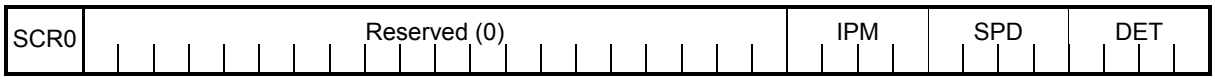
Serial ATA provides an additional block of registers to control the interface and to retrieve interface state information. There are 16 contiguous registers allocated of which the first five are defined and the remaining 11 are reserved for future definition. Table 85 defines the Serial ATA Status and Control registers.

Table 85 – SCR Definition

SCR[0]	SStatus register
SCR[1]	SError register
SCR[2]	SControl register
SCR[3]	SActive register
SCR[4]	SNotification register
SCR[5]	Reserved
…	…
SCR[15]	Reserved

14.1.1 SStatus register

The Serial ATA interface Status register - SStatus - 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.



DET The DET value indicates the interface device detection and Phy state.
- 0000b No device detected and Phy communication not established
- 0001b Device presence detected but Phy communication not established
- 0011b Device presence detected and Phy communication established
- 0100b 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
- 0000b No negotiated speed (device not present or communication not established)
- 0001b Gen1 communication speed negotiated
- 0010b Gen2 communication speed negotiated
- 0011b Gen3 communication speed negotiated

All other values reserved

IPM The IPM value indicates the current interface power management state
- 0000b Device not present or communication not established
- 0001b Interface in active state
- 0010b Interface in Partial power management state
- 0110b Interface in Slumber power management state

All other values reserved

Reserved All reserved fields shall be cleared to zero.

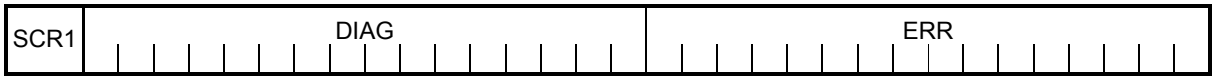
NOTE – The interface needs to be in the active state for the interface device detection value (DET field) to be accurate. When the interface is in the Partial or Slumber state no communication between the host and target is established resulting in a DET value corresponding to no device present or no 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 the Partial or Slumber state. This field alone may therefore be insufficient to satisfy all the requirements for device attach or detatch detection during all possible interface states.

NOTE – The SStatus register's IPM value may not correctly represent the current interface low power state (Partial or Slumber) when the host, device, or both are enabled to support Automatic Partial to Slumber transitions. The IPM value is guaranteed to indicate that the interface has entered a low power state, however, it may not represent the interface low power state of the host or device currently. Please refer to 13.16 for further information regarding Automatic Partial to Slumber transitions.

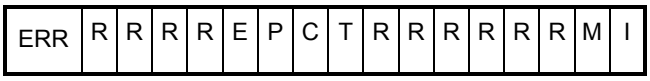
14.1.2 SError register

The Serial ATA interface Error register - SError - is a 32-bit register that conveys supplemental Interface error information to complement the error information available in the Shadow Register Block Error register. The register represents all the detected errors accumulated since the last time the SError register was cleared (whether recovered by the interface of 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.



ERR The ERR field contains error information for use by host software in determining the appropriate response to the error condition. The field is bit significant as defined in the following figure.



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 may 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 potentially step down the negotiated communication speed.

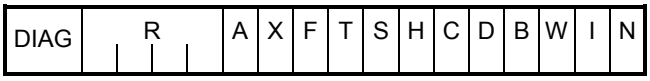
M Recovered communications error: Communications between the device and host was temporarily lost but was re-established. This may arise from a device temporarily being removed, from a temporary loss of Phy synchronization, or from other causes and may be derived from the PHYRDYn signal between the Phy 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 potentially step down the negotiated communication speed.

P Protocol error: A violation of the Serial ATA protocol was detected. This may 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 bit for future use: Shall be cleared to zero.

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. The field is bit significant as defined in the following figure.



A Port Selector presence detected: This bit is set to one when COMWAKE is received while the host is in state HP2: HR_AwaitCOMINIT. On power-up reset this bit is cleared to zero. The bit is cleared to zero when the host writes a one to this bit location.

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 to zero.

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 to zero.

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 to zero.

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 to zero.

N PHYRDY change: When set to one, this bit indicates that the PHYRDY signal changed state since the last time this bit was cleared to zero.

H Handshake error: When set to one, this bit indicates that one or more R_ERR_P 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 bit for future use: Shall be cleared to zero.

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 to zero. 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 to zero.

W COMWAKE Detected: When set to one this bit indicates that a COMWAKE signal was detected by the Phy since the last time this bit was cleared to zero.

X Exchanged: When set to one this bit indicates that device presence has changed since the last time this bit was cleared to zero. The means by which the implementation determines that the device presence has changed is vendor specific. This bit may be set to one anytime a Phy reset initialization sequence occurs as determined by reception of the COMINIT signal whether in response to a new device being inserted, in response to a COMRESET having been issued, or in response to power-up.

14.1.3 SControl register

The Serial ATA interface Control register - SControl - is a 32-bit read-write register that provides the interface by which software controls Serial ATA interface capabilities. Writes to the SControl register result in an action being taken by the host adapter or interface. Reads from the register return the last value written to it.

DET The DET field controls the host adapter device detection and interface initialization.

0000b No device detection or initialization action requested

0001b Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications reinitialized. Upon a write to the SControl register that sets the DET field to 0001b, the host interface shall transition to the HP1: HR_Reset state and shall remain in that state until the DET field is set to a value other than 0001b by a subsequent write to the SControl register.

0100b Disable the Serial ATA interface and put Phy in offline mode.
All other values reserved

SPD The SPD field represents the highest allowed communication speed the interface is allowed to negotiate when interface communication speed is established
0000b No speed negotiation restrictions
0001b Limit speed negotiation to a speed not greater than Gen1 communication speed
0010b Limit speed negotiation to a speed not greater than Gen2 communication speed
0011b Limit speed negotiation to a speed not greater than Gen3 communication speed
All other values reserved

IPM The IPM field represents the enabled interface power management states that may be invoked via the Serial ATA interface power management capabilities
0000b No interface power management state restrictions
0001b Transitions to the Partial power management state disabled
0010b Transitions to the Slumber power management state disabled
0011b Transitions to both the Partial and Slumber power management states disabled
All other values reserved

SPM The Select Power Management (SPM) field is used to select a power management state. A non-zero value written to this field shall cause the power management state specified to be initiated. A value written to this field is treated as a one-shot. This field shall be read as 0000b.
0000b No power management state transition requested
0001b Transition to the Partial power management state initiated
0010b Transition to the Slumber power management state initiated
0100b Transition to the active power management state initiated
All other values reserved

PMP The Port Multiplier Port (PMP) field represents the 4-bit value to be placed in the PM Port field of all transmitted FISes. This field is '0000' upon power-up. This field is optional and an HBA implementation may choose to ignore this field if the FIS to be transmitted is constructed via an alternative method.

Reserved All reserved fields shall be cleared to zero.

14.1.4 SActive register

The SActive register is a 32-bit register that conveys the information returned in the SActive field of the Set Device Bits FIS. If NCQ is not supported, then the SActive register does not need to be implemented.

The host may set bits in the SActive register by a write operation to the SActive register. The value written to set bits shall have ones encoded in the bit positions corresponding to the bits that are to be set. Bits in the SActive register are not cleared as a result of a register write operation by the host, and host software cannot directly clear bits in the SActive register.

Set bits in the SActive register are cleared as a result of data returned by the device in the SActive field of the Set Device Bits FIS. The value returned in the SActive field of the Set Device Bits FIS shall have ones encoded in the bit positions corresponding to the bits that are to be cleared in the SActive register. The device cannot set bits in the SActive register.

The host controller shall clear all bits in the SActive register to zero upon issuing a COMRESET signal or as a result of issuing a software reset by transmitting a Control Register FIS with the SRST bit set to one.

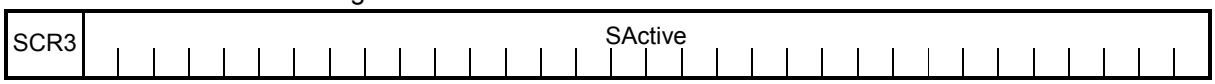


Figure 248 – SActive register definition

SActive For the Native Command Queuing protocol, the SActive value represents the set of outstanding queued commands that have not completed successfully yet. See section 13.6.

14.1.5 SNotification register (Optional)

The Serial ATA interface notification register - SNotification - is a 32-bit register that conveys the devices that have sent the host a Set Device Bits FIS with the Notification bit set, as specified in section 10.3.6. When the host receives a Set Device Bits FIS with the Notification bit set to one, the host shall set the bit in the SNotification register corresponding to the value of the PM Port field in the received FIS. For example, if the PM Port field is set to 7 then the host shall set bit 7 in the SNotification register to one. After setting the bit in the SNotification register, the host shall generate an interrupt if the Interrupt bit is set to one in the FIS and interrupts are enabled.

Set bits in the SNotification register are explicitly cleared by a write operation to the SNotification register, or a power-on reset operation. The register is not cleared due to a COMRESET, software is responsible for clearing the register as appropriate. The value written to clear set bits shall have ones encoded in the bit positions corresponding to the bits that are to be cleared.

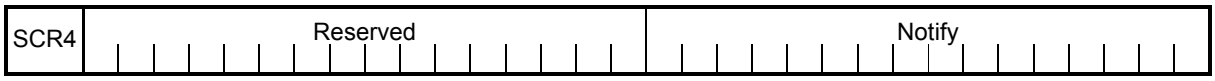


Figure 249 – SNotification register definition

Notify The field represents whether a particular device with the corresponding PM Port number has sent a Set Device Bits FIS to the host with the Notification bit set to one.

Reserved The reserved field shall be cleared to zero.

15 Error handling

15.1 Architecture

The layered architecture of Serial ATA extends to error handling as well. As indicated in Figure 250, each layer in the Serial ATA stack has as inputs error indication from the next lower layer (except for the Phy 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.

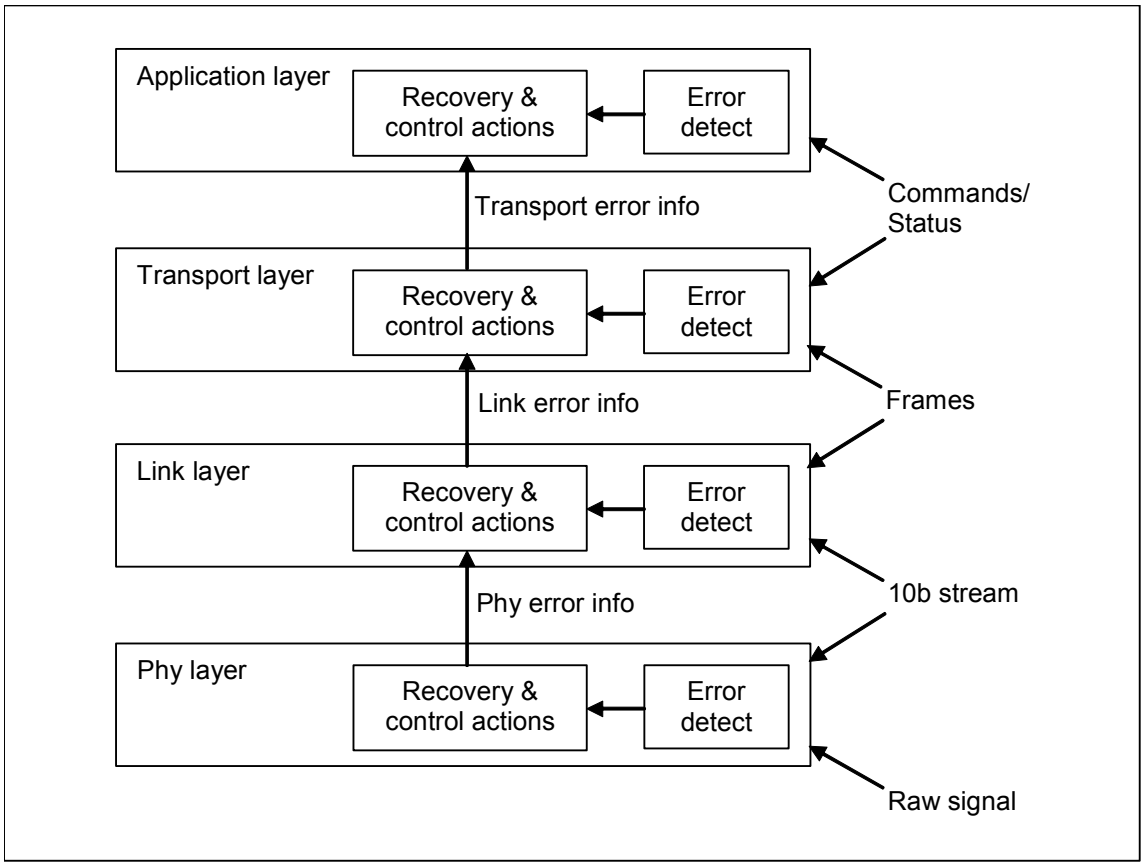


Figure 250 – Error handling architecture

Error responses are generally classified into four categories

Freeze
Abort
Retry
Track/ignore

The error handling responses described in this section are not comprehensive and are included to cover specific known error scenarios as well as to illustrate typical error control and recovery actions. This section is therefore descriptive and supplemental to the error reporting interface

defined in section 14.1.2 and implementations may vary in their internal error recovery and control actions.

For the most severe error conditions in which state has been critically perturbed in a way that it is not recoverable, the appropriate error response is to freeze and rely on a reset or similar operation to restore all necessary state 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 may 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.

15.2 Phy error handling overview

15.2.1 Error detection

There are three primary categories of error that the Phy layer detects internally:

- No device present
- OOB signaling sequence failure
- 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 controller and device, whether intermittent or persistent. The host controller shall detect device presence as part of the interface reset sequence defined in section 7.5.

An OOB signaling failure condition arises when the sequence of OOB signaling events cannot be completed, prior to the PHYRDY signal being asserted. OOB signaling sequences are required when emerging from a power management Partial or Slumber state, from a loopback BIST test state, or from initial power-up. OOB signaling sequences are used to achieve a specifically ordered exchange of COMRESET, COMINIT, COMWAKE, and $ALIGN_P$ patterns to bring the communications link up between host controller and device. The specific sequences are detailed in section 7.5.

A Phy Internal Error may 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 always results 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 speed frequency variations from the nominal expected speed, thus they usually have elasticity buffers. Elasticity buffers are used to accommodate the difference between input data speed frequency, and the local receive PLL frequency -- overrun/underrun conditions may 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 speed 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 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.

15.2.2 Error control actions

15.2.2.1 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 should respond appropriately.

During the interface initialization sequence, an internal state bit “Device Detect” shall be cleared in the host controller when the COMRESET signal is issued. The “Device Detect” state bit shall be set in the host controller when the host controller detects a COMINIT signal from the attached device. The “Device Detect” state bit may be set in the host controller when the host controller detects a COMWAKE signal while recovering from the Partial or Slumber state. The “Device Detect” state bit corresponds to the device presence detect information in the SStatus register defined in section 14.1.1.

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.

15.2.2.2 OOB signaling sequence failure

The Phy does not have any timeout conditions for the interface OOB reset signaling sequence as defined in section 8.1. If a device is present, the Phy shall detect device presence within 10 ms of a power-on reset (i.e. COMINIT shall be returned within 10 ms 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 shall be ready, active, and synchronized, and the SStatus register bits shall reflect this as defined in section 14.1.1.

15.2.2.3 Phy internal error

As described in section 15.2.1, 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 section 14.1.2. This requirement does not preclude each vendor from implementing their own level of error diagnostic bits, but those bits shall reside in vendor specific register locations.

Phy internal errors shall result in the Phy becoming not ready (the PHYRDY signal being negated) and the corresponding SStatus register and SError register bits shall be updated as defined in section 14.1.

The "PHYRDY change" bit, as defined in the SError register, shall be updated as per its definition in section 14.1.2

15.2.3 Error reporting

Phy errors are generally reported to the Link layer in addition to being reflected in the SStatus register and SError register as defined in section 14.1.

15.3 Link layer error handling overview

15.3.1 Error detection

There are two primary categories of errors that the Link layer detects internally:

Invalid state transitions
Data integrity errors

Invalid state transition errors may arise from a number of sources and the Link layer responses to many such error conditions are defined in section 1. Data integrity errors generally arise from noise in the physical interconnect.

15.3.2 Error control actions

Errors detected by the Link layer during a transmission are generally handled by accumulating the errors until the end of the transmission and reflecting the reception error condition in the final R_ERR_P /R_OK_P handshake. Specific scenarios are listed in the following sections.

15.3.2.1 Invalid state transitions

Invalid state transitions are generally handled through the return to a known state, for example where the Link state diagrams in section 9.6 define the responses for invalid state transition attempts. Returning to a known state is generally 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_P), the Link layer shall signal negative acknowledgement (R_ERR_P) to the transmitting agent.

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_P at the receiver interface, if the next control character received is not SOF_P, the Link layer shall notify the Transport layer of the condition and transition to the idle state.

Following transmission of X_RDY_P, if there is no returned R_RDY_P received, no Link layer recovery action shall be attempted. The higher-level layers should eventually time out, and reset the interface.

On receipt of an unexpected SOF_P, when the receiving interface had not yet signaled readiness to receive data with R_RDY_P, that receiving interface shall remain in the idle state issuing $SYNC_P$ primitives until the transmitting interface terminates the transmission and also returns to the idle state.

If the transmitter closes a frame with EOF_P and $WTRM_P$, and receives neither a R_OK_P nor an R_ERR_P within a predetermined timeout, no Link layer recovery action shall be attempted. The higher-level layers should eventually time out, and reset the interface.

If the transmitter signals EOF_P, and a primitive other than $SYNC_P$, R_OK_P, or R_ERR_P is received, the Link layer shall persistently continue to await reception of a proper terminating primitive.

Data integrity errors:
Data integrity errors are generally 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_P), 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_P, 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 shall retain this error information, and at the close of the received frame the Link layer shall provide the negative acknowledgement, R_ERR_P, as the frame handshake, and shall notify the Transport layer of the error.

The control actions are essentially the same for coding violations as for CRC errors.

15.3.3 Error reporting

Link layer error conditions are reported to the Transport layer via a private interface between the Link layer and Transport layer. Additionally, Link layer errors are reported in the SError register as defined section 14.1.2.

15.4 Transport layer error handling overview

The Transport layer is the highest level layer in the Serial ATA interface. The Transport layer communicates errors to the software and/or performs local error recovery, and initiates control actions, such as retrying a class of FIS transmissions

The Transport layer informs the Link layer of detected errors so that the Link layer reflects Transport errors in the R_ERR_P/R_OK_P handshake at the end of each frame. Devices shall reflect any R_ERR_P frame handshakes in the command ending status reflected in the transmitted Register FIS that conveys the operation ending status. The Transport layer also reflects any encountered error information in the SError register.

The Transport layer may retry any 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 should eventually time out the corresponding command and perform recovery operations.

15.4.1 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 generally 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 register frames that are not the correct length). Frame errors are generally 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 often stem from ill-behaved devices not following the proper Serial ATA 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 generally 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).

15.4.2 Error control actions

15.4.2.1 Internal errors

Internal errors are normally 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 layer 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_P 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_P and CRC value that is forced to be incorrect in order to ensure the receiver of the corrupted frame also executes appropriate error control actions.

15.4.2.2 Frame errors

Frame errors generally are 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 generally to fail the transaction and report the failure.

The following are specific frame error scenarios and their corresponding Transport layer error control actions.

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:

Register – Host to Device
Register – Device to Host
DMA Activate – Device to Host
DMA Setup – Device to Host
PIO Setup – Device to Host
Set Device Bits – Device to Host
BIST Activate – Host to Device or Device to Host

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 should never 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 should eventually time out the transaction and reset the interface.

If the Transport layer detects reception of a FIS with unrecognized TYPE value, the Transport layer shall signal the Link layer to negatively acknowledge the frame reception by asserting R_ERR_P 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 transmitting R_ERR_P during the frame acknowledgement handshake.

15.4.2.3 Protocol and state transition errors

Protocol and state errors stem from ill-behaved devices not following the Serial ATA protocol. Such errors are generally 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 layer error control actions.

If the PIO transfer count expires, and two Dwords later is not EOF_P (the CRC falls between the last data Dword and EOF_P), 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 transmitting R_ERR_P during the frame acknowledgement handshake.

15.4.3 Error reporting

The Transport layer reports errors to host software via the Serial ATA Status and Control registers. Devices communicate Transport layer error information to host software via transmitting a Register FIS to update the ATA Shadow Register Block Status and Error register values.

All Transport layer error conditions that are not handled/recovered by the Transport layer shall set the error bit in the Shadow Register Block Status register, and update the value in the Error register through transmission of an appropriate Register FIS.

Host Transport layer error conditions shall result in the status and error values in the SStatus and SError registers 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 frame acknowledgement handshake.

15.5 Application layer error handling overview

The Application layer error handling is in part defined by behavior of software written for Parallel ATA. Superset error reporting capabilities are supported by the Transport layer through the Status and Control registers, and software should take advantage of those error reporting capabilities to improve error handling for Serial ATA.

15.5.1 Error detection

There are three overall error detection mechanisms by which software identifies and responds to Serial ATA errors

- Bad status in the Command Block Status register
- Bad status in the SError register
- Command failed to complete (timeout)

Conditions that return bad status in the Command Block Status register, but for which no Serial ATA interface error information is available, correspond to the error conditions specified in the ATA standard. Such error conditions and responses are defined in the ATA standard and there is no unique handling of those in Serial ATA. 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 sector number. These failures are not related to the Serial ATA interface, and thus no Serial ATA 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 section 14.1.2. 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 Shadow Register Block Status and Error registers through transmission of appropriate Register FISes.

Commands that fail to complete are detected by host driver software through a timeout mechanism. Generally such timeouts result in no 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.

15.5.2 Error control actions

Conditions that return bad status in the Shadow Register Block Status register but for which no interface error information in the SError register is available shall be handled as defined in the ATA standard.

Conditions that return interface error information in the SError register are handled through four basic responses:

Freeze
Abort/Fail
Retry (possible after reset)
Track/ignore

Error conditions that result in catastrophic system perturbation that is not recoverable should result in the system halting. Serial ATA does not define any explicit halting conditions, however, software may be able to infer such conditions.

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 speed.

16 Port Multiplier

16.1 Introduction

A Port Multiplier is a mechanism for one active host connection to communicate with multiple devices. A Port Multiplier may be thought of as a simple multiplexer where one active host connection is multiplexed to multiple device connections, as shown in Figure 251.

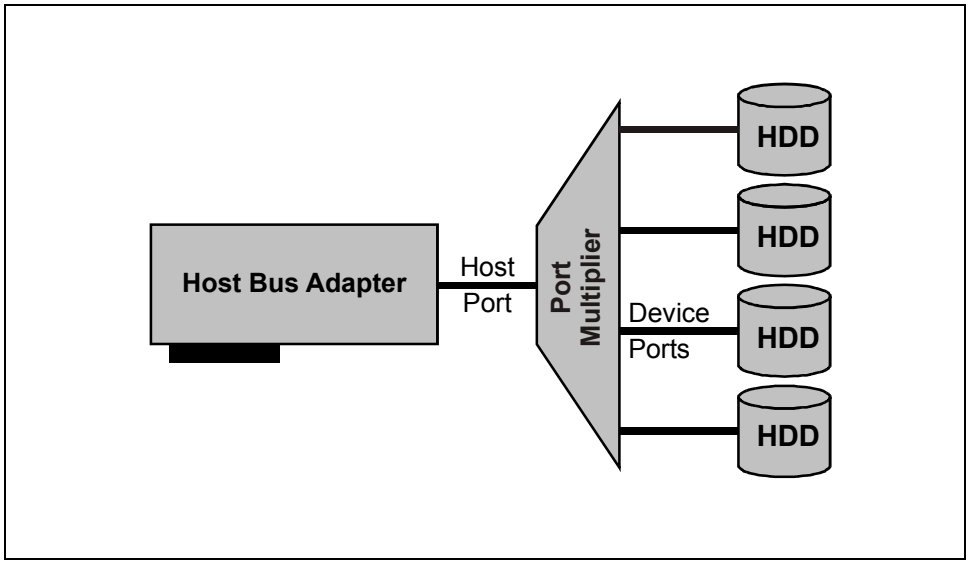


Figure 251 – Port Multiplier Overview

Only one active host connection to the Port Multiplier is supported. The Port Multiplier is an extensible design that supports up to 15 endpoint device connections and utilizes the full bandwidth of the host connection. A Port Multiplier shall not be connected to another Port Multiplier (i.e. no cascading).

16.2 Overview

The Port Multiplier uses four bits, known as the PM Port field, in all FIS types to route FISes between the host and the appropriate device. Using the PM Port field, the Port Multiplier routes FISes to up to 15 Serial ATA devices from one active host. The PM Port field is filled in by the host on a host-to-device FIS with the port address of the device to route the FIS to. For a device-to-host FIS, the PM Port field is filled in by the Port Multiplier with the port address of the device that is transmitting the FIS. Device port addresses start at zero and are numbered sequentially higher until the last device port address has been defined. The control port, port address Fh, is used for control and status communication with the Port Multiplier itself.

In order to utilize all devices connected to a Port Multiplier, the host needs to have a mechanism to set the PM Port field in all transmitted FISes.

The Port Multiplier maintains a set of general purpose registers and also maintains the Serial ATA superset Status and Control registers for each device port. The control port supports two commands, READ PORT MULTIPLIER and WRITE PORT MULTIPLIER, that are used to read and write these registers.

Some additional Port Multiplier features include:

- Supports booting with legacy mode software on device port 0h.
- Supports staggered spin-up.
- Supports hot plug.

16.3 Definition

16.3.1 Addressing Mechanism

The Port Multiplier uses four bits, known as the PM Port field, in all FIS types to route FISes between the host and the appropriate device. Using the PM Port field, the Port Multiplier routes FISes to up to 15 Serial ATA devices from one active host. The PM Port field is filled in by the host on a host-to-device FIS with the port address of the device to route the FIS to. For a device-to-host FIS, the PM Port field is filled in by the Port Multiplier with the port address of the device that is transmitting the FIS.

The PM Port field directly follows the FIS Type field in all FISes, refer to section 10.3.

16.3.2 Device Port Requirements

A device port is a port that may be used to connect a device to the Port Multiplier. The Port Multiplier may support up to 15 device ports. The device port addresses shall start at zero and shall be numbered sequentially until all device ports have port addresses. A valid device port shall have a port address that has a value less than the total number of device ports supported by the Port Multiplier.

16.3.2.1 Transmission from Host to Device

To transmit a FIS to a device connected to a Port Multiplier, the host shall set the PM Port field in the FIS to the device's port address. Then the host shall start transmitting the FIS to the Port Multiplier in accordance with the Transport, Link, and Phy state machines.

When a Port Multiplier receives a FIS over the host port, the Port Multiplier shall check the PM Port field in the FIS to determine the port address that the FIS should be transmitted over. If the FIS is destined for the control port, the Port Multiplier shall receive the FIS and perform the command or operation requested. If the FIS is destined for a device port, the Port Multiplier shall perform the following procedure:

1. The Port Multiplier shall determine if the device port is valid. If the device port is not valid, the Port Multiplier shall issue a SYNC Escape to the host and terminate reception of the FIS. Refer to section 16.3.3.8.3.

2. The Port Multiplier shall determine if the X bit in the DIAG field of the device port's PSCR[1] (SError) register is set to one. If set to one, the Port Multiplier shall issue a SYNC Escape to the host and terminate reception of the FIS. Refer to section 16.3.3.8.2.

3. The Port Multiplier shall determine if a collision has occurred. A collision is when a reception is already in progress from the device that the host wants to transmit to. If a collision has occurred, the Port Multiplier shall finish receiving the FIS from the host and shall issue R_ERR_P to the host as the ending status. The Port Multiplier shall follow the procedures outlined in section 16.3.3.2 to clear the collision condition.

4. The Port Multiplier shall initiate the transfer with the device by issuing X_RDY_P to the device. A collision may occur as the Port Multiplier is issuing the X_RDY_P to the device if the device has just decided to start a transmission to the host. If the device starts

transmitting X_RDY_P to the Port Multiplier, a collision has occurred. If a collision has occurred, the Port Multiplier shall finish receiving the FIS from the host and shall issue R_ERR_P to the host as the ending status. The Port Multiplier shall follow the procedures outlined in section 16.3.3.2 to clear the collision condition.

5. After the device issues R_RDY_P to the Port Multiplier, the Port Multiplier shall transmit the FIS from the host to the device. The Port Multiplier shall not send R_OK_P status to the host until the device has issued an R_OK_P for the FIS reception. The R_OK_P status handshake shall be interlocked from the device to the host. Refer to section 16.3.3.1.

If an error is detected during any part of the FIS transfer, the Port Multiplier shall ensure that the error condition is propagated to the host and the device.

The transfer between the host and Port Multiplier is handled separately from the transfer between the Port Multiplier and device; only the end of frame R_OK_P handshake is interlocked. The Port Multiplier shall ensure that the flow control signaling latency requirement specified in section 9.4.7 is met for all FIS transfers on a per link basis. Specifically, the Port Multiplier shall ensure that the flow control signaling latency is met between:

1. The host port and the host it is connected to
2. Each device port and the device that it is connected to

If there is not an error detected during the FIS transfer, the Port Multiplier shall not alter the FIS transmitted to the device. The Port Multiplier is not required to check or recalculate the CRC.

16.3.2.2 Transmission from Device to Host

To transmit a FIS to the host, the device shall proceed with the transmission in accordance with the Transport, Link, and Phy state machines. The device behavior is the same whether it is connected directly to the host or is connected to the host via a Port Multiplier.

When a device wants to transmit a FIS to the host, the Port Multiplier shall perform the following procedure:

1. After receiving X_RDY_P from the device, the Port Multiplier shall determine if the X bit is set in the DIAG field of the device port's PSCR[1] (SError) register. The Port Multiplier shall not issue R_RDY_P to the device until the X bit in the DIAG field is cleared to zero.

2. The Port Multiplier shall receive the FIS from the device. The Port Multiplier shall fill in the PM Port field with the port address of the transmitting device. The Port Multiplier shall transmit the modified FIS to the host with a recalculated CRC. The Port Multiplier shall check the CRC received from the device. If the CRC from the device is invalid the Port Multiplier shall corrupt the CRC sent to the host to ensure that the error condition is propagated. Refer to section 16.3.3.8.4 for specific details on how the Port Multiplier corrupts the CRC in this error case.

3. The Port Multiplier shall issue X_RDY_P to the host to start the transmission of the FIS to the host. After the host issues R_RDY_P to the Port Multiplier, the Port Multiplier shall transmit the FIS from the device to the host. The Port Multiplier shall not send R_OK_P status to the device until the host has issued an R_OK_P for the FIS reception. The R_OK_P status handshake shall be interlocked from the device to the host. Refer to section 16.3.3.1.

If an error is detected during any part of the FIS transfer, the Port Multiplier shall ensure that the error condition is propagated to the host and the device.

The Port Multiplier may wait for an X_RDY_P/R_RDY_P handshake with the host prior to issuing an R_RDY_P to the device to minimize buffering. The transfer between the device and Port Multiplier

is handled separately from the transfer between the Port Multiplier and the host; only the end of frame R_OK_P handshake is interlocked. The Port Multiplier shall ensure that the flow control signaling latency requirement specified in section 9.4.7 is met for all FIS transfers on a per link basis. Specifically, the Port Multiplier shall ensure that the flow control signaling latency is met between:

1. The host port and the host it is connected to.
2. Each device port and the device that it is connected to.

16.3.3 Policies

16.3.3.1 FIS Delivery

The end of frame handshake shall be interlocked between the host and the device. Specifically, the Port Multiplier shall not issue an R_OK_P to the initiator of a FIS before the target of a FIS has issued an R_OK_P. The Port Multiplier shall propagate R_OK_P and R_ERR_P from the target of the FIS to the initiator of a FIS.

If a transmission fails before the R_OK_P handshake is delivered to the initiator, the Port Multiplier is responsible for propagating the error condition. Specifically, the Port Multiplier shall propagate $SYNC_P$ primitives received during a FIS transmission end-to-end to ensure that any error condition encountered in the middle of a FIS is propagated. Reference sections 16.3.3.8.4 and 16.3.3.8.5 on the appropriate actions to take when CRC calculation errors or possible data corruption occurs in a FIS transmission.

If there is a FIS transfer ongoing and the link between the Port Multiplier and the active device becomes inoperable, the Port Multiplier should issue $SYNC_P$ primitives to the host until the host responds with $SYNC_P$ in order to fail the transfer. Failing the transfer upon detecting an inoperable link allows the host to proceed with recovery actions immediately, thereby eliminating latency associated with a timeout.

16.3.3.1.1 Port Priority

The Port Multiplier shall ensure that an enabled and active device port is not starved. The specific priority algorithm used is implementation specific.

The control port shall have priority over all device transfers. While a command is outstanding to the control port, no device transmissions shall be started by the Port Multiplier until the command outstanding to the control port is completed.

16.3.3.1.2 FIS Delivery Mechanisms (Informative)

This section provides an informative reference for one method that a Port Multiplier may use to satisfy the FIS Delivery policies outlined.

Starting a FIS Transmission: If a device on a Port Multiplier asserts X_RDY_P and the Port Multiplier has selected that device for transmission next and the host port is not busy, the Port Multiplier shall:

1. Issue X_RDY_P to the host.
2. Wait for the host to respond with R_RDY_P.
3. After the host issues R_RDY_P, the Port Multiplier issues R_RDY_P to the device.

Then the transmission to the host may proceed.

If the host asserts X_RDY_P to the Port Multiplier and the Port Multiplier does not have X_RDY_P asserted to the host, the Port Multiplier responds with R_RDY_P to the host. The Port Multiplier receives the first Dword of the FIS payload from the host. If the Port Multiplier Port specified is a

device port that is enabled on the Port Multiplier, the Port Multiplier issues an X_RDY_P over the device port specified and proceeds to transmit the entire FIS to the device. If a collision occurs during this process, the Port Multiplier shall follow the procedures outlined in section 16.3.3.2.

Status Propagation: If there is an on-going FIS transmission between the host and a device, the Port Multiplier only issues R_OK_P, R_ERR_P and $SYNC_P$ if it has first received that primitive from the host or device, unless a collision occurs or an invalid port is specified.

The Port Multiplier shall not convey R_OK_P or R_ERR_P to the initiator of a FIS until the target of the FIS has issued R_OK_P or R_ERR_P once the end-to-end transmission has commenced.

If the initiator or target of a FIS transmission issues $SYNC_P$ during a FIS transfer, this primitive shall be propagated in order to ensure that the error condition is propagated to either end.

16.3.3.2 Collisions

A collision is when the Port Multiplier has already started a reception from the device that the host wants to transmit to. A collision also occurs when the device issues X_RDY_P at the same time that the Port Multiplier is issuing X_RDY_P to that device. All collisions are treated as an X_RDY_P/X_RDY_P collision; in accordance with the Link layer state machine, the host loses all such collisions and should retransmit its FIS at a later time.

A collision only occurs when the host is trying to issue another native queued command to a device that has native queued commands outstanding. The Native Command Queuing protocol guarantees that the host is never transmitting a Data FIS when the collision occurs. This means that the Port Multiplier may safely issue an error to the host and that the host should retry the failed FIS transmission at a later point in time.

When the Port Multiplier detects a collision, the Port Multiplier shall finish reception of the FIS from the host and shall issue an R_ERR_P to the host for the end of frame handshake. The Port Multiplier shall discard the FIS received from the host. The host should attempt to retry the FIS transfer that failed.

Table 86 – State Diagram Collisions

PMC1: PmColl_Idle	Perform normal operation. Wait for FIS reception from host.		
1. Reception started from host for device port X.		→	PmColl_ChkRecv
2. Reception not started from host for device port X.		→	PmColl_Idle
PMC2: PmColl_ChkRecv	Determine if a FIS reception is in progress on device port X.		
1. Reception from device port X in progress.		→	PmColl_Collision
2. Reception from device port X not in progress.		→	PmColl_XmitToDP
PMC3: PmColl_Collision	Collision is detected. Finish reception of the FIS from the host and discard the FIS contents.		
1. $WTRM_P$ or $SYNC_P$ not received from host.		→	PmColl_Collision
2. $WTRM_P$ received from host.		→	PmColl_FailHostFIS
3. $SYNC_P$ received from host.		→	PmColl_Idle

PMC4: PmColl_FailHostFIS	Transmit R_ERR_P to the host.		
	1. Unconditional	→	PmColl_Idle

PMC5: PmColl_XmitToDP	Transmit X_RDY_P to the device port X.		
	1. X_RDY_P or R_RDY_P not received from device port X.	→	PmColl_XmitToDP
	2. X_RDY_P received from device port X.	→	PmColl_Collision
	3. R_RDY_P received from device port X.	→	PmColl_Idle

X_RDY_P/X_RDY_P collisions that occur on the host port shall be handled in accordance with the Link layer state diagrams in section 9.6.

16.3.3.3 Booting with Software That Is Not Port Multiplier Aware

Booting is accommodated off of the first port of the Port Multiplier, device port 0h, without any special software or hardware support. An HBA that does not support attachment of a Port Multiplier shall work with the device on device port 0h.

If a system requires fast boot capability it should ensure that both the BIOS and OS driver software are Port Multiplier aware. If software that is not Port Multiplier aware is used in the presence of a Port Multiplier and there is no device present on device port 0h, that software detects a device as present but never receives a Register FIS with the device signature. Waiting for the device signature may cause a BIOS that is not Port Multiplier aware to not meet fast boot timing requirements.

16.3.3.4 Staggered Spin-Up Support

The Port Multiplier shall disable all device ports on power-up or upon receiving a COMRESET signal over the host port. This feature allows the host to control when each device spins up. Refer to section 13.15.4.2 on how to enumerate a device, including devices that may be spun down because the port is disabled.

16.3.3.5 Hot Plug Events

Port Multiplier handling of hot plug events is defined by the hot plug state machines for the host port and device port, in sections 16.3.3.5.1 and section 16.3.3.5.2 respectively.

Upon receiving a COMRESET signal from the host, the Port Multiplier shall perform an internal reset. As part of the internal reset, the Port Multiplier shall update the Serial ATA superset Status and Control registers for each device port. A more detailed description of COMRESET handling is in section 13.15.2.1.

Upon receiving a COMINIT signal from a device, the Port Multiplier shall update the Serial ATA superset Status and Control registers for that port as defined in section 13.16. The Port Multiplier shall set the X bit in the DIAG field of the device port's PSCR[1] (SError) register to mark that device presence has changed as defined in section 14.1.2. If the Port Multiplier has not received a FIS for the control port after the last COMRESET and an unsolicited COMINIT signal was received over device port 0h, the Port Multiplier shall propagate the COMINIT signal to the host as specified in the hot plug state machine for the host port in section 16.3.3.5.1. For all other cases, the COMINIT signal shall not be propagated to the host.

If the X bit in the DIAG field of the PSCR[1] (SError) register is set for a device port, the Port Multiplier shall disallow FIS transfers with that port until the X bit has been cleared. When the Port Multiplier is disallowing FIS transfers with a device port, the Port Multiplier shall ensure FISes the device is attempting to transmit are not dropped.

It is recommended that host software frequently query GSCR[32], defined in section 16.4.1.2, to determine if there has been a device presence change on a device port.

16.3.3.5.1 Hot Plug State Machine for Host Port

Cabled hot plug of the Port Multiplier host port shall be supported since Port Multipliers may not share the same power supply as the host. Therefore the Port Multiplier shall periodically poll for host presence by sending periodic COMINIT signals to the host after transitioning to the HPHP1:NoComm state.

The state machine enables device port 0h after communication has been established over the host port and also clears the X bit in the DIAG field of the PSCR[1] (SError) register for device port 0h after it is set when operating with software that is not Port Multiplier aware. In addition, the state machine shall propagate unsolicited COMINIT signals received from device port 0h to the host when no control port register accesses have yet occurred. These accommodations allow device port 0h to work with host software that is not Port Multiplier aware.

When operating with Port Multiplier aware software, the state machine treats device port 0h exactly like every other device port.

Table 87 – State Diagram Hot Plug State Machine for Host Port

HPHP1: NoComm[1]	Set Port Multiplier to initial state as specified in section 13.15.1 and section 13.15.2.1.	
1. Unconditional	→	StartComm
NOTES: 1. This state is entered upon power-on reset or in response to a received COMRESET.		

HPHP2: StartComm	Transition Phy state machine to state DP1: DR_RESET. Communications retry interval reset to initial value.[1]	
1. Unconditional	→	WaitComm
NOTES: 1. The communications retry interval shall be greater than or equal to 10 ms.		

HPHP3: WaitComm		
1. PHYRDY asserted.	→	EnablePort0
2. PHYRDY not asserted and communications retry interval not expired.	→	WaitComm
3. PHYRDY not asserted and communications retry interval expired.	→	NoComm

HPHP4: EnablePort0	Enable device port 0h if device port 0h is currently disabled.	
1. PHYRDY asserted.	→	Port0ComInit
2. PHYRDY not asserted.	→	NoComm

HPHP4b: Port0ComInit		
1. Communications lost and interface not in power management state (i.e., unplug) and FIS not received for control port.	→	NoComm
2. FIS received for control port.	→	CommOK[1]
3. Communications established and FIS not received for control port and (COMINIT received from device port 0h or >= 10ms elapsed since entry into HPHP4b:Port0ComInit state).	→	Port0ComInitWait
4. Communications established and FIS not received for control port and COMINIT not received from device port 0h and < 10 ms elapsed since entry into HPHP4b:Port0ComInit state.	→	Port0ComInit
NOTES: 1. If bit 0 of the DET field in PSCR[0] (SStatus) for Port 0 is set to one then the X bit in the DIAG field of PSCR[1] (SError) for Port 0 shall be set prior to making the transition.		

HPHP4c: Port0ComInitWait		
1. Communications lost and interface not in power management state (i.e. unplug) and FIS not received for control port.	→	NoComm
2. FIS received for control port.	→	CommOK[1]
3. Communications established and FIS not received for control port and COMINIT negated from device port 0h.	→	LegacyCommOK[2]
4. Communications established and FIS not received for control port and COMINIT asserted from device port 0h.	→	Port0ComInitWait
NOTES: 1. If bit 0 of the DET field in PSCR[0] (SStatus) for Port 0 is set to one then the X bit in the DIAG field of PSCR[1] (SError) for Port 0 shall be set prior to making the transition. 2. The X bit in PSCR[1] (SError) for device port 0h shall be cleared to zero prior to making the transition.		

HPHP5: LegacyCommOK		
1. Communications lost and interface not in power management state (i.e. unplug) and FIS not received for control port.	→	NoComm
2. Communications established or interface in power management state and COMINIT not received on device port 0h and FIS not received for control port.	→	LegacyCommOK
3. Communications established or interface in power management state and COMINIT received on device port 0h and FIS not received for control port.	→	StartComm[1]
4. FIS received for control port.	→	CommOK[2]
NOTES: 1. The X bit in PSCR[1] (SError) for Port 0 shall be set to one prior to making the transition. 2. If bit 0 of the DET field in PSCR[0] (SStatus) for Port 0 is set to one then the X bit in the DIAG field of PSCR[1] (SError) for Port 0 shall be set to one prior to making the transition.		

HPHP6: CommOK		
1. Communications lost and interface not in power management state (i.e. unplug).	→	NoComm
2. Communications established or interface in power management state.	→	CommOK

16.3.3.5.2 Hot Plug State Machine for Device Port

The device port hot plug behavior is exactly the same as the behavior for a host controller supporting device hot plug directly. There is no change to the device and there is no change to the usage model.

Table 88 – State Diagram Hot Plug State Machine for Device Port

DPHP1: NoComm		
1. PHYRDY asserted.	→	CommOK
2. PHYRDY not asserted.	→	NoComm

DPHP2: CommOK		
1. Communications lost and interface not in power management state.	→	NoComm
2. Communications established or interface in power management state.	→	CommOK

16.3.3.6 Link Power Management

The Port Multiplier shall support reception of PMREQ_P_P and PMREQ_S_P from the host and from attached devices, in accordance with the Link layer state machine. If the Port Multiplier does not support the power management state requested, the Port Multiplier shall respond to the PMREQ_P_P or PMREQ_S_P with $PMNAK_P$.

Before the Port Multiplier delivers a FIS, the Port Multiplier shall check the state of the link and issue a COMWAKE signal if the link is in Partial or Slumber state. The Port Multiplier shall accurately reflect the current state of each device port link in the port specific registers (specifically PSCR[0] (SStatus) for each port) defined in section 16.4.2.

The Port Multiplier shall not propagate PMREQ_P$_P$ or PMREQ_S$_P$ received on a device port. PMREQ_P$_P$ or PMREQ_S$_P$ received from a device only affects the link between that device and the Port Multiplier. If the Port Multiplier receives PMREQ_P$_P$ or PMREQ_S$_P$ over a device port, the Port Multiplier shall perform the following actions:

- The Port Multiplier shall respond to the device with PMACK$_P$ or PMNAK$_P$
- If the Port Multiplier responds to the device with PMACK$_P$:
    - The Port Multiplier shall transition the link with that device to the power state specified
    - The Port Multiplier shall update the port specific registers for that device port to reflect the current power management state of that link

The Port Multiplier shall propagate PMREQ_P$_P$ or PMREQ_S$_P$ received from the host to all active device ports if the Port Multiplier responds to PMREQ_P$_P$ or PMREQ_S$_P$ with PMACK$_P$. In this case, the Port Multiplier shall propagate PMREQ_P$_P$ or PMREQ_S$_P$ to all device ports that have PHYRDY asserted. If a device responds to PMREQ_P$_P$ or PMREQ_S$_P$ with PMNAK$_P$, this event shall only affect the link with that device. The host may interrogate the Port Multiplier port specific registers to determine which device ports are in a power managed state.

If the Port Multiplier receives PMREQ_P$_P$ or PMREQ_S$_P$ from the host, the specific actions the Port Multiplier shall take are:

- The Port Multiplier shall respond to the host with PMACK$_P$ or PMNAK$_P$
- If the Port Multiplier responds to the host with PMACK$_P$:
    - The Port Multiplier shall transition the link with the host to the power state specified
    - The Port Multiplier shall issue PMREQ_P$_P$ or PMREQ_S$_P$ to all device ports that have PHYRDY asserted
        - If a device responds with PMACK$_P$, the Port Multiplier shall transition the link with that device to the power state specified and shall update the Port Multiplier port specific registers for that port
        - If a device responds with PMNAK$_P$, the Port Multiplier shall not take any action with that device port

The Port Multiplier shall wake device links on an as-needed basis. A COMWAKE signal from the host is not propagated to the devices. The Port Multiplier shall issue a COMWAKE signal to a device when a FIS needs to be delivered to that device.

The Port Multiplier may issue a PMREQ_P$_P$ or PMREQ_S$_P$ to the host if all device ports are in the Partial state, Slumber state, or are disabled.

16.3.3.7 Reducing Context Switching Complexity

It may complicate some host controller designs if traffic from another device is received in the middle of certain FIS sequences. For example, if a host receives a DMA Activate FIS from one device, it may be awkward for the host to receive a FIS from another device before it is able to issue the Data FIS to the first device.

The Port Multiplier shall provide the host with the opportunity to transmit before initiating any pending device transmissions. The Port Multiplier shall not assert X_RDY$_P$ to the host until the Port Multiplier has received at least two consecutive SYNC$_P$ primitives from the host (regardless of whether the host or Port Multiplier was the transmitter of the preceding FIS). In the case of a

collision, the Port Multiplier shall ignore this requirement and perform the actions detailed in section 16.3.3.2.

If the host wants to transmit prior to receiving another FIS, the host should issue one $SYNC_P$ between the end of the last FIS transmission and the start of the next FIS transmission. One possible implementation is to transition to the host Link layer state HL_SendChkRdy defined in section 9.6.3 immediately, regardless of whether the host is ready to proceed with the next transmission. The host would then only transition out of HL_SendChkRdy state when the host is ready to proceed with the transmission and R_RDY_P is received.

16.3.3.8 Error Handling and Recovery

The host is responsible for handling error conditions in the same way it handles errors when connected directly to a device. The host is responsible for detecting commands that do not finish and performing error recovery procedures as needed. The exact host software error recovery procedures are implementation specific.

The Port Multiplier is not responsible for performing any error recovery procedures. The Port Multiplier shall return R_ERR_P for certain error conditions as described by the Link layer state machine. The Port Multiplier shall update GSCR[32], defined in section 16.4.1.2, and PSCR[1] (SError) for the device port that experiences an error.

16.3.3.8.1 Command Timeout

If a command times out, the host may check PSCR[1] (SError), defined in section 16.4.2, to determine if there has been an interface error condition on the port that had the error. If there has been a device presence change on that port as indicated by the X bit in the DIAG field of PSCR[1] (SError) for the port, the host should re-enumerate the device on that port. The host software error recovery mechanism after a command timeout is implementation specific.

16.3.3.8.2 Disabled Device Port

If the host transmits a FIS to a device port that is disabled or that has FIS transfers disallowed due to the X bit being set in the DIAG field of PSCR[1] (SError) for that port, the Port Multiplier shall not perform an R_OK_P or R_ERR_P handshake at the end of FIS reception and shall instead terminate the FIS reception by issuing SYNC Escape to the host. The host is responsible for detecting that the command did not finish and performing error recovery procedures, including clearing the X bit in the DIAG field of the PSCR[1] register as necessary.

16.3.3.8.3 Invalid Device Port Address

An invalid device port address is a device port address that has a value greater than or equal to the number of device ports that the Port Multiplier supports. If the host specifies an invalid device port address as part of a FIS transmission, the Port Multiplier shall issue a SYNC Escape and terminate reception of the FIS. The host is responsible for detecting that the command did not finish and performing normal error recovery procedures.

16.3.3.8.4 Invalid CRC for Device Initiated Transfer

On a device initiated transfer, the Port Multiplier shall recalculate the CRC since it modifies the first Dword of the FIS. The Port Multiplier shall check the original CRC sent by the device. If the original CRC is invalid, the Port Multiplier shall invert the recalculated CRC to ensure that the CRC error is propagated to the host. The inversion may be done by XORing the recalculated CRC with FFFFFFFFh. The Port Multiplier shall also update the error information in PSCR[1] (SError) for the device port that experienced the error.

16.3.3.8.5 Data Corruption

If the Port Multiplier encounters an 8b/10b decoding error or any other error that could affect the integrity of the data passed between the transmitter and receiver of a FIS, the Port Multiplier shall ensure that the error is propagated to the receiver. The Port Multiplier may propagate the error by corrupting the CRC for the FIS. The Port Multiplier shall also update the error information in PSCR[1] (SError) for the device port that experienced the error.

16.3.3.8.6 Unsupported Command Received on Control Port

If an unsupported command is received on the control port, the Port Multiplier shall respond with a Register FIS that has the values shown in Table 89 for the Status and Error registers.

Table 89 – Register Values for an Unsupported Command

Register	7	6	5	4	3	2	1	0
Error	Reserved					ABRT	Reserved	
Status	BSY	DRDY	DF	na	DRQ	0	0	ERR

Register Values for an Unsupported Command:

ABRT = 1
BSY = 0
DRDY = 1
DF = 0
DRQ = 0
ERR = 1

16.3.3.9 BIST Support

A Port Multiplier may optionally support BIST. A Port Multiplier that supports BIST shall only support BIST in a point-to-point manner. A Port Multiplier that supports BIST shall not propagate a BIST Activate FIS received on one port over another port. The host determines that a Port Multiplier supports BIST by checking GCSR[64], defined in section 16.4.1.3.

To enter BIST mode over the host connection with a Port Multiplier, the host shall issue a BIST Activate FIS to the Port Multiplier control port. The host shall not issue a BIST Activate FIS to a device port.

To enter BIST mode over a device connection, the device shall issue a BIST Activate FIS to the Port Multiplier. The Port Multiplier shall intercept the BIST Activate FIS and enter BIST mode. Upon entering BIST mode with the device, the Port Multiplier shall update the PSCR[0] (SStatus) register for that port to reflect that the link has entered BIST mode, as specified in section 14.1.1.

Initiation of BIST by a Port Multiplier is vendor unique.

16.3.3.10 Asynchronous Notification

A Port Multiplier may optionally support asynchronous notification as defined in section 13.8. If asynchronous notification is enabled, a Port Multiplier shall send an asynchronous notification to the host when a bit transitions from zero to one in GSCR[32] of the Global Status and Control Registers. The Port Multiplier may send one notification for multiple zero to one bit transitions. The asynchronous notification shall only be sent when there is no command currently outstanding to the control port. If there is a command outstanding to the control port when an asynchronous notification needs to be sent, the Port Multiplier shall first complete the command and then send the asynchronous notification. The Port Multiplier shall set the PM Port field in the Set Device Bits FIS to the control port to indicate that the Port Multiplier itself needs attention.

Support for the asynchronous notification feature is indicated in GSCR[64] and it is enabled using GSCR[96].

16.3.3.10.1 Command-Based Switching (Informative)

Host designs should give careful consideration to support of asynchronous notification in command-based switching designs (outlined in section 16.7.1). If a command-based switching HBA has no explicit accommodation for asynchronous notification, then the host should not enable asynchronous notification on the control port or on any attached device.

16.3.3.11 Phy Event Counters

A Port Multiplier may optionally support the Phy event counters feature that is defined in section 13.9. If Phy event counters is enabled, a Port Multiplier shall store supported counter information for all of the ports that are enabled, including the host port. It is not required that the same list of Phy event counters be implemented on every port.

Support for the Phy event counters feature is indicated in GSCR[64] and it is enabled using bit 0 in GSCR[34]. The counter values shall not be retained across power cycles. The counter values shall be preserved across COMRESET and software resets that occur on any port.

16.3.3.11.1 Counter Identifiers

A Port Multiplier may support any of the counter identifiers described in Figure 223 in section 13.9.2. Support for some counters may not be logical for all ports due to the Port Multiplier architecture; examples of this would be the counters with identifiers 001h and 00Ah. For the Port Multiplier implementation, all counters other than 000h are optional. A counter identifier value of 000h shall indicate the end of the Phy event counter list implemented in the GSCR or PSCR registers. The counter with identifier 000h shall have no counter value.

All counter values consume a multiple of 16-bits, with a maximum of 64-bits. Each counter is allocated a single register location. The register location contains both the identifier for the counter implemented along with the value of the Phy event counter.

16.3.3.11.2 Reading Counter Values

Initially, the host may obtain the mapping of counters implemented along with their sizes by submitting reads starting at GSCR[256] (or PSCR[256]) to obtain the identifiers for the counters on a per port basis. Once the identifier of 000h is reached, this signifies the end of the list of counters implemented. The format of this read is a READ PORT MULTIPLIER command with the RS1 bit of the Device register cleared to zero. The value in the PortNum field determines which set of counters is to be accessed. Device port counter information for ports 0h-Eh is retrieved by using port numbers 0h-Eh respectively. Host port counter information is retrieved by using the control port number (Fh). RegNum[15:0] shall be set to the specific register location to be read. The output of the READ PORT MULTIPLIER command contains the identifier of the counter. When Phy event counters are enabled, a Port Multiplier shall return identifier 000h as the first counter for a port on which no Phy event counters are implemented.

To read the counter value itself, the READ PORT MULTIPLIER command is sent with the RS1 bit of the Device register set to one. The value in the PortNum field determines which set of counters are to be accessed. Device port counter information for ports 0h-Eh is retrieved by using port numbers 0h-Eh respectively. Host port counter information is retrieved by using the control port number (Fh). RegNum[15:0] shall be set to the specific register location to be read. The output of the READ PORT MULTIPLIER command contains the value of the counter, up to 64-bits in length. If the counter value read is less than 64-bits in length, the value returned by the Port Multiplier shall have the upper bits padded with zeroes.

All counters are one-extended up to a 16-bit multiple (i.e. 16, 32, 48, 64-bit), once the maximum counter value has been reached. For example, when returned as a 16-bit counter, if a counter is only physically implemented as 8-bits when it reaches the maximum value of FFh, it shall be one-extended to FFFFh. The counter shall stop (and not wrap to zero) after reaching its maximum value.

Upon any read to Phy event counter register space that is at or beyond the identifier 000h location, the Port Multiplier shall return error status with the ERR bit set to one, and the BSY and DRQ bits cleared to zero in the Status field of the FIS. The ABRT bit shall also be set to one in the Error field.

16.3.3.11.3 Counter Reset Mechanisms

There are three mechanisms the host may use to explicitly cause the Phy event counters to be reset. The first mechanism uses the WRITE PORT MULTIPLIER command to clear counters on an individual counter basis. The PortNum field determines which set of counters are to be accessed. Device port counter information for ports 0h-Eh is written by using port numbers 0h-Eh respectively. Host port counter information is written by using the control port number (Fh). The RegNum field shall be set to the register location (counter) which is to be written. The Value field within the command contains what shall be written into the register, in this case all zeroes.

The second mechanism allows for a counter to be reset following a read to that counter register. If the RS2 bit in the Device register is set to one for a read to any counter, the Port Multiplier shall return the current counter value for the command and then reset that value upon successful transmission of the Register FIS. If retries are required, upon unsuccessful transmission of the FIS it is possible that the counter value may be changed before the re-transmission of the counter value.

The third mechanism is a global reset function by writing appropriate bits in GSCR[34] to reset all Phy event counters for specific ports. A host may reset all Phy event counters by writing FFFFh to bits 31-16 of GSCR[34].

16.4 Port Multiplier Registers

The Port Multiplier registers are accessed using READ PORT MULTIPLIER and WRITE PORT MULTIPLIER commands issued to the control port.

16.4.1 General Status and Control Registers

The control port address is specified in the PortNum field of the READ PORT MULTIPLIER and WRITE PORT MULTIPLIER commands defined in section 16.5.1 and section 16.5.2 in order to read/write the General Status and Control registers.

16.4.1.1 Static Configuration Information

The Static Configuration Information section of the General Status and Control Registers contains registers that are static throughout the operation of the Port Multiplier. These registers are read-only and may not be modified by the host.

Table 90 – Static Information Registers

Register	O/M	F/V	Description
GSCR[0]	M		Product Identifier
		F	31-16 Device ID allocated by the vendor.
		F	15-0 Vendor ID allocated by the PCI-SIG.
GSCR[1]	M		Revision Information
		F	31-16 Reserved
		F	15-8 Revision level of the Port Multiplier.
		F	7-4 Reserved
		F	3 1 = Supports Port Multiplier specification 1.2.
		F	2 1 = Supports Port Multiplier specification 1.1.
		F	1 1 = Supports Port Multiplier specification 1.0.
		F	0 Reserved
GSCR[2]	M		Port Information

		F F	31-4 Reserved 3-0 Number of exposed device fan-out ports.
GSCR[3] – GSCR[31]	O	F	Reserved

Key:
O/M = Mandatory/optional requirement.
M = Support of the register is mandatory.
O = Support of the register is optional.
F/V = Fixed/variable content.
F = the content of the register is fixed and does not change.
V = the contents of the register is variable and may change.
X = the content of the register may be fixed or variable.

Register 0: Product Identifier

Support for this register is mandatory. The register identifies the vendor that produced the Port Multiplier and the specific device identifier.

Bit 15-0 shall be set to the vendor identifier allocated by the PCI-SIG of the vendor that produced the Port Multiplier.

Bit 31-16 shall be set to a device identifier allocated by the vendor.

Register 1: Revision Information

Support for this register is mandatory. The register specifies the specification revision that the Port Multiplier supports; the Port Multiplier may support multiple specification revisions. The register also specifies the revision level of the specific Port Multiplier product identified by Register 0.

Bit 0 is reserved and shall be cleared to zero.

Bit 1 when set to one indicates that the Port Multiplier supports Port Multiplier specification version 1.0.

Bit 2 when set to one indicates that the Port Multiplier supports Port Multiplier specification version 1.1.

Bit 3 when set to one indicates that the Port Multiplier supports Port Multiplier specification version 1.2.

Bit 7-4 are reserved and shall be cleared to zero.

Bit 15-8 identifies the revision level of the Port Multiplier product identified by Register 0. This identifier is allocated by the vendor.

Bit 31-16 are reserved and shall be cleared to zero.

Register 2: Port Information

Support for this register is mandatory. The register specifies information about the ports that the Port Multiplier contains, including the number of exposed device fan-out ports. The number of exposed device ports is the number of device ports that are physically connected and available for use on the product.

Bit 3-0 specifies the number of exposed device fan-out ports. A value of zero is invalid. The control port shall not be counted.

Bit 31-4 are reserved and shall be cleared to zero.

Register 3-31: Reserved

Registers 3-31 are reserved for future Port Multiplier definition and shall be cleared to zero.

16.4.1.2 Status Information and Control

The Status Information section of the General Status and Control Registers contains registers that convey status information and control operation of the Port Multiplier.

Table 91 – Status Information and Control Registers

Register	O/M	F/V	Description
GSCR[32]	M		Error Information
		F	31-15 Reserved
		V	14 OR of selectable bits in Port 14 PSCR[1] (SError)
		V	13 OR of selectable bits in Port 13 PSCR[1] (SError)
		V	12 OR of selectable bits in Port 12 PSCR[1] (SError)
		V	11 OR of selectable bits in Port 11 PSCR[1] (SError)
		V	10 OR of selectable bits in Port 10 PSCR[1] (SError)
		V	9 OR of selectable bits in Port 9 PSCR[1] (SError)
		V	8 OR of selectable bits in Port 8 PSCR[1] (SError)
		V	7 OR of selectable bits in Port 7 PSCR[1] (SError)
		V	6 OR of selectable bits in Port 6 PSCR[1] (SError)
		V	5 OR of selectable bits in Port 5 PSCR[1] (SError)
		V	4 OR of selectable bits in Port 4 PSCR[1] (SError)
		V	3 OR of selectable bits in Port 3 PSCR[1] (SError)
		V	2 OR of selectable bits in Port 2 PSCR[1] (SError)
		V	1 OR of selectable bits in Port 1 PSCR[1] (SError)
		V	0 OR of selectable bits in Port 0 PSCR[1] (SError)
GSCR[33]	M		Error Information Bit Enable
		V	31-0 If set, bit is enabled for use in GSCR[32]
GSCR[34]	O	V	Phy Event Counter Control 31 Host port global counter reset 30 Port 14 global counter reset 29 Port 13 global counter reset 28 Port 12 global counter reset 27 Port 11 global counter reset 26 Port 10 global counter reset 25 Port 9 global counter reset 24 Port 8 global counter reset 23 Port 7 global counter reset 22 Port 6 global counter reset 21 Port 5 global counter reset 20 Port 4 global counter reset 19 Port 3 global counter reset 18 Port 2 global counter reset 17 Port 1 global counter reset 16 Port 0 global counter reset 15-1 Reserved 0 Phy event counters enabled
GSCR[35] – GSCR[63]	O	F	Reserved
Key: O/M = Mandatory/optional requirement.			

Register	O/M	F/V	Description
M = Support of the register is mandatory. O = Support of the register is optional. F/V = Fixed/variable content. F = the content of the register is fixed and does not change. V = the contents of the register is variable and may change. X = the content of the register may be fixed or variable.			

Register 32: Error Information

Support for this register is mandatory. This register reflects whether specific bits have been set in any of the device port's PSCR[1] (SError) register. A set bit may reflect an error or that device presence has changed. The host selects the device port's PSCR[1] (SError) bits to reflect using GSCR[33]. The Port Multiplier algorithm for updating the register contents is:

```
for (n=0; n<NumPorts; n++)
{
      if (Port[n].PSCR[1] & GSCR[33]) == 0)
            GSCR[32].Bit[n]=0
      else
            GSCR[32].Bit[n]=1
}
```

This register is read-only. The specific device port's PSCR[1] (SError) register shall be written in order to clear values in the affected PSCR[1] (SError) register and by reflection in this register. Refer to section 14.1.2 for the definition of the SError register.

Bit 0 shall be set to the logically OR-ed value of selected bits in Port 0 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33].

Bit 1 shall be set to the logically OR-ed value of selected bits in Port 1 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 1 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 2 shall be set to the logically OR-ed value of selected bits in Port 2 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 2 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 3 shall be set to the logically OR-ed value of selected bits in Port 3 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 3 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 4 shall be set to the logically OR-ed value of selected bits in Port 4 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 4 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 5 shall be set to the logically OR-ed value of selected bits in Port 5 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 5 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 6 shall be set to the logically OR-ed value of selected bits in Port 6 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 6 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 7 shall be set to the logically OR-ed value of selected bits in Port 7 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 7 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 8 shall be set to the logically OR-ed value of selected bits in Port 8 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 8 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 9 shall be set to the logically OR-ed value of selected bits in Port 9 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 9 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 10 shall be set to the logically OR-ed value of selected bits in Port 10 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 10 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 11 shall be set to the logically OR-ed value of selected bits in Port 11 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 11 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 12 shall be set to the logically OR-ed value of selected bits in Port 12 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 12 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 13 shall be set to the logically OR-ed value of selected bits in Port 13 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 13 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 14 shall be set to the logically OR-ed value of selected bits in Port 14 PSCR[1] (SError). The bits to be OR-ed are selected using GSCR[33]. If Port 14 is not implemented by the Port Multiplier, this bit shall be cleared to zero.

Bit 31-15 are reserved and shall be cleared to zero.

Register 33: Error Information Bit Enable

Support for this register is mandatory. This register selects/enables bits to be used for the OR operation in GSCR[32]. If a bit is set to one, that bit shall be reflected for each device port in GSCR[32]. This is a global enable and is not device port specific. The default value of this register shall be 0400FFFFh; that corresponds to all bits in the ERR field and the DIAG field's X bit being OR-ed together for each device port in GSCR[32].

Register 34: Phy Event Counter Control

Support for this register is mandatory if Phy event counters is supported by the Port Multiplier as indicated by bit 4 in GSCR[64]. This register is used for Phy event counter control mechanisms.

Bit 0 if set to one indicates that the Port Multiplier supports Phy event counters and that the feature is currently enabled. If the Port Multiplier supports Phy event counters, it shall support counters for all ports implemented on the Port Multiplier, including the host port. It is not required for the same list of Phy event counters be implemented on every port. If this bit is cleared to zero, then the current values within the counters shall be retained and counting shall stop. This bit shall be cleared to zero on power-up. This bit shall not be affected by a COMRESET or software reset that has occurred on any port. The values within the Phy event counters are not affected by this bit.

Bits 1-15 are reserved and shall be cleared to zero.

Bit 16 if set to one shall result in an immediate reset of all Phy event counters associated with Port 0. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 17 if set to one shall result in an immediate reset of all Phy event counters associated with Port 1. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 18 if set to one shall result in an immediate reset of all Phy event counters associated with Port 2. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 19 if set to one shall result in an immediate reset of all Phy event counters associated with Port 3. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 20 if set to one shall result in an immediate reset of all Phy event counters associated with Port 4. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 21 if set to one shall result in an immediate reset of all Phy event counters associated with Port 5. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 22 if set to one shall result in an immediate reset of all Phy event counters associated with Port 6. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 23 if set to one shall result in an immediate reset of all Phy event counters associated with Port 7. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 24 if set to one shall result in an immediate reset of all Phy event counters associated with Port 8. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 25 if set to one shall result in an immediate reset of all Phy event counters associated with Port 9. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 26 if set to one shall result in an immediate reset of all Phy event counters associated with Port 10. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 27 if set to one shall result in an immediate reset of all Phy event counters associated with Port 11. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 28 if set to one shall result in an immediate reset of all Phy event counters associated with Port 12. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 29 if set to one shall result in an immediate reset of all Phy event counters associated with Port 13. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 30 if set to one shall result in an immediate reset of all Phy event counters associated with Port 14. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Bit 31 if set to one shall result in an immediate reset of all Phy event counters associated with the host port. Once the reset of all counters is complete, this bit shall be cleared to zero automatically by the Port Multiplier. If written to zero, no action is taken.

Register 35-63: Reserved

Registers 34-63 are reserved for future Port Multiplier definition and shall be cleared to zero.

16.4.1.3 Features Supported

The Features Supported section of the General Status and Control Registers contains registers that convey the optional features that are supported by the Port Multiplier. All Features Supported registers are read-only.

Table 92 – Features Supported Registers

Register	O/M	F/V	Description
GSCR[64]	M		Port Multiplier Revision 1.X Features Support
		F	31-5 Reserved
		F	4 1 = Supports Phy event counters
		F	3 1 = Supports asynchronous notification
		F	2 1 = Supports dynamic SSC transmit enable
		F	1 1 = Supports issuing $PMREQ_P$ to host
		F	0 1 = Supports BIST
GSCR[65] – GSCR[95]	O	F	Reserved

Key:
O/M = Mandatory/optional requirement.
M = Support of the register is mandatory.
O = Support of the register is optional.
F/V = Fixed/variable content.
F = the content of the register is fixed and does not change.
V = the contents of the register is variable and may change.
X = the content of the register may be fixed or variable.

Register 64: Port Multiplier Revision 1.X Features Support

Support for this register is mandatory. This register specifies the optional Port Multiplier specification revision 1.X features that the Port Multiplier supports.

Bit 0 if set to one indicates that the Port Multiplier supports BIST as outlined in section 16.3.3.9. If this bit is cleared to zero the Port Multiplier does not support reception of the BIST Activate FIS.

Bit 1 if set to one indicates that the Port Multiplier supports issuing $PMREQ\_P_P$ and $PMREQ\_S_P$ requests to the host when all device ports are disabled or in a Partial/Slumber state. If this bit is cleared to zero, the Port Multiplier does not support issuing $PMREQ\_P_P$ or $PMREQ\_S_P$ requests to the host.

Bit 2 if set to one indicates that the Port Multiplier supports dynamically enabling and disabling spread-spectrum clocking transmit. If this bit is cleared to zero, the Port Multiplier does not support dynamically enabling and disabling spread-spectrum clocking transmit.

Bit 3 if set to one indicates that the Port Multiplier supports asynchronous notification. If the Port Multiplier supports asynchronous notification, it shall be capable of sending a Set Device Bits FIS to the host with the Interrupt bit set to one and the Notification bit set to one when a bit in GSCR[32] transitions from zero to one. If this bit is cleared to zero then the Port Multiplier does not support asynchronous notification.

Bit 4 if set to one indicates that the Port Multiplier supports Phy event counters, and GSCR[34] shall be implemented. If this bit is cleared to zero, the Port Multiplier does not support Phy event counters.

Bit 31-5 are reserved and shall be cleared to zero.

Register 65-95: Reserved

Registers 65-95 are reserved for future Port Multiplier definition and shall be cleared to zero.

16.4.1.4 Features Enabled

The Features Enabled section of the General Status and Control Registers contains registers that allow optional features to be enabled. The Features Enabled registers have a one-to-one correspondence with the Features Supported registers defined in section 16.4.1.3. All Features Enabled registers are read/write.

Table 93 – Features Enabled Registers

Register	O/M	F/V	Description
GSCR[96]	M		Port Multiplier Revision 1.X Features Enable
		F	31-4 Reserved
		V	3 1 = Asynchronous notification enabled
		V	2 1 = Dynamic SSC transmit is enabled
		V	1 1 = Issuing $PMREQ_P$ to host is enabled
		V	0 1 = BIST support is enabled
GSCR[97] – GSCR[127]	O	F	Reserved

Key:
O/M = Mandatory/optional requirement.
M = Support of the register is mandatory.
O = Support of the register is optional.
F/V = Fixed/variable content.
F = the content of the register is fixed and does not change.
V = the contents of the register is variable and may change.
X = the content of the register may be fixed or variable.

Register 96: Port Multiplier Revision 1.X Features Enable

Support for this register is mandatory. This register controls whether the optional Port Multiplier specification revision 1.X features that the Port Multiplier supports are enabled.

Bit 0 if set to one indicates that the Port Multiplier supports BIST and that BIST support is enabled. If this bit is cleared to zero reception of the BIST Activate FIS is not enabled. If the Port Multiplier supports BIST, this bit shall be set to one on power-up or after a reset, otherwise it shall be cleared to zero on power-up and reset.

Bit 1 if set to one indicates that the Port Multiplier supports issuing $PMREQ\_P_P$ and $PMREQ\_S_P$ requests to the host when all device ports are disabled or in a Partial/Slumber

state and the feature is enabled. If this bit is cleared to zero, the Port Multiplier shall not issue PMREQ_P_P or PMREQ_S_P requests to the host. This bit shall be cleared to zero on power-up and reset.

Bit 2 if set to one indicates that the Port Multiplier supports dynamically enabling and disabling spread-spectrum clocking transmit and that the feature is currently enabled. If this bit is cleared to zero and the feature is supported as specified in GSCR[64], the Port Multiplier shall not use spread-spectrum clocking transmit. If this feature is supported as specified in GSCR[64], the bit shall be set to one on power-up and reset. If this feature is not supported as specified in GSCR[64], then the bit shall be cleared to zero on power-up and reset.

Bit 3 if set to one indicates that the Port Multiplier supports asynchronous notification and that the feature is currently enabled. If the Port Multiplier supports asynchronous notification, it shall be capable of sending a Set Device Bits FIS to the host with the Interrupt bit set to one and the Notification bit set to one when a bit in GSCR[32] transitions from zero to one. If this bit is cleared to zero and the corresponding bit in GSCR[64] is set to one, then asynchronous notification is not enabled. This bit shall be cleared to zero on power-up and reset.

Bit 31-4 are reserved and shall be cleared to zero.

Register 97-127: Reserved

Registers 97-127 are reserved for future Port Multiplier definition and shall be cleared to zero.

16.4.1.5 Vendor Unique

The Vendor Unique section of the General Status and Control Registers contains registers that are vendor unique.

Table 94 – Vendor Unique Registers

Register	O/M	F/V	Description
GSCR[128] – GSCR[255]	O	X	Vendor Unique
Key: O/M = Mandatory/optional requirement. M = Support of the register is mandatory. O = Support of the register is optional. F/V = Fixed/variable content. F = the content of the register is fixed and does not change. V = the contents of the register is variable and may change. X = the content of the register may be fixed or variable.			

Register 128-255: Vendor unique

16.4.1.6 Phy Event Counters

The Phy event counters section of the General Status and Control Registers contains registers that store the data for each of the Phy event counters supported by the Port Multiplier for the host port. The Phy event counters registers contain both the identifier and counter values. All Phy event counter registers are read/write.

A value of zero returned for a counter means that there have been no instances of that particular event.

Table 95 – Phy Event Counter Registers

Register	O/M	F/V	Description
GSCR[256] – GSCR[2303]	O	V	Phy event counter registers.

Key:
O/M = Mandatory/optional requirement.
M = Support of the register is mandatory.
O = Support of the register is optional.
F/V = Fixed/variable content
F = the content of the register is fixed and does not change.
V = the contents of the register is variable and may change.
X = the content of the register may be fixed or variable.

Register 256-2303: Used for Phy event counters. Phy event counters are not allocated specific registers. Each register contains both the identifier and value for the counter implemented.

16.4.1.7 Reserved

The section of the General Status and Control Registers starting at address 2304 are reserved.

Table 96 – Reserved Registers

Register	O/M	F/V	Description
GSCR[2304] – GSCR[65535]	O	F	Reserved

Key:
O/M = Mandatory/optional requirement.
M = Support of the register is mandatory.
O = Support of the register is optional.
F/V = Fixed/variable content.
F = the content of the register is fixed and does not change.
V = the contents of the register is variable and may change.
X = the content of the register may be fixed or variable.

Register 2304-65535: Reserved for future Port Multiplier definition and shall be cleared to zero.

16.4.2 Port Status and Control Registers

The Port Multiplier shall maintain a set of Port Status and Control registers (PSCRs) for each device port that it supports. These registers are port specific and contain the Serial ATA superset Status and Control registers, along with the Phy event counters information for each port. The host specifies the device port to read or write registers for in the PortNum field of the READ PORT MULTIPLIER command or WRITE PORT MULTIPLIER command defined in section 16.5.1 and section 16.5.2. The registers are defined in the following table.

Table 97 – PSCR Definition

Register	Definition
PSCR[0]	SStatus register (see section 14.1.1)
PSCR[1]	SError register (see section 14.1.2)
PSCR[2]	SControl register (see section 14.1.3)
PSCR[3]	SActive register (not implemented)
PSCR[4]	SNotification register (not implemented)
PSCR[5] - PSCR[255]	Reserved
PSCR[256] – PSCR[2303]	Phy event counter registers
PSCR[2304] – PSCR[65535]	Reserved

The reset value for the DET field of PSCR[2] (SControl) register shall be 4h (Phy disabled).

16.4.2.1 Phy Event Counters

The Phy event counter information for each of the device ports within a Port Multiplier is contained in the Port Status and Control registers starting at PSCR[256]. The Phy event counters registers contain both the identifier and counter values. All Phy event counter registers are read/write.

A value of zero returned for a counter means that there have been no instances of that particular event.

16.5 Port Multiplier Command Definitions

16.5.1 READ PORT MULTIPLIER

The READ PORT MULTIPLIER command is used to read a register on a Port Multiplier. The READ PORT MULTIPLIER command shall be issued to the control port.

16.5.1.1 Inputs

Table 98 – READ PORT MULTIPLIER Command Definition

Register	7	6	5	4	3	2	1	0
Features(7:0)	RegNum[7:0]							
Features(15:8)	RegNum[15:8]							
Count(7:0)	Reserved							
Count(15:8)	Reserved							
LBA(7:0)	Reserved							
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	na	RS1	RS2	na	PortNum			
Command	E4h							

PortNum Set to the port address that has register to be read.

RegNum Set to number of register to read.

RS1 Register Specific 1: This bit is register specific.
Phy event counter usage: Used to determine access to Phy event counter identifier or value. If cleared to zero upon a read to a Phy event counter register, the Port Multiplier shall return the Phy event counter identifier for that register. If set to one, the Port Multiplier shall return the Phy event counter value for that register, up to 64-bits in length.
All other usage: This bit shall be treated as 'na' on all READ PORT MULTIPLIER commands to registers other than Phy event counter registers.

RS2 Register Specific 2: This bit is register specific.
Phy event counter usage: Used in reads to Phy event counter values. If set to one upon a read to a Phy event counter register, the Port Multiplier shall return the value of the counter, followed by a reset of the counter value for the register supplied in RegNum.
All other usage: This bit shall be treated as 'na' on all READ PORT MULTIPLIER commands to registers other than Phy event counter registers.

16.5.1.2 Success Outputs

Upon successful completion, the ERR bit in the Status register shall be cleared to zero and the value in the Error register shall be zero, and the value of the specified register shall be returned.

Table 99 – READ PORT MULTIPLIER Success Status Result Values

Register	7	6	5	4	3	2	1	0
Error	0							
Count(7:0)	Value [7:0]							
Count(15:8)	Value [39:32]							
LBA(7:0)	Value [15:8]							
LBA(31:24)	Value [47:40]							
LBA(15:8)	Value [23:16]							
LBA(39:32)	Value [55:48]							
LBA(23:16	Value [31:24]							
LBA(47:40)	Value [63:56]							
Device	Reserved							
Status	BSY	DRDY	DF	na	DRQ	0	0	ERR

Key:
Value Set to the 64-bit value read from the register.

BSY = 0
DRDY = 1
DF = 0
DRQ = 0
ERR = 0

16.5.1.3 Error Outputs

Upon encountering an error, the Port Multiplier shall set the ERR bit to one in the Status register and identify the error code in the Error register.

Table 100 – READ PORT MULTIPLIER Error Status Result Values

Register	7	6	5	4	3	2	1	0
Error	Reserved					ABRT	REG	PORT
ount(7:0)	Reserved							
Count(15:8)	Reserved							
LBA(7:0)	Reserved							
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	Reserved							
Status	BSY	DRDY	DF	na	DRQ	0	0	ERR

Key:
ABRT = 0
REG Set to one if the register specified is invalid.
PORT Set to one if the port specified is invalid.

BSY = 0
DRDY = 1
DF = 0
DRQ = 0
ERR = 1

16.5.2 WRITE PORT MULTIPLIER

The WRITE PORT MULTIPLIER command is used to write a register on a Port Multiplier. The WRITE PORT MULTIPLIER command shall be issued to the control port.

16.5.2.1 Inputs

Table 101 – WRITE PORT MULTIPLIER Command Definition

Register	7	6	5	4	3	2	1	0
Features(7:0)	RegNum[7:0]							
Features(15:8)	RegNum[15:8]							
Count(7:0)	Value [7:0]							
Count(15:8)	Value [39:32]							
LBA(7:0)	Value [15:8]							
LBA(31:24)	Value [47:40]							
LBA(15:8)	Value [23:16]							
LBA(39:32)	Value [55:48]							
LBA(23:16	Value [31:24]							
LBA(47:40)	Value [63:56]							

Register	7	6	5	4	3	2	1	0
Device	na	RS1	na	na	PortNum			
Command	E8h							

Key:
Value — Set to the 64-bit value to write to the register.
RegNum — Set to number of register to write.
PortNum — Set to the port address that has register to be written.
RS1 — Register Specific 1: This bit is register specific.
Phy event counter usage: The WRITE PORT MULTIPLIER command may not be used to write to a Phy event counters' identifier value. If this bit is set to one during a write to a Phy event counter register, the counter addressed by RegNum shall be written with Value. If cleared to zero during a write to a Phy event counter register, the Port Multiplier shall return error status with the ERR bit set to one, and the BSY and DRQ bits cleared to zero in the Status field of the FIS. The ABRT bit shall also be set to one in the Error field.
All other usage: This bit shall be treated as 'na' on all READ PORT MULTIPLIER commands to registers other than Phy event counter registers.

16.5.2.2 Success Outputs

Upon successful completion, the Port Multiplier shall write the specified value to the specified register. The ERR bit in the Status register shall be cleared and the value in the Error register shall be zero.

Table 102 – WRITE PORT MULTIPLIER Success Status Result Values

Register	7	6	5	4	3	2	1	0
Error	0							
Count(7:0)	Reserved							
Count(15:8)	Reserved							
LBA(7:0)	Reserved							
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	Reserved							
Status	BSY	DRDY	DF	na	DRQ	0	0	ERR

Key:

BSY = 0
DRDY = 1
DF = 0
DRQ = 0
ERR = 0

16.5.2.3 Error Outputs

Upon encountering an error, the Port Multiplier shall set the ERR bit to one in the Status register and identify the error code in the Error register.

Table 103 – WRITE PORT MULTIPLIER Error Status Result Values

Register	7	6	5	4	3	2	1	0
Error	Reserved					ABRT	REG	PORT
Count(7:0)	Reserved							
Count(15:8)	Reserved							
LBA(7:0)	Reserved							
LBA(31:24)	Reserved							
LBA(15:8)	Reserved							
LBA(39:32)	Reserved							
LBA(23:16	Reserved							
LBA(47:40)	Reserved							
Device	Reserved							
Status	BSY	DRDY	DF	na	DRQ	0	0	ERR

Key:
ABRT = 0
REG Set to one if the register specified is invalid.
PORT Set to one if the port specified is invalid.

BSY = 0
DRDY = 1
DF = 0
DRQ = 0
ERR = 1

16.5.3 Interrupts

The Port Multiplier shall generate an interrupt in the status response to a command for the control port by setting the Interrupt bit to one in the Device-to-Host Register FIS. The Port Multiplier shall not generate an interrupt in response to a software reset for the control port.

16.6 Controlling PM Port Value and Interface Power Management

The host controller needs to provide a means by which software may set the PM Port field in all transmitted FISes. This capability may be exposed to software by supporting the PMP field in the SControl register defined in section 14.1.3. In addition a means by which software may cause a specific device port to transition to a low power management state needs to be provided. This capability may be exposed to software by supporting the SPM field in the SControl register defined in section 14.1.3.

16.7 Switching Types (Informative)

The host may use two different switching types depending on the capabilities of the host controller. If the host controller supports hardware context switching based on the value of the PM Port field in a received FIS, then the host may have commands outstanding to multiple devices at the same time. This switching type is called FIS-based switching and results in FISes being delivered to the host from any of the devices that have commands outstanding to them. If the host controller does not support hardware context switching based on the value of the PM Port field in a received FIS, then the host may only have commands outstanding to one device at any point in time. This switching type is called command-based switching and results in FISes being delivered to the host from only the one device that has commands outstanding to it. The Port Multiplier's operation is the same regardless of the switching type used by the host.

16.7.1 Command-Based Switching

Host controllers that do not support hardware context switching utilize a switching type called command-based switching. To use command-based switching, the host controller has commands outstanding to only one device at any point in time. By only issuing commands to one device at a time, the result is that the Port Multiplier only delivers FISes from that device.

A host controller may support command-based switching by implementing the Port Multiplier Port (PMP) field in the SControl register as detailed in section 14.1.3. In order to use this mechanism, host software would set the PMP field appropriately before issuing a command to a device connected to the Port Multiplier. When host software had completed the commands with a particular device port, it would modify the PMP field before issuing commands to any other device port. Note that the PMP field shall be set to the control port when host software issues commands to the Port Multiplier itself (such as READ PORT MULTIPLIER or WRITE PORT MULTIPLIER).

16.7.2 FIS-Based Switching

Host controllers that support hardware context switching may utilize a switching type called FIS-based switching. FIS-based switching allows the host controller to have commands outstanding to multiple devices at any point in time. When commands are outstanding to multiple devices, the Port Multiplier may deliver FISes from any device with commands outstanding.

16.7.2.1 Host Controller Requirements

To support FIS-based switching, the host controller shall have context switching support. The host controller needs to provide a means for exposing a programming interface for up to 16 devices on a single port. The context switching support needs to comprehend not only Control Block register context, but also DMA engine context and the SActive register. The host controller needs to be able to update context for a particular device even if the programming interface for that device is not currently selected by host software.

The host controller needs to fill in the PM Port field in hardware assisted FIS transmissions. For example, if the host controller receives a DMA Activate from a device, it needs to construct a Data FIS with the PM Port field set to the value in the received DMA Activate FIS.

Refer to section 16.3.3.7 for a mechanism to reduce the complexity of host context switching.

17 Port Selector

A Port Selector allows two different host ports to connect to the same device in order to create a redundant path to that device. In combination with RAID, the Port Selector allows system providers to build fully redundant solutions. The upstream ports of a Port Selector may also be attached to a Port Multiplier to provide redundancy in a more complex topology. A Port Selector may be thought of as a simple multiplexer as shown in Figure 252.

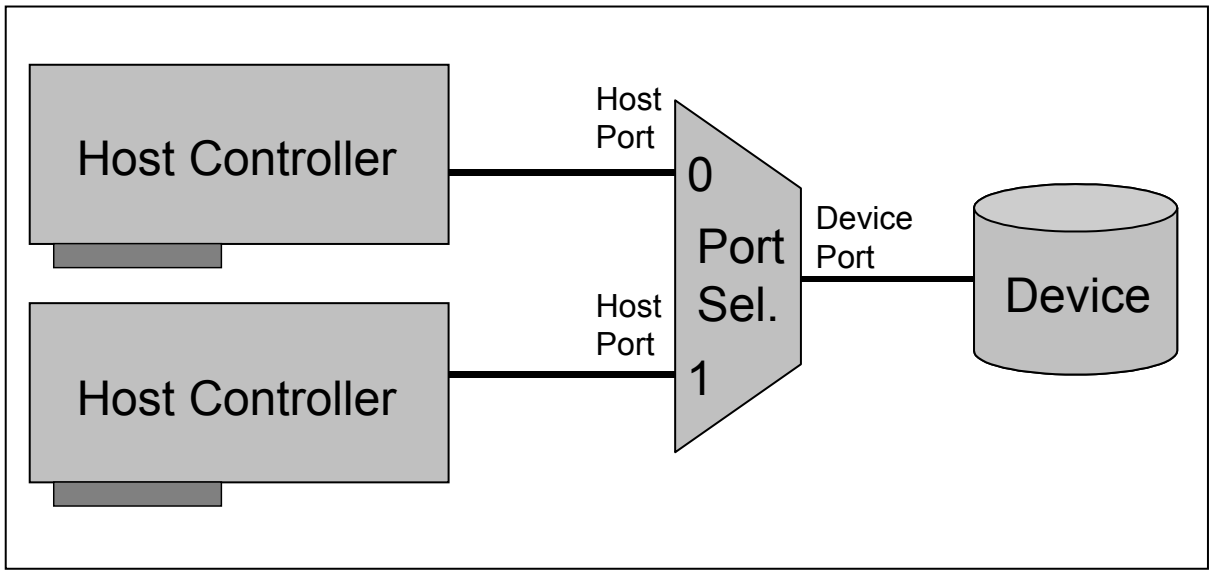


Figure 252 – Port Selector Overview

Exactly two host connections are provided by a Port Selector. Only one of the two host ports is active at a time – the specification does not define mechanisms for active/active solutions. Cascading Port Selectors to eachother is not supported.

17.1 Example Applications

One example application of a Port Selector, as shown in Figure 253, is to provide a means for redundant access to a device. This ingredient, along with RAID, allows a system with no single point of failure to be built. Typically the Port Selector would be packaged in the hard drive carrier to create a single serviceable unit in case the hard drive failed. The total system would consist of two hosts each connected to a RAID array where each device in the system had a Port Selector attached that was connected to each host. One host could be considered the live host and the other host may be the spare. In this configuration, the live host would maintain access to all of the devices and the spare host would only take over access to the devices if the live host had a failure.

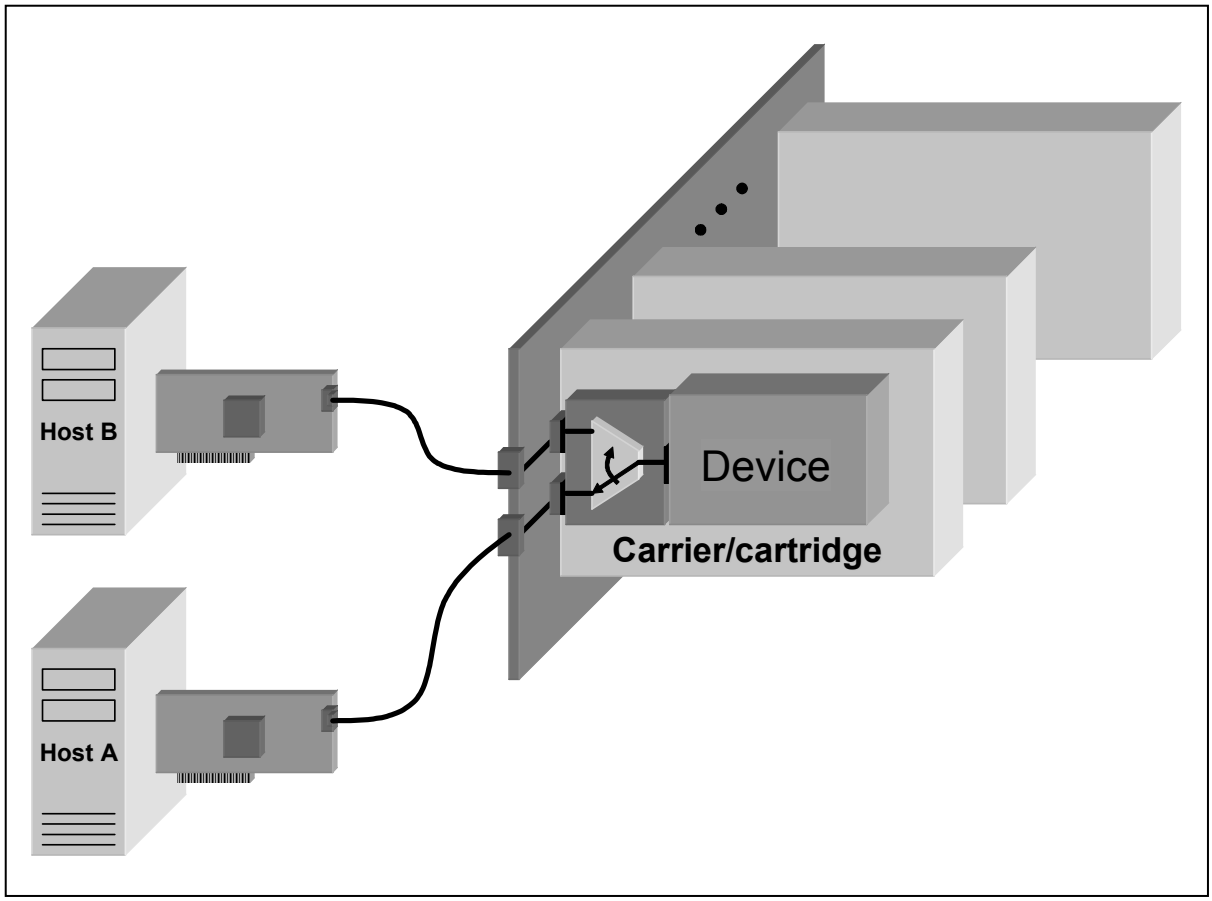


Figure 253 – Example Failover Application with Two Hosts

17.2 Overview

Port Selector is a mechanism that allows two different host ports to connect to the same device in order to create a redundant path to that device. Only one host connection to the device is active at a time. Effective use of a Port Selector requires coordinated access to the device between the two host ports. The host(s) shall coordinate to determine which host port should be in control of the device at any given point in time. Definition of the coordination mechanism or protocol is beyond the scope of this specification.

Once the host(s) determines the host port that should be in control of the device, the host that contains the host port to be made active takes control of the device by selecting that host port to be active. The active host selects a port to be active by using either a protocol-based or side-band port selection mechanism. A side-band port selection mechanism may be as simple as a hardware select line that is pulled high to activate one host port and low to activate the other. The side-band port selection mechanism is outside the scope of this specification. A protocol-based port selection mechanism uses the Serial ATA protocol to cause a switch of active port. This specification defines a protocol-based port selection mechanism that uses a particular Morse coding of COMRESET signals to cause a switch of active host port. A Port Selector shall only support one selection mechanism at any point in time. The externally visible behavior of a Port Selector is the same regardless of whether a protocol-based or side-band port selection mechanism is used.

A Port Selector that supports protocol-based port selection is detected in the signal path if the optional presence detection feature is supported by the Port Selector and the host has an enhanced SError register that latches this event. The detection mechanism for a Port Selector that supports side-band port selection is outside the scope of this specification.

17.3 Active Port Selection

The Port Selector has a single active host port at a time. The Port Selector shall support one of two mechanisms for determining which of the two host ports is active. The first mechanism is called side-band port selection. Side-band port selection uses a mechanism outside of the Serial ATA protocol for determining which host port is active. The second mechanism is called protocol-based port selection. Protocol-based port selection uses a sequence of Serial ATA OOB Phy signals to select the active host port.

Whether a protocol-based or side-band port selection mechanism is used, the Port Selector shall exhibit the behavior defined within this specification.

After selection of a new active port, the device and is in an unknown state. The device may have active commands outstanding from the previous active host that need to be flushed. After an active port switch has been performed it is strongly suggested that the active host issue a COMRESET to the device to ensure that the device is in a known state.

17.3.1 Protocol-based Port Selection

Protocol-based port selection is an active port selection mechanism that uses a sequence of Serial ATA OOB Phy signals to select the active host port. A Port Selector that supports protocol-based port selection shall have no active host port selected upon power-up. The first COMRESET or COMWAKE received over a host port shall select that host port as active. The host may then issue explicit switch signals to change the active host port.

Reception of the protocol-based port selection signal on the inactive host port causes the Port Selector to deselect the currently active host port and select the host port over which the selection signal is received. The protocol-based port selection signal is defined such that it is generated using the Status and Control registers and such that it is received and decoded without the need for the Port Selector to include a full Link or Transport layer (i.e. direct Phy detection of the signal).

17.3.1.1 Port Selection Signal Definition

The port selection signal is based on a pattern of COMRESET OOB signals transmitted from the host to the Port Selector. As illustrated in Figure 254, the Port Selector shall qualify only the timing from the assertion of a COMRESET signal to the assertion of the next COMRESET signal in detecting the port selection signal.

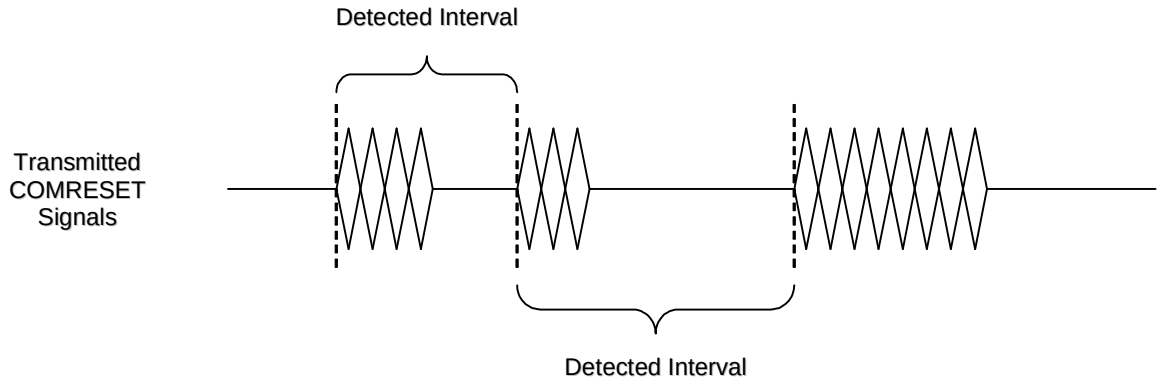


Figure 254 – Port selection signal based on assertion of COMRESET to assertion of following COMRESET

The port selection signal is defined as a series of COMRESET signals with the timing from the assertion of one COMRESET signal to the assertion of the next as defined in Table 104 and illustrated in Figure 255. The Port Selector shall select the port, if inactive, on the negation of COMRESET after receiving two complete back-to-back sequences with specified inter-burst spacing over that port (i.e. two sequences of two COMRESET intervals comprising a total of five COMRESET bursts with four inter-burst delays). Specifically, after receiving a valid port selection signal, the Port Selector shall not select that port to be active until the entire fifth COMRESET burst is no longer detected. The Port Selector is only required to recognize the port selection signal over an inactive port. Reception of COMRESET signals over an active port is propagated to the device without any action taken by the Port Selector, even if the COMRESET signals constitute a port selection signal. This may result in multiple device resets.

The timings detailed in Table 104 shall be independent of the signaling speed used on the link. For example, the inter-reset timings are the same for links using Gen1 or Gen2 speeds.

	Nom	Min	Max	Units	Comments
T1	2.0	1.6	2.4	ms	Inter-reset assertion delay for first event of the selection sequence
T2	8.0	7.6	8.4	ms	Inter-reset assertion delay for the second event of the selection sequence

Table 104 – Port selection signal inter-reset timing requirements

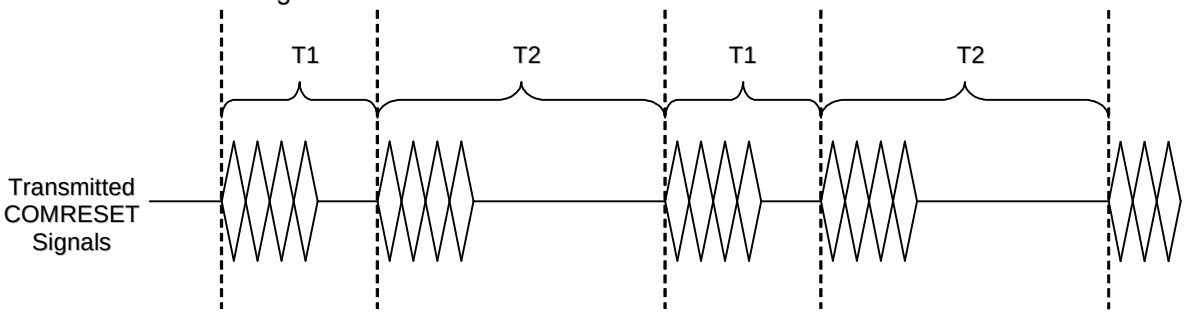


Figure 255 – Complete port selection signal consisting of two sequences with requisite inter-reset spacings

The interpretation and detection of the COMRESET signal by the Port Selector is in accordance with the Phy layer electrical specifications, i.e. the COMRESET signal is detected upon receipt of the fourth burst that complies with the COMRESET signal timing definition. The inter-reset timings referred to here for the port selection signal are from the detection of a valid COMRESET signal to the next detection of such a signal, and are not related to the bursts that comprise the COMRESET signal itself.

17.3.1.2 Presence Detection

Presence detection is the ability for a host to detect that a Port Selector is present on a port. If a Port Selector supports presence detection capabilities, a host will be able to determine whether a Port Selector is connected to a host port. The host may determine this regardless of whether the Port Selector port to which it is connected is the active or inactive link.

Presence detection capabilities are defined for Port Selectors utilizing protocol-based port selection only. Systems utilizing side-band port selection shall be preconfigured for side-band port selection and therefore those systems may also be preconfigured to support presence detection. Presence detection is an optional feature of protocol-based Port Selectors.

17.3.1.2.1 Host port Phy state machine enhancements

If presence detection is supported, the Port Selector host port Phy state machine as described in the section 8.4.2 shall be modified as shown. A Port Selector shall remain in the DR_PS_Wait state when the Phy is offline and presence detection is enabled.

DP1: DR_Reset[1]	Interface quiescent	
1. COMRESET not detected and power-on reset negated ***and presence detection not enabled and Phy not offline***	→	DR_COMINIT
2. COMRESET not detected and power-on reset negated and presence detection enabled	→	***DR_PS_Presence***
3. COMRESET detected or power-on reset asserted	→	DR_Reset
NOTE : 1. This state is entered asynchronously any time in response to power-on reset or receipt of a COMRESET signal from the host.		

DP12: DR_PS_Presence	***Transmit COMWAKE***	
1. Phy online	→	***DR_COMINIT***
2. Phy offline	→	***DR_PS_Wait***

DP13: DR_PS_Wait	*Interface quiescent*		
	1. Phy online	→	***DR_COMINIT***
	2. Phy offline	→	***DR_PS_Wait***

17.3.1.2.2 Host Phy Initialization state machine impact (Informative)

A host connected to a Port Selector performing presence detection receives a COMWAKE signal while in the HP2: HR_AwaitCOMINIT state of the Phy state machine defined in section 8.4.1. This state is insensitive to receiving a COMWAKE and only performs a transition to a new state when COMINIT is received. Host designers should ensure that their implementations are insensitive to receiving a COMWAKE in this state. If the host design is capable of latching the event of receiving a COMWAKE while in this state, the host should expose that event using the SError register enhancement detailed in section 17.3.1.2.3.

17.3.1.2.3 SError Register Enhancement for Presence Detection

The Serial ATA interface error register (SError) as defined in section 14.1.2 includes indications for various events that may have occurred on the interface such as a change in the PHYRDY state, detection of disparity, errors, and so on. In order to facilitate a means for notifying host software that a Port Selector presence detection signal was received, the A bit in the DIAG field of the SError register is set to one when COMWAKE is received while the host is in state HP2: HR_AwaitCOMINIT.

17.3.1.3 Host Transmission Considerations (Informative)

In order to ensure the port selection signal is reliably conveyed to the Port Selector, the host should account for any other interface activity that may interfere with the transmitted COMRESET port selection sequence. For example, if the host periodically issues a COMRESET signal as part of a hardware-polled device presence detection mechanism, a periodic COMRESET signal could occur during the port selection signaling sequence, thereby corrupting the port selection sequence. In order to avoid such interactions, the host may elect to continually transmit the port selection sequence while monitoring the Phy status in the associated superset Status and Control register. When the port selection signal is recognized by the Port Selector and has taken effect, the host detects a change in the PHYRDY status since the associated port is activated and communications with it are established. It is recommended that the host check the PHYRDY signal immediately before issuing each COMRESET burst in the protocol-based selection signal and only issue the next COMRESET burst if PHYRDY is not asserted.

17.3.2 Side-band Port Selection

The active host port may be selected by a side-band mechanism. Side-band port selection uses a mechanism outside of the Serial ATA protocol for selecting which host port is active. One example of a side-band port selection mechanism is a hardware select line. The side-band selection mechanism used is outside the scope of this specification. A Port Selector that supports side-band port selection shall exhibit the behavior defined within this specification.

17.3.3 Behavior during a change of active port

During a change of active port, the previous host connection is broken and all internal state other than the active host port is initialized before the connection with the new active host is made. When a new active host port is selected, the Port Selector shall perform the following procedure:

1. The Port Selector shall stop transmitting and enter the quiescent power condition on the previously active host port Phy (now the inactive host port).
2. The Port Selector shall initialize all internal state other than the state of the selection bit for the active host port.
3. The Port Selector shall enter the active power condition on the new active host port.
4. The Port Selector shall allow OOB and normal traffic to proceed between the new active host port and the device.

17.3.3.1 Device State after a change of active port (Informative)

A Port Selector may support an orderly switch to a new active host port. A Port Selector that supports an orderly switch ensures that primitive alignment with the device Phy is maintained during the switch to the new active host port. Maintaining primitive alignment ensures that PHYRDY remains present between the Port Selector and the device throughout the switch to the new active host port.

After selection of a new active port, the device may be in an unknown state. The device may have active commands outstanding from the previously active host port that need to be flushed. The new active host should issue a COMRESET to the device in order to return the device to a known state.

17.4 Behavior and Policies

17.4.1 Control State Machine

The Port Selector Control state machine is based on a model of a Port Selector consisting of three Serial ATA Phys interconnected and controlled by an overall control logic block as depicted in Figure 256. For convenience, the three ports of a Port Selector are abbreviated “A,” “B,” and “D” corresponding to Host Port A, Host Port B, and Device Port respectively.

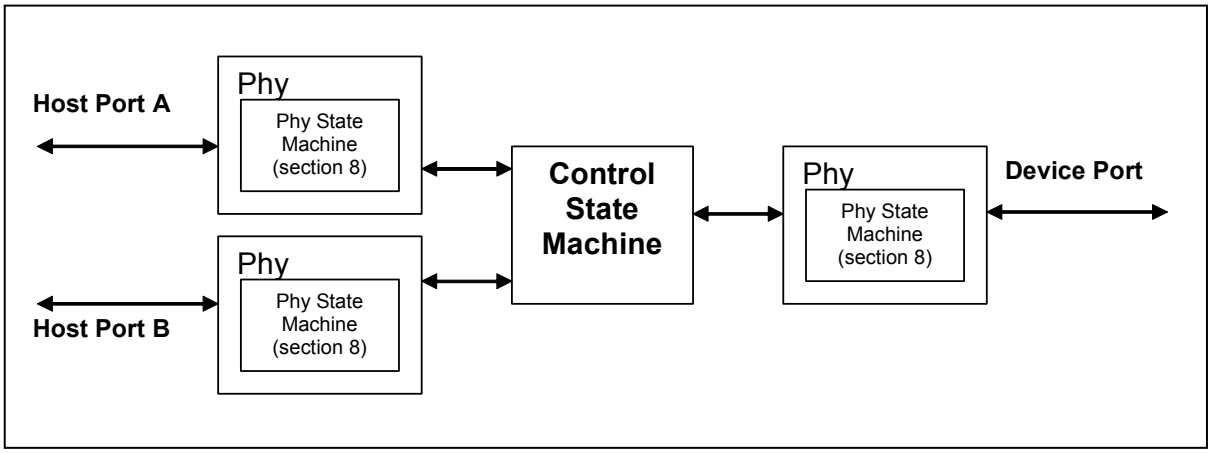


Figure 256 – Control State Machine

The Phys depicted in Figure 256 are presumed to have the basic capabilities and controls indicated in Figure 257. Figure 257 is a variant of Figure 111. The differences are:

- Loopback controls were removed since those controls were not relevant in this state machine.
- Explicit ONLINE and OFFLINE signals were added, including a register latch for these signals so that the signals do not need to be asserted in every state. The SControl register specifies a mechanism for the host to put the Phy in offline mode. Therefore, it is reasonable to expect that the Phy has signals ONLINE and OFFLINE that may be utilized.
- A PORTSELECT signal was added. A Port Selector using protocol-based port selection shall set the PORTSELECT signal to the output of the Sequence Detect block. The Sequence Detect block is asserted when the protocol-based selection signal is received, otherwise the signal is negated. A Port Selector using side-band port selection shall set the PORTSELECT signal when a change in active port is requested.

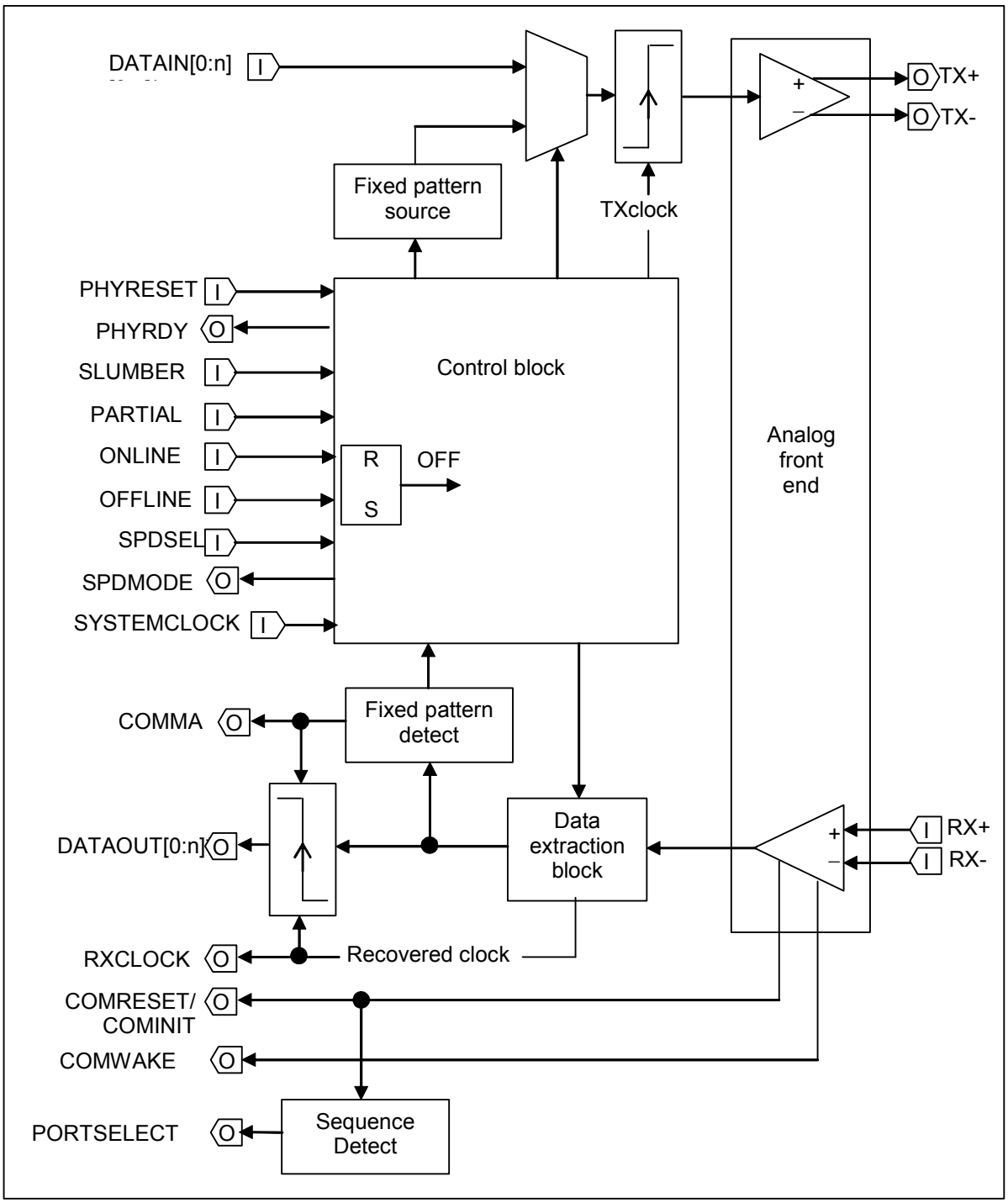


Figure 257 – Phy Block Diagram

The following state diagram specifies the behavior of the Port Selector control logic block. The Port Selector shall have the externally visible behavior described by this state machine.

Reception of a COMRESET signal from the selected host port shall unconditionally force the control state machine to transition to state PS15: ResetDevice. Reception of a COMINIT signal

from the device shall unconditionally force the control state machine to transition to state PS14: ResetHost if not in the ResetDevice state. For the sake of brevity, this implied transition has been omitted from most states.

PS1: PORReset[1]	Set internal PS state to initial conditions including MaxNegSpeed=Max and SelHostPort=none. Assert A.OFFLINE. Assert B.OFFLINE. Assert D.OFFLINE.	
1. Power-on reset condition and explicit reset request negated and Mode = SideBand	→	SetHostPortSideBand
2. Power-on reset condition and explicit reset request negated and Mode = ProtocolBased	→	AwaitHostSelection
3. Power-on reset or explicit reset request asserted	→	PORReset
NOTE : 1. This state is entered asynchronously any time in response to power-on reset or an explicit reset request. An explicit reset request is a reset line/button on the Port Selector itself, not a COMRESET signal from a host.		

PS2: SetHostPortSideBand	Set SelHostPort based on value of side-band selection signal.	
1. COMRESET received from SelHostPort		ResetDevice
2. COMINIT received from D and COMRESET not received from SelHostPort		ResetHost
3. COMRESET not received from SelHostPort and COMINIT not received from D	→	SetHostPortSideBand

PS3: ComInitPropToBoth	Assert A.ONLINE. Assert A.PHYRESET. Assert B.ONLINE. Assert B.PHYRESET.	
1. Unconditional	→	AwaitHostSelection

PS4: AwaitHostSelection		
1. COMINIT received from D and (COMRESET or COMWAKE) not received from (A or B)	→	ComInitPropToBoth
2. (COMRESET or COMWAKE) received from A	→	SelectA
3. (COMRESET or COMWAKE) not received from A and (COMRESET or COMWAKE) received from B	→	SelectB
4. No OOB signal detected	→	AwaitHostSelection

PS5: SelectA	Assert B.OFFLINE. Set SelHostPort=A. Assert SelHostPort.ONLINE. Assert D.ONLINE. Assert SelHostPort.PHYRESET. Assert D.PHYRESET.	
1. Unconditional	→	WaitforComm

PS6: SelectB	Assert A.OFFLINE. Set SelHostPort=B. Assert SelHostPort.ONLINE. Assert D.ONLINE. Assert SelHostPort.PHYRESET. Assert D.PHYRESET.	
1. Unconditional	→	WaitforComm

PS7: WaitForComm		
1. SelHostPort.PHYRDY and D.PHYRDY and no change in PORTSELECT signal	→	CheckSpeeds
2. (!SelHostPort.PHYRDY or !D.PHYRDY) and timeout[1] not exceeded and no change in PORTSELECT signal	→	WaitForComm
3. (!SelHostPort.PHYRDY or !D.PHYRDY) and timeout[1] exceeded and no change in PORTSELECT signal	→	PowerManageCheck
4. PORTSELECT signal received for non-selected host port	→	ChangePort
NOTE: 1. The timeout is vendor specific but shall be larger than 1760 microseconds.		

PS8: CheckSpeeds[1]		
1. SelHostPort.SPDMODE = D.SPDMODE	→	Online
2. SelHostPort.SPDMODE > D.SPDMODE	→	SetDeviceSpeed
3. SelHostPort.SPDMODE < D.SPDMODE	→	SetHostSpeed
NOTE: 1. A larger value for the SPDMODE signal shall indicate a higher speed than a smaller value for SPDMODE.		

PS9: Online	Transfer DATA received on D to SelHostPort. Transfer DATA received on SelHostPort to D.	
1. PORTSELECT signal received for non-selected host port	→	ChangePort
2. SelHostPort.PHYRDY negated or D.PHYRDY negated	→	PowerManageCheck

PS10: SetDeviceSpeed	Set MaxNegSpeed=D.SPDMODE	
1. Unconditional	→	ReComm

PS11: SetHostSpeed	Set MaxNegSpeed=SelHostPort.SPDMODE	
1. Unconditional	→	ReComm

PS12: ReComm	Assert D.PHYRESET	
1. Unconditional	→	WaitForComm

PS13: ChangePort	Assert SelHostPort.OFFLINE. Assert !SelHostPort.ONLINE. Set SelHostPort=!SelHostPort. Set MaxNegSpeed=Max.	
1. Unconditional	→	WaitForComm

PS14: ResetHost	Assert SelHostPort.ONLINE. Assert D.ONLINE. Assert SelHostPort.PHYRESET	
1. Unconditional	→	WaitForComm
NOTE: 1. This state is entered unconditionally upon receipt of D.COMINIT if SelHostPort != none and the Port Selector is not in state ResetDevice.		

PS15: ResetDevice[1]	Assert SelHostPort.ONLINE. Assert D.ONLINE. Assert D.PHYRESET.		
1. Unconditional		→	WaitForComm
NOTE: 1. This state is entered unconditionally upon receipt of SelHostPort.COMRESET if SelHostPort != none.			

PS16: PowerManageCheck			
1. Port Selector determined that low power state entered is SLUMBER		→	PowerManageSlumber
2. Port Selector has not determined that low power state entered is SLUMBER		→	PowerManagePartial

PS17: PowerManagePartial	Assert SelHostPort.Partial. Assert D.Partial.		
1. SelHostPort.COMWAKE detected or D.COMWAKE detected		→	WaitForComm
2. PORTSELECT signal received for non-active host port		→	ChangePort
3. COMWAKE not received and no change in PORTSELECT signal		→	PowerManagePartial

PS18: PowerManageSlumber	Assert SelHostPort.Slumber. Assert D.Slumber.		
1. SelHostPort.COMWAKE detected or D.COMWAKE detected		→	WaitForComm
2. PORTSELECT signal received for non-active host port		→	ChangePort
3. COMWAKE not received and no change in PORTSELECT signal		→	PowerManageSlumber

17.4.2 BIST support

A Port Selector is not required to support the BIST Activate FIS. The resultant behavior of sending a BIST Activate FIS through a Port Selector is undefined.

17.4.3 Flow control signaling latency

The Port Selector shall satisfy the flow control signaling latency specified in section 9.4.7. The Port Selector shall ensure that the flow control signaling latency is met on a per link basis. Specifically, the Port Selector shall ensure that the flow control signaling latency is met between:

1. The Port Selector active host port and the host it is connected to
2. The Port Selector device port and the device it is connected to

The Port Selector shall not reduce the flow control signaling latency budget of the active host it is connected to or the device it is connected to.

17.4.4 Power Management

The Port Selector shall maintain the active host port across power management events and only allow an active host port change after receiving a valid port selection signal.

The Phy on the inactive host port shall be in the quiescent power condition. Upon detecting that the PHYRDY signal is not present for the active host port or device port, the Port Selector shall place that Phy in a quiescent power condition.

If the PHYRDY signal is not present between the device and the Port Selector, the Phy connected to the active host port shall enter the quiescent power condition and squelch the Phy transmitter. If the PHYRDY signal is not present between the active host and the Port Selector,

the Phy connected to the device shall enter the quiescent power condition and squelch the Phy transmitter. During these periods while PHYRDY is not present, OOB signals shall still be propagated between the active host and the device to ensure that communication is established.

If the Port Selector is able to determine that the active host and device negotiated a Slumber power management transition, the Port Selector may recover from the quiescent power condition in the time defined by the Slumber power state. If the Port Selector is not able to determine the power state entered by the host and device, the Port Selector shall recover from the quiescent power condition in the time defined by the Partial power state.

17.4.4.1 Wakeup Budget

The wakeup budget out of Partial or Slumber may increase when a Port Selector is connected to a device. When the active host Phy comes out of low power condition, the Port Selector active host Phy may wakeup before causing the Port Selector device Phy to wakeup which in turn wakes the device. The host shall allow the device at least 20 microseconds to wakeup from the Partial power management state.

17.4.5 OOB Phy signals

The Port Selector shall propagate COMRESET received from the active host to the device as specified in the Control State Machine in section 17.4.1. The Port Selector shall propagate COMINIT received from the device to the active host port as specified in the Control State Machine in section 17.4.1. If no active host port is selected, the Port Selector shall propagate COMINIT received from the device to the active host port as specified in the Control State Machine in section 17.4.1. The Port Selector is allowed to delay delivery of propagated OOB signals.

The Port Selector shall not respond to COMRESET signals received over the inactive host port. The inactive host port Phy shall remain in the quiescent power condition when COMRESET is received over the inactive host port.

17.4.6 Hot Plug

The Port Selector shall only generate a COMINIT over a host port when a COMINIT signal is received from the device or as part of an active speed negotiation as specified in the Control State Machine in section 17.4.1. If a device connected to a Port Selector is hot plugged, the device issues a COMINIT sequence as part of its normal power-up sequence in accordance with the Phy layer state machine. The Port Selector shall propagate the COMINIT over the active host port (or both host ports if both host ports are inactive). When the host detects the COMINIT signal, the host then interrogates the port to determine whether a device is attached.

If a device connected to a Port Selector is hot unplugged, the Port Selector shall squelch the transmitter for the active host port as detailed in section 17.4.4. The active host should then determine that the PHYRDY signal is no longer present and therefore that a device is no longer present.

17.4.7 Speed Negotiation

Speed is negotiated on a per link basis. Specifically, the Port Selector shall negotiate speed between:

1. The Port Selector active host port and the host to which it is connected
2. The Port Selector device port and the device to which it is connected

The Port Selector starts speed negotiation at the highest speed speed that it supports. The Port Selector then negotiates speed on each link to the appropriate supported speed. After negotiating speed on the active host link and on the device link, the Port Selector shall check whether the two speeds match. If the speeds do not match, the Port Selector limits the maximum speed speed it supports to the lower of the two speeds negotiated. Then the Port Selector forces speed to be renegotiated to reach a common speed speed.

17.4.8 Spread spectrum clocking

The Port Selector shall support spread spectrum clocking receive on all of its ports. The Port Selector may support spread spectrum transmit. It is recommended that a configuration jumper be used to enable/disable spread spectrum clocking if it is settable. There is no means within the Serial ATA protocol provided to enable/disable spread spectrum clocking if it is statically configurable.

If spread spectrum clocking is used, the spreading domain between the host and the Port Selector is not required to be the same as the spreading domain between device and the Port Selector. The signals passing through a Port Selector may be re-spread.

17.5 Power-up and Resets

17.5.1 Power-up

Upon power-up, the Port Selector shall reset all internal state, including the active host port. This causes no active host port to be selected when protocol-based port selection is used.

17.5.1.1 Presence detection of Port Selector

For protocol-based port selection, presence detection may be performed using the optional mechanism outlined in section 17.3.1.2. Presence detection for Port Selectors implementing side-band port selection is outside the scope of this specification.

17.5.2 Resets

17.5.2.1 COMRESET

When COMRESET is received over the active host port the Port Selector shall reset all internal state, the active host port shall remain unchanged, the maximum speed shall remain unchanged, and the COMRESET signal shall be propagated to the device. The Port Selector shall take no reset action upon receiving a COMRESET signal over the inactive host port except as specified in section 17.3.1 when there is no active host port selected after power-up.

17.5.2.2 Software reset and DEVICE RESET

The Port Selector shall not reset in response to receiving a Software reset or the DEVICE RESET command.

17.6 Host Implementation (Informative)

17.6.1 Software Method for Protocol-based Selection (Informative)

The preferred software method for producing a protocol-based port selection signal is detailed in this section. Software for HBAs that implement the SControl and SStatus registers may use this method to create the protocol-based port selection signal.

If the Phy is left on for long periods during the generation of the sequence, a hardware based COMRESET polling algorithm may interfere and corrupt the sequence. This method tries to minimize any impact of host based COMRESET polling algorithms by leaving the Phy online for very short sequences during the creation of the signal. The procedure outlined is appropriate for any HBA design that has a COMRESET polling interval greater than 25 microseconds.

1. Set Phy to offline by writing SControl.DET to 4h.
2. Set Phy to reset state by writing SControl.DET to 1h.
3. Wait 5 microseconds to allow charging time for DC coupled Phy designs.
4. Set Phy to online state by writing SControl.DET to 0h.
5. Wait 20 microseconds to allow COMRESET burst to be transmitted to the device.
6. Set Phy to offline by writing SControl DET to 4h.

7. Wait 1.975 milliseconds to satisfy T1 timing as specified in Table 104.

8. Repeat steps 2-6.
9. Wait 7.975 milliseconds to satisfy T2 timing as specified in Table 104.

10. Repeat steps 2-6.
11. Wait 1.975 milliseconds to satisfy T1 timing as specified in Table 104.

12. Repeat steps 2-6
13. Wait 7.975 milliseconds to satisfy T2 timing as specified in Table 104.

14. Set Phy to reset state by writing SControl.DET to 1h.
15. Wait 5 microseconds to allow charging time for DC coupled Phy designs.
16. Set Phy to online state by writing SControl.DET to 0h.
17. Wait up to 10 milliseconds for SStatus.DET = 3h
18. If SStatus.DET != 3h, go to step 1 to restart the process.

The procedure is also outlined in pseudocode on the following page.

```
//
// Continue to perform this procedure until SStatus.DET == 3.
//
while (SStatus.DET != 3)
{
    SControl.DET = 4;             // Turn off Phy

    //
    // Mimic out the COMRESET bursts at the appropriate T1/T2 timing intervals.
    //
    for (i = 0; i < 4; i++)
    {
        SControl.DET = 1;         // Place HBA in reset state
        Sleep(5);                 // Wait for Phy to charge, in microseconds
        SControl.DET = 0;         // Issue COMRESET
        Sleep(20);                // Wait for COMRESET to be sent
        SControl.DET = 4;         // Turn off Phy

        if ((i == 0) || (i == 2))
        {
            Sleep(1975);          // Wait T1 time minus time already waited
        }
        else if ((i == 1) || (i == 3))
        {
            Sleep(7975);          // Wait T2 time minus time already waited
        }
    }

    //
    // Issue final COMRESET of the burst.
    //
    SControl.DET = 1;             // Place HBA in reset state
    Sleep(5);                     // Wait for Phy to charge, in microseconds
    SControl.DET = 0;             // Issue COMRESET


    //
    // Wait up to 10 milliseconds for PHYRDY.
    //
    for (i = 0; i < 10000; i++)
    {
        if (SStatus.DET == 3)
        {
            break;                // Stop procedure if SStatus.DET == 3
        }
        Sleep(1);                 // Wait 1 microsecond
    }
```

Appendix A. Sample Code for CRC and Scrambling (Informative)

A.1 CRC calculation

A.1.1 Overview

The following section provides an informative implementation of the CRC polynomial. The example is intended as an aid in verifying a HDL implementation of the algorithm.

A.1.2 Maximum frame size

The 32-bit CRC used by Serial ATA may be shown to provide detection of two 10-bit errors up to a maximum frame size of 16384 bytes. This provides for future expansion of 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 Register Host to Device 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 EOF_P.. The code displays both the resulting command FIS and the intermediate CRC polynomial values. To compile the code with the GNU tool chain, use the following command:

"gcc –o crc.exe crc.c"

This code is supplied for illustrative purposes only.

A.1.4 Example code for CRC algorithm

```
/*****************************************************************************/
/*                                                                           */
/* 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 + x + 1 */
/*                                                                           */
/* This sample code uses a parallel implementation of the CRC calculation   */
/* circuit that is suitable for implementation in hardware.  A block        */
/* diagram of the circuit being emulated is shown below.                    */
/*                                                                           */
/*                     +---+          +---+          +---+                  */
/*   Data_In --------->|   |          |   |          | R |                  */
/*                     | + |--------->| * |--------->| e |----+             */
/*                +--->|   |          |   |          | g |    |             */
/*                |    +---+          +---+          +---+    |             */
/*                |                                           |             */
/*                |                                           |             */
/*                +-------------------------------------------+             */
/*                                                                           */
/* The CRC value is initialized to 0x52325032 as defined in the Serial ATA  */
/* specification.                                                            */
/*                                                                           */
/*****************************************************************************/

#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%8x", &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.1.5 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 section 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 Register Host to Device 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. To compile the code with the GNU tool chain, use the following command:

"gcc –o scramble.exe scramble.c"

This code is supplied for illustrative purposes only.

A.2.3 Example scrambler implementation

```
/*****************************************************************************/
/*                                                                           */
/* scramble.c                                                                */
/*                                                                           */
/* This sample code generates the entire sequence of 65535 Dwords produced   */
/* by the scrambler defined in the Serial ATA specification.  The            */
/* specification calls for an LFSR to generate a string of bits that are     */
/* then packaged into 32 bit Dwords to be XORed with the data Dwords.  The   */
/* generator polynomial specified is:                                        */
/*                        16   15   13   4                                   */
/*                  G(x) = x  + x  + x  + x + 1                              */
/*                                                                           */
/* Parallelized versions of the scrambler are initialized to a value         */
/* derived from the initialization value of 0xFFFF defined in the            */
/* specification.  This implementation is initialized to 0xF0F6.  Other      */
/* parallel implementations have different initial values.  The              */
/* important point is that the first Dword output of any implementation      */
/* needs to equal 0xC2D2768D.                                                */
/*                                                                           */
/* This code does not represent an elegant solution for a C implementation,  */
/* but it does demonstrate a method of generating the sequence that can be   */
/* easily implemented in hardware.  A block diagram of the circuit emulated  */
/* by this code is shown below.                                              */
/*                                                                           */
/*         +-----------------------------------+                             */
/*         |                                   |                             */
/*         |                                   |                             */
/*         |     +---+               +---+     |                             */
/*         |     | R |               | * |     |                             */
/*         +---->| e |---------+---->| M |-----+----> Output(31 downto 16)  */
/*               | g |         |     | 1 |                                   */
/*               +---+         |     +---+                                   */
/*                             |                                             */
/*                             |     +---+                                   */
/*                             |     | * |                                   */
/*                             +---->| M |---------> Output(15 downto 0)     */
/*                                   | 2 |                                   */
/*                                   +---+                                   */
/*                                                                           */
/* The register shown in the block diagram is a 16 bit register.  The two    */
/* boxes, *M1 and *M2, each represent a multiply by a 16 by 16 binary        */
/* matrix.  A 16 by 16 matrix times a 16 bit vector yields a 16 bit vector.  */
/* The two vectors are the two halves of the 32 bit scrambler value.  The    */
/* upper half of the scrambler value is stored back into the context         */
/* register to be used to generate the next value in the scrambler           */
/* sequence.                                                                 */
```

```
/*                                                                          */
/****************************************************************************/

#include <stdlib.h>
#include <stdio.h>


main(argc,argv)
int argc;
char *argv[];


{
   int               i, j;
   unsigned short    context;       /* The 16 bit register that holds the context or state */
   unsigned long     scrambler;     /* The 32 bit output of the circuit    */
   unsigned char     now[16];       /* The individual bits of context      */
   unsigned char     next[32];      /* The computed bits of scrambler      */



   /* Parallelized versions of the scrambler are initialized to a value     */
   /* derived from the initialization value of 0xFFFF defined in the        */
   /* specification.  This implementation is initialized to 0xF0F6.  Other  */
   /* parallel implementations have different initial values.  The          */
   /* important point is that the first Dword output of any implementation  */
   /* needs to equal 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.2.4 Example scrambler implementation output

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(7:0) = 2

Table 105 – CRC and scrambler calculation example - PIO Write Command

FIS Data (hex)	Scrambler Value (hex)	Scrambled Data (hex)	Accumulated CRC Value (hex)	Comments
3737B57C	na	3737B57C	na	SOF_P, primitives not scrambled, scrambler reset
00308027	C2D2768D	C2E2F6AA	11E353FD	Register – Host to Device FIS, Command = 30, Cbit set
E1234567	1F26B368	FE05F60F	0F656DA7	LBA 1234567
00000000	A508436C	A508436C	3D14369C	Extended LBA = 0
00000002	3452D354	3452D356	92D0D681	Device Control Register = 0, 2 sectors
00000000	8A559502	8A559502	319FFF6F	reserved = 0
319FFF6F	BB1ABE1B	8A854174	na	CRC
D5D5B57C	na	D5D5B57C	na	EOF, primitives not scrambled

Note: All values in hexadecimal, shown 31:0

The transmitted Dwords for this Register FIS, prior to 8b/10b encoding, are :

SOF_P
C2E2F6AAh
FE05F60Fh
A508436Ch
3452D356h
8A559502h
8A854174h
EOF_P

Appendix B. Command processing overview (Informative)

B.1 Non-data commands

When the Command register is written by the BIOS or software driver, the host adapter sets the BSY bit to one in the shadow Status register and transmits a Command frame to the device.

When command actions are complete, the device transmits a Register frame to set ending content of the shadow registers.

B.2 DMA read by host from device

- 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).
- The host driver software issues the command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command Shadow Register Block Register being written, the host adapter sets the BSY bit to one in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device has processed the command and is ready, 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.
- 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.
- Upon completion of the transfer, the device transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit to zero in the Status register, and if the interrupt flag is set in the header an interrupt is asserted to the host.

In some error conditions there may be no Data FIS transmitted prior to the Register FIS being transmitted with the status information. If so, the host software driver shall abort the setup of the DMA controller.

B.3 DMA write by host to device

- 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). As a result the DMA controller becomes armed but remains paused pending a signal from the device to proceed with the data transfer.
- The host driver software issues the command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command Shadow Register Block Register being written, the host adapter sets the BSY bit in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- 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 FIS. 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 and every one of the subsequent Data FISes. The amount of data

transmitted to the device is determined by the transfer count programmed into the host adapter's DMA controller by the host driver software during the command setup phase.
- Upon completion of the transfer, the device transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit to zero in the Status register, and if the interrupt flag is set in the header an interrupt is asserted to the host.

In some error conditions the device may signal an ending status by transmitting a register frame to the host without having transmitted a DMA Activate FIS. In such cases the host driver software should abort and clean up the DMA controller.

B.4 PIO data read from the device

- The host driver software issues a PIO read command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit to one in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device has processed the command and is ready to begin transferring data to the host, it first transmits a PIO Setup FIS to the host. Upon receiving the PIO Setup FIS, the host adapter holds the FIS contents in a temporary holding buffer.
- 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, the host adapter transfers the register contents from the PIO Setup FIS into the shadow registers including the initial status value, resulting in DRQ bit getting set to one and BSY bit getting cleared to zero in the Status register. Also, if the Interrupt bit is set to one, an interrupt is generated to the host.
- The host controller receives the incoming data that is part of the Data FIS into a speed matching FIFO that is conceptually attached to the Shadow Register data Block Register.
- As a result of the issued interrupt and DRQ being set to one in the Status register, host software does a REP INSW on the data register and pulls data from the head of the speed matching FIFO while the serial link is adding data to the tail of the FIFO. The flow control scheme handles data throttling to avoid underflow/overflow of the receive speed matching FIFO that feeds the data Shadow Register Block Register.
- When the number of words read by host software from the Data shadow register reaches the value indicated in the PIO Setup FIS, the host transfers the ending status value from the earlier PIO Setup FIS into the Shadow Status register resulting in DRQ being cleared to zero and the ending status reported.
- If there are more data blocks to be transferred, the ending status indicates BSY bit being set to one, and the process repeats from the device sending the PIO Setup FIS to the host.

B.5 PIO data write to the device

- The host driver software issues a PIO write command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit to one in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device is ready to receive the PIO write data, it transmits a PIO Setup FIS to the host to indicate that the target is ready to receive PIO data and the number of words of data that are to be transferred.
- In response to a PIO Setup FIS with the D bit indicating a write to the device, the host transfers the beginning Status register contents from the PIO Setup FIS into the Shadow Status register, resulting in DRQ bit getting set to one and BSY bit cleared to zero. Also, if the Interrupt bit is set to one, an interrupt is generated to the host.

- As a result of DRQ bit being set to one in the Shadow Status register, the host driver software starts a REP OUTSW to the data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The REP OUTSW pushes data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO Setup frame into the shadow Status register resulting in DRQ being cleared to zero, and closes the frame with a CRC and EOF_P. If additional sectors of data are to be transferred, the ending status value transferred to the Shadow Status register would have the BSY bit set to one and the state is the same as immediately after the command was first issued to the device.
- If there are more data blocks to be transferred, the ending status indicates BSY bit being set to one, and the process repeats from the device sending the PIO Setup FIS to the host.
- When the number of sectors indicated in the Sector Count register have been transferred, the device shall send a Register – Device to Host FIS with the command complete interrupt and BSY bit cleared to zero.
- In the case of a write error, the device may, on any sector boundary include error status and a command complete interrupt in the PIO setup FIS, and there is no need to send the Register – Device to Host FIS.

B.6 ATA Tagged Command Queuing DMA read from device

- 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 host driver software issues the command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command Shadow Register Block Register being written, the host adapter sets the BSY bit to one in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device has queued the command and wishes to release the bus, it transmits a Register FIS to the host resulting in the BSY bit being cleared to zero and the REL bit being set to one in the Status register.
- When the device is ready 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 Status register. If no other command is active (i.e. BSY bit set to one), then an interrupt is also generated.
- In response to the service request, the host software deactivates the DMA controller (if activated) and issues a SERVICE command to the device by writing the Shadow Register Block Registers, resulting in the BSY bit getting set to one and a Register FIS being transmitted to the device.
- In response to the SERVICE request, the device transmits a Register FIS to the host conveying the TAG value to the host and clearing the BSY bit to zero and setting the DRQ bit to one.
- When the DRQ bit is set to one, the host software reads the TAG value from the Shadow Register Block and restores the DMA controller context appropriate for the command that is completing.
- The device transmits the read data to the host in the form of one or more Data FIS. This transfer proceeds in response to flow control signals/readiness. Any DMA data arriving before the DMA controller has its context restored backs up into the inbound speed-matching FIFO until the FIFO is filled and thereafter is flow controlled to throttle the incoming data until the DMA controller has its context restored by host software.
- The host controller recognizes that the incoming packet is a Data FIS and the DMA controller is programmed, and directs the incoming data to the host controller’s DMA controller that forwards the incoming data to the appropriate host memory locations.

- Upon completion of the transfer, the target transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit to zero in the Status register, and if the interrupt flag is set in the header an interrupt is asserted to the host.

B.7 ATA Tagged Command Queuing DMA write to device

- 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 host driver software issues the command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the Shadow Register command Block Register being written, the host adapter sets the BSY bit to one in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- When the device has queued the command and wishes to release the bus, it transmits a Register FIS to the host resulting in the BSY bit being cleared to zero and the REL bit being set to one in the Status register.
- When the device is ready 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 Status register. If no other command is active (i.e. BSY bit set to one), then an interrupt is also generated.
- In response to the service request, the host software deactivates the DMA controller (if activated) and issues a SERVICE command to the device by writing the Shadow Register Block Registers, resulting in the BSY bit getting set to one and a Register FIS being transmitted to the device.
- In response to the SERVICE request, the device transmits a Register FIS to the host conveying the TAG value to the host and clearing the BSY bit to zero and setting the DRQ bit to one.
- When the DRQ bit is set to one, the host software reads the TAG value from the Shadow Register Block and restores the DMA controller context appropriate for the command that is completing.
- 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 and every one 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. If the DMA Activate FIS arrives at the host prior to host software restoring the DMA context, the DMA Activate FIS results in the DMA controller starting the transfer as soon as the host software completes programming it (i.e. the controller is already activated, and the transfer starts as soon as the context is restored).
- Upon completion of the transfer, the target transmits a Register – Host to Device FIS to indicate ending status for the command, clearing the BSY bit to zero in the Status register, and if the Interrupt bit is set to one an interrupt is asserted to the host.

B.8 ATAPI Packet commands with PIO data in

- The host driver software issues a PACKET command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit to one in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.

- 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 target is ready to receive PIO data and the number of words of data that are to be transferred.
- The host transfers the beginning Status register contents from the PIO Setup FIS into the Shadow Status register, resulting in BSY bit getting cleared to zero and DRQ bit getting set to one.
- As a result of BSY bit being cleared to zero and DRQ bit being set to one in the Shadow Status register, the host driver software writes the command packet to the Shadow Register Block data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The writes to the data register push data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO setup frame into the Shadow Status register resulting in DRQ bit being cleared to zero and BSY bit being set to one, and closes the frame with a CRC and EOF_P. This completes the transmission of the command packet to the device.
- When the device has processed the command and is ready to begin transferring data 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.
- 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 register contents from the PIO Setup FIS into the shadow registers including the initial status value, resulting in DRQ bit getting set to one and BSY bit getting cleared to zero in the Status register. Also, if the Interrupt bit is set to one, an interrupt is generated to the host.
- The host controller receives the incoming data that is part of the Data FIS into a speed matching FIFO that is conceptually attached to the data Shadow Register Block Register.
- As a result of the issued interrupt and DRQ bit being set to one in the Status register, host software reads the byte count and does a REP INSW on the data register to pull data from the head of the speed matching FIFO while the serial link is adding data to the tail of the FIFO. The flow control scheme handles data throttling to avoid underflow/overflow of the receive speed matching FIFO that feeds the data Shadow Register Block Register.
- 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 transfers the ending status value from the earlier PIO Setup into the Shadow Status register resulting in DRQ bit being cleared to zero and BSY bit being set to one.
- The device transmits final ending status by sending a Register FIS with BSY bit cleared to zero and the ending status for the command and the Interrupt bit set to one.
- The host detects an incoming register frame that contains BSY bit cleared to zero and ending status for the command and the Interrupt bit set to one and places the frame content into the shadow registers to complete the command.

B.9 ATAPI Packet commands with PIO data out

- The host driver software issues a PACKET command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit to one in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- 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 target is ready to receive PIO data and the number of words of data that are to be transferred.

- The host transfers the beginning Status register contents from the PIO Setup – Device to Host FIS into the shadow Status register, resulting in BSY bit being cleared to zero and DRQ bit being set to one.
- As a result of BSY bit being cleared to zero and DRQ bit being set to one in the Shadow Status register, the host driver software writes the command packet to the Shadow Register Block data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The writes to the data register push data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO Setup frame into the Shadow Status register resulting in DRQ bit being cleared to zero and BSY bit being set to one, and closes the frame with a CRC and EOF_P. This completes the transmission of the command packet to the device.
- 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 target is ready to receive PIO data and the number of words of data that are to be transferred.
- In response to a PIO Setup FIS with the D bit cleared to zero indicating a write to the device, the host transfers the beginning Status register contents from the PIO Setup FIS into the Shadow Status register, resulting in DRQ bit getting set to one. Also, if the Interrupt bit is set to one, an interrupt is generated to the host.
- As a result of DRQ bit being set to one in the Shadow Status register, the host driver software reads the byte count and starts a REP OUTSW to the data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – host to device FIS. The REP OUTSW pushes data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO Setup FIS into the Shadow Status register resulting in DRQ bit being cleared to zero, BSY bit being set to one, and closes the frame with a CRC and EOF_P.
- The device transmits final ending status by sending a Register – Device to Host FIS with BSY bit cleared to zero and the ending status for the command and the Interrupt bit set to one.
- The host detects an incoming Register – Device to Host FIS that contains BSY bit cleared to zero and ending status for the command and the Interrupt bit set to one and places the frame content into the shadow registers to complete the command.

B.10 ATAPI Packet commands with DMA data in

- 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 host driver software issues a PACKET command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit to one in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- 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 target is ready to receive PIO data and the number of words of data that are to be transferred.
- The host transfers the beginning Status register contents from the PIO Setup FIS into the Shadow Status register, resulting in BSY bit getting cleared to zero and DRQ bit getting set to one.

- As a result of BSY bit getting cleared to zero and DRQ bit being set to one in the Shadow Status register, the host driver software writes the command packet to the Shadow Register Block data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The writes to the data register push data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO setup frame into the Shadow Status register resulting in DRQ bit being cleared to zero and BSY bit being set to one, and closes the frame with a CRC and EOF_P. This completes the transmission of the command packet to the device.
- 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.
- The host controller recognizes that the incoming packet is a Data FIS and the DMA controller is programmed, and directs the incoming data to the host controller's DMA controller that forwards the incoming data to the appropriate host memory locations.
- Upon completion of the transfer, the target transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit to zero in the Status register, and if the Interrupt bit is set to one an interrupt is asserted to the host.

B.11 ATAPI Packet commands with DMA data out

- 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). As a result the DMA controller becomes armed but remains paused pending a signal from the device to proceed with the data transfer.
- The host driver software issues a PACKET command to the device by writing the Shadow Register Block Registers (command register last).
- In response to the command register being written, the host adapter sets the BSY bit to one in the Shadow Status register and transmits a Register – Host to Device FIS to the device with the Shadow Register Block contents.
- 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 target is ready to receive PIO data and the number of words of data that are to be transferred.
- The host transfers the beginning Status register contents from the PIO Setup FIS into the Shadow Status register, resulting in BSY bit getting cleared to zero and DRQ bit getting set to one.
- As a result of BSY bit getting cleared to zero and DRQ bit being set to one in the Shadow Status register, the host driver software writes the command packet to the Shadow Register Block data register.
- The data written to the data register is placed in an outbound speed matching FIFO and is transmitted to the device as a Data – Host to Device FIS. The writes to the data register push data onto the tail of the FIFO and the serial link pulls data from the head. The flow control scheme handles data throttling to avoid underflow of the transmit FIFO.
- When the number of words indicated in the PIO Setup FIS have been written to the transmit FIFO, the host controller transfers the final status value indicated in the PIO Setup frame into the shadow Status register resulting in DRQ bit being cleared to zero and BSY bit being set to one, and closes the frame with a CRC and EOF_P. This completes the transmission of the command packet to the device.
- 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 FIS. 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.
- Upon completion of the transfer, the device transmits a Register – Device to Host FIS to indicate ending status for the command, clearing the BSY bit to zero in the Status register, and if the Interrupt bit is set to one an interrupt is asserted to the host.

B.12 Odd word count considerations

This section 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 Serial ATA is made for the transfer of an odd number of 8-bit byte quantities.

B.12.1 DMA read from target for odd word count

- 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.
- 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 Dword.
- The host controller 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 ATA/ATAPI Host Adapters standard).
- 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.

B.12.2 DMA write by host to target for odd word count

- 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.
- 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.
- 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 Dword 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 Dword.
- 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.

B.13 PIO data read from the device

- 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 Dword.
- Host driver software responsible for retrieving the PIO data is aware of the number of words of data it expects to retrieve from the Shadow Command Block Register Data register and performs a REP INSW operation for an odd number of repetitions.
- Upon exhaustion of the REP INSW 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 Shadow Command Block Register Command register or upon the receipt of the next data FIS from the device.

B.14 PIO data write to the device

- 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.
- Host driver software responsible for transmitting the PIO data is aware of the number of words of data it needs to write to the Shadow Command Block Register Data register and performs a REP OUTSW operation for an odd number of repetitions.
- After the final write by the software driver to the Shadow Command Block Register Data register, the transfer count indicated in the PIO Setup packet is exhausted, which signals the host controller 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 controller 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 Dword.
- 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.

B.15 Native Command Queuing Examples (Informative)

The following is an overview of macro operations and their sequencing for two typical Native Command Queuing scenarios. These illustrative sequences presume a host controller DMA implementation equivalent to current mainstream desktop implementations (hence references to PRD tables and other data structures typically referenced for such implementations) and these illustrative sequences are not intended to exclude other possible host controller implementations.

B.15.1 Queued Commands with Out of Order Completion

HOST Actions	*DEVICE Actions*
Host issues Read Command Tag=0 by presetting bit 0 in the SActive register by writing the value 00000001h (…00000001b) to it and transmitting a Register FIS to the device.	
	Device clears BSY bit to zero by transmitting a Register FIS to the host.
Host issues Read Command Tag=5 (if BSY bit not yet cleared to zero, host needs to wait) by presetting bit 5 in the SActive register by writing the value 00000020h (…00100000b) to it and transmitting a Register FIS to device. The resultant SActive register value is 21h.	
	Device clears BSY bit to zero.
	Device sends DMA Setup FIS, DMA Buffer ID=5 (in this example the second issued command is serviced first).
Host loads PRD pointer into DMA engine corresponding to buffer 5.	
	Device sends data for command corresponding to TAG=5.
Host DMA engine directs incoming data into buffer 5.	
	Device sends Set Device Bits FIS with Interrupt bit set to one and with SActive value of 00000020h (…00100000b), indicating that TAG=5 has finished.
Host receives Set Device Bits FIS with SActive field value of 20h and Interrupt bit set to one. Results in bit 5 in SActive register getting cleared yielding a value of 01h in the SActive shadow register and interrupt getting triggered.	
	Device sends DMA Setup FIS, DMA Buffer ID=0
Host loads PRD pointer into DMA engine corresponding to buffer 0. Host software processes the received interrupt. Reads SActive shadow register and determines that bit 5 is de-asserted and retires command with TAG=5	
	Device sends data for command corresponding to TAG=0
Host DMA engine directs incoming data into Buffer 0	
	Device sends Set Device Bits FIS with Interrupt bit set to one and with SActive value of 00000001h (…00000001b), indicating that TAG=0 has finished.

HOST Actions	***DEVICE Actions***
Host receives Set Device Bits FIS with SActive field value of 01h and Interrupt bit set to one. Results in bit 0 in SActive register getting cleared yielding a value of 00h in SActive shadow register and interrupt getting triggered.	
	Device idle
Host software processes the received interrupt. Reads SActive shadow register and determines that bit 0 is de-asserted and retires command with TAG=0. Host idle.	

B.15.2 Interrupt Aggregation

HOST Actions	***DEVICE Actions***
Host issues Read Command Tag=0 by presetting bit 0 in the SActive register by writing the value 00000001h (…00000001b) to it and transmitting a Register FIS to the device.	
	Device clears BSY bit to zero by transmitting a Register FIS to the host.
Host issues Read Command Tag=5 (if BSY bit not yet cleared to zero, host needs to wait) by presetting bit 5 in the SActive register by writing the value 00000020h (…00100000b) to it and transmitting a Register FIS to device. The resultnant SActive register value is 21h.	
	Device clears BSY bit to zero.
	Device sends DMA Setup FIS, DMA Buffer ID=5 (in this example the second issued command is serviced first).
Host loads PRD pointer into DMA engine corresponding to buffer 5.	
	Device sends data for command corresponding to TAG=5.
Host DMA engine directs incoming data into buffer 5.	
	Device sends Set Device Bits FIS with Interrupt bit set to one and with SActive value of 00000020h (…00100000b), indicating that TAG=5 has finished.
Host receives Set Device Bits FIS with SActive field value of 20h and Interrupt bit set to one. Results in bit 5 in SActive register getting cleared yielding a value of 01h in the SActive shadow register and interrupt getting triggered.	
	Device sends DMA Setup FIS, DMA Buffer ID=0

HOST Actions	*DEVICE Actions*
Host loads PRD pointer into DMA engine corresponding to buffer 0	
	Device sends data for command corresponding to TAG=0
Host DMA engine directs incoming data into buffer 0 Host is busy and is slow to respond to and clear the received interrupt. Host interrupt response time is slow relative to device completion speed for this example.	
	Device sends Set Device Bits FIS with Interrupt bit set to one and with SActive value of 00000001h (…00000001b), indicating that TAG=0 has finished.
Host receives Set Device Bits FIS with SActive field value of 01h and Interrupt bit set to one. Results in bit 0 in SActive register getting cleared yielding value of 00h in SActive shadow register and interrupt getting triggered. If the host had not already reset the pending interrupt from the completion of TAG=5, no new interrupt is triggered. If the previous interrupt has already been reset, then a new interrupt is triggered which is the previous example illustrated in section B.15.1	
Host had a long latency, but is now processing the interrupt and has reset the interrupt pending flag. Host is now doing command completion processing. The second interrupt issued by the device got aggregated because the first interrupt didn't get reset soon enough. Host reads SActive shadow register and sees that TAG=0 and TAG=1 commands have both completed (neither have their SActive bit set). Host processes command completions and retires both commands.	
	Device idle

Appendix C. Device Emulation of nIEN with Interrupt Pending (Informative)

This specification defines the Interrupt 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 bit in received Host to Device FISes. In this specification, devices ignore the nIEN bit in received Host to Device FISes and always perform as if nIEN bit is cleared to zero. See sections 10.3.4 and 10.3.5.

Some devices implemented to prior Serial ATA specification revisions used the nIEN bit of the Register Host to Device FIS as a pre-condition to setting the Interrupt bit to one of the Register Device to Host and Set Device Bits FISes. The purpose of using nIEN bit to enable the Interrupt bit was to emulate the operation of the parallel implementation of ATA. In the parallel implementation, when the nIEN bit is cleared to zero, the device is enabled for the INTRQ line to the host. When the nIEN bit 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 section 13.1.3), and it is also required for Overlapped operation.

In this specification, the implementation of Device 0 / Device 1 emulation is performed exclusively by the host (see section 13.1.3).

One serious side effect of device emulation of nIEN is the possibility of lost interrupts. In the parallel implementation, a host may 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 bit, a Register Device to Host FIS may be received with Interrupt bit cleared to zero (since it is masked by the nIEN bit). The device, however, may have an interrupt pending at that time. When the host writes the Device Control register with nIEN bit 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 Interrupt bit set to one condition to the host. In this instance, the host has to resort to a polling operation to resume the operation.

The system designer should be aware of the following:

1) The Interrupt bit is the interrupt pending flag for Serial ATA devices.

2) The behavior of the Interrupt bit may be modified by the state of the nIEN bit in devices implemented to prior versions of the Serial ATA specification. Such devices may change the behavior of the Interrupt bit based on the current state of the nIEN bit, as last written by a Register Host to Device FIS with C bit cleared to zero or C bit set to one. Some devices do not observe the nIEN bit when C bit is set to one. Devices that do not observe the nIEN bit when C bit is set to one have compatibility issues with hosts that only transmit a Control Register FIS when the SRST bit changes state.

3) There is no defined behavior for a device when nIEN bit changes and the modification of the behavior of the Interrupt bit by nIEN is vendor specific.

4) The host should clear the nIEN bit to zero in all Register Host to Device FISes. This results in the highest compatibility with devices that use the nIEN bit as a pre-condition to setting the interrupt pending flag.

5) If a device supports nIEN emulation, and the nIEN bit is set to one by the host, the host driver should accommodate the case of no interrupt generation when nIEN bit is cleared to zero and the device has a pending interrupt.

Appendix D. I/O Controller Module (Informative)

The purpose of this optional feature is to specify the IO Connectors (shown with heavy outline below) that provide interconnection for power, data and enclosure services between I/O controller(s) and a Serial ATA disk backplane. This specification defines interoperability between I/O modules from one manufacturer which work with backplanes from a different manufacturer. This interoperability includes electrical, mechanical and enclosure management connections. Compliant I/O controllers and Serial ATA backplanes shall adhere to the specified connector placement and connector pinout.

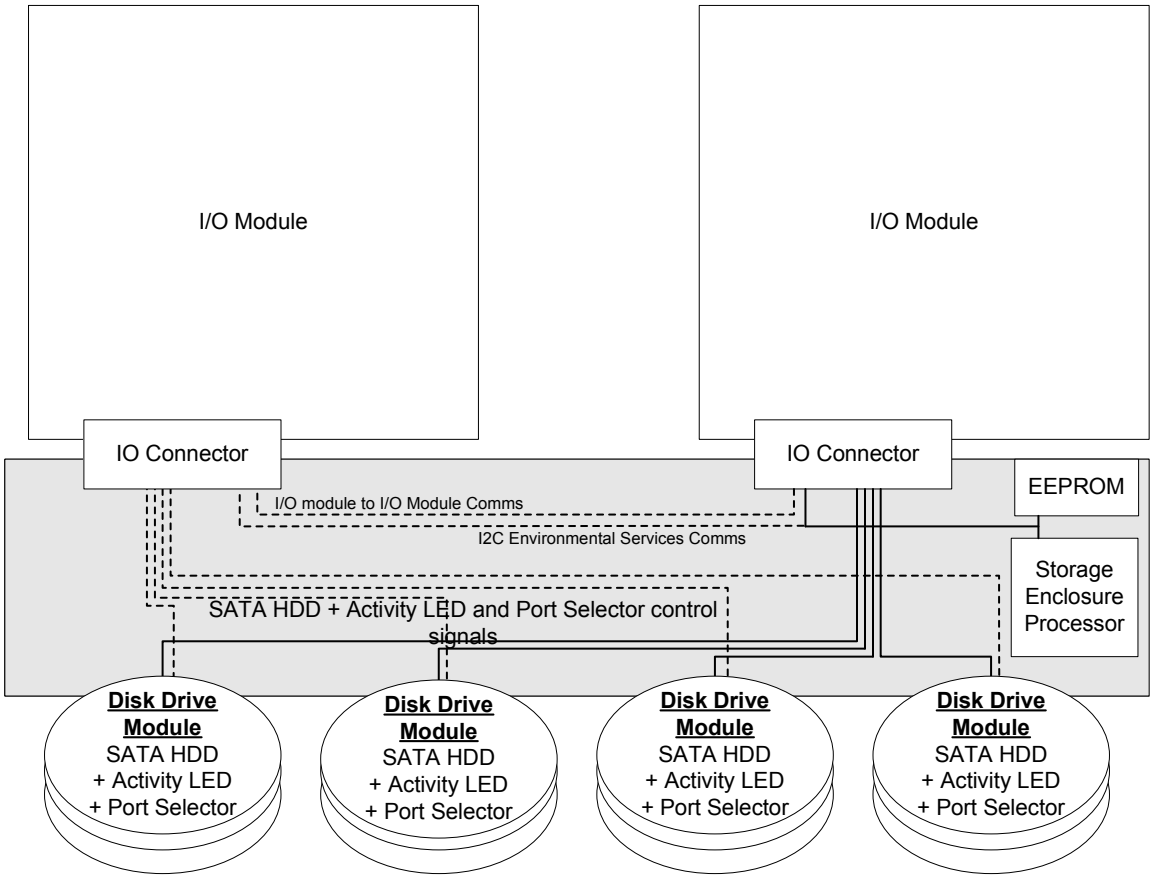


Figure 258 – Concept summary interconnect structure

The I/O connector(s), shown in Figure 258 with the heavy outline, are the central items being specified. The I/O Module in dotted line format represents a secondary and optional I/O module. The shaded area represents the backplane. The Disk Drive Module is envisioned as a standard Serial ATA HDD within a carrier, including:

- power connection between the module and the backplane
- data connection between the module and the backplane
- HDD activity LED (see section 6.7.1), optional
- Port Selector with a secondary connection to the backplane, optional

D.1 Supported Configurations

Two configuration goals are supported by the connector specification:

1. Single I/O Controller System
2. Dual I/O Controller System with redundancy features (optional)

Each implementation has mandatory and optional features supported through the use of the available signals.

D.1.1 Single I/O Controller Signals

- Serial ATA interface
- I/O controller to enclosure processor (SEP) communication
- Disk activity LED signals.
- I/O controller power and ground
- I/O controller identification and control (RESET)

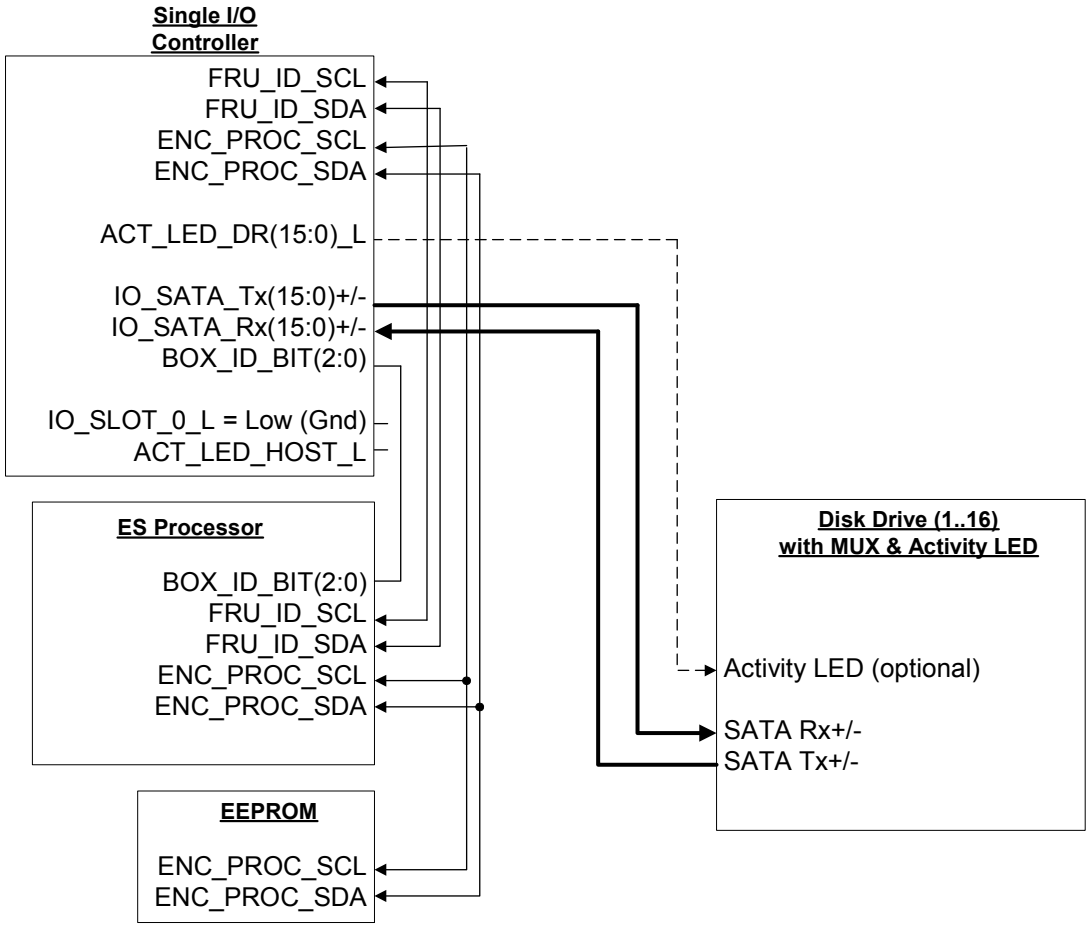


Figure 259 – An example of signal connections with one I/O Controller

D.1.2 Dual I/O Controller Signals

As for single I/O controller and in addition:

- Multiplexer control to provide dual access for I/O controller failover capabilities
- I/O controller to I/O controller communication

D.1.3 Further optional features

- RAID battery backup support
- Additional low and high speed communications for optional board-to-board communications. As this specification is not intended to define board-to-board for boards from different manufacturers; the actual implementation of signaling and protocol is left to the discretion of the I/O controller manufacturer.

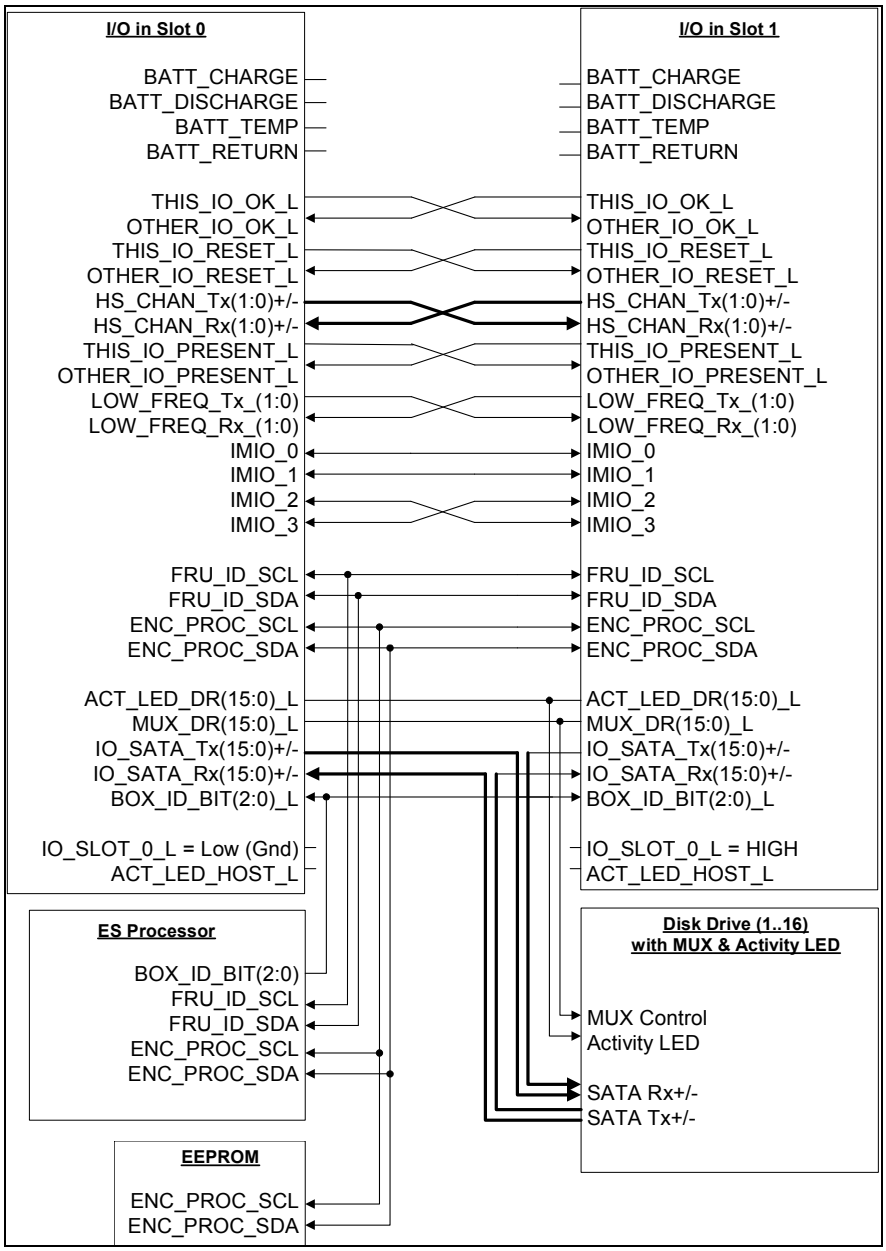


Figure 260 – Example of signal connections with two I/O Modules

D.2 Optional High Speed Channel configurations

The optional High speed communications channels may be used for high speed differential communications between the two I/O controllers, or for I/O controller to host communications such as inclusion into a Fibre Channel loop. It is recommended that these signals should be 100 ohms differential characteristic impedance.

Actual usage is open to the user definition however the normative backplane routing should be either:

Configuration 0: Both channels link I/O controller to I/O controller

Configuration 1: One channel links the two I/O Controllers, and the other is router to a host connector.

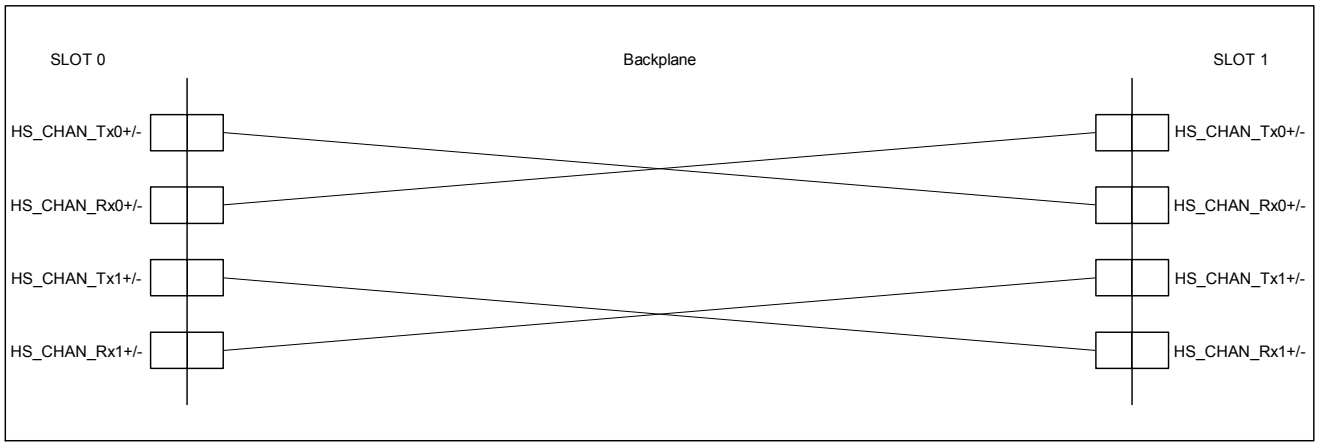


Figure 261 – High Speed Channels – Configuration 0

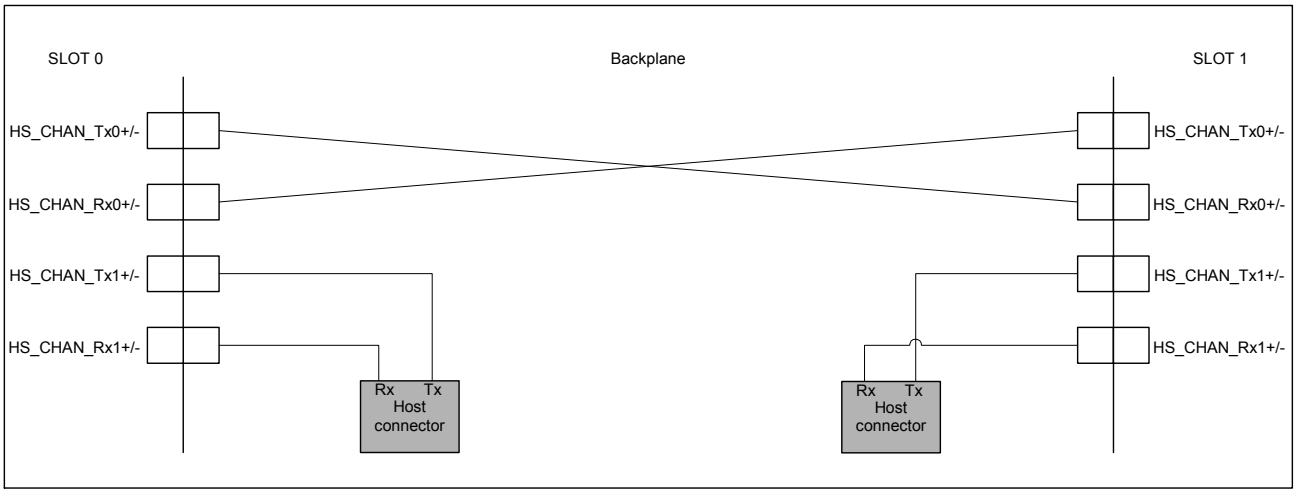


Figure 262 – High-Speed Channels – Configuration 1

D.3 Optional Low Speed Channel configurations

The optional Low speed communications channels may be used for end-to-end I/O controller to I/O controller communications. There is a pair transmit and receive signals for low frequency (<500 kHz) I/O controller to I/O controller communications.

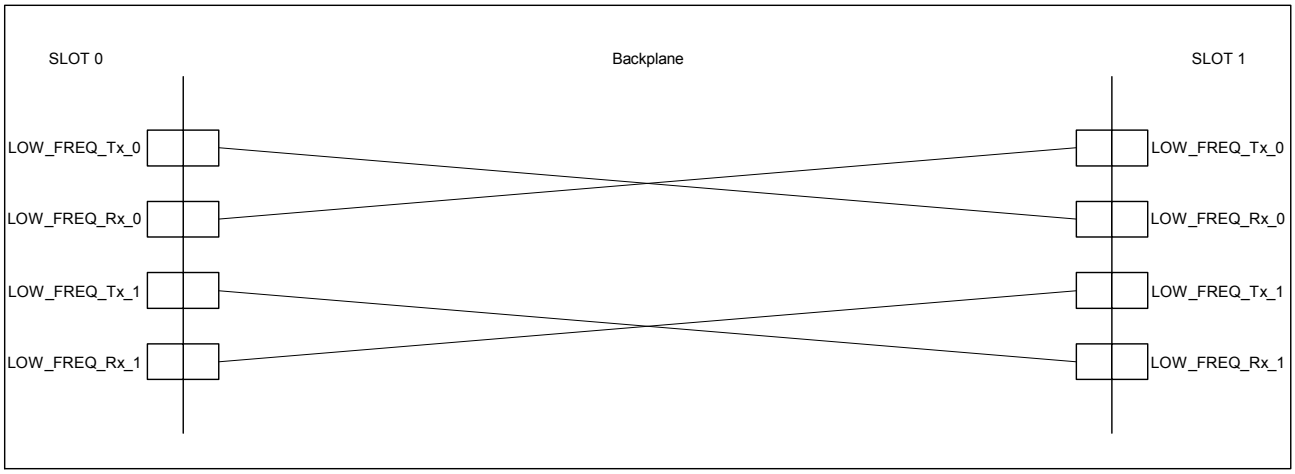


Figure 263 – Low Speed Channels

There are 4 end-to-end signals that may be used for intermodule I/O communications.

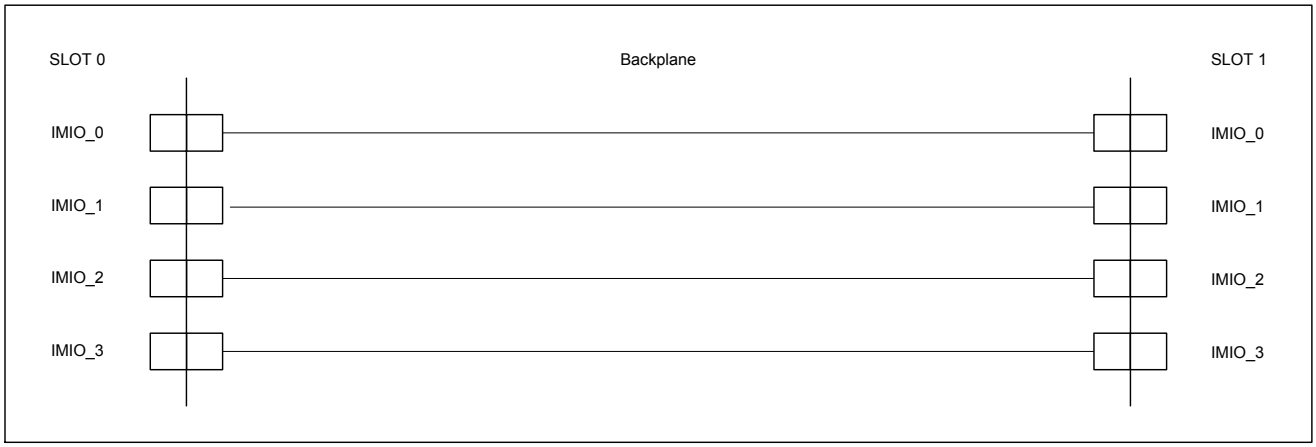


Figure 264 – Interconnect Channels

D.4 I/O Controller Module Connectors

The interface connector that shall be used on the backplane is based on a Berg-FCI High Speed Metral 5x6 30-pin male header connector part number 59566-1001 or equivalent. Equivalent parts from ITT Cannon: on the backplane 5-row 4000 male CBC20HS4000-030WXP5-5yy-x-VR; and on the I/O controller 5-row 4000 std-female CBC20HS4000-030FDP5-500-x-VR.

D.4.1 I/O Controller Module Connector

The interface connector that shall be used on the I/O controller is based on a Berg-FCI High Speed Metal 5x6 30-pin right angle female receptacle part number 52057-102. This connector is used for the 6 high-speed connectors.

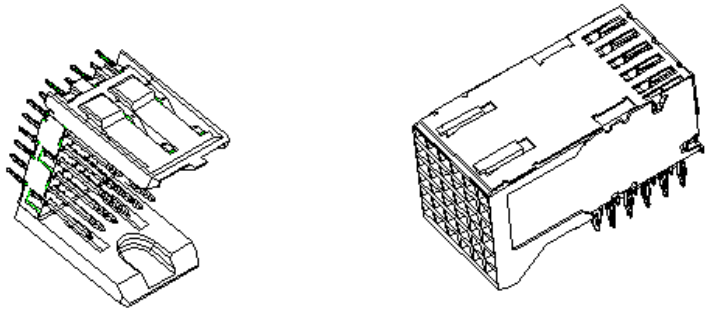


Figure 265 – I/O Controller Module Connector Rendering

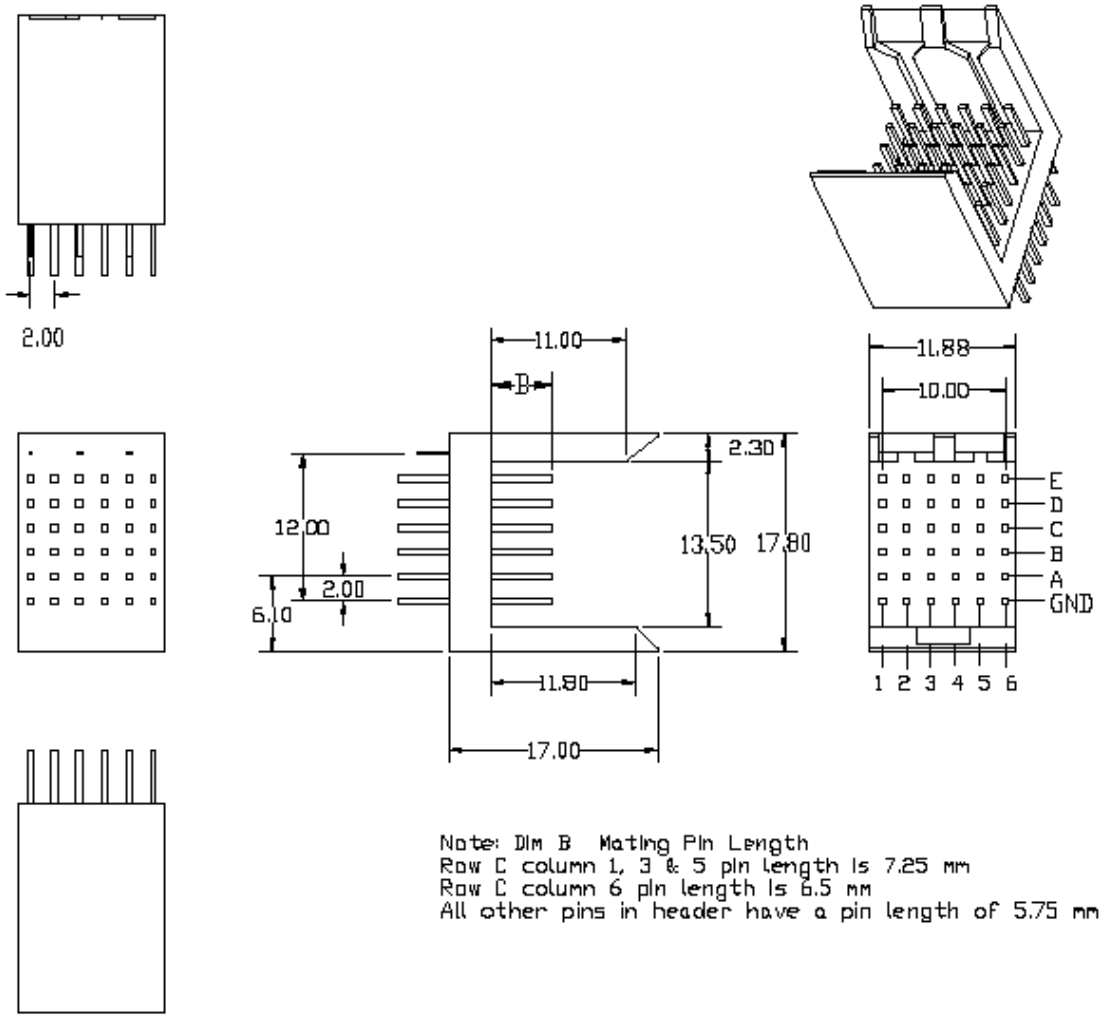


Figure 266 – Connector Pin Layout and Pin Lengths

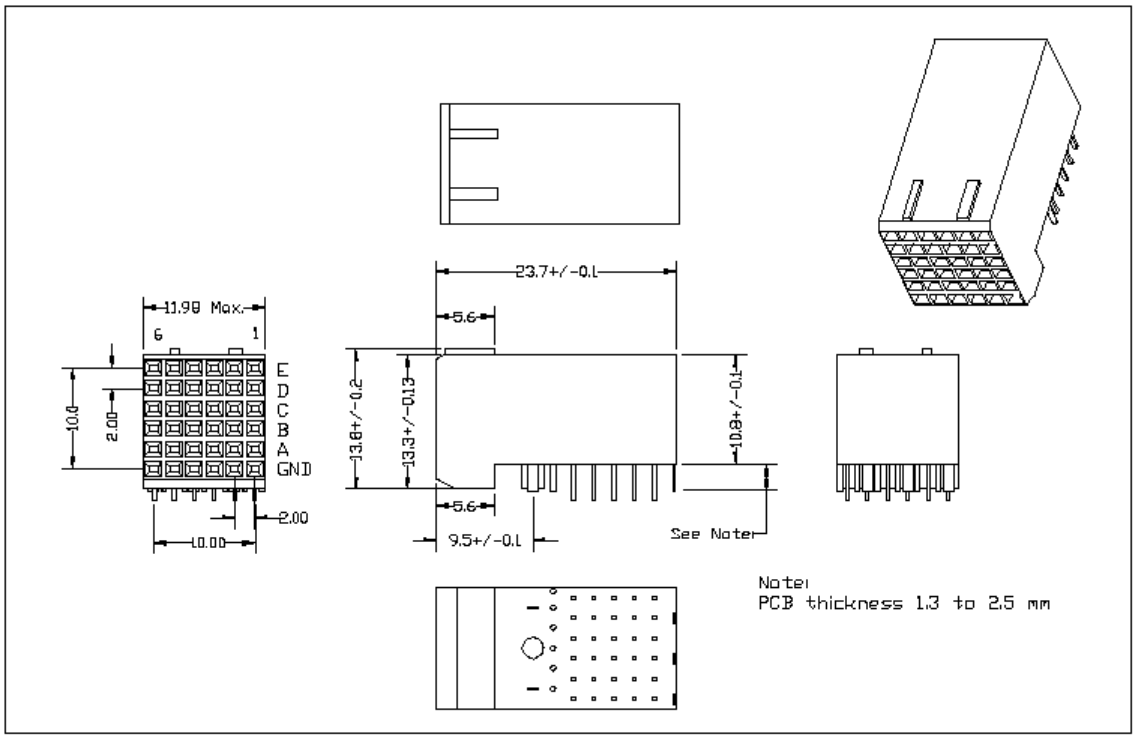


Figure 267 – I/O Controller Module Connector Receptacle Engineering Drawing

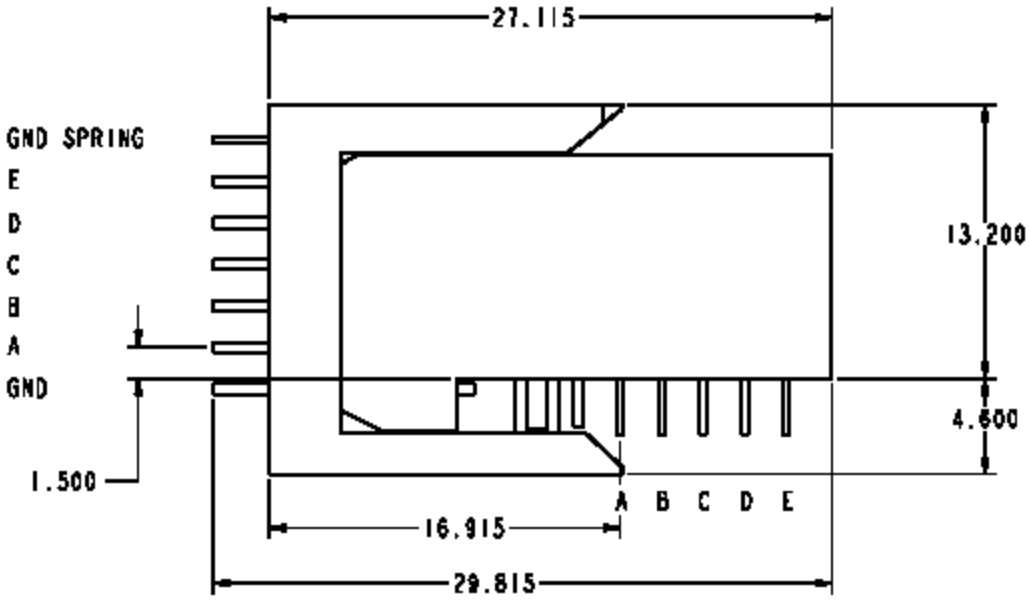


Figure 268 – Side View of Connector

D.5 I/O Controller Module Connector Locations

D.5.1 Purpose

This section defines the mating connector locations and connector alignment between I/O modules from one manufacturer and backplanes from a different manufacturer. This is an optional feature of the Serial ATA specification; however I/O controllers and Serial ATA backplanes in support of this industry normal Serial ATA I/O Controller-to-Backplane interface definition shall adhere to this specified connector placement, and connector pinout.
The width and length dimensions of the I/O Module shown in Figure 269 and Figure 270 are presented as examples and not mandatory requirements.

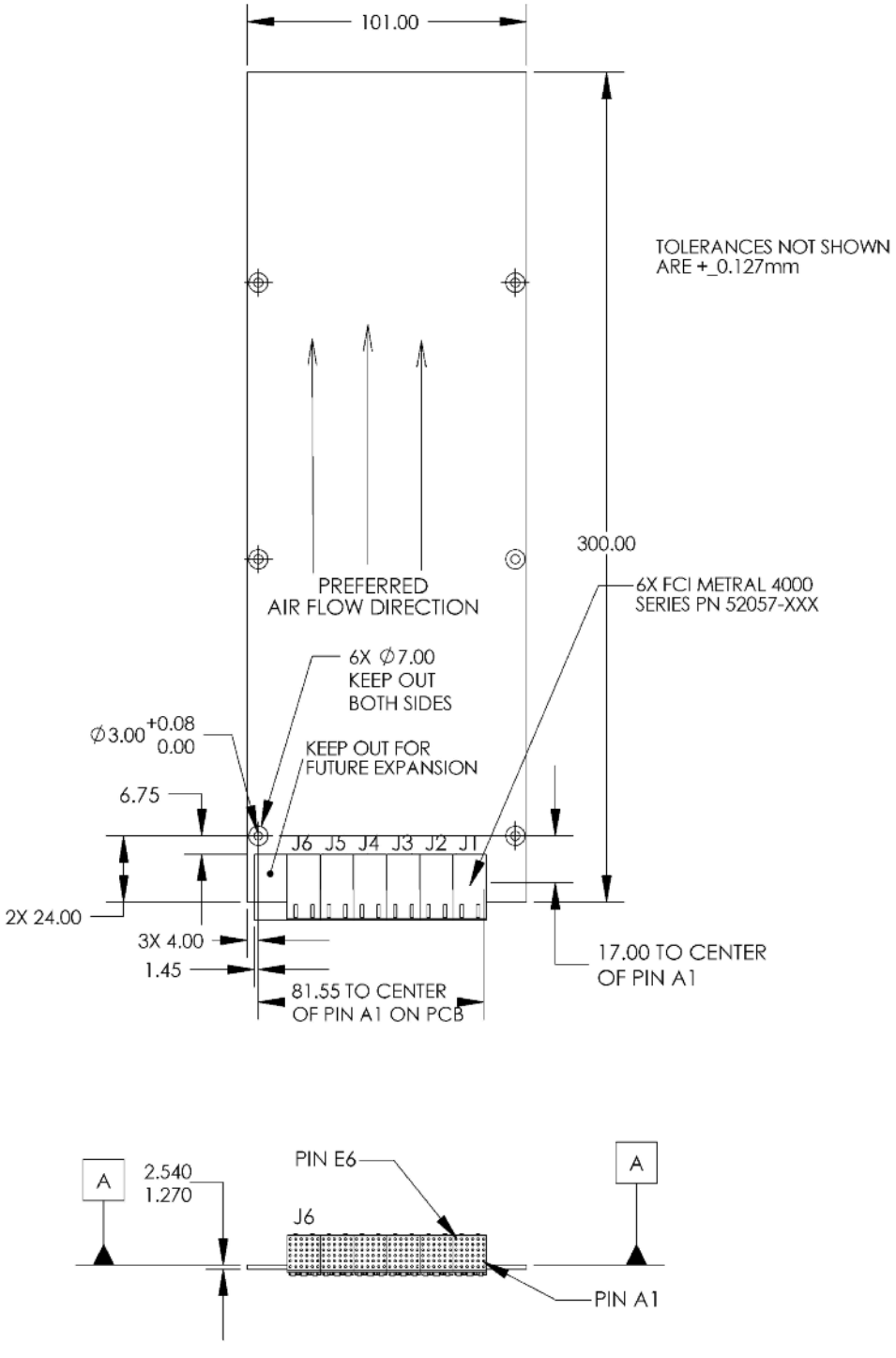


Figure 269 – I/O Controller Module Connector Locations on 1xWide I/O module

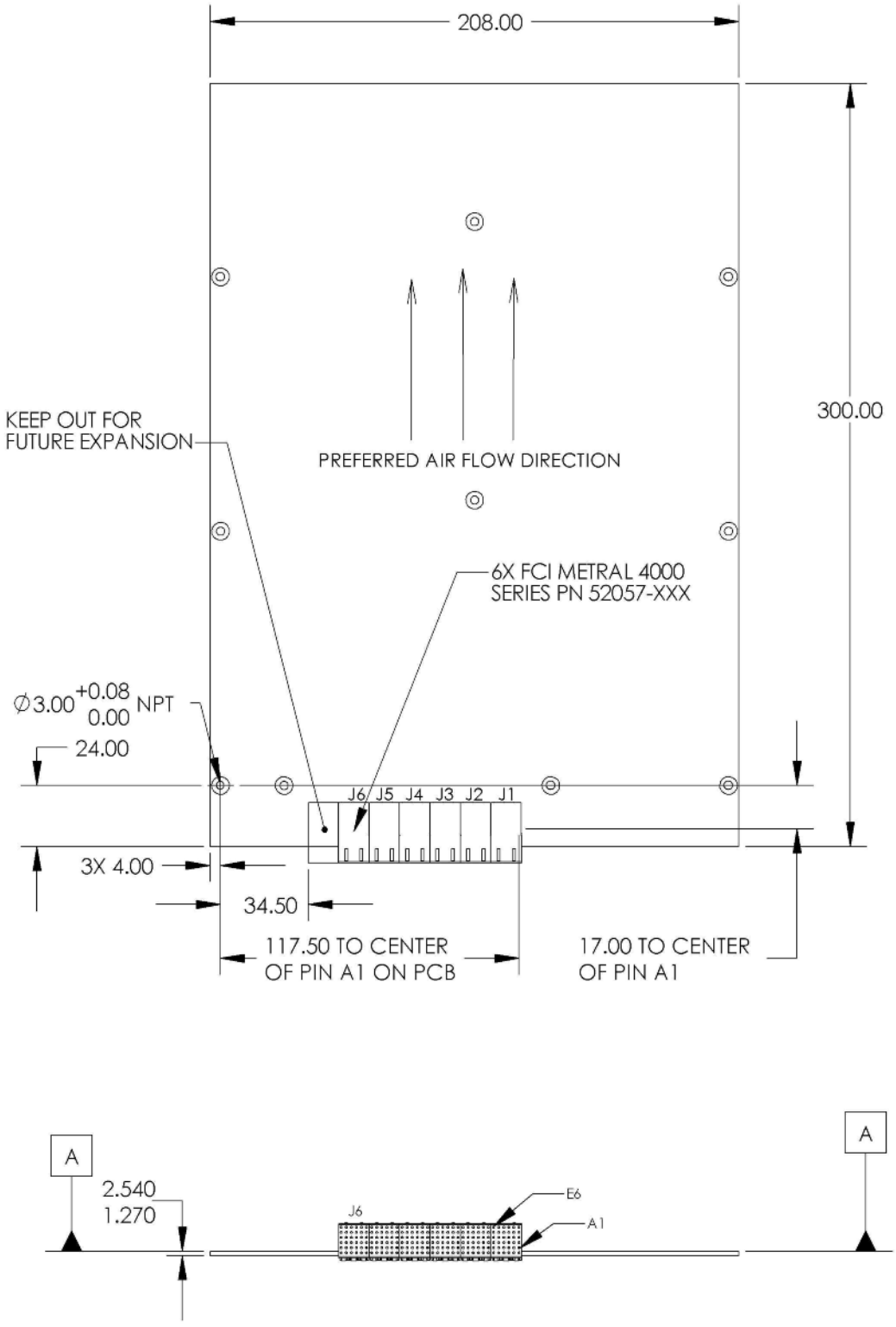


Figure 270 – I/O Controller Module Connector Locations on 2xWide I/O Module

D.6 Pinout Listing

These tables show the pin listing for the Metral 4000 connector system. Note that there are grounded shields between each connector row and signal, which are not shown in this matrix to aid clarity.

Table 106 – J1, J2, J3 Pin Assignments

J1	Row E	Row D	Row C	Row B	Row A	Ground
1	IO_SATA_RX0+	IO_SATA_RX0-	Ground	IO_SATA_TX0+	IO_SATA_TX0-	Shield
2	ACT_LED_DR0_L	MUX_DR0_L	5 V	5 V	IO_SLOT_0_L	Shield
3	IO_SATA_RX1+	IO_SATA_RX1-	Ground	IO_SATA_TX1+	IO_SATA_TX1-	Shield
4	ACT_LED_DR1_L	MUX_DR1_L	5 V	5 V	OTHER_IO_PRESENT_L	Shield
5	IO_SATA_RX2+	IO_SATA_RX2-	Ground	IO_SATA_TX2+	IO_SATA_TX2-	Shield
6	ACT_LED_DR2_L	MUX_DR2_L	5 V PRECHARGE	ENC_PROC_SDA	ENC_PROC_SCL	Shield
J2	Row E	Row D	Row C	Row B	Row A	Ground
1	IO_SATA_RX3+	IO_SATA_RX3-	Ground	IO_SATA_TX3+	IO_SATA_TX3-	Shield
2	ACT_LED_DR3_L	MUX_DR3_L	5 V	5 V	BOX_ID_BIT_0	Shield
3	IO_SATA_RX4+	IO_SATA_RX4-	Ground	IO_SATA_TX4+	IO_SATA_TX4-	Shield
4	ACT_LED_DR4_L	MUX_DR4_L	5 V	5 V	BOX_ID_BIT_1	Shield
5	IO_SATA_RX5+	IO_SATA_RX5-	Ground	IO_SATA_TX5+	IO_SATA_TX5-	Shield
6	ACT_LED_DR5_L	MUX_DR5_L	5 V PRECHARGE	FRU_ID_SDA	FRU_ID_SCL	Shield
J3	Row E	Row D	Row C	Row B	Row A	Ground
1	IO_SATA_RX6+	IO_SATA_RX6-	Ground	IO_SATA_TX6+	IO_SATA_TX6-	Shield
2	ACT_LED_DR6_L	MUX_DR6_L	5 V	5 V	BATT_CHARGE	Shield
3	IO_SATA_RX7+	IO_SATA_RX7-	Ground	IO_SATA_TX7+	IO_SATA_TX7-	Shield
4	ACT_LED_DR7_L	MUX_DR7_L	5 V	THIS_IO_PRESENT_L	BATT_DISCHARGE	Shield
5	IO_SATA_RX8+	IO_SATA_RX8-	Ground	IO_SATA_TX8+	IO_SATA_TX8-	Shield
6	ACT_LED_DR8_L	MUX_DR8_L	12 V PRECHARGE	12 V	12 V	Shield

Table 107 – J4, J5, J6 Pin Assignments

J4	Row E	Row D	Row C	Row B	Row A	Ground
1	IO_SATA_RX9+	IO_SATA_RX9-	Ground	IO_SATA_TX9+	IO_SATA_TX9-	Shield
2	ACT_LED_DR9_L	MUX_DR9_L	5 V	12 V	BATT_RETURN	Shield
3	IO_SATA_RX10+	IO_SATA_RX10-	Ground	IO_SATA_TX10+	IO_SATA_TX10-	Shield
4	ACT_LED_DR10_L	MUX_DR10_L	5 V	THIS_IO_OK_L	BATT_TEMP	Shield
5	IO_SATA_RX11+	IO_SATA_RX11-	Ground	IO_SATA_TX11+	IO_SATA_TX11-	Shield
6	ACT_LED_DR11_L	MUX_DR11_L	Reserved (3_3 V PRECHARGE)	Reserved (3_3 V)	BOX_ID_BIT_2	Shield
J5	Row E	Row D	Row C	Row B	Row A	Ground
1	IO_SATA_RX12+	IO_SATA_RX12-	Ground	IO_SATA_TX12+	IO_SATA_TX12-	Shield
2	ACT_LED_DR12_L	MUX_DR12_L	5 V	LOW_FREQ_RX_2	LOW_FREQ_TX_2	Shield
3	IO_SATA_RX13+	IO_SATA_RX13-	Ground	IO_SATA_TX13+	IO_SATA_TX13-	Shield
4	ACT_LED_DR13_L	MUX_DR13_L	5 V	ACT_LED_HOST0	ACT_LED_HOST1	Shield

5	IO_SATA_RX14+	IO_SATA_RX14-	Ground	IO_SATA_TX14+	IO_SATA_TX14-	Shield
6	ACT_LED_DR14_L	MUX_DR14_L	Reserved (3_3 V PRECHARGE)	Reserved	Reserved	Shield
J6	Row E	Row D	Row C	Row B	Row A	Ground
1	IO_SATA_RX15+	IO_SATA_RX15-	Ground	IO_SATA_TX15+	IO_SATA_TX15-	Shield
2	ACT_LED_DR15_L	MUX_DR15_L	5 V	5 V	OTHER_IO_RESET_L	Shield
3	HS_CHAN_TX(0)+	HS_CHAN_TX(0)-	Ground	HS_CHAN_RX(0)+	HS_CHAN_RX(0)-	Shield
4	LOW_FREQ_RX_1	LOW_FREQ_TX_1	5 V	IMIO_0	IMIO_1	Shield
5	HS_CHAN_TX(1)+	HS_CHAN_TX(1)-	Ground	HS_CHAN_RX(1)+	HS_CHAN_RX(1)-	Shield
6	OTHER_IO_OK_L	THIS_IO_RESET_L	5 V PRECHARGE	IMIO_2	IMIO_3	Shield

NOTES:

1 Connector as seen from side 1 of backplane. (I.e. Disk Side)

2 Long pins shaded black - Length 7.25 mm

3 Medium pins shaded grey - Length 6.50 mm

4 All other pins short - Length 5.75 mm

D.7 Signal Descriptions

ACT_LED_DR(15:0)_L

Signal is used to drive the activity LED associated with each disk drive. This is an active low signal, i.e. the LED is on when the signal is low. The LEDs are pulled up to +5 Vdc. The LED control circuitry shall be capable of sinking 15 mA at 0.4 V steady-state.

This signal should be driven by an open-drain output in order to prevent damage should two I/O modules inadvertently attempt to drive the signal at the same time. During RESET the I/O controller output should be set to a high impedance state.

The number of activity LED signals shall match the number of SATA ports supported by the controller. It is not mandatory to support 16 ports, any number between 1 and 16 SATA ports may be supported by this configuration.

MUX_DR(15:0)_L

Signal is used to control the Serial ATA path through a front-end multiplexer to a disk drive (if implemented). When this signal is low, the multiplexer selects the Serial ATA path to the I/O controller in the option slot 0 identified with IO_Slot_0 connected to GND. This is a standard 3.3 VTTL level signal. This signal should be driven by an open-drain output in order to prevent damage should two I/O modules inadvertently attempt to drive the signal at the same time.

Serial ATA is a point-to-point, controller to disk drive, single initiator interface. Current Serial ATA devices do not support dual Serial ATA ports, therefore, no native capability exists to implement I/O controller failover (Active-Active or Active-Passive). An alternative method to provide a dual I/O controller access to the same disk drive is to implement a front-end multiplexer between the I/O controller and disk drive. This multiplexer allows only one I/O controller to own the complete access to the disk drive and implement an Active-Passive failover.

This signal shall not be implemented in single I/O controller enclosures, as all Serial ATA signals shall be directly connected to the single slot. The I/O controller needs to ensure it is capable of operating in a single I/O controller configuration without the presence of these signals.

The MUX_DRx_L controls on the I/O controller shall be capable of being reset to an OPEN or high impedance state using the I/O controller RESET signal. This is to allow the failover I/O controller to have direct RESET capability over the MUX_DRx_L controls in the event of the I/O controller failure.

The number of activity MUX_DR signals shall match the number of SATA ports supported by the controller. It is not mandatory to support 16 ports, any number between 1 and 16 SATA ports may be supported by this specification.

IO_SATA_Tx(15:0)±

Differential SATA signal pair that originates at the I/O controller (Tx) and is received by the SATA disk drive (Rx). It is not mandatory to support 16 ports, any number between 1 and 16 SATA ports may be supported by this specification.

These signals shall be 100 Ohms differential characteristic impedance as per the nominal differential impedance documented in Table 29.

IO_SATA_Rx(15:0)±

Differential SATA signal pair that originates at the SATA disk drive (Tx) and is received by the I/O controller (Rx). It is not mandatory to support 16 ports, any number between 1 and 16 SATA ports may be supported by this specification.

These signals shall be 100 Ohms differential characteristic impedance. As per the nominal differential impedance documented in Table 29.

IO_Slot_0_L

Active low signal that identifies the I/O controller location, 0 or 1. This signal shall be pulled low (GND) on the backplane for IO Slot 0 and high (3.3 Vdc through a 10 kOhms resistor) for IO Slot 1.

In a single I/O controller enclosure, this signal shall be connected to GND. In a dual controller configuration, the I/O controller should sense this line to determine in which slot it is located.

Ground

Signal and power ground of the I/O module.

5 V PRECHARGE

+5 Vdc power that is available on the extended pins. This is used for pre-charging the I/O module.

The enclosure shall provide for a current limit of 4.5 A peak on each 5 V pre-charge pin (R=1.1 Ohms).

5 V

+5 Vdc power that is available on standard length pins.

The enclosure shall be capable to supplying 10 A of 5 V current per I/O module.

3V3 PRECHARGE

+3.3 Vdc power that is available on the extended pin. This is used for pre-charging the I/O controller 3.3 V circuitry. The enclosure shall provide for a current limit of 0.75 A on each 3.3 V pre-charge pin (R=4.4 Ohms).

3V3

+3.3 Vdc power that is available on standard length pins.

The enclosure shall be capable of supplying 0.75 A of 3.3 V current per I/O module.

12 V PRECHARGE

+12 Vdc power that is available on the extended pins. This is used for pre-charging the 12 V circuitry in the I/O Option slot module.

The enclosure shall be capable of supplying 2.4 A peak on each 12 V pre-charge pin (R=5 Ohms).

12 V

+12 Vdc power that is available on standard length pins.

The enclosure shall be capable of supplying 1.0 A of current per I/O module.

FRU_ID_SCL

Clock signal of the FRU Identification 2-wire, serial bus. The backplane has a 1.1 kOhms resistor pulled up to 3.3 V on this signal. The devices on this bus shall have open collector outputs. This 2-wire serial bus is used by the enclosure processor to gather information from all the FRUs located within the enclosure.

FRU_ID_SDA

Data signal of the FRU Identification 2-wire, serial bus. The backplane has a 1.1 kOhms resistor pulled up to 3.3 V on this signal. The devices on this bus shall have open collector outputs. This 2-wire serial bus is used by the enclosure processor to gather information from all the FRUs located within the enclosure.

ENC_PROC_SCL

Clock signal of the SES processor 2-wire, serial bus. The SES processor has a 1.1 kOhms resistor pulled up to 3.3 V on this signal. The devices on this bus shall have open collector outputs. This 2-wire serial bus is used by the I/O controller to talk to the enclosure processor.

ENC_PROC_SDA

Data signal of the SES processor 2-wire, serial bus. The SES processor has a 1.1 kOhms resistor pulled up to 3.3 V on this signal. The devices on this bus shall have open collector outputs. This 2-wire serial bus is used by the I/O controller to talk to the enclosure processor.

HS_CHAN_Tx(1:0)

Differential transmit pair that may connect the I/O controller to either the other I/O controller or an I/O connector receive pair. Details of the uses of these signals are provided in Figure 261 and Figure 262. It is recommended that these signals be 100 Ohms differential characteristic impedance.

HS_CHAN_Rx(1:0)

Differential receive pair that may connect the I/O controller to either the other I/O controller or an I/O connector transmit pair. Details of the uses of these signals are provided in Figure 261 and Figure 262. It is recommended that these signals be 100 Ohms differential characteristic impedance.

ACT_LED_HOST_L

Host link activity LED control signal from the I/O controller for front panel mounted LED. Each port shall have a separate LED for the intended use of indication which controller port is active. Exact functionality may be defined by the vendor.

BATT_CHARGE

Signal is used to charge a battery back-up unit (BBU). The backplane shall support a maximum of 2 A on this trace.

BATT_DISCHARGE

Signal is used to discharge a BBU. The backplane supports a maximum of 2 A on this trace.

BATT_TEMP

Signal is used to report an analog voltage level which corresponds to a temperature level within a BBU. The backplane supports a maximum of 100 mA on this trace.

BATT_RETURN

Signal is used as the return (ground) path for a BBU. The backplane supports a maximum of 2 A on this trace.

THIS_IO_OK_L

Active low signal is used to determine if the I/O Option slot module is performing within specification. This signal is driven by the I/O controller. The I/O Option Slot module shall provide a pull-up for this line when it is inactive. The backplane should support a maximum of 100 mA on this trace.

OTHER_IO_OK_L

Active low signal is used to determine if the other I/O Option slot module in a dual controller configuration is performing within specification. This signal is an input to the I/O controller. The backplane should support a maximum of 100 mA on this trace.

OTHER_IO_RESET_L

Active low output signal which is used to reset the other I/O Option slot module in a dual controller system, located in the opposite slot. The backplane shall provide a pull-up for this line

when it is inactive. This signal is cross-wired with the THIS_IO_RESET_L signal on dual controller backplanes. The backplane should support a maximum of 100 mA on this trace.

THIS_IO_RESET_L

Active low input signal which is used to reset the I/O Option slot module located in this slot. This signal is cross-wired with the OTHER_IO_RESET_L signal on dual controller backplanes. The backplane should support a maximum of 100 mA on this trace.

LOW_FREQ_RX_(1:0)

Signal line used for intermodule, 2-wire communications. This signal is cross-wired with the LOW_FREQ_TX signal on dual controller backplanes as shown in Figure 263. This is a low-speed signal line. Maximum frequency is 500 kHz. The actual implementation of this signal is to be decided by the I/O controller vendor. The backplane should support a maximum of 100 mA on this trace.

LOW_FREQ_TX_(1:0)

Signal line used for intermodule, 2-wire communications. This signal is cross-wired with the LOW_FREQ_RX signal on dual controller backplanes as shown in Figure 263. This is a low-speed signal line. Maximum frequency is 500 kHz. The actual implementation of this signal is to be decided by the I/O controller vendor. The backplane should support a maximum of 100 mA on this trace.

OTHER_IO_PRESENT_L

Active low signal is used to denote the presence of an option slot module in the opposite I/O option slot module. The I/O controller shall provide a pull-up for this line when it is inactive. This signal is cross-wired with the THIS_IO_PRESENT_L signal on the dual backplane only. This is a low-speed signal line. The actual implementation of this signal is to be decided by the I/O controller vendor. The backplane should support a maximum of 100 mA on this trace.

THIS_IO_PRESENT_L

Active low signal is used to denote the presence of an option slot module in the local I/O option slot module. The I/O controller shall provide a pull-up for this line when it is inactive. This signal is cross-wired with the OTHER_IO_PRESENT_L signal on the dual backplane only. This is a low-speed signal line. The actual implementation of this signal is to be decided by the I/O controller vendor. The backplane should support a maximum of 100 mA on this trace.

IMIO_(3:0)

These are direct connecting signals used for intermodule communication. The backplane should support a maximum of 100mA on these traces. See Figure 264 for details.

BOX_ID_BIT_(2:0)

Signal is used to determine the ID of the enclosure in which it is installed.

Reserved (2 pins)

Two undefined pins J5 Row A pin 6, and J5 Row B pin 6, which are not available for use at this time. No signals should be connected to these signals on either the backplane or the I/O Option Slot Module.

Appendix E. Jitter Formulas without SSC (Informative)

E.1 Clock to Data

Consider the times when the clock edges occur. A perfect clock has edges that occur at multiples of a given period T. Associate an integer index with each clock edge. The times of ideal clock edges is expressed by

$$t_C(i) = i\,T$$

Data transitions always occur on a clock edge. Ideal data transitions occur at the same times as clock edges. In real systems, the data transitions do not occur at ideal times. The time error from ideal of the data transitions is called the "clock to edge jitter". This is expressed by

$$t_D(i) = i\,T + \varepsilon_i$$

where the ε_i are time deviations from ideal, the clock to edge jitter. Over time, these perturbations are constrained by

$$\lim_{N\to\infty} \sum_{i=0}^{N-1} \frac{\varepsilon_i}{N} = 0$$

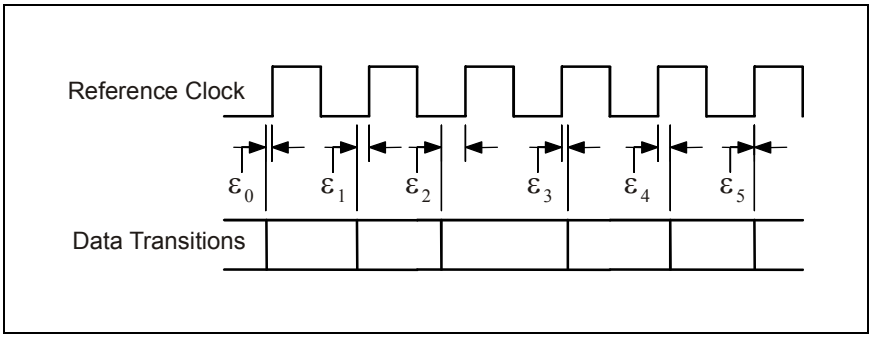


Figure 271 – Jitter Deviations

E.2 Data to Data (shown for historical reasons)

The time difference between data transitions is shown in **Error! Reference source not found.** and given by

$$t_D(j) - t_D(i) = (j - i)\,T + \varepsilon_j - \varepsilon_i$$

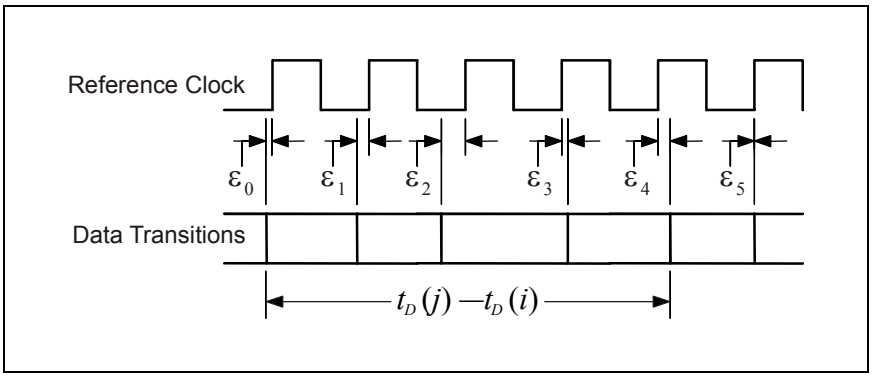


Figure 272 – Edge to Edge Timing

A specification shall be invariant to when the measurement is taken so introduce a new quantity $k = (j - i)$ for the spacing between bits. The time difference between data transitions is

$$t_D(i+k) - t_D(i) = k\,T + \varepsilon_j - \varepsilon_i$$

Define a new quantity for the limits of this time difference – this is the jitter definition for Gen1i and Gen1m. The maximum and minimum is taken over all bit positions i, which makes the jitter a function of only the bit spacing k.

$$t_J(k) = \max\left[t_D(i+k) - t_D(i)\right]_{\forall i} - \min\left[t_D(i+k) - t_D(i)\right]_{\forall i}$$

$$t_J(k) = \max\left[k\,T + \varepsilon_{i+k} - \varepsilon_i\right]_{\forall i} - \min\left[k\,T + \varepsilon_{i+k} - \varepsilon_i\right]_{\forall i}$$

Note that kT is a constant for each k, and is present in both the maximum and minimum terms. Since the difference is taken, the terms cancel giving

$$t_J(k) = \max\left[\varepsilon_{i+k} - \varepsilon_i\right]_{\forall i} - \min\left[\varepsilon_{i+k} - \varepsilon_i\right]_{\forall i}$$

A distinct advantage of this jitter definition is the absence of dependence on the bit time T. This is not true when the clock is frequency modulated as in spread spectrum clocking (SSC).

For a given separation of data transitions expressed in clock cycles, the maximum peak to peak deviation of the data transition spacing is the jitter. It is a function of the transition separation only, not the position of any particular transition. This jitter definition expresses the extreme separations of data transition times.