Serial ATA II:

Extensions to Serial ATA 1.0a

Revision 1.2
27-August-2004

Dell Computer Corporation

Intel Corporation

Maxtor Corporation

Seagate Technology

Vitesse Semiconductor Corporation

This 1.2 revision of the Serial ATA II: Extensions to Serial ATA 1.0a specification ("Final Specification") is available for download at www.serialata.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 Working Group website at www.serialata.org.

Serial ATA II Workgroup Extensions to Serial ATA 1.0a Technical Editor:

Amber Huffman
Intel Corporation
2111 NE 25th Ave   M/S JF2-53
Hillsboro, OR 97124 USA
Tel: (503) 264-7929
Email: amber.huffman@intel.com

Table of Contents

1. Introduction

1.1. Goals, Objectives, & Constraints

This specification is one of a series of specifications that comprise Serial ATA II. This specification defines enhancements to the Serial ATA 1.0a specification that provide additional capabilities while retaining Serial ATA 1.0a compatibility and desktop cost structure

Additional features and capabilities are defined in a way that allow them to be selectively deployed, as business and market conditions require. Capabilities are defined in a way that costs associated with new feature support are incurred if and when the feature is implemented in order to realize customer benefit/value.

Some of the goals and requirements for the specification include:

- Compatibility must be maintained with Serial ATA 1.0a
- New features/capabilities must be separable/optional proper supersets of Serial ATA 1.0a
- Support for command queuing must be provided
- Means for supporting industry-standard enclosure services must be provided
- Device management expected of storage subsystems needs to be supported

1.2. References

This specification is an extension to the Serial ATA 1.0a specification. The Serial ATA 1.0a specification is presumed as the underlying baseline for this specification. This specification makes reference to the following specifications:

Serial ATA: High Speed Serialized AT Attachment revision 1.0a. Available for download at www.serialata.org.

Serial ATA II: Port Multiplier revision 1.0. Available for download at www.serialata.org.

AT Attachment with Packet Interface – 6 (ATA/ATAPI-6). Draft available at www.T13.org. Published ATA/ATAPI specifications available from ANSI at webstore.ansi.org or from Global Engineering.

SAF-TE – SCSI Accessed Fault-Tolerant Enclosure version 1.00 [revision R041497, April 14, 1997].

ANSI INCITS 305-1998, Information Technology – SCSI-3 Enclosure Services (SES) Command Set. Available from ANSI at webstore.ansi.org or from Global Engineering. Additional material and draft versions available from www.T10.org.

ANSI INCITS 230-1994 (R1999), Information Technology – Fibre Channel – Physical and Signaling Interface (FC-PH). Available from ANSI at webstore.ansi.org or from Global Engineering.

ANSI INCITS 301-1997, Information Technology – SCSI-3 Primary Commands (SPC). Available from ANSI at webstore.ansi.org or from Global Engineering.

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

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

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

1.3. Definitions, abbreviations, and conventions

1.3.1. Definitions and Abbreviations

The terminology used in this specification is consistent with the terminology used in the Serial ATA 1.0a specification, and all definitions and abbreviations defined in that specification are used consistently in this document. Additional terms and abbreviations introduced in this specification are defined in the following sections.

1.3.1.1. 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.

1.3.1.2. FPDMA Data Phase

The FPDMA 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.

1.3.1.3. HBA (Host Bus Adapter)

A Host Bus Adapter is a component that connects to the host system’s expansion bus to provide connectivity for devices. Host Bus Adapters are also often referred to as controller cards or merely controllers.

1.3.1.4. PRD (Physical Region Descriptor)

A Physical Region Descriptor table is a data structure used by DMA engines that comply with the SFF 8038i specification use to describe memory regions for transferring data to/from. A PRD table is also often referred to as a scatter/gather list.

1.3.1.5. 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.

1.3.1.6. 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.

1.3.2. Conventions

1.3.2.1. Register Naming Conventions

This specification uses the same naming conventions for the Command Block Registers as the Serial ATA 1.0a specification. However, the register naming convention is different from that used in the ATA/ATAPI-6 standard. Figure 1 defines the correspondence of the register names used in this specification with those used in the ATA/ATAPI-6 standard.

Serial ATA register name	***ATA/ATAPI-6 register name when writing registers***	***ATA/ATAPI-6 register name when reading registers***
Features	Features Current	
Features (exp)	Features Previous	
Sector Count	Sector Count Current	Sector Count HOB=0
Sector Count (exp)	Sector Count Previous	Sector Count HOB=1
Sector Number	LBA Low Current	LBA Low HOB=0
Sector Number (exp)	LBA Low Previous	LBA Low HOB=1
Cylinder Low	LBA Mid Current	LBA Mid HOB=0
Cylinder Low (exp)	LBA Mid Previous	LBA Mid HOB=1
Cylinder High	LBA High Current	LBA High HOB=0
Cylinder High (exp)	LBA High Previous	LBA High HOB=1
Device/Head	Device	Device
Command	Command	na
Control	Control	na
Status	na	Status
Error	na	Error

Figure 1 Register naming conventions and correspondence

2. Physical Layer

2.1. Backplane Interconnect Reference

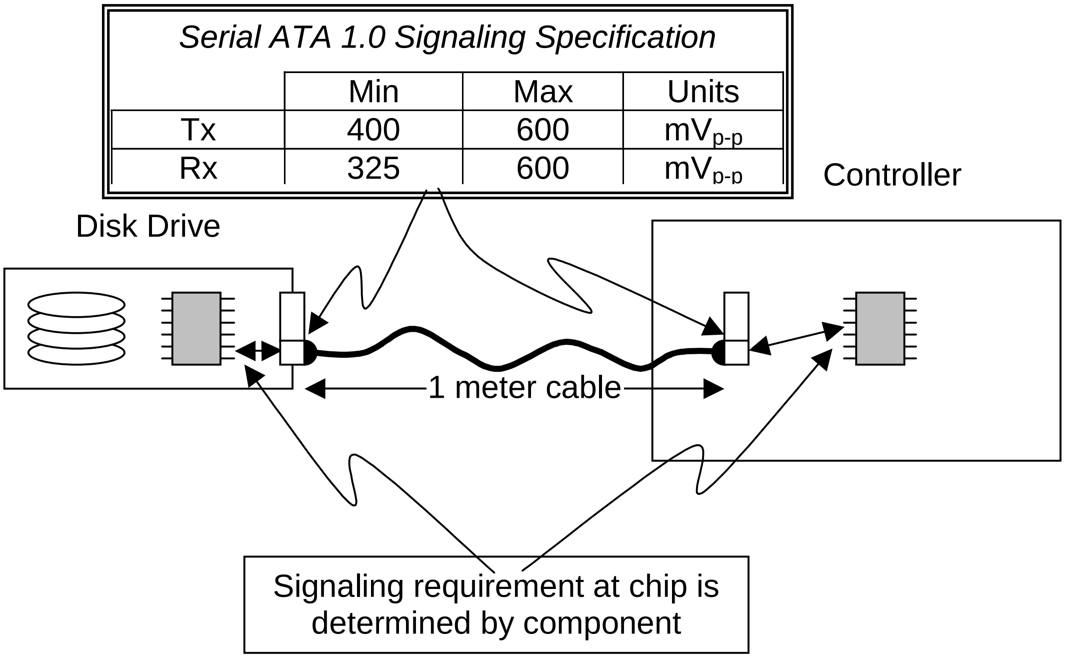
2.1.1. Goals, Objectives, & Constraints

The Serial ATA 1.0a specification defines a cabled interconnect configuration and the corresponding Phy signaling parameters for that specific configuration. The parameters defined in the Serial ATA 1.0a specification do not directly apply to a backplane interconnect that has transmission characteristics substantially different from the cabled interconnect defined in the Serial ATA 1.0a specification. This section defines the host Phy parameters to accommodate a backplane interconnect of up to 18 inches in length. The solution is constrained to isolate all compensation for the losses and reflection characteristics of the backplane interconnect to the host side Phy, while leaving the device-side Phy wholly unchanged from that defined in the 1.0a specification.

2.1.2. Introduction

The Serial ATA 1.0a specification defines required signaling levels at the Serial ATA connector (see Figure 2). These parameters do not directly apply to the pins of the controller electronics and it is the responsibility of the interface component supplier to account for losses between the controller IC and the connector interface as part of the design collateral that accompanies the component.

This section provides additional host controller Phy parameter recommendations in order to promote timely and cost-effective solutions addressing the specific backplane requirements of storage subsystems. The overriding premise is that the device-side signal specification (at its connector) is an immutable given. All burdens for these new applications are placed upon the host-side controller.



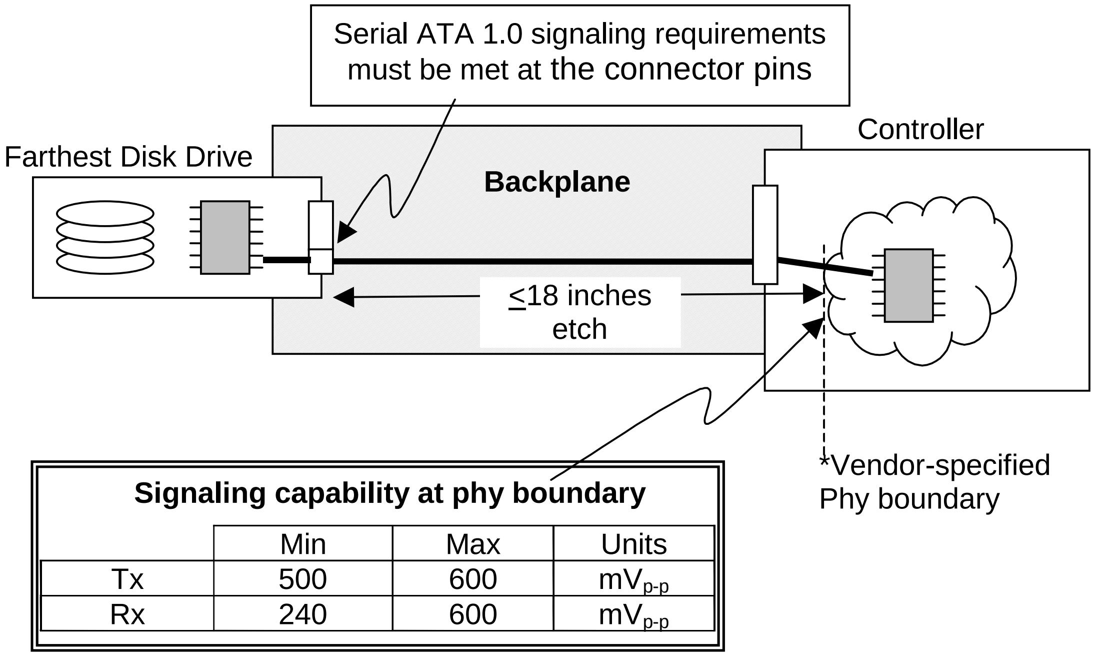
Serial ATA 1.0 Signaling Specification

	Min	Max	Units
Tx	400	600	mV_{p-p}
Rx	325	600	mV_{p-p}

Figure 2 Interconnect configuration defined in Serial ATA 1.0a

2.1.3. Storage Arrays

The application of Serial ATA to storage subsystems may include integration into backplane-based designs, typically 19-inch standard racks. A storage system may consist of an array of devices mounted side by side along the front panel of a 19-inch rack in order to allow any one of the devices to be removed. A backplane routes signals to each device from a central controller. In the worst case this controller could be positioned at one end of the device array, requiring routing from the controller silicon device to a connector on the backplane and then across the backplane. This type of application therefore defines a physical interconnect between the controller chip and the Serial ATA 1.0a device connector that consists of up to 18 inches of etch and one connector.


Serial ATA 1.0 signaling requirements must be met at the connector pins
Controller
Farthest Disk Drive
Backplane
≤18 inches etch
*Vendor-specified Phy boundary

Signaling capability at phy boundary

	Min	Max	Units
Tx	500	600	mV_{p-p}
Rx	240	600	mV_{p-p}

Figure 3 Definition reference configuration for multi-disk enclosure with internal controller using up to 18" of FR4 interconnect

*The vendor-specified Phy boundary encompasses design specific support elements including but not limited to coupling capacitors, compensating resistors, and tuned PCB trace geometries as part of the supplied controller component.

2.1.4. Conclusion & Recommendation

In order for a host controller to be able to reliably transmit to and receive from a Serial ATA 1.0a device within a backplane environment consisting of up to 18 inches of FR4 0.012 inch trace, greater minimum transmit levels and smaller minimum acceptable receive sensitivity levels are required. However, when that same host controller is applied in less lossy environments, the controller may run the risk of exceeding the maximum specified signal levels. This implies that it may be necessary for such a host controller to have a means of adjusting its signaling level as appropriate for the particular application. For example, the host controller may use an external set resistor that determines the transmit current level.

	Nom	Min	Max	Units	Comments
Vdiff,tx	-	~~400~~ 500	600*	mVp-p	At host controller Phy boundary
Vdiff,rx	-	~~325~~ 240	600	mVp-p	At host controller Phy boundary

* Maximum transmit voltage may be increased above specified maximum if means for ensuring that the specified maximum receiver voltage is not exceeded at the far end of the interconnect is provided. All other parameters are as specified in the Serial ATA 1.0a specification.

Figure 4 Host-controller 1.5 Gbps Phy specification recommendations for backplane application

2.2. Asynchronous Signal Recovery (Optional)

The Serial ATA 1.0a specification does not explicitly call out the Phy behavior for asynchronous signal recovery since the intended usage model for hot-plugging was insertion of a device into a receptacle where power would be applied as part of the insertion process.

Phys may optionally support Serial ATA II asynchronous signal recovery for those applications where the Serial ATA 1.0a 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. If the device attempts to recover the signal before the host by issuing a COMINIT, it is unclear what state the drive is in and whether the drive will return its signature. 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. As a consequence of the COMRESET, the device shall return its signature and will be in an unambiguous state.

When a COMRESET is sent to the device in response to an unsolicited COMINIT, the host shall set the Status register to 0x7F and shall set all other taskfile registers to 0xFF. When the COMINIT is received in response to the COMRESET, the Status register value shall be updated to either 0xFF or 0x80 to reflect that a device is attached.

2.2.1. Host Phy Initialization State Machine

Hosts that support asynchronous signal recovery have a modified Phy initialization state machine as defined in the following state diagram. Material depicted in bold italic typeface represents the changes from the behavior defined in the Serial ATA 1.0a specification.

As described in section 6.7.4.3 of the Serial ATA 1.0a specification, 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 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 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 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, these implied transitions have 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 must allow the device to retransmit COMWAKE and shall not initiate asynchronous signal recovery operations unless a COMRESET is explicitly triggered from a higher layer.

Designs that support asynchronous signal recovery have a state variable referred to as RetryInterval that determines the rate at which optional signal recovery polling is attempted. The value for RetryInterval is design specific but shall be no shorter than 10 ms. Implementations that do not implement optional retry polling may consider the RetryInterval value to be infinite.

HP1: HR_Reset[1]	Transmit COMRESET[2, 3, ***4***] ***If asynchronous signal recovery is supported then clear ResumePending to 0.***		
	1. Power-on reset and explicit reset request deasserted	→	HR_AwaitCOMINIT
	2. Power-on reset or explicit reset request asserted	→	HR_Reset
	NOTE : 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. Must transmit COMRESET for a minimum of 6 bursts (and a multiple of 6) 3. As described in section 6.7.4.2 of the 1,0a specification, 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
	NOTE : 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
	NOTE : 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 or ResumePending = 1)***	→	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***

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 detected from device (at any supported speed)[3]	→	HR_AdjustSpeed
	2. ALIGN not detected from device and 880us (32768 Gen1 dwords) has elapsed since entry to HR_AwaitAlign	→	HR_Reset[1,4]
	3. ALIGN not detected from device and less than 880us (32768 Gen1 dwords) has elapsed since entry to HR_AwaitAlign	→	HR_AwaitAlign
	NOTE : 1. Host retries the power-on sequence indefinitely unless explicitly turned off by the application layer 2. Host must start transmitting d10.2 characters no later than 533ns (20 Gen1 dwords) after COMWAKE is deasserted as specified in the out of band signaling section 3. Host designers should be aware that the device is allowed 53.3ns (2 Gen1 dwords) after releasing COMWAKE (by holding the idle condition for more than 175ns) to start sending characters. Until this occurs, the bus will be at an idle condition and may be susceptible to crosstalk from other devices. Care must be taken so that crosstalk during this window doesn't result in a false detection of an ALIGN. For example: a compliant host may detect the deassertion of COMWAKE in as little as 112ns, such a host should wait at least 116.3ns (175+53.3-112) after detecting the release of COMWAKE to start looking for ALIGNs. 4. The Host PHY state machine may use the transition to HR_Reset as a method of speed negotiation.		

HP7: HR_SendAlign	Transmit ALIGN at speed detected		
	1. Three back-to-back non-ALIGN primitives[2] detected from device	→	HR_Ready
	2. Three back-to-back non-ALIGN primitives not detected from device	→	HR_SendAlign[1]
	NOTE : 1. Host retries indefinitely unless explicitly turned off by the application layer 2. Non-ALIGN primitives can be detected by the presence of the k28.3 control character in the byte0 position		

HP8: HR_Ready	Transmit word from Link[1]. ***If asynchronous signal recovery is supported then clear ResumePending to 0.***	
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***
NOTE : 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 transmit interval.***		

HP9: HR_Partial	Interface quiescent. ***If asynchronous signal recovery is supported then set ResumePending to 1.***	
1. Partial signal from Link deasserted and no COMWAKE detected from device[1]	→	HR_COMWAKE
2. Partial signal from Link deasserted and COMWAKE detected from device[1]	→	HR_AwaitNoCOMWAKE
3. Partial signal from Link asserted	→	HR_Partial
NOTE : 1. Host Phy must remember if COMWAKE was detected during Partial to determine if the wakeup request originated from the host or the Phy.		

HP10: HR_Slumber	Interface quiescent. ***If asynchronous signal recovery is supported then set ResumePending to 1.***	
1. Slumber signal from Link deasserted and no COMWAKE detected from device[1,2]	→	HR_COMWAKE
2. Slumber signal from Link deasserted and COMWAKE detected from device[1,2]	→	HR_AwaitNoCOMWAKE
3. Slumber signal from Link asserted	→	HR_Slumber
NOTE : 1. Host Phy must 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
NOTE : 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 53ns (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.		

2.2.2. Device Phy Initialization State Machine

Devices that support asynchronous signal recovery have a modified Phy initialization state machine as defined in the following state diagram. Material depicted in bold italic typeface represents the changes from the behavior defined in the Serial ATA 1.0a specification.

As described in section 6.7.4.2 of the 1.0a specification, reception of a COMRESET signal shall be treated by the device as a hard reset signal and shall unconditionally force the Device Phy state machine to transition to the DP1:DR_Reset initial state regardless of other conditions. Reception of 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.

DP1: DR_Reset[1]	Interface quiescent		
	1. COMRESET not detected and power-on reset deasserted	→	DR_COMINIT
	2. 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		

DP2: DR_COMINIT	Transmit COMINIT[1, ***2***]		
	1. Unconditional	→	DR_AwaitCOMWAKE
	NOTE : 1. COMINIT transmitted for a 6 bursts duration ***2. As indicated in section 5.2 of the 1.0a specification, devices shall respond with a COMINIT signal within 10 ms of the deassertion 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
	NOTE : 1. Device must 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
	NOTE : 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[1,2,3,5]		
	1. ALIGN detected from host (device locked to incoming data)[4]	→	DR_Ready
	2. ALIGN not detected from host and ALIGN primitives transmitted for 54.6us (2048[5] Gen1 ALIGN primitives) at speed other than lowest[6]	→	DR_ReduceSpeed
	3. ALIGN not detected from host and ALIGN primitives transmitted for 54.6us (2048[5] Gen1 ALIGN primitives) at lowest speed[6]	→	DR_Error
	4. ALIGN not detected from host and ALIGN primitives transmitted for less than 54.6us (2048 Gen1 ALIGN primitives)	→	DR_SendAlign
	NOTE : 1. ALIGN should be sent at the devices fastest supported speed first 2. ALIGNS should be sent only at valid frequencies (if PLL not locked, send D10.2) 3. After COMWAKE is released as specified in the out of band signaling section, the device must ensure the interface is active (not quiescent). 4. Device designers should be aware that the host is allowed 533ns (20 Gen1 dwords) after detecting the deassertion of COMWAKE to start sending d10.2 characters. Until this occurs, the bus will be at an idle condition and may be susceptible to crosstalk from other devices. Care must be taken so that crosstalk during this window doesn't result in a false detection of an ALIGN. Devices may extend this timeout up to an additional 54.6us (2048 Gen1 dwords) (for a max total of 109.2us), as necessary to allow their receiver time to lock to the host ALIGN. 5. Device must not leave the bus idle more than 53.3ns (2 Gen1 dwords) longer than the required 175ns to deassert COMWAKE. 6. If this is part of a device-initiated recovery from the Slumber or Partial power management state, the device Phy should resume at the speed previously negotiated and should 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 re-try 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***
	NOTE : 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 deasserted	→	DR_COMWAKE
2. Partial signal from Link deasserted and COMWAKE detected from host	→	DR_AwaitNoCOMWAKE
3. Partial signal from Link asserted	→	DR_Partial

DP9: DR_Slumber	Interface quiescent	
1. Slumber signal from Link deasserted	→	DR_COMWAKE
2. Slumber signal from Link deasserted and COMWAKE detected from host	→	DR_AwaitNoCOMWAKE
3. Slumber signal from Link asserted	→	DR_Slumber

DP10: DR_ReduceSpeed	Interface quiescent	
1. Transition to legacy (slower) speed complete	→	DR_SendAlign[1]
2. Transition to legacy speed not complete	→	DR_ReduceSpeed
NOTE : 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***

2.3. Device Power Connector Pin 11 Definition (optional)

Pin 11 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 11 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 11 is primarily for use in backplane applications. Reference section 6.3 for information on activity LED generation for desktop applications.

A device may optionally support activity indication via pin 11, staggered spin-up control via pin 11, or both features. If neither feature is supported, then pin 11 is a no-connect at the device as specified in Serial ATA 1.0a.

A host may only support one pin 11 feature, either receiving activity indication or staggered spin-up disable control. If a host supports receiving activity indication via pin 11, then the host shall not use pin 11 to disable staggered spin-up. If a host does not support receiving activity indication via pin 11, then the host may use pin 11 to disable staggered spin-up.

2.3.1. Device Activity Signal (optional)

2.3.1.1. 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 must 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 1 and Table 2 define the electrical parameters and requirements for the activity signal for both the device and the host. Figure 5 is an example of an activity signal implementation for illustrative purposes. Note that the host cannot rely on a particular resistor pull-up value on the device side, nor can the device rely on particular host resistor values. No direct support for wire-ORing 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.

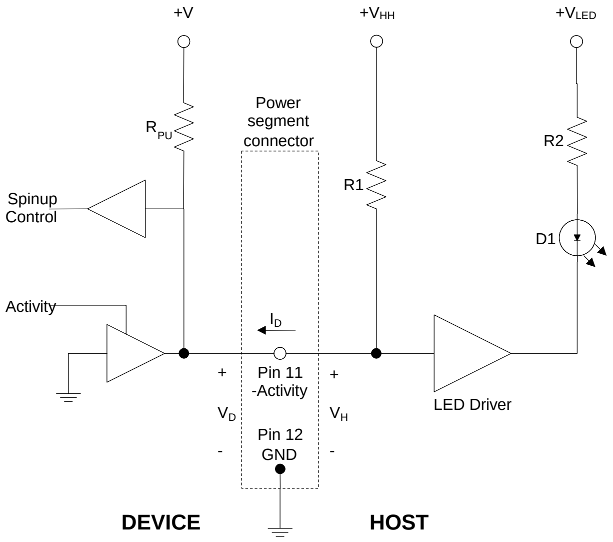


Figure 5 Example activity signal electrical block diagram

All voltage references in Table 1 and Table 2 are to ground pin 12 on the device connector. All voltages and currents in Table 1 and Table 2 are measured at pin 11 on the device connector.

Table 1 Device connector pin 11 activity signal electrical parameters

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

Table 2 Host activity signal electrical parameters

Parameter	Min Value	Max Value	Description & Conditions
V_{HIn}	-0.5V	3.3V	Tolerated input voltage
V_{HH}		2.1V	Host voltage presented to device when device not driving signal low. (Also see Section 2.3.2 for staggered spin-up control).
V_{HL}	-0.1V		Minimum allowable host voltage that may be presented to the device.
I_{HAct}		300uA	Host current delivered to device when device driving signal low. Value specified at V_{DAct} voltage of 0V.

2.3.1.2. LED Driver Circuit (informative)

The LED driver circuit provided by the host to drive an activity LED is vendor specific. Figure 6 illustrates two conceptual driver circuits that would satisfy the electrical requirements and provide a signal suitable for driving an activity LED. Variations in the driver circuits can 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.

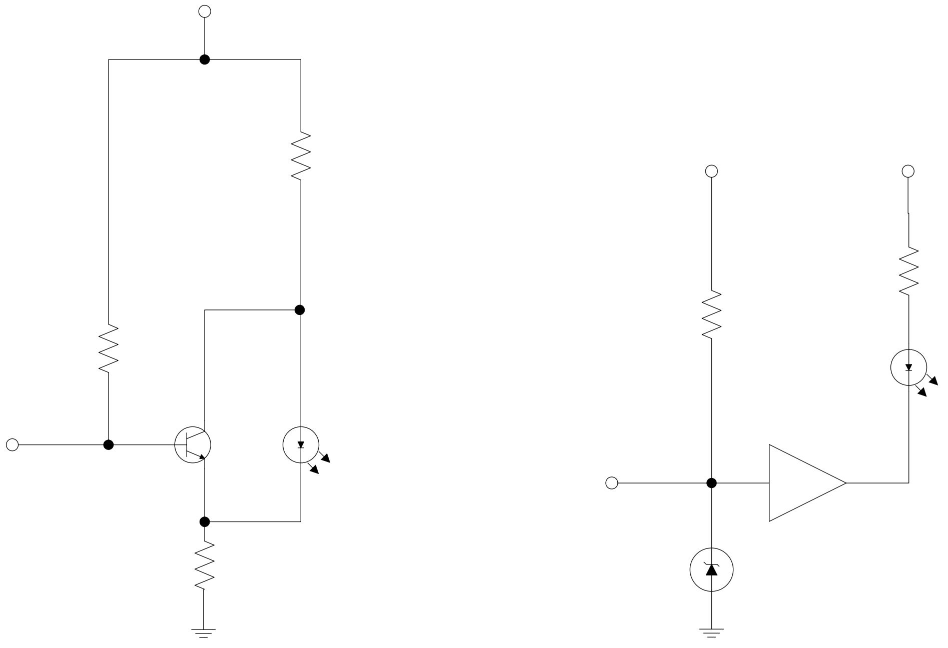


Figure 6 Example host LED driver circuits

2.3.1.3. Functional Definition

Table 3 defines the two activity signal states and the corresponding conditions.

+V

Table 3 Activity signal functional states

State	Condition
Signal asserted (driven low)	Command(s) outstanding[1]
Signal de-asserted (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 11 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.	

2.3.2. Staggered Spin-up Disable Control (optional)

2.3.2.1. Electrical and Functional Definition

The staggered spin-up feature is defined in section 6.2. Devices may optionally provide support to disable staggered spin-up through pin 11 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 11 is asserted low by the host. If pin 11 is asserted low the device shall disable staggered spin-up and immediately initiate media spin-up. If pin 11 is not connected in the host (floating), devices that support staggered spin-up disable through pin 11 shall enable staggered spin-up. Table 4 defines the electrical signal requirements for the device detection of staggered spin-up disable.

Table 4 Host staggered spin-up control electrical requirements

Parameter	Min Value	Max Value	Description & Conditions
V_{HEnb}	1.8V	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}	-0.1V	225mV	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 optional 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.

2.3.2.2. Staggered Spin-up Disable Circuit (informative)

The host circuit for signaling staggered spin-up disable by pulling pin 11 low is vendor specific. Figure 7 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 11 to ground or for the host to actively drive the signal low.

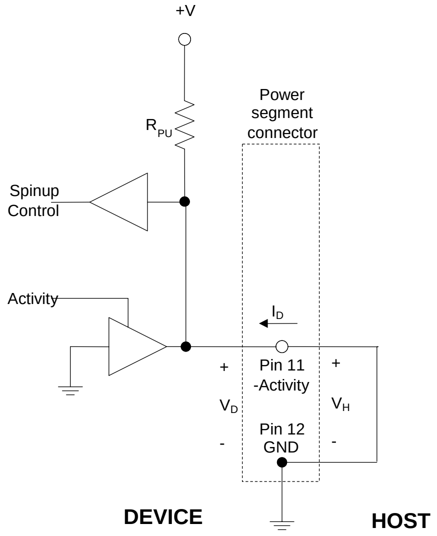


Figure 7 Example host circuit for signaling staggered spin-up disable

3. Transport Layer

The following section provides information on extensions and enhancements to the Serial ATA 1.0a transport layer.

3.1. Auto-Activate in DMA Setup FIS

In the Serial ATA 1.0a specification, 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 can 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 can be eliminated by merely having the DMA Setup FIS automatically activate the DMA controller.

In the Serial ATA 1.0a specification, the high-order bit of byte 1 (the second byte) of the DMA Setup FIS is reserved and cleared to zero. In order to eliminate the need for the DMA Activate FIS immediately following a DMA Setup FIS for a host-to-device transfer, as described in section 4.2.3.2 this bit is defined as the Auto-Activate bit. The modified definition of the DMA Setup FIS is illustrated in Figure 8.

0	**Reserved (0)**	**Reserved (0)**	**A**	**I**	**D**	**Reserved (0)**	**FIS Type (41h)**
1	**DMA Buffer Identifier Low**						
2	**DMA Buffer Identifier High**						
3	**Reserved (0)**						
4	**DMA Buffer Offset**						
5	**DMA Transfer Count**						
6	**Reserved (0)**						

Figure 8 DMA Setup FIS with Auto-Activate bit location identified

The Auto-Activate bit in the DMA Setup FIS is defined in Figure 9.

Auto-Activate bit value	Host action
1	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.
0	Behavior as defined in the Serial ATA 1.0a specification.

Figure 9 Auto-Activate bit definition

All other fields of the DMA Setup field are as defined in the Serial ATA 1.0a specification.

Host controllers that support the Auto-Activate capability shall have modified behavior for the HTDS1:HT_DS_FIS host transport state defined in section 8.6.10 of the Serial ATA 1.0a specification. The modified behavior shall be as defined in Figure 10. In the figure, text depicted in italic typeface is as defined in the Serial ATA 1.0a specification.

HTDS1: HT_DS_FIS	*Initialize the DMA controller for a first party DMA with the content of the request FIS.*		
	1. *No error detected* and Auto-Activate = 0*.*	→	*HT_HostIdle*
	2. *Error detected.*	→	*HT_HostIdle*
	3. *Notification of illegal transition error received from Link layer*	→	*HT_HostIdle*
	4. No error detected and Auto-Activate = 1.	→	HT_DMAOTrans2

Figure 10 Modified state HTDS1:HT_DS_FIS defining behavior for Auto-Activate

The modification to the definition of the DMA Setup FIS is fully compatible with Serial ATA 1.0a-compliant implementations. Devices that comply with the Serial ATA 1.0a specification have the Auto-Activate field in the DMA Setup FIS always cleared to zero, which yields behavior that is the same as that defined for the Serial ATA 1.0a specification. Devices that support this new behavior can elect to set the Auto-Activate bit in the DMA Setup FIS in order to eliminate an extra transaction for data transfers from host to device. Devices shall not attempt to utilize this capability prior to the optimization having been explicitly enabled by the host as defined in Section 4.5.4.2. 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.

3.2. Asynchronous Notification

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.

3.2.1. Set Device Bits FIS Enhancement

The Set Device Bits FIS is modified to allow devices to notify the host that attention is needed. In the Serial ATA 1.0a specification, the high-order bit of byte 1 (the second byte) of the Set Device Bits FIS is reserved and cleared to zero. In order to allow a device to notify the host that it needs attention, this bit is defined as the Notification bit. The 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 must 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.

0	Error	R	Status Hi	R	Status Lo	***N***	I	R	Reserved (0)	FIS Type (A1h)
1	Reserved (0)									

Figure 11 Set Device Bits FIS with Notification bit location identified

Field Definitions

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.

All other fields of the Set Device Bits FIS are as defined in the Serial ATA 1.0a specification.

3.2.2. 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. It is the responsibility of the device to ensure that all Reserved fields specified in Figure 11 are cleared to zero before sending the Set Device Bits FIS to the host.

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 5.1.2. A host may support Asynchronous Notification without supporting the SNotification register. The requirement for a host to support Asynchronous Notification is that it can 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.

3.2.3. 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 can 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 4.1. 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 4.1.

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.

3.2.4. ATAPI Notification

An ATAPI device shall indicate whether it supports asynchronous notification in Word 78 of IDENTIFY PACKET DEVICE, refer to section 4.5.3. The feature is enabled by using SET FEATURES, as defined in section 4.5.4.5.

3.2.4.1. 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.

3.3. Reserved FIS Types and Assignments

Figure 12 summarizes the FIS type value assignments. The FIS type values are enumerated and described in Appendix B of the Serial ATA 1.0a specification.

Type field value (hex)	Assignment	Description
0x27	SATA 1.0a	Register FIS – host to device
0x34	SATA 1.0a	Register FIS – device to host
0x39	SATA 1.0a	DMA Activate FIS – Device to host
0x41	SATA 1.0a	DMA Setup FIS – Bi-directional
0x46	SATA 1.0a	Data FIS – Bi-directional
0x58	SATA 1.0a	BIST Activate FIS – Bi-directional
0x5F	SATA 1.0a	PIO Setup FIS – Device to host
0xA1	SATA 1.0a	Set Device Bits FIS – Device to host
0xA6		Reserved for future Serial ATA definition
0xB8		Reserved for future Serial ATA definition
0xBF		Reserved for future Serial ATA definition
0xC7	SATA II	Vendor specific
0xD4	SATA II	Vendor specific
0xD9	SATA II	Reserved for Serial ATA Switch

Figure 12 FIS type value assignments

4. Command Layer

This section includes definitions of new command layer features and capabilities for the Serial ATA specification. All new capabilities defined here are evolutionary, incremental, and optional additions to the Serial ATA 1.0a specification.

4.1. Device command layer protocol enhancements

The state diagram below describes the changes to the Device Idle Protocol state diagram defined in Section 9.2 of the Serial ATA 1.0a specification.

In the following state tables, the states and transitions depicted using italic typeface correspond to states and transitions as defined in the Serial ATA 1.0a specification and are unchanged from their 1.0a definitions. In order to avoid reproducing the entire Serial ATA 1.0a state tables here, the destination states for such transitions are not included in this specification. The Serial ATA 1.0a states and transitions reproduced here are for the sake of indicating additions to those 1.0a states defined as part of this specification. Transitions preceded by an * are utilized only when queuing is implemented.

DI0: Device_idle	*Wait.*	
1. FIS receipt	→	*DI1: Check_FIS*
*2. *Ready to complete released command and FIS receipt not indicated*	→	*DI4: Set_service*
3. *Ready to receive data for WRITE FPDMA QUEUED command and FIS receipt not indicated and no error encountered	→	DFPDMAQ4: DataPhasePreWriteSetup
4. *Ready to transmit data for READ FPDMA QUEUED command and FIS receipt not indicated and no error encountered	→	DFPDMAQ3: DataPhaseReadSetup
5. *One or more FPDMA QUEUED commands completed successfully and FIS receipt not indicated and no error encountered	→	DFPDMAQ10: SendStatus[1]
6. *FPDMA QUEUED 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, and SRST set.	→	*DSR0: Software_reset_ asserted*
2. Register type, C bit cleared, and SRST cleared	→	*DI0: Device_idle*
3. Register type and C bit set.	→	*DI2: Check_command*
4. First Party Setup FIS Received	→	*DI0 : Device_idle*
5. Unexpected FIS type.	→	*DI0: Device_idle*

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 (legacy)* and no native queued command outstanding.	→	*DDMAI0: DMA_in*
5. *WRITE DMA command protocol (legacy)* and no native queued command outstanding.	→	*DDMAO0: DMA_out*
6. *PACKET command protocol* and no native queued command outstanding.	→	*DP0: PACKET*
7. **READ DMA QUEUED command protocol (legacy)* and no native queued command outstanding.	→	*DDMAQI0: DMA_queued_in*
8. **WRITE DMA QUEUED command protocol (legacy)* and no native queued command outstanding.	→	*DDMAQIO: 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*
13. *READ FPDMA QUEUED command protocol.	→	DFPDMAQ1: AddCommandToQueue
14. *WRITE FPDMA QUEUED command protocol.	→	DFPDMAQ1: AddCommandToQueue
15. *Not READ FPDMA QUEUED and not WRITE FPDMA QUEUED and not DEVICE RESET and native queued command(s) outstanding	→	DFPDMAQ12: BrokenHostClearBusy
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.*		

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, a queued command has been released, or an asynchronous notification has been sent.*

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 an 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 command, the device shall transition to the DFPDMAQ10: SendStatus state.

*** Transition DI0:6:** When the device has encountered an error in a FPDMA QUEUED 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: AN_Notify state.

DI1: Check_FIS state: *This state is entered when the device receives an FIS from the Transport layer.*

When in this state, the device shall check the FIS type.

Transition DI1:1: *If the FIS type is a Register FIS, the C bit in the FIS is cleared, and the SRST bit in the FIS is set, the device shall transition to the DSR0: Software_reset_asserted state.*

Transition DI1:2: *If the FIS type is a Register FIS, the C bit in the FIS is cleared, and the SRST bit in the FIS is cleared, the device shall transition to the DI0: Device_idle state.*

Transition DI1:3: *If the FIS type is a Register FIS and the C bit in the FIS is set, the device shall transition to the DI2: Check_command state.*

Transition DI1:4: *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 0 if asynchronous notification is supported. Clearing NotifyPending to 0 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 DP0: 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 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 not a READ FPDMA QUEUED; and not a WRITE FPDMA QUEUED; 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: BrokenHostClearBusy state.

4.2. Native Command Queuing

This section defines a simple and streamlined command queuing model for Serial ATA. The command queuing model is derived in part from the legacy queuing definition but with some of the legacy queuing aspects modified to take advantage of the capabilities of Serial ATA as well as to streamline the overall mechanism. A race-free status return mechanism is defined that minimizes the required protocol round trips in order to reduce the incurred overhead.

4.2.1. Goals, Objectives & Constraints

The native queuing definition has the following objectives and constraints:

- The queuing model must require no protocol additions or changes to the Serial ATA 1.0a specification in order to be implemented
- The queuing model must have minimal overhead in order to avoid potentially optimizing the exceptional deeply-queued case at the expense of the typical lightly-queued case
- The impact to the HBA implementation must be minimized to avoid dramatic controller redesign
- The software programming interface changes must be incremental to allow easy software migration from existing ATA software drivers

This section is limited in scope to the interface between the device and the host controller and does not define the host controller’s interface to host software. Native command queuing only makes sense in the context of new driver software and therefore does not comprehend the needs of master/slave emulation configurations that are used for supporting legacy software only.

4.2.2. Overview

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 deasserts the BSY bit before proceeding to execute received commands. The 32 reserved 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 FPDMA 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 before processing is again explicitly restarted.

4.2.3. Definition

4.2.3.1. Command Issue Mechanism

The Serial ATA transmission protocol is sensitive to the state of the BSY bit in the Status shadow register that provides write protection to the shared Shadow Register Block Registers. Since the Shadow Register Block Registers can be safely written only when the BSY bit is cleared to zero, the BSY bit conventions defined in Serial ATA 1.0a must 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 Status shadow register is cleared to zero, another command may be issued to the device.

The state of the BSY bit in the Status shadow 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 (the device queue depth is limited to 32 outstanding entries), and is conveyed in the Register FIS when the command is issued. For devices that report maximum queue depth less than 32 in their IDENTIFY DEVICE word 75, the host shall issue only unique tag values that have a value less than the value reported. For example, for devices reporting a maximum queue depth of 16, the host shall not issue a tag value greater than 15.

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 5.1.1 describes the SActive register and the access conventions for it.

4.2.3.2. Data Delivery Mechanism

The Serial ATA 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 First Party DMA mechanism defined in the Serial ATA 1.0a specification already specifies that the passed buffer identifier is only a "token" and that it shall not be construed to be a physical address. Use of the tag for the buffer identifier is therefore consistent with the existing Serial ATA specification.

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 must account for the DMA Setup FIS buffer identifier being a value between 0 and 31 and the host controller must 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 3.1.

If non-zero buffer offsets in the DMA Setup FIS are not enabled (see section 4.5.4.1) or not supported (see section 4.5.2), 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, it will not be possible for a device to 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 sections 4.5.2 or 4.5.3) or is disabled (see section 4.5.4.4), 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 4.5.4.1 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 Setups to satisfy a particular I/O process, but if multiple DMA setups are used, the data must 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 13 (see also the Serial ATA 1.0a specification).

0	**Reserved (0)**	**Reserved (0)**	**A**	**I**	**D**	**Reserved (0)**	**FIS Type (41h)**
1	**0**						**TAG**
2	**0**						
3	**Reserved (0)**						
4	**DMA Buffer Offset**						
5	**DMA Transfer Count**						
6	**Reserved (0)**						

Figure 13 DMA Setup FIS definition for memory buffer selection

Field Definitions

FIS Type Set to a value of 41h. Defines the rest of the FIS fields. Defines the total length of the FIS as seven Dwords.

D Indicates whether subsequent data transferred after this FIS is from transmitter to receiver or from receiver to transmitter. 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 For a DMA Setup with transfer direction from host to device indicates whether the host should immediately proceed with the data transfer without awaiting a subsequent DMA Activate to start the transfer. See section 3.1 for details. 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 as defined in the Serial ATA 1.0a specification 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 Serial ATA 1.0a specification is used to convey a TAG value that occupies the five least-significant bits of the field.

DMA Buffer Offset

This is the byte offset into the buffer for the transfer. Bits <1:0> shall be zero. 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 4.5.4. Data is transferred to/from sequentially increasing logical addresses starting at the specified offset in the specified buffer.

DMA Transfer Count

This is the number of bytes that will be transferred. Bit zero must be zero and the value must accurately reflect the length of the data transfer to follow. Refer to section 8.5.9.2 of the Serial ATA 1.0a specification for special considerations if the transfer count is for an odd number of words. Devices shall not set this field to ‘0’; a value of ‘0’ for this field is illegal and will result in indeterminate behavior.

I Interrupt The queuing model defined here does not make use of an interrupt following the data transfer phase (after the transfer count is exhausted) as defined in the Serial ATA 1.0a spec. The I bit shall be cleared to zero.

R/Reserved All reserved fields shall be cleared to zero.

4.2.3.3. Success Status 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 the Serial ATA 1.0a specification 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 must 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 section 4.2.3.4 to convey error information as well as the affected command tag.

4.2.3.3.1. Outputs

Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Error	0							
Sector Count	na							
Sector Count (exp)	na							
Sector Number	na							
Sector Number (exp)	na							
Cylinder Low	na							
Cylinder Low (exp)	na							
Cylinder High	na							
Cylinder High (exp)	na							
Device/Head	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	**31**	**30**	**29**	**…**	**2**	**1**	**0**
SActive*	ACT 31:0						

Figure 14 Register values for successful result

*The SActive register is defined in section 5.1.1.

ACT Bit positions set to one for each command TAG still outstanding. Device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set for each bit position to be cleared in the SActive register.

BSY	0:device is prepared to receive another command 1: device is not prepared to receive another command
DRDY	1
DF	0
DRQ	0
ERR	0
na	As defined in the ATA/ATAPI-6 standard

Upon successful completion of an outstanding command, the device shall transmit a Set Device Bits FIS with bits set in the SActive 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 and the value in the Error register shall be zero.

Only the registers that are updated as part of the Set Device Bits FIS are modified. Other Shadow Register Block Registers are left unchanged as indicated by "na" in the table.

The SActive field occupies the last 32 bits of the Set Device Bits FIS as defined in Figure 15.

0	Error	R	Status Hi	R	Status Lo	R	I	R	Reserved (0)	FIS Type (A1h)
1	SActive 31:0									

Figure 15 Set Device Bits FIS and SActive field

SActive The SActive 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. The device shall only indicate completion notification for a command if it has completed successfully.

Other All other fields as defined in the Serial ATA 1.0a specification.

4.2.3.4. Error Handling Mechanism

If the device has received a command that has not yet been acknowledged by clearing the BSY bit to zero and an error condition is detected, the device shall transmit a Register FIS to the host with the ERR bit set to one and the BSY bit cleared to zero in the Status field of the FIS and stop all further command processing until host software can respond to the error condition with appropriate action. 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 to the host with the ERR bit set to one in the Status field of the FIS and stop all further command processing until host software can respond to the error condition with appropriate action. All outstanding commands issued to the device at the time of an error are aborted as part of the error response and may be re-issued as appropriate by the host. Upon detecting an error when there are one or more queued commands outstanding, the device shall stop processing commands until a READ LOG EXT command with a log page of 10h is issued. When a READ LOG EXT command with a log page of 10h is issued, the device shall abort any outstanding queued commands and 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 section 4.2.5.3. The READ LOG

EXT page reflects the error information for the last recorded erring queued command until such time as another erring queued command is encountered. Issuing the READ LOG EXT command thus not only retrieves extended error information but also results in the device aborting any remaining queued commands and performing any cleanup tasks necessary to again be ready to process commands.

The host may perform the following functions to detect and handle an error condition for a queued command:

1. When the host receives status notification for a command, by polling or interrupt, the host checks if any outstanding queued command has completed successfully and indicates successful completion to the operating system for those commands. The host can determine which commands have successfully completed since the last time status was checked by comparing the value in the SActive register with the previously stored value.
2. The host checks the ERR bit in the Status register to determine if any queued command has failed. At this point the host is aware that an error has occurred.
3. The host issues a READ LOG EXT command with a log address of 10h. The READ LOG EXT command causes the device to abort any commands remaining in the queue. The command returns detailed information about the error condition encountered including:
   - The tag value of the queued command that failed
   - The ATA Shadow Register Block image including error code for the queued command that failed

   The information returned by the READ LOG EXT with a log address of 10h is persistent and any subsequent read of this log information returns the same data until such time as a new queued error condition is encountered.
4. The host can now re-issue any aborted commands and/or begin sending new commands to the device.

The device shall only signal completion notification for commands that have completed successfully or for queued commands that are aborted as a result of the host issuing a READ LOG EXT command with log page of 0x10. This means that a failed queued command will always have its corresponding bit in the SActive shadow register set to one until cleared as a consequence of the host issuing a READ LOG EXT command. A device will clear all SActive shadow register bits in response to a received READ LOG EXT command by transmitting a Set Device Bits FIS to the host with all the bits in the SActive field set to one. This policy avoids the host inadvertently completing/retiring a failed command with successful status.

4.2.3.4.1. Outputs

Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Error	ERROR							
Sector Count	na							
Sector Count (exp)	na							
Sector Number	na							
Sector Number (exp)	na							
Cylinder Low	na							
Cylinder Low (exp)	na							
Cylinder High	na							
Cylinder High (exp)	na							
Device/Head	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	**31**	**30**	**29**	**…**	**2**	**1**	**0**
SActive	ACT 31:0						

Figure 16 Register values for error result

ERROR	ATA error code
ACT	Bit positions set to one for each command TAG still outstanding. Only successfully completed commands have bit positions zeroed. Device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set for each bit position to be cleared in the SActive register.
BSY	0
DRDY	1
DF	0
DRQ	0
ERR	1
na	As defined in the ATA/ATAPI-6 standard

Only the registers that are updated as part of the Set Device Bits FIS are modified. Other Shadow Register Block Registers are left unchanged as indicated by “na” in the table.

4.2.4. Intermixing Legacy Commands and Queued Commands

The host shall not issue a legacy ATA command while a native queued command is outstanding. Upon receiving a legacy ATA 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 4.2.3.4 except as noted below. Legacy ATA commands include all commands other than the READ FPDMA QUEUED and WRITE FPDMA QUEUED commands defined in section 4.2.5.1 and section 4.2.5.2.

The legacy ATA *ReadLogExt* command with a specified log page of 10h 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 and shall clear all bits in the SActive shadow register by transmitting a Set Device Bits FIS to the host with bits set to one in the SActive field for each bit position to be cleared in the SActive register. After completing a *ReadLogExt* command with a specified log page of 10h, the device must be prepared to execute subsequently issued queued commands regardless of any previous errors on a queued command.

4.2.5. Command Definitions

4.2.5.1. 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 17.

4.2.5.1.1. Inputs

Register	7	6	5	4	3	2	1	0
Features	Sector Count 7:0							
Features (exp)	Sector Count 15:8							
Sector Count	TAG					Reserved		
Sector Count (exp)	Reserved							
Sector Number	LBA 0:7							
Sector Number (exp)	LBA 31:24							
Cylinder Low	LBA 15:8							
Cylinder Low (exp)	LBA 39:32							
Cylinder High	LBA 23:16							
Cylinder High (exp)	LBA 47:40							
Device/Head	FUA	1	Res	0	Reserved			
Command	60h							

Figure 17 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.

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 parallel ATA, including the Sector Count 15:0 convention where a value of zero specifies that 65,536 sectors are to be transferred.

Upon accepting the command, the device shall clear the BSY bit if/when it is prepared to receive another command by transmitting a Register FIS to the host with the BSY bit cleared in the Status field of the FIS. The ability for the device to quickly clear the BSY bit will allow the host to issue another queued command without blocking on this bit. The host shall check the BSY bit in the shadow Status register before attempting to issue a new command in order to determine whether the device is ready to receive another command (and determine that the host has write access to the Shadow Register Block Registers). The device shall not trigger an interrupt in response to having successfully received the command, so the Register FIS that the device transmits to clear BSY shall have the I bit cleared to zero.

4.2.5.1.2. Success Outputs

Upon successful completion of an outstanding command, the device shall clear the bit in the SActive shadow register that corresponds to the bit position for the command TAG completing, and alert the host by transmitting a Set Device Bits FIS that has the I bit set to one and has bit positions in the SActive field set to one for each bit position to be cleared in the host SActive shadow register (as defined in 5.1.1). The ERR bit in the Status register shall be cleared and the value in the Error register shall be zero.

Register	7	6	5	4	3	2	1	0
Error	0							

Sector Count	na							
Sector Count (exp)	na							
Sector Number	na							
Sector Number (exp)	na							
Cylinder Low	na							
Cylinder Low (exp)	na							
Cylinder High	na							
Cylinder High (exp)	na							
Device/Head	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	**31**	**30**	**29**	**…**		**2**	**1**	**0**
SActive*	ACT 31:0							

Figure 18 READ FPDMA QUEUED success status result values

*The SActive register is defined in section 5.1.1.

ACT Bit positions set to one for each command TAG still outstanding. The device clears bit positions in the host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set for each bit position to be cleared in the SActive register.

BSY Unchanged from its previous value (Set Device Bits FIS does not write this field)

DRDY 1

DF 0

DRQ Unchanged from its previous value (Set Device Bits FIS does not write this field)

ERR 0

na As defined in the ATA/ATAPI-6 standard

Only the registers that are updated as part of the Set Device Bits FIS are modified. Other Shadow Register Block Registers are left unchanged, as shown by “na” in the table.

4.2.5.1.3. Error Outputs

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 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, and the I bit set to one, and halt further processing of commands until a READ LOG EXT command with a specified log page of 10h is received as described in section 4.2.3.4. If all commands have been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Set Device Bits FIS to the host with the ERR bit set to one in the Status field, the ATA error code in the Error field, and the I bit set to one, and halt further processing of commands until a READ LOG EXT command with a specified log page of 10h is received as described in section 4.2.3.4. The device shall leave the bit positions in the SActive shadow register corresponding to the erring tag value and any uncompleted queued commands set.

Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Error	ERROR							
Sector Count	na							
Sector Count (exp)	na							

Sector Number	na							
Sector Number (exp)	na							
Cylinder Low	na							
Cylinder Low (exp)	na							
Cylinder High	na							
Cylinder High (exp)	na							
Device/Head	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	**31**	**30**	**29**	**…**	**2**	**1**	**0**
SActive	ACT 31:0						

Figure 19 READ FPDMA QUEUED error status result values

ACT Bit positions set to one for each command TAG outstanding. The failed command is considered as still outstanding since it has not completed successfully. The device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set to one for each bit position to be cleared in the SActive register.

ERROR ATA error code for the failure condition of the failed command

BSY 0

DRDY 1

DF 0

DRQ 0

ERR 1

na As defined in the ATA/ATAPI-6 standard

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, leaving the other Shadow Register Block Registers unchanged, as shown by “na” in the table. If the device signals an error condition when the BSY bit in the shadow Status register is set, the device clears the BSY bit with a Register FIS which updates all registers in the Shadow Register Block, but the corresponding error information for the command is still retrieved using the READ LOG EXT command with a log page of 0x10.

4.2.5.2. WRITE FPDMA QUEUED

Queued native write commands make use of a new command. The format of the new command is defined in Figure 20.

4.2.5.2.1. Inputs

Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Features	Sector Count 7:0							
Features (exp)	Sector Count 15:8							
Sector Count	TAG					Reserved		
Sector Count (exp)	Reserved							
Sector Number	LBA 0:7							
Sector Number (exp)	LBA 31:24							

Cylinder Low	LBA 15:8							
Cylinder Low (exp)	LBA 39:32							
Cylinder High	LBA 23:16							
Cylinder High (exp)	LBA 47:40							
Device/Head	FUA	1	0	0	Reserved			
Command	61h							

Figure 20 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.

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 parallel ATA, including the Sector Count 15:0 convention where a value of zero specifies that 65,536 sectors are to be transferred.

Upon accepting the command, the device shall clear the BSY bit if/when it is prepared to receive another command by transmitting a Register FIS to the host with the BSY bit cleared to zero in the Status field of the FIS. The ability for the device to quickly clear the BSY bit will allow the host to issue another queued command without blocking on this bit. The host shall check the BSY bit in the shadow Status register before attempting to issue a new command in order to determine that the device is ready to receive another command (and determine that the host has write access to the Shadow Register Block Registers). The device shall not trigger an interrupt in response to having successfully received the command, so the initial status return that clears BSY shall not have an interrupt associated with it.

4.2.5.2.2. Success Outputs

Upon successful completion of an outstanding command, the device shall clear the bit in the SActive field that corresponds to the bit position for the command tag completing by transmitting a Set Device Bits FIS that has bit positions in the SActive field set to one for each bit position to be cleared in the host SActive shadow register as defined in 5.1.1, and shall trigger an interrupt. The ERR bit in the Status register shall be cleared and the value in the Error register shall be zero.

Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Error	0							
Sector Count	na							
Sector Count (exp)	na							
Sector Number	na							
Sector Number (exp)	na							
Cylinder Low	na							
Cylinder Low (exp)	na							
Cylinder High	na							
Cylinder High (exp)	na							
Device/Head	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	**31**	**30**	**29**	**…**	**2**	**1**	**0**

SActive*	ACT 31:0

Figure 21 WRITE FPDMA QUEUED success status result values

*The SActive register is defined in section 5.1.1.

ACT Bit positions set to one for each command TAG still outstanding. Device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set for each bit position to be cleared in the SActive register.

BSY Unchanged from its previous value (Set Device Bits FIS does not write this field)

DRDY 1

DF 0

DRQ Unchanged from its previous value (Set Device Bits FIS does not write this field)

ERR 0

na As defined in the ATA/ATPI-6 standard

Only the registers that are updated as part of the Set Device Bits FIS are modified. Other Shadow Register Block Registers are left unchanged, as shown by "na" in the table.

4.2.5.2.3. Error Outputs

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 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, and the I bit set to one, and halt further processing of commands until a READ LOG EXT command with a specified log page of 10h is received as described in section 4.2.3.4. If all commands have been acknowledged by clearing the BSY bit to zero and an error is encountered, the device shall transmit a Set Device Bits FIS to the host with the ERR bit set to one in the Status field, the ATA error code in the Error field, and the I bit set to one, and halt further processing of commands until a READ LOG EXT command with a specified log page of 10h is received as described in section 4.2.3.4. The device shall leave the bit positions in the SActive shadow register corresponding to the erring tag value and any uncompleted queued commands set.

Register	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Error	ERROR							
Sector Count	na							
Sector Count (exp)	na							
Sector Number	na							
Sector Number (exp)	na							
Cylinder Low	na							
Cylinder Low (exp)	na							
Cylinder High	na							
Cylinder High (exp)	na							
Device/Head	na							
Status	BSY	DRDY	DF	na	DRQ	na	na	ERR

Register	**31**	**30**	**29**	**…**	**2**	**1**	**0**
SActive	ACT 31:0						

Figure 22 WRITE FPDMA QUEUED error status result values

ACT	Bit positions set to one for each command TAG outstanding. The failed command is considered as still outstanding since it has not completed successfully. The device clears bit positions in host shadow register by transmitting a bitmask in the SActive field of the Set Device Bits FIS with bits set to one for each bit position to be cleared in the SActive register.
ERROR	ATA error code for the failure condition of the failed command
BSY	0
DRDY	1
DF	0
DRQ	0
ERR	1
na	As defined in the ATA/ATAPI-6 standard

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, leaving the other Shadow Register Block Registers unchanged, as shown by “na” in the table. If the device signals an error condition when the BSY bit in the shadow Status register is set, the device clears the BSY bit with a Register FIS which updates all registers in the Shadow Register Block.

4.2.5.3. READ LOG EXT Command Error Page

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 page has been added in order for the host to be able to retrieve additional information for erring queued commands.

The READ LOG EXT command is a general facility as defined in the ATA/ATAPI standards. Reading the queued error log page (page 10h) has the additional side effect defined in section 4.2.4 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 (i.e. after completion of the READ LOG EXT command the device returns to state DI0:Device_idle state as defined in section 4.1). Log page 10h for READ LOG EXT returns extended command error information.

4.2.5.3.1. READ LOG EXT Log Page 10h

Devices supporting the native queued capability shall support READ LOG EXT log page 10h. Page 10h is one page in length and is defined in Figure 23.

Byte	7	6	5	4	3	2	1	0
0	NQ	Reserved		TAG				
1	Reserved							
2	Status							
3	Error							
4	Sector Number							
5	Cylinder Low							
6	Cylinder High							
7	Dev/Head							
8	Sector Number Exp							
9	Cylinder Low Exp							
10	Cylinder High Exp							
11	Reserved							
12	Sector Count							
13	Sector Count Exp							
14	Reserved							
15	Reserved							
16	Reserved							
17	Reserved							
18	Reserved							
19	Reserved							
20	Reserved							
…								
255								
256	Vendor Specific							
…								
510								
511	Data Structure Checksum							

Figure 23 READ LOG EXT Log Page 10h data structure definition

TAG If the NQ bit is cleared, the TAG field contains the TAG corresponding to the queued command that failed.

NQ If set 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 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 as defined in the ATA/ATAPI-6 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 in the error register and all other bits cleared).

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 will be zero when the checksum is correct.

Reserved All reserved fields shall be cleared to zero

4.2.6. State Diagrams for Command Queuing

This state machine is in addition to the state diagrams already defined in section 9 of the Serial ATA 1.0a specification and in section 4.1 of this specification. These are high-level state machines used to describe behavior only.

This class includes:

– READ FPDMA QUEUED
– WRITE FPDMA QUEUED

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: BrokenHostClearBusy

DFPDMAQ2: ClearInterfaceBsy	Transmit Register FIS with BSY = 0 and DRQ=0 and interrupt bit I=0 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 = 1 (host memory write) and interrupt bit I=0	
1. First Party DMA Setup FIS transmission complete	→	DFPDMAQ8: DataXmitRead

DFPDMAQ4: DataPhasePreWriteSetup		
1. First Party DMA Setup AutoActivate option supported and enabled	→	DFPDMAQ5: DataPhaseWriteSetup
2. First Party DMA Setup Autoactivate option not supported or not enabled	→	DFPDMAQ6: DataPhaseOldWriteSetup

DFPDMAQ5: DataPhaseWriteSetup	Transmit a DMA Setup FIS to the Host with the DMA Buffer Identifier = TAG and D = 0 (host memory read) and AutoActivate = 1 and interrupt bit I=0	
1. DMA Setup FIS transmission complete	→	DFPDMAQ9: DataXmitWrite

DFPDMAQ6: DataPhaseOldWriteSetup	Transmit a DMA Setup FIS to the Host with the DMA Buffer Identifier = TAG and D = 0 (host memory read) and AutoActivate=0 and interrupt bit I=0	
1. DMA Setup FIS transmission complete	→	DFPDMAQ7: DataPhaseXmitActivate

DFPDMAQ7: DataPhaseXmitActivate	Transmit a DMA Activate FIS to the Host	
1. DMA Activate FIS transmission complete	→	DFPDMAQ9: DataXmitWrite

DFPDMAQ8: DataXmitRead	Transmit Data FIS to 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: DataPhaseXmitActivate
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 = 0, interrupt bit I = 1, and bit ***n*** in SActive 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, interrupt bit I = 1, ATA error code set in Error field, and bits in SActive field not set for any outstanding queued commands and bits set for any successfully completed queued commands for which completion notification not yet delivered.		
	1. Set Device Bits FIS transmission complete	→	DFPDMAQ13: WaitforClear

DFPDMAQ12: BrokenHostClearBusy	Halt command processing and transmit Register FIS to Host with ERR bit in Status field set, interrupt bit I = 1, BSY = 0, DRQ=0, and Error field = 0x04		
	1. Register FIS transmission complete	→	DFPDMAQ13: WaitforClear

DFPDMAQ13: WaitforClear	Wait for Host to either issue READ LOG EXT command with log page=0x10 or issue SRST		
	1. READ LOG EXT command with log page=0x10 received	→	DFPDMAQ14: SendQueueCleanACK
	2. SRST received	→	*DSR0: Software_reset_asserted*
	3. Any command other than READ LOG EXT with log page=0x10 received	→	DFPDMAQ12: BrokenHostClearBusy

DFPDMAQ14: SendQueueCleanACK	Discard all commands in the pending device queue. Transmit Set Device Bits FIS with ERR in Status field = 0, ERROR=0, SActive field = 0xFFFFFFFF, and interrupt bit I=0.		
	1. Set Device Bits FIS transmission complete	→	DPIOI0: PIO_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 will check the TAG validity and verify that it not already assigned to an outstanding command. If valid, the device will 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: BrokenHostClearBusy 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, or as a result of the received command being a non-queued command while queued commands are outstanding.

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=0 and DRQ=0 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 DI: 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 will transmit a First Party DMA setup FIS to the host with the DMA buffer identifier set to the queued TAG value and the direction set to D=1 (host memory write).

Transition DFPDMAQ3:1: When the device completes the transmission of the DMA Setup FIS, the device shall transition to the DFPDMAQ8: DataTransmitRead 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 will determine if First Party DMA AutoActivate option is supported and enabled, and then make the appropriate state transition.

Transition DFPDMAQ4:1: If the First Party DMA AutoActivate option is supported and enabled, the device shall transition to the DFPDMAQ5: DataPhaseWriteSetup state.

Transition DFPDMAQ4:2: If the First Party DMA AutoActivate option is not supported or not enabled, the device shall transition to the DFPDMAQ6: DataPhaseOldWriteSetup state.

DFPDMAQ5: DataPhaseWriteSetup: 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 will transmit a DMA setup FIS to the host with the DMA buffer identifier set to the queued TAG value and the direction set to D=0 (host memory read), and AutoActivate=1.

Transition DFPDMAQ5:1: When the device completes the transmission of the DMA Setup FIS, the device shall transition to the DFPDMAQ9: DataXmitWrite state.

DFPDMAQ6: DataPhaseOldWriteSetup: 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 AutoActivate, or it is not enabled.

When in this state, the device will transmit a DMA setup FIS to the host with the DMA buffer identifier set to the queued TAG value and the direction set to D=0 (host memory read), and AutoActivate=0.

Transition DFPDMAQ6:1: When the device completes the transmission of the DMA Setup FIS, the device shall transition to the DFPDMAQ7: DataPhaseXmitActivate state.

DFPDMAQ7: DataPhaseXmitActivate: 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 will transmit a DMA Activate FIS to the host indicating readiness to receive data FIS’s 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 will transmit 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 will remain 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 will receive 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: DataPhaseXmitActivate 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 will transmit a Set Device Bits FIS to the host with ERR = 0, interrupt bit I = 1, and bit ***n*** in SActive 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 will halt command processing and transmit a Set Device Bits FIS to the host with ERR = 1, interrupt bit I = 1, ATA error code set in the Error field, and bits in SActive field not set for any outstanding queued commands (including the erring command) and bits set 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: BrokenHostClearBusy: 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 will halt command processing and transmit a Register FIS to the host with ERR = 1 in the status field, interrupt bit I = 1, BSY = 0, DRQ = 0, and ATA error code set in the Error field.

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 READ LOG EXT command with log page = 0x10 or a soft reset. Any other commands will return an error register FIS.

Transition DFPDMAQ13:1: If the device receives a READ LOG EXT command with log page = 0x10, the device shall transition to the DFPDMAQ14: SendQueueCleanAck state.

Transition DFPDMAQ13:2: If the device receives a SRST, the device shall transition to the DSR0: Software_reset_asserted state defined in the Serial ATA 1.0a state machines.

Transition DFPDMAQ13:3: If the device receives command other than READ LOG EXT command with log page = 0x10, the device shall transition to the DFPDMAQ12: BrokenHostClearBusy state.

DFPDMAQ14: SendQueueCleanAck: This state is entered when the host has responded to an error FIS with a READ LOG EXT command with log page = 0x10.

The device shall discard all commands in the pending queue and transmit a Set Device Bits FIS with ERR in the status field = 0, ERROR = 0, SActive field = 0xFFFFFFFF, and Interrupt bit I = 0.

Transition DFPDMAQ14:1: when the Set Device Bits FIS transmission is complete, the device shall transition to the DPIOI0: PIO_in state.

4.2.7. Host command layer protocol for command queuing

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 or hardware/software partitioning.

This class includes:

- READ FPDMA QUEUED
- WRITE FPDMA QUEUED

HFPI0: Idle			
	1. Free TAG location and command waiting to have TAG assigned	→	HFPDMAQ2: PresetACTBit
	2. Command with assigned TAG awaiting issue and BSY=0 and not FPDMA 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 design-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 FPDMA 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 1 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 = 0	→	HFPI0: Idle
2. ERR = 1	→	HFPDMAQ8: ResetQueue

HFPDMAQ8: ResetQueue	Issue a READ LOG EXT command with log page = 0x10 in a Register FIS to device	
1. Unconditional	→	HFPDMAQ9: CleanupACK

HFPDMAQ9: CleanupACK	Wait for DRQ=1 and BSY=0 [1]	
1. DRQ=1 and BSY=0	→	HFPDMAQ10: RetrieveRequestSense
2. DRQ=0 or BSY=1	→	HFPDMAQ9: CleanupACK
NOTE: 1. The host may wait for this condition using any means including awaiting an interrupt and checking the DRQ and BSY status, spinning, or periodic timer.		

HFPDMAQ10: RetrieveRequestSense	Receive PIO data FIS with request sense information	
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

HFPI0: Idle: When in this state, if queuing is supported and enabled, the Command Layer is awaiting a READ FPDMA QUEUED or WRITE FPDMA QUEUED 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 DMA Read or Write 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=0 and the interface is not in the FPDMA 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 or WRITE FPDMA QUEUED 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 will attempt 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 will read 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 Command 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=0, the queue is maintained and it shall transition to the HFPI0: Idle state.

Transition HFPDMAQ7:2: If the SavedStatus value ERR=1, 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 Command will issue a READ LOG EXT command with log page = 0x10 in a Register FIS to the device.

Transition HFPDMAQ8:1: After the command has been issued, it shall transition to the HFPDMAQ9: CleanupACK state.

HFPDMAQ9: CleanupACK: When in this state, the Command Layer tests the Device for DRQ = 1 and BSY = 0 in preparation for a PIO data FIS transfer.

Transition HFPDMAQ9:1: If DRQ = 1 and BSY = 0, it shall transition to the HFPDMAQ10: ReceiveRequestSense state.

Transition HFPDMAQ9:2: If DRQ = 01 or BSY = 1, it shall transition to the HFPDMAQ9: CleanupACK state.

HFPDMAQ10: RetrieveRequestSense: When in this state, the Command Layer completes the PIO data FIS that retrieves the Request Sense information.

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.

4.3. Phy Event Counters (Optional)

Phy event counters are an optional feature to obtain more information about Phy level events that occur on 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 issuing the READ LOG EXT command with a log page of 11h. The counter values shall not be retained across power cycles. The counter values shall be preserved across COMRESET and software resets.

4.3.1. Counter Reset Mechanisms

There are two mechanisms by which the host can 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. The second mechanism uses the READ LOG EXT command. When the device receives a READ LOG EXT command for log page 11h and bit 0 in the Features register is set to one, the device shall return the current counter values for the command and then reset all Phy event counter values.

4.3.2. Counter Identifiers

Each counter begins with a 16-bit identifier. Figure 24 defines the counter value for each identifier. Any unused counter slots in the log page should have a counter identifier value of 0h. Optional counters that are not implemented shall not be returned in log page 11h. 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 page; the order is arbitrary and selected by the device vendor.

For all counter descriptions, 'transmitted' refers to items sent by the device to the host and 'received' refers to items received by the device from the host.

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 page
001h	Mandatory	Command failed due to an ICRC error
002h	Optional	Data FIS R_ERR ending status (transmitted and received)
003h	Optional	Data FIS R_ERR ending status (transmitted only)
004h	Optional	Data FIS R_ERR ending status (received only)
005h	Optional	Non-data FIS R_ERR ending status (transmitted and received)
006h	Optional	Non-data FIS R_ERR ending status (transmitted only)
007h	Optional	Non-data FIS R_ERR ending status (received only)
008h	Optional	Non-data FIS retries (transmitted)
009h	Optional	Transitions from drive PhyRdy to drive PhyNRdy
00Ah	Mandatory	Signature D2H Register FISes sent due to a COMRESET
00Bh	Optional	CRC errors within the FIS (received)
00Dh	Optional	Non-CRC errors within the FIS (received)
00Fh	Optional	Data FIS R_ERR ending status due to CRC errors (received)
010h	Optional	Data FIS R_ERR ending status due to non-CRC errors (received)
012h	Optional	Non-data FIS R_ERR ending status due to CRC errors (received)
013h	Optional	Non-data FIS R_ERR ending status due to non-CRC errors (received)

Figure 24 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 ending status counters.

4.3.3. READ LOG EXT Log Page 11h

READ LOG EXT log page 11h is one page (512 bytes) in length. The first Dword of the log page contains information that applies to the rest of the log page. Software should continue to process counters until a counter identifier with value 0h is found or the entire page has been read. A counter identifier with value 0h indicates that the log page contains no more counter values past that point. Log page 11h is defined in Figure 25.

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 25 READ LOG EXT Log Page 11h 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 24.

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 4.3.2). 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 0xFF, it shall be one-extended to 0xFFFF. 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 will be zero when the checksum is correct.

Reserved All reserved fields shall be cleared to zero

4.4. Non-512 Byte Sector Size (Informative)

ATA disk drives today generally support 512-byte sector sizes exclusively. 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. Some classes of applications have a need for sector sizes of 520 bytes or 528 bytes.

Since there is no inherent sector size sensitivity in the Serial ATA interface, the command set extensions for supporting the reporting of sector sizes other than 512 bytes is relegated to the INCITS T13 committee for resolution.

Regardless of the physical sector size of storage devices, the maximum Data FIS length is 8192 bytes as defined in the Serial ATA 1.0a specification. 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.

4.5. IDENTIFY DEVICE/SET FEATURES

4.5.1. Overview

New Serial ATA features and capabilities include a means by which their presence and support can be determined, and a means for enabling them if optionally supported. All new features and capabilities are be supported in a way that does not compromise Serial ATA 1.0a compatibility, and all new features are be optional supersets of the Serial ATA 1.0a definition.

4.5.2. IDENTIFY DEVICE Definition

ATA devices report their capabilities using the IDENTIFY DEVICE command. Figure 26defines the additional capabilities reported in IDENTIFY DEVICE for Serial ATA devices.

Word	O/M	F/V	Description
0-74			As defined in the ATA reference
75	O		Queue depth
		F	15-5 Reserved
		F	4:0 Maximum queue depth-1
76	O		Serial ATA capabilities
		F	15-11 Reserved
		F	10 Supports Phy event counters
		F	9 Supports receipt of host-initiated interface power management requests
		F	8 Supports native command queuing
		F	7-4 Reserved
		F	3 Reserved for future Serial ATA
		F	2 1 = Supports Serial ATA Gen-2 signaling speed
		F	1 1 = Supports Serial ATA Gen-1 signaling speed (1.5Gbps)
		F	0 Reserved (cleared to zero)
77			Reserved for future Serial ATA definition
78	O		Serial ATA features supported
		F	15-7 Reserved
		F	6 1 = supports software settings preservation
		F	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 Reserved (cleared to zero)
79	O		Serial ATA features enabled
		F	15-7 Reserved
		V	6 1 = software settings preservation enabled
		F	5 Reserved
		V	4 1 = in-order data delivery enabled
		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
		V	0 Reserved (cleared to zero)
80-255			As defined in the ATA reference
Key: O/M = Mandatory/optional requirement. M = Support of the word is mandatory. O = Support of the word is optional. F/V = Fixed/variable content 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 may be fixed or variable.			

Figure 26 IDENTIFY DEVICE field definitions

WORD 75: Queue depth

This word is as defined in the ATA reference. The native queuing scheme supports at most 32 queued commands, which coincides with the reporting capabilities of the ATA specification. In the native queuing scheme, the host is required to issue only unique tag values for queued commands that have a value less than or equal to the value reflected in this field (i.e. 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 queued commands).

WORD 76: Serial ATA capabilities

If not 0000h or FFFFh, the device claims compliance with the Serial ATA specification and supports the signaling rate indicated in bits 1-3. Since Serial ATA will support generational compatibility, multiple bits may be set. Bit 0 is reserved and shall be set 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 is reserved and shall be cleared to zero

Bit 1 when set to one indicates that the device is a Serial ATA device and supports the Gen-1 signaling rate of 1.5Gbps.

Bit 2 when set to one indicates that the device is a Serial ATA device and supports the Gen-2 signaling rate.

Bit 3-7 are reserved 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 4.1.

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 READ LOG EXT page 11h.

Bit 11-15 are reserved and shall be cleared to zero

WORD 77: Reserved

Word 77 is reserved for future Serial ATA definition and shall be zero.

WORD 78: Serial ATA features supported

If word 76 is not 0000h or FFFFh, 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 is reserved and shall be set 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 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 4.6. 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

WORD 79: Serial ATA features enabled

If word 76 is not 0000h or FFFFh, 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 is reserved and shall be set 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 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 7-15 are reserved and shall be cleared to zero

4.5.3. IDENTIFY PACKET DEVICE Definition

ATAPI devices report their capabilities using the IDENTIFY PACKET DEVICE command. Figure 27defines the additional capabilities reported in IDENTIFY PACKET DEVICE for Serial ATA ATAPI devices.

Word	O/M	F/V	Description
0-75			As defined in the ATA reference
76	O		Serial ATA capabilities

		F	15-11 Reserved
		F	10 Supports Phy event counters
		F	9 Supports receipt of host-initiated interface power management requests
		F	8-4 Reserved
		F	3 Reserved for future Serial ATA
		F	2 1 = Supports Serial ATA Gen-2 signaling speed
		F	1 1 = Supports Serial ATA Gen-1 signaling speed (1.5Gbps)
		F	0 Reserved (cleared to zero)
77			Reserved for future Serial ATA definition
78	O		Serial ATA features supported
		F	15-7 Reserved
		F	6 1 = supports software settings preservation
		F	5 1 = supports asynchronous notification
		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 Reserved (cleared to zero)
79	O		Serial ATA features enabled
		F	15-7 Reserved
		V	6 1 = software settings preservation enabled
		V	5 1 = asynchronous notification enabled
		V	4 1 = in-order data delivery enabled
		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
		V	0 Reserved (cleared to zero)
80-255			As defined in the ATA reference

Key:
O/M = Mandatory/optional requirement.
M = Support of the word is mandatory.
O = Support of the word is optional.
F/V = Fixed/variable content
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 may be fixed or variable.

Figure 27 IDENTIFY PACKET DEVICE field definitions

WORD 76: Serial ATA capabilities

If not 0000h or FFFFh, the device claims compliance with the Serial ATA specification and supports the signaling rate indicated in bits 1-3. Since Serial ATA will support generational compatibility, multiple bits may be set. Bit 0 is reserved and shall be set 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 is reserved and shall be cleared to zero

Bit 1 when set to one indicates that the device is a Serial ATA device and supports the Gen-1 signaling rate of 1.5Gbps.

Bit 2 when set to one indicates that the device is a Serial ATA device and supports the Gen-2 signaling rate.

Bit 3-8 are reserved and shall be cleared to zero

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 READ LOG EXT page 11h.

Bit 11-15 are reserved and shall be cleared to zero

WORD 77: Reserved

Word 77 is reserved for future Serial ATA definition and shall be zero.

WORD 78: Serial ATA features supported

If word 76 is not 0000h or FFFFh, 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 is reserved and shall be set 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 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 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 3.2.

Bit 6 indicates whether the device supports software settings preservation as defined in section 4.6. 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.

WORD 79: Serial ATA features enabled

If word 76 is not 0000h or FFFFh, 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 is reserved and shall be set 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 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 indicates whether device support for asynchronous notification to indicate to the host that attention is required is enabled. When set to one the device may initiate notification events. When cleared to zero the device shall not initiate notification events. This field shall be cleared to zero by default. An example of an event that the device may need attention for includes a media change. Asynchronous notification is described in section 3.2.

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 7-15 are reserved and shall be cleared to zero

4.5.4. SET FEATURES Definition

Devices are informed of host capabilities and have optional features enabled/disabled through the SET FEATURES command defined in ATA/6. Serial ATA features are controlled using a new features value as defined in Figure 28.

Features Value	Description
10h	Enable use of Serial ATA feature
90h	Disable use of Serial ATA feature

Figure 28 Features enable/disable values

The Sector Count register contains the specific Serial ATA feature to enable or disable. The specific Serial ATA features are defined as defined in Figure 29.

Sector Count Value	Description
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

Figure 29 Feature identification values

4.5.4.1. Enable/Disable Non-Zero Offsets in DMA Setup

A Sector Count 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 4.2.3.2). 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 FPDMA 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.

4.5.4.2. Enable/Disable DMA Setup FIS Auto-Activate Optimization

A Sector Count 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, the Serial ATA 1.0a specification indicates that first-party 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 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.

4.5.4.3. Enable/Disable Device-Initiated Interface Power State Transitions

A Sector Count 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 the PM_REQ protocol primitive 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. 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 (PACKET) DEVICE data structure.

4.5.4.4. Enable/Disable Guaranteed in-Order Data Delivery

A Sector Count 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, 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.

4.5.4.5. Enable/Disable Asynchronous Notification

A Sector Count 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.

4.5.4.6. Enable/Disable Software Settings Preservation

A Sector Count value of 06h is used by the host to enable or disable software settings preservation, as defined in section 4.6. 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 hard reset in Parallel ATA).

4.5.5. READ LOG EXT Log Directory

Devices supporting READ LOG EXT log page 10h reflect this support in the General Purpose Log Directory page (log page 0) by having the value 1 at offset 20h and the value 0 at offset 21h of that log page to indicate existence of a log page at address 10h of 1-page in length.

Devices supporting READ LOG EXT log page 11h reflect this support in the General Purpose Log Directory page (log page 0) by having the value 1 at offset 22h and the value 0 at offset 23h of that log page to indicate existence of a log page at address 11h of 1-page in length.

Byte	Value
0 - 1Fh	As defined in the ATA reference
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

4.6. Software Settings Preservation

When a device is enumerated, software will configure the device using SET FEATURES and other commands. These software settings are often preserved across software reset but not necessarily across hardware reset. In Parallel ATA, only commanded hardware resets can occur, thus legacy software only reprograms settings that are cleared for the particular type of reset it has issued. In Serial ATA, COMRESET is equivalent to hard 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 software knowledge. In order to avoid losing important software settings without legacy 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 4.5.4.6). If a device supports software settings preservation, the feature shall be enabled by default.

4.6.1. COMRESET Preservation Requirements

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.

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 READ LOG EXT and WRITE LOG EXT).

Security mode state: The security mode state established by Security Mode feature set commands (refer to section 6.13 of the ATA/6 specification). 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 ADDRESS MAX (EXT): The maximum LBA specified in SET ADDRESS MAX or SET ADDRESS MAX 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 Sector Count 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.

4.7. Defect Management (Informative)

4.7.1. 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 ReadDefectData (37h) and ReassignBlocks (07h) have been used to allow storage administrators or applications to proactively address problematic sectors on a disk surface.

Serial ATA drives leverage the economies of the desktop drive market, and as a consequence inherit both the design philosophy and implementation of desktop drives. From a design philosophy perspective, desktop drives have never yielded the low-level of control, such as reassignment of logical blocks that is commonplace in enterprise-class drives. Instead, desktop drives 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 drive performance and awareness of defect management activity in general.

As “black box” providers, Serial ATA drive 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 drive 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 drives provide similar information via S.M.A.R.T. (Self-Monitoring, Analysis and Reporting Technology) commands as the information returned with SCSI ReadDefectData commands. Subsystem designers make no provisions for reassignment of blocks; rather reassignment of defective blocks is performed automatically when indicated by desktop-class drives. Additionally, disk drive manufacturers provide a spectrum of tools and embedded features that help prevent occurrence of non-recoverable read errors, that help detect disk drive degradation that might cause catastrophic loss of data, and that help administrators diagnose and isolate the cause of error events.

4.7.2. 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 drive 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.

4.7.3. 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 drive manufacturers to determine the nature of a defect. If a defect is determined to be of a permanent nature, drive 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 drive 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 drive vendors and drive models and reflects tradeoffs between drive 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 will 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 drive 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 S.M.A.R.T. in section 4.7.5for 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 can 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.

4.7.4. Continuous Background Defect Scanning (Informative)

Serial ATA drive 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.

4.7.5. 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 ATA/ATAPI Spec version 6, section 6.14 describes S.M.A.R.T. command support in detail. Individual drive vendors provide unique S.M.A.R.T. capabilities on their drives and documentation on those capabilities so that host or controller developers can effectively use predictive failure information

Design Recommendation: Use capabilities provided through ATA S.M.A.R.T. commands to predict disk failure when possible. Predictive knowledge can be used to swap in spare disk drives in RAID configurations, to trigger data backup or protection routines, or to schedule storage subsystem maintenance.

4.8. Device Configuration Overlay

4.8.1. Definition

Figure 30 defines additional features and capabilities that support can be controlled for using the Device Configuration Overlay feature in the ATA reference. 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 ATA reference
8	Serial ATA command / feature sets supported 15-5 Reserved (0)

	4 1 = Supports software settings preservation 3 1 = Supports asynchronous notification 2 1 = Supports interface power management 1 1 = Supports non-zero buffer offsets in DMA Setup FIS 0 1 = Supports native command queuing
9	Reserved for Serial ATA
10-255	As defined in the ATA reference

Figure 30 Device Configuration Overlay data structure

WORD 8: Serial ATA command / feature sets supported

This word enables configuration of command sets and feature sets.

Bit 0 indicates whether native command queuing shall be supported by the device. When set to one, the drive shall support native command queuing. When cleared to zero, drive support for native command queuing shall be disabled and Word 76 bit 8, Word 78 bit 1, Word 78 bit 2, Word 78 bit 4, Word 79 bit 1, Word 79 bit 2, and Word 79 bit 4 of IDENTIFY DEVICE shall all be cleared to zero. 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 register and the ABRT bit set to one in the Error register.

Bit 1 indicates whether the drive supports non-zero buffer offsets in the DMA Setup FIS. When set to one, the drive shall support non-zero buffer offsets in the DMA Setup FIS. When cleared to zero, drive support for non-zero buffer offsets in the DMA Setup FIS shall be disabled and Word 78 bit 1, Word 78 bit 4, Word 79 bit 1, and Word 79 bit 4 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE shall all be cleared to zero. 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.

Bit 2 indicates whether the drive supports interface power management requests. When set to one, the drive shall support receiving host initiated power management requests and shall support device initiated power management requests. When cleared to zero, the drive shall not support receiving host initiated power management requests and shall not support device initiated power management requests and Word 76 bit 9, Word 78 bit 3, and Word 79 bit 3 of IDENTIFY DEVICE or IDENTIFY PACKET DEVICE shall all be cleared to zero. If interface power management requests are disabled, the drive shall respond with PMNAK$_P$ to any interface power management requests and the device shall not initiate PMREQ_P$_P$ or PMREQ_S$_P$ primitives to the host.

Bit 3 indicates whether the drive supports asynchronous notification. When set to one, the drive shall support asynchronous notification. When cleared to zero, the drive shall not support asynchronous notification and Word 78 bit 5 and Word 79 bit 5 of IDENTIFY PACKET DEVICE shall both be cleared to zero. When asynchronous notification is disabled, the device shall not initiate a Set Device Bits FIS with the Notification bit set to one.

Bit 4 indicates whether the device supports software settings preservation. When set to one, the device shall support software settings preservation. When cleared to zero, the device shall not support software settings preservation and Word 78 bit 6 and Word 79 bit 6 of IDENTIFY (PACKET) DEVICE shall both be cleared to zero. When software settings preservation is disabled, the device shall not preserve any software settings that are normally cleared upon an ATA hardware reset.

Bit 5-15 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.

5. Host Controller Registers and Hardware Requirements

The native queuing model defines the 32-bit reserved field in the Set Device Bits FIS as the SActive field. The host controller must provide some mechanism by which the SActive field of the Set Device Bits FIS can be exposed to host software. Exposing these bits to the host is accomplished through addition of a fourth register to the Serial ATA status and control register block.

5.1. Register Definition

Figure 31 defines the register added to the Serial ATA Status and Control register (SCR) block to expose the information returned in the SActive field of the Set Device Bits FIS.

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

Figure 31 Serial ATA Status and Control (SRC) registers

5.1.1. 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.

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 upon issuing a hard reset (COMRESET) signal or as a result of issuing a software reset by transmitting a Control Register FIS with the SRST bit asserted.

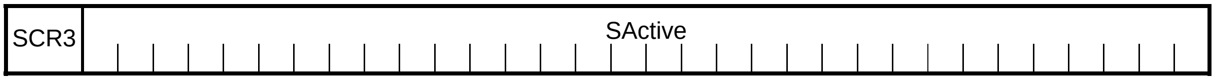


Figure 32 SActive register definition

SActive For the native command queuing protocol, the SActive value represents the set of outstanding queued commands that have not completed successfully yet. The value is bit significant and each bit position represents the status of a pending queued command with a corresponding TAG value.

5.1.2. SNotification register

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 3.2. 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; the PM Port field is defined in the Port Multiplier 1.0 specification. For example, if the PM Port field is set to 7 then the host shall set bit 7 in the SNotification register to 1. After setting the bit in the SNotification register, the host shall generate an interrupt if the I 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 1’s encoded in the bit positions corresponding to the bits that are to be cleared.

SCR4	Reserved	Notify

Figure 33 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.

Reserved The reserved field shall be cleared to zero.

5.1.3. SError Register Enhancement for Device Change Detection

The Serial ATA interface error register (SError) as defined in the Serial ATA 1.0a specification 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. There is no direct indication in Serial ATA 1.0a of a device change condition, and in a hot-plug scenario where devices may be removed and inserted, host notification of a device change condition is essential in order to ensure that data intended for one device is not inadvertently written to another device when the operator changes the devices.

In order to facilitate a means for notifying host software of the potential of a device having been changed, an additional bit to the DIAG field of the SError superset register is defined as indicated in Figure 34.

DIAG	Reserved	***X***	F	T	S	H	C	D	B	W	I	N

Figure 34 Location of X bit in DIAG field

N As defined in Serial ATA 1.0a

I As defined in Serial ATA 1.0a

W As defined in Serial ATA 1.0a

B As defined in Serial ATA 1.0a

D As defined in Serial ATA 1.0a

C As defined in Serial ATA 1.0a

H As defined in Serial ATA 1.0a

S As defined in Serial ATA 1.0a

T As defined in Serial ATA 1.0a

F As defined in Serial ATA 1.0a

X Exchanged: When set to one this bit indicates that device presence has changed since the last time this bit was cleared. The means by which the implementation determines that the device presence has changed is vendor specific. This bit may be set 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.

5.1.4. SStatus Register Enhancement for Gen-2 Signaling Speed

The Serial ATA interface status register (SStatus) as defined in the Serial ATA 1.0a specification includes reporting of the negotiated interface signaling speed, but does not directly specify the values reported for signaling rates other than the first generation (Gen-1) signaling rate of 1.5Gbps. The SStatus register's SPD field that reports the negotiated signaling rate is further defined to convey the negotiated signaling speed between the host controller and the device.

SCR0	Reserved (0)	IPM	SPD	DET

Figure 35 SStatus register definition

DET As defined in Serial ATA 1.0a

SPD The SPD value indicates the negotiated interface communication speed established
0000b As defined in 1.0a (No negotiated speed, communication not established)
0001b As defined in 1.0a (Generation 1 communication rate negotiated)
0010b Generation 2 communication rate negotiated
All other values reserved

IPM As defined in Serial ATA 1.0a

Reserved All reserved fields shall be cleared to zero.

5.1.5. SControl Register Enhancement for Gen-2 Signaling Speed

The Serial ATA interface control register (SControl) as defined in the Serial ATA 1.0a specification includes selection of maximum negotiated interface signaling speed, but does not directly specify the values reported for signaling rates other than the first generation (Gen-1) signaling rate of 1.5Gbps. The SControl register's SPD field that selects the maximum signaling rate to be negotiated is further defined as follows:

SCR2	Reserved (0)	IPM	SPD	DET

Figure 36 SControl register definition

DET As defined in Serial ATA 1.0a

SPD The SPD field represents the highest allowed communication speed the interface is allowed to negotiate when interface communication speed is established
0000b As defined in Serial ATA 1.0a (No speed negotiation restrictions)
0001b As defined in Serial ATA 1.0a (Limit speed negotiation to a rate not greater than Gen-1 communication rate)
0010b Limit speed negotiation to a rate not greater than Gen-2 communication rate
All other values reserved

IPM As defined in Serial ATA 1.0a

Reserved All reserved fields shall be cleared to zero.

5.2. HBA Enforcement of FPDMA Data Phase Atomicity

The host bus adapter shall ensure the FPDMA 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 is transmitted from the host to device between the reception of a DMA Setup FIS and the exhaustion of the associated transfer count.

5.3. Host Transport Layer Accommodation for Asynchronous FIS Reception

Host bus adapters that accommodate asynchronous FIS reception as is required for the native command queuing protocol, require a modification to the automatic FIS transmission retry mechanism defined in the Serial ATA 1.0a specification.

All automatic FIS retry transitions in the Serial ATA 1.0a Host Transport layer shall transition through the host Transport layer Idle state HTI1:HT_HostIdle prior to performing each retry attempt. In the host Transport layer Idle state, the transition for reception of a FIS from the device has priority over all other transitions in that state. In the host Transport layer Idle state, 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 and the Control Register FIS transmission initiated. 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.

For host adapters capable of asynchronous FIS reception, the HTI1:HT_HostIdle state is defined as follows:

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 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 can 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.		

For host adapters capable of asynchronous FIS reception, the Command Register FIS transmission retry state, Control Register FIS transmission retry state, DMA Setup FIS transmission retry state, and BIST FIS transmission retry state are defined as follows. Note that in response to detection of an error condition for these FISes, the associated FIS remains pending for transmission upon return to the HT_HostIdle state.

HTCM2: HT_TransStatus	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		

HTCR2: HT_TransStatus	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		

HTPDMASTUP1: HT_TransStatus	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		

HTXBIST2: HT_TransBISTStatus	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		

5.4. 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 37 illustrates these concepts (the figure is intended as illustrative and does not exclude other possible host controller implementations).

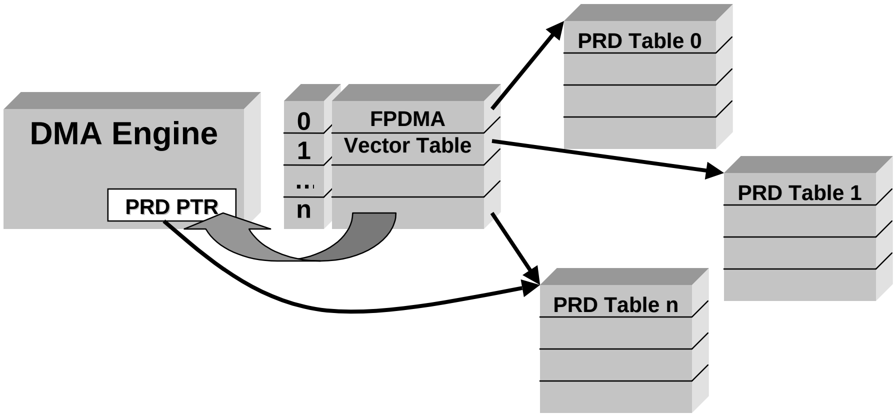


Figure 37 Example DMA engine indirection for FPDMA support

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 4.5.4 for information on non-zero buffer offsets.

6. Subsystem

6.1. Enclosure Services/Management

6.1.1. Goals, Objectives, & Constraints

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. In order to accommodate management of Serial ATA 1.0a devices, no modification to the facilities defined in the Serial ATA 1.0a specification is required.

In this specification, the host (the Serial ATA RAID controller or HBA) uses either the SAF-TE or SES command protocol to communicate control/status with the storage enclosure processor (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 hosts (or host controllers) that are capable of 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 standard ATA Command Block Register set (or Serial ATA Register FIS) using the READ SEP/WRITE SEP commands associated with this interface. The standard 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.

6.1.2. Topology

The enclosure services support mechanisms must support the configurations and topologies expected for storage subsystems that require such enclosure services (such as external storage enclosures). Figure 38 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 Serial ATA router/switch/mux as described separately), 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. For a Serial ATA router/switch/mux, the host interface would be Serial ATA.

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 Serial ATA fan-out component, 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.

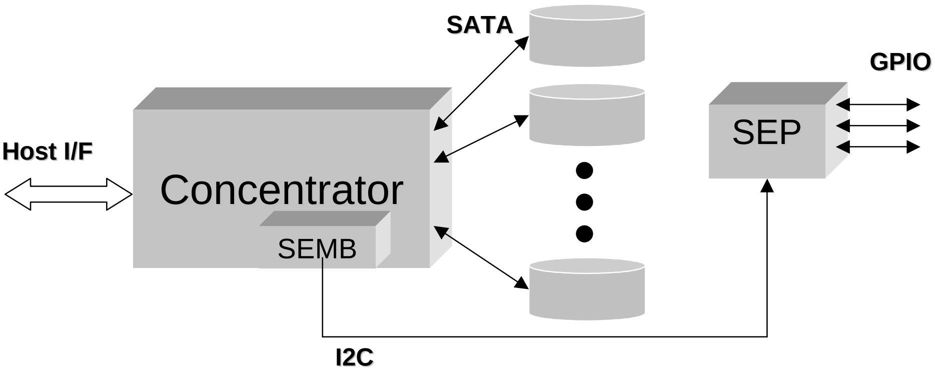


Figure 38 Generic enclosure services topology

6.1.2.1. Definition Configuration

The definition configuration is distilled from the generic topology in Figure 38 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 fan-out component). 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.

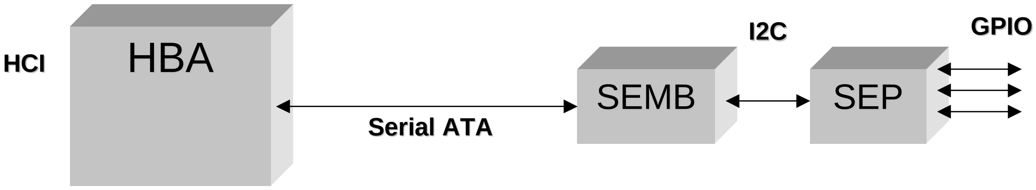


Figure 39 Simplified view of generic topology

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 can be further reduced to the definition configuration in Figure 40 that presents the enclosure services facility as an ATA device with a Command Block Register interface.

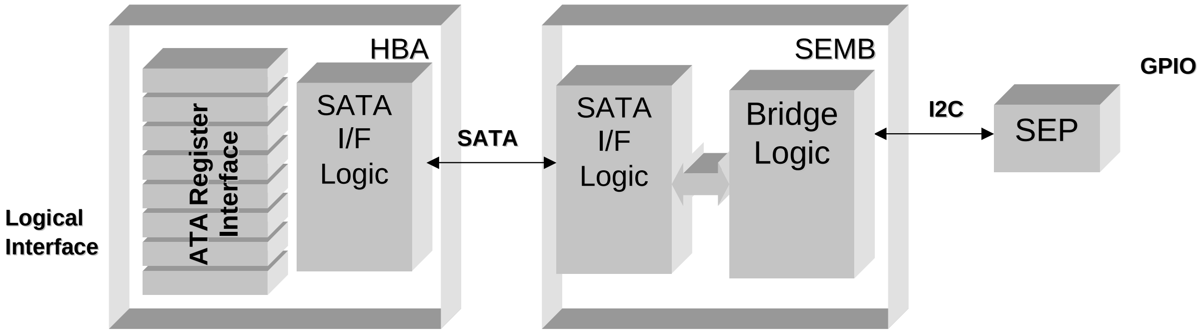


Figure 40 Enclosure services definition configuration

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 6.1.4.2. 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).

6.1.3. 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.

6.1.4. 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 1.2 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 register. Section 6.1.4.3 and 6.1.4.4 define the host-to-SEP and SEP-to-host command protocols.

6.1.4.1. Discovery

Following a hardware reset or software reset, the SEMB shall place the unique SEMB-specific signature as identified in Figure 41 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, hard reset, 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							
Sector Count	01h							
Sector Count (exp)	00h							
Sector Number	01h							
Sector Number (exp)	00h							
Cylinder Low	3Ch							
Cylinder Low (exp)	00h							
Cylinder High	C3h							
Cylinder High (exp)	00h							
Device/Head	00h							
Status	BSY	DRDY	DF	DSC	DRQ	0	0	ERR

Status =50h

Figure 41 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 42 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 hardware reset 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							
Sector Count	FFh							
Sector Count (exp)	FFh							
Sector Number	FFh							
Sector Number (exp)	FFh							
Cylinder Low	FFh							
Cylinder Low (exp)	FFh							
Cylinder High	FFh							
Cylinder High (exp)	FFh							
Device/Head	FFh							
Status	0	1	1	1	1	1	1	1

Status=7Fh

Figure 42 Register Signature for Absent Enclosure Processor

In addition to the device reset signature, SEPs shall support the IDENTIFY SEP command described in section 6.1.5.1 in order to allow host software to determine its capabilities and to identify whether it supports the SAF-TE or SES command set.

6.1.4.2. Logical Command Block Registers to I^{2}C 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.

6.1.4.2.1. Command Delivery

Commands are delivered to the SEP by the SEMB via the Command Block Register interface as a result of the Command or 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^{2}C 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^{2}C interconnect, the SEP command fields are packaged as an I^{2}C frame and forwarded to the SEP via the I^{2}C 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^{2}C packet based on the contents of the Command Block Registers or Register FIS. The mapping of Command Block Registers to transmitted I^{2}C 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^{2}C packet shall be as indicated in Figure 43.

Register	7	6	5	4	3	2	1	0
Features	SEP_CMD							
Features (exp)	Reserved							
Sector Count	LEN							
Sector Count (exp)	Reserved							
Sector Number	CMD_TYPE							
Sector Number (exp)	Reserved							
Cylinder Low	Reserved							
Cylinder Low (exp)	Reserved							
Cylinder High	Reserved							
Cylinder High (exp)	Reserved							
Device/Head	0	1	0	0	Reserved			
Command	SEP_ATTN (67h)							

Figure 43 Command Block Register Fields Used in Enclosure Processor Communications

SEP_CMD The SAF-TE or SES command code to be issued. See the SAF-TE and SES references for the command codes and their functions.

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 an SES command code and whether the data transfer protocol 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 (SAF-TE ReadBuffer)
80h	SAF-TE command code with host-to-SEP data transfer (SAF-TE WriteBuffer)
02h	SES command code with SEP-to-host data transfer (SES ReceiveDiagnostic)
82h	SES command code with host-to-SEP data transfer (SES SendDiagnostic)

All other values reserved

The resulting I^{2}C frame for delivering a command is as indicated in Figure 44.

S	SEP ADDRESS	R/W (0)	A	CMD_TYPE	A	CHK SUM	A	SEMB ADDRESS	A	SEQ (0)	A	SEP_CMD	A	..rest of transaction…

▨ **From Master to Slave** □ **From Slave to Master**

Figure 44 I^{2}C 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^{2}C traffic for that illegal command. The SEMB shall support Control register writes where the state of the SRST bit changes, but shall take no action in response to 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 6.1.4.2.2.

6.1.4.2.2. Status Mechanism

The SEMB indicates command completion to the host by clearing the BSY bit 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^{2}C 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 BSY 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.

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.

6.1.4.3. 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 the Features register as illustrated in the Command Block Registers image of Figure 45. 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	SEP_CMD							
Features (exp)	Reserved							
Sector Count	LEN							
Sector Count (exp)	Reserved							
Sector Number	CMD_TYPE							
Sector Number (exp)	Reserved							
Cylinder Low	Reserved							
Cylinder Low (exp)	Reserved							
Cylinder High	Reserved							
Cylinder High (exp)	Reserved							
Device/Head	Reserved			0	Reserved			
Command	SEP_ATTN (67h)							

Figure 45 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^{2}C interface, the transmitted I^{2}C packets shall be of the form indicated in Figure 46.

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 46 I^{2}C Transactions Corresponding to a WRITE SEP Command

The I^{2}C transport for Serial ATA enclosure service traffic shall be Master to Slave (IPMB) compliant. See the IPMB and I^{2}C 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.

6.1.4.4. 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 the Features register as illustrated in the Command Block Registers image of Figure 47. 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	SEP_CMD							
Features (exp)	Reserved							
Sector Count	LEN							
Sector Count (exp)	Reserved							
Sector Number	CMD_TYPE							
Sector Number (exp)	Reserved							
Cylinder Low	Reserved							
Cylinder Low (exp)	Reserved							
Cylinder High	Reserved							
Cylinder High (exp)	Reserved							
Device/Head	Reserved			0	Reserved			
Command	SEP_ATTN (67h)							

Figure 47 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^{2}C interface, the transmitted I^{2}C packets shall be of the form indicated in Figure 48.

SEMB (master) to SEP (slave) transfer – transfer the command

S	SEP ADDRESS	R/W (0)	A	CMD TYPE	A	CHK SUM	A	SEMB ADDRESS	A	SEQ (0)	A	SEP CMD	A	CHK SUM	A	P

SEP (master) to SEMB (slave) transfer – transfers both the data and 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	Data[0]	A	Data[1]	A	…	Data[LEN*4-1]	A	CHKSUM	A	P

▨ **From Master to Slave** □ **From Slave to Master**

Figure 48 I^{2}C Transactions Corresponding to READ SEP Command

The host shall use the DMA protocol for transferring data from the SEMB to the host.

6.1.5. SES and SAF-TE Extensions

6.1.5.1. 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 0x00 or 0x02 depending on the command protocol being used (see section 6.1.4.2.1). 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.

6.1.5.1.1. IDENTIFY SEP Data Structure

Figure 49 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 can be controlled (i.e. the transfer time can 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.

Identify Data Page

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 49 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 shall use one of the 8-byte worldwide name formats defined by FC-PH. 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, will uniquely identify any SEP unit from any manufacturer. The worldwide name should have an NAA field of 0x05, 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 ANSI X3.301). 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 ANSI X3.301). 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 ANSI X3.301). 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 will be 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.

6.1.5.2. LED Activity 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 6.1.5.2.1 and 6.1.5.2.2 to accommodate a means by which an enclosure processor can be used to provide operator activity indication.

6.1.5.2.1. 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 drive slot on the channel and the associated data structure is defined in Figure 50.

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 50 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.

6.1.5.2.2. 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 51. The data structure is modified with the addition of the drive 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 51 SES Device Element data structure

6.1.5.2.3. Activity Indication Behavior and Operation

The host writes a 1 to the DR_ACT bit 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 will stop blinking).

6.1.5.3. 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 52 illustrates a configuration where there is no direct correspondence between host connection and enclosure device slot.

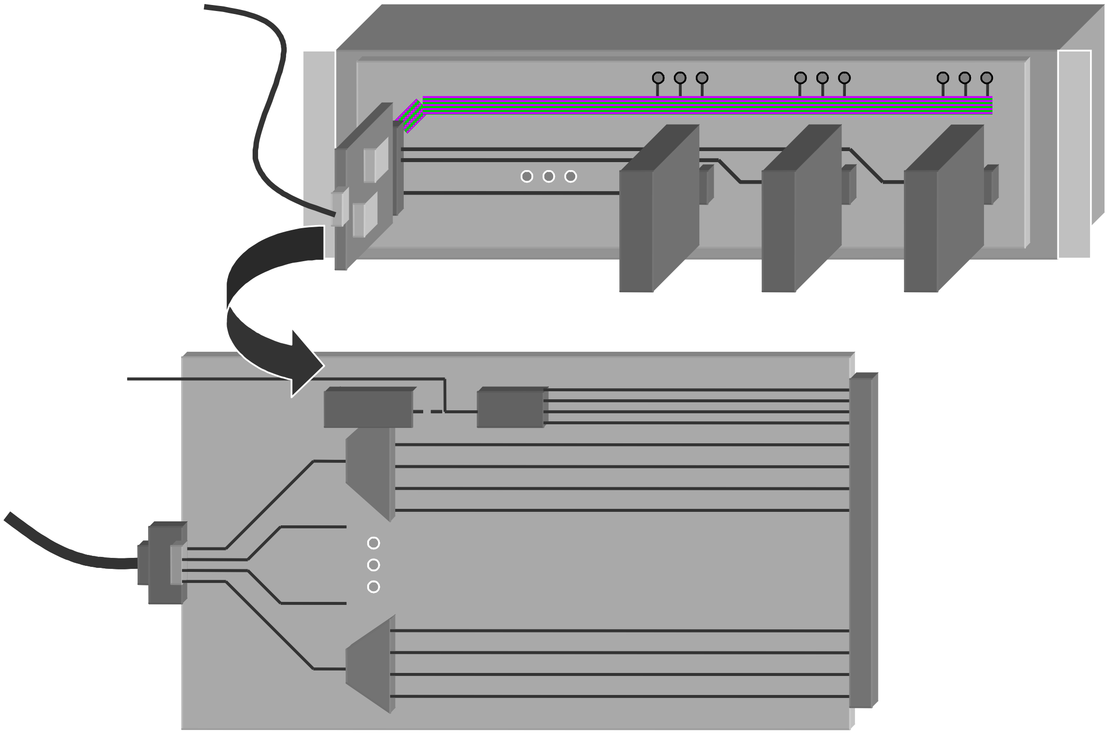


Figure 52 Example Subsystem

6.1.5.3.1. SAF-TE Correspondence Definition (Optional)

Figure 53 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 0xFF; a SCSI ID of 0xFF 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 53 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 52, the SCSI ID value corresponding to Slot 0 in the enclosure would be 0x33, while the SCSI ID value corresponding to Slot 3 in the enclosure would be 0x03.

6.1.5.3.2. SES Correspondence Definition (Optional)

Figure 54 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 54. The enclosure status page can be read with the Receive Diagnostic Results command.

Bits **Field**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Slot Address	**Port**				**Channel**			

Figure 54 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 52, the Slot Address value corresponding to Slot 0 in the enclosure would be 0x33, while the Slot Address value corresponding to Slot 3 in the enclosure would be 0x03.

6.1.6. 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 6.1.6.1 defines the connector and cable interconnect between the enclosure/backplane and the storage subsystem controller.

6.1.6.1. I^{2}C 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^{2}C 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^{2}C interface originating on the PCI controller can be connected to the backplane that interfaces with the various storage devices and operator indicators (LEDs).

Products that utilize I^{2}C 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.

6.1.6.2. SEP Discovery and Enumeration

For implementations where the SEP is an I^{2}C device and is not provided as part of a complete storage solution (i.e. will communicate with a SEMB provided by another vendor), the SEP should have a dedicated I^{2}C interface with the SEMB and should not share the I^{2}C 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^{2}C bus attached to a SEMB in a storage subsystem shall be assigned I^{2}C address 0xC0 (note that the LSB of the I^{2}C address field is the R/W bit effectively resulting in the I2C 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^{2}C interface in a storage subsystem shall be assigned consecutively higher I^{2}C addresses.

6.2. Staggered Spin-up

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) can sequence disk drive 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 drive and completing drive 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 drive 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 must 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.

6.3. HDD Activity Indication

Serial ATA devices do not provide support for an activity LED indication. Operator notification/indication of storage device status and activity may be driven by the host through the enclosure services facilities defined in Section 6.1 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 6.1.5.2.1 for methods to support a SEP controlling HDD Activity Indication. As this reference suggests, these implementations may be vender specific.

6.3.1. 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 55.

```
//          POR - Power On Reset
//          HRESET - Hardware RESET
RESET = POR  || HRESET;


if ((BSY || SActive) && DEVICE_TYPE != ATAPI)        // How !ATAPI determined is implementation
            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 55 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 0xA0 and 0xA1), 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.

6.3.1.1. Desktop HDD Activity Signal Electrical Requirements

For implementations that provide an activity signal in accordance with section 6.3.1, 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.

6.3.2. 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 drive activity LEDs without need for changes to the Serial ATA specification. It is likely that the methods outlined in this section will become obsolete as enclosure management facilities for Serial ATA mature and become readily available.

Figure 56 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 57 is identical in configuration to that shown in Figure 56, except that it shows the use of a ribbon cable for ease of assembly.

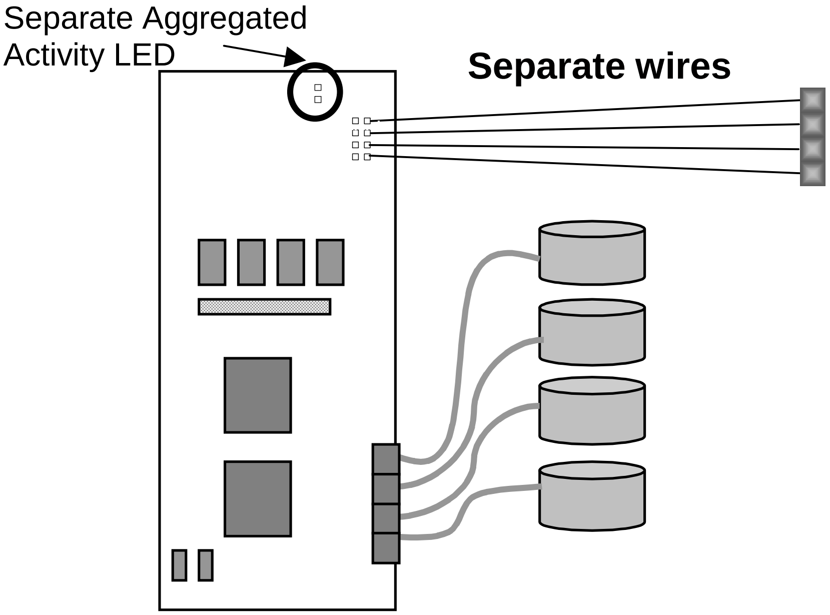


Figure 56 Device Activity LEDs w/ Separate Wires

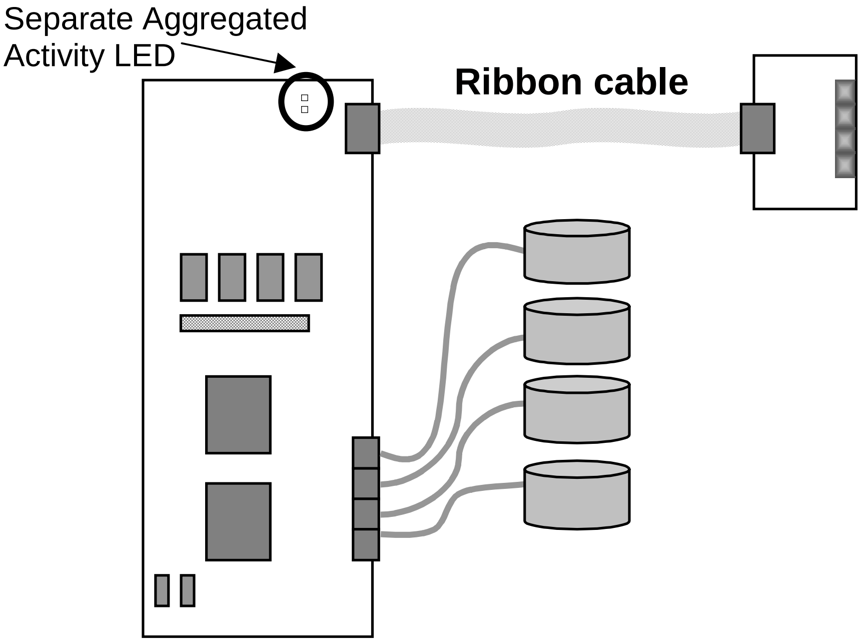


Figure 57 Device Activity LEDs w/ Ribbon Cable

Note also the activity outputs have been designed to support wired-OR configurations. They can then support a configuration where all the outputs can be connected together for the purpose of generating an aggregate activity indication.

Taking advantage of the wired-OR functionality, Figure 58 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.

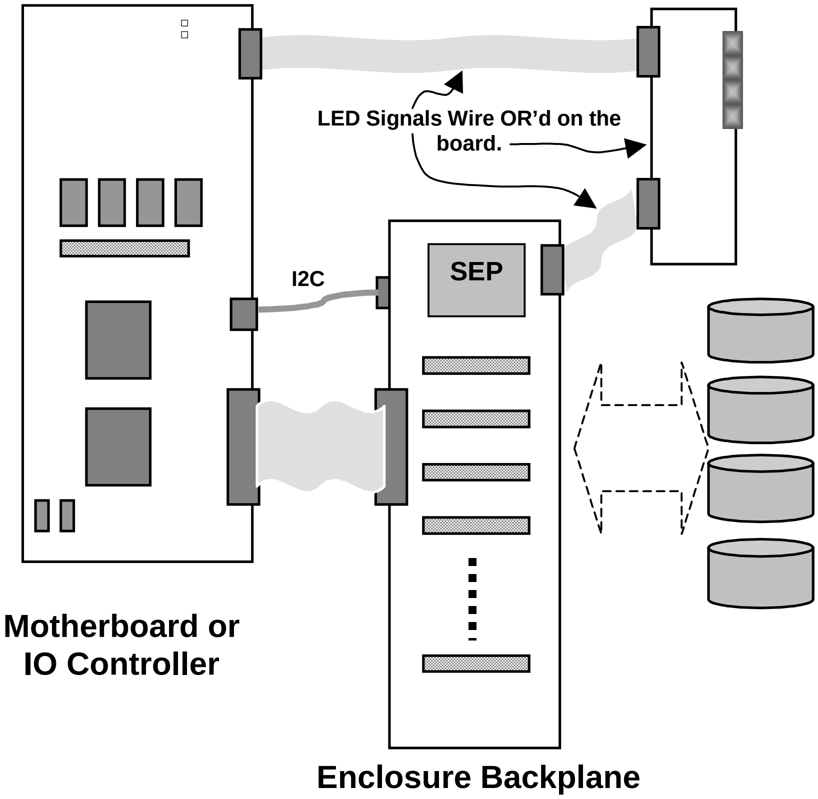

LED Signals Wire OR'd on the board.
I2C
SEP
Motherboard or IO Controller
Enclosure Backplane

Figure 58 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 fan-out component, 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 fan-out component 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 asserted if the BSY bit is asserted 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.

6.4. Hot-Plug and Presence Detect

The Serial ATA 1.0a specification provides the capabilities necessary to deliver a hot-plug implementation but provides no additional guidelines for delivering a complete solution. For a storage subsystem, a hot-plug capability is required as well as the ability to seamlessly handle presence detection in those cases where the storage subsystem can remove power to individual receptacles.

6.4.1. 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. In the power segment of the Serial ATA connector, devices shall connect together the following sets of pins regardless of whether the corresponding supply voltages are actually utilized by the device:

P1, P2, and P3	3.3V power delivery and precharge
P7, P8, and P9	5V power delivery and precharge
P13, P14, and P15	12V power delivery and precharge

6.4.2. 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 59 illustrates one hot-plug power delivery scheme that utilizes the precharge connector feature.

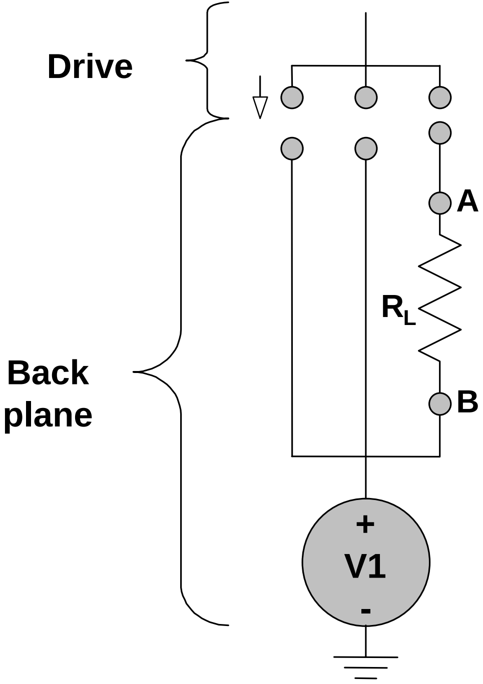


Figure 59 Typical precharge configuration

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 20uF. For illustrative purposes, these values will be presumed in an example scenario for estimating precharge resistor dimensioning. 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 20uF, the resultant precharge resistor size is approximately:

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

Because the Serial ATA power conductors can support currents up to 1.5A, the computed resistor size can be substantially reduced without adverse consequence in order to reduce the sensitivity to the device's actual effective capacitance. For the 12V 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.

6.4.3. Presence Detection (Informative)

The Serial ATA 1.0a solution for presence detection relies on the device signaling the host using the out of band (OOB) sequence to signal 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 6.4.2. 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 will be R_L with no device present and will be effectively zero with a device inserted in the receptacle.

Figure 60 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.

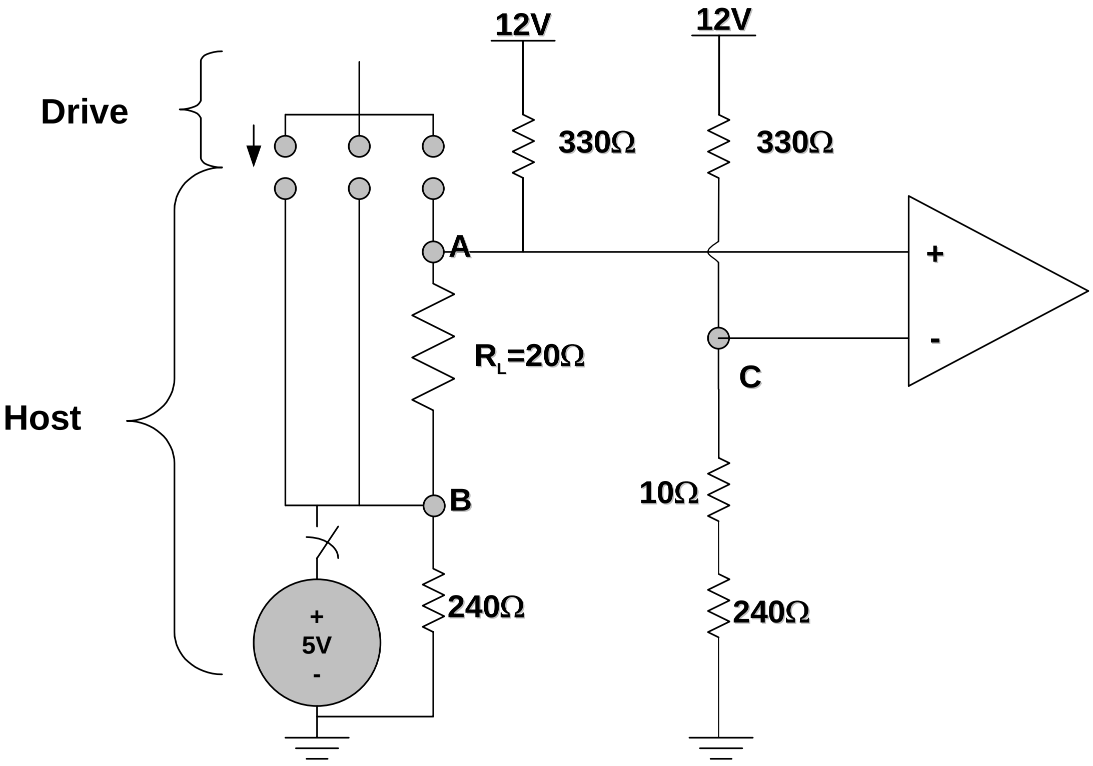


Figure 60 Example presence detection implementation

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

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

Figure 61 Comparator voltages for alternate example presence detection circuit

Appendix A. Backplane Interconnect Losses Estimations (Informative)

A.1 Methodology and Assumptions

The backplane interconnect loss estimations presented here deals only with signal amplitude and does not include other signal integrity considerations such as skew, jitter, etc.

A.1.1 Considered Losses

The following effects are modeled in detail by this analysis:

- Resistive Losses
- Skin Effect
- Dielectric absorption

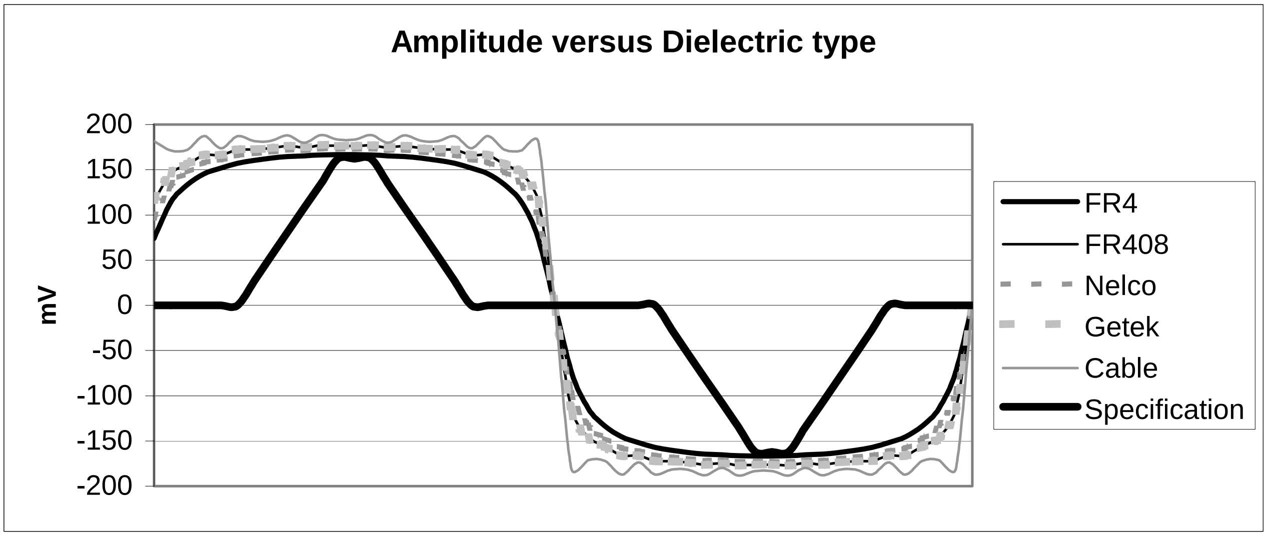
The following effects are not modeled in detail, but are instead accounted for as a percent of signal budget:

- Impedance mismatch
- Cross-talk
- Reflective losses

The percentage figure used for 18-inch trace environments is 27%. Within this tolerance is an allowance for +/- 5% impedance mismatch across three medium transitions and termination.

A.1.2 Dielectric materials

FR4 is the most commonly used printed circuit board material. A comparison with other lower loss tangent materials shows very little difference between them at a 1.5 Gbps signaling rate (see Figure 62). Therefore, for the purposes of this document FR4 dielectric will be assumed.

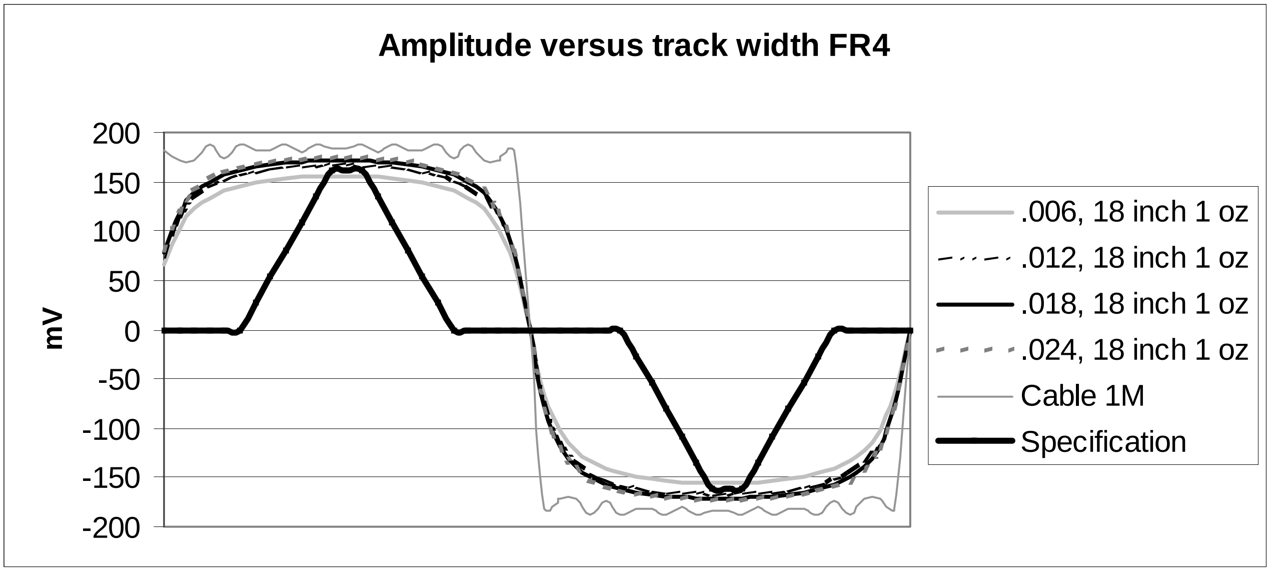


(1.5 Gbps, Resistive, Dielectric and Skin effect losses)

Figure 62: Signal loss versus PCB dielectric material comparison for an 18-inch 12mil trace

A.1.3 Trace widths

A commonly used backplane trace dimension is 0.012 inch wide 1oz copper. A comparison of different trace dimensions shows a significant reduction in signal transmission for a 0.006 inch trace width, but very similar characteristics for 0.012 inch and larger (see Figure 63). Therefore, for the purposes of this document 0.012 inch 1oz trace will be assumed.



(1.5 Gbps, Resistive, Dielectric and Skin effect losses)

Figure 63: Signal loss versus trace width comparison

A.1.4 Electrical Specifications: Serial ATA 1.0a specification.

A storage enclosure system must be compatible with devices (disk drives) that meet the Serial ATA 1.0a specification. These devices will have the following signal characteristics at the device connector.

General electrical specifications (Table 11 of the Serial ATA specification Page 76)

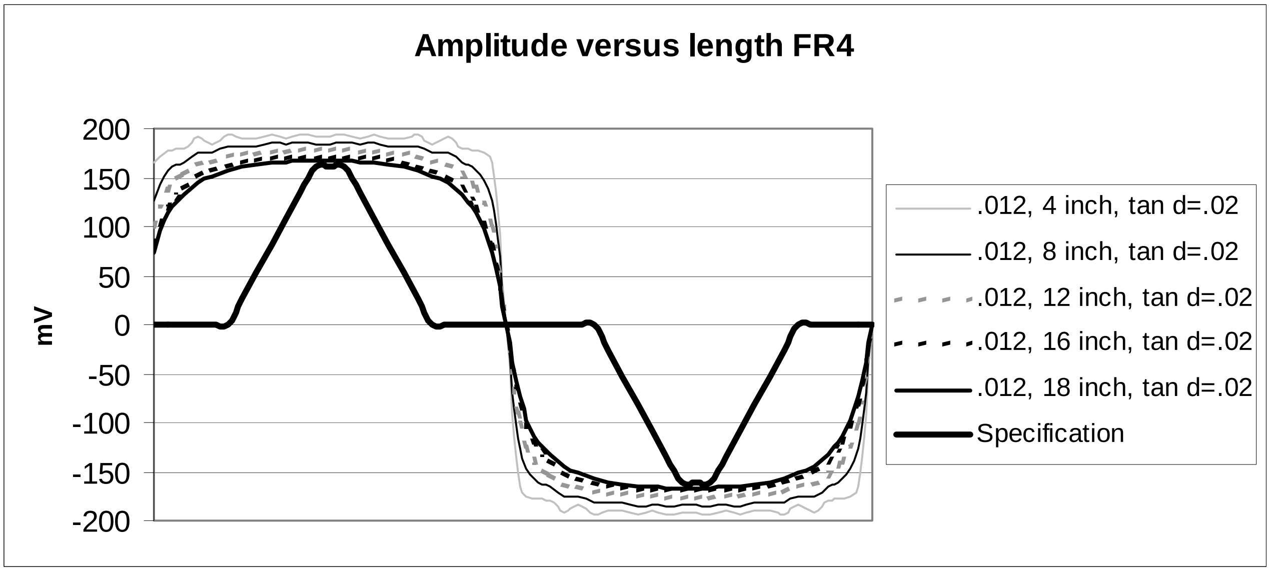
	Nom	Min	Max	Units	Comments
Vdiff,tx	500	400	600	mVp-p	+/- 250 mV differential nominal. Measured at Serial ATA connector on transmit side
Vdiff,rx	400	325	600	mVp-p	+/- 200 mV differential nominal. Measured at Serial ATA connector on receive side
T,UI		666.43	670.12	ps	Operating Data period
trise	0.3	0.2	0.41	UI	20%-80% at transmitter
tfall	0.3	0.2	0.41	UI	20%-80% at transmitter

According to the jitter requirements, the allowed total jitter is 0.355UI (equating to a 430ps eye width) at the transmitter connector and 0.43UI (equating to a 380ps eye width) at the receiver PCB connector.

A.2 Signal Transmission deterioration over a backplane

A.2.1 Receiver Sensitivity

Figure 64 shows the receive amplitude of a 1.5Gbps signal from a 400mV differential transmitter down various lengths of 12 mil, 1 oz copper etch on FR4 material.



(1.5 Gbps, Resistive, Dielectric and Skin effect losses)

Figure 64: Effect of length on 1.5Gbps Serial ATA signal eye

Figure 65 shows the received signal level versus trace length—without an additional margin for connector losses, reflections and manufacturing tolerances—from a SATA 1.0a minimum transmit level device. In order to detect this signal, the host-side receiver sensitivity needs to be at least 240mV differential.

In addition to data signaling levels, Serial ATA uses a lower level of signaling to initialize the Phy and control sleep modes. These squelch levels are specified in the Serial ATA 1.0a Specification Table 11 Electrical Specification define the maximum signal level that can be rejected as *no activity* on the wire. This therefore sets a limit on the sensitivity level that can be defined for the active signal receiver. Serial ATA 1.0a specifies that any signal below 200 mVp-p be rejected as inactivity. This is a close margin to the recommended requirement that 240mVp-p be accepted as an active signal.

	Nominal	Minimum	Maximum		
Squelch detector Threshold (Serial ATA 1.0a Specification)	100mV	50mV	200mV	mVp-p	Minimum differential Signal amplitude

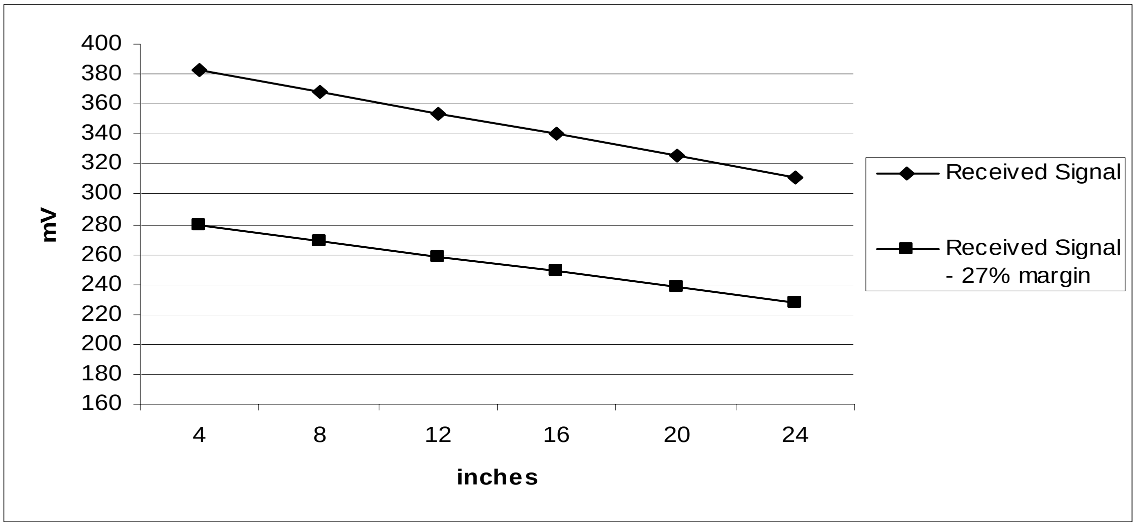


Figure 65: Expected minimum host-side controller receive signal level versus trace length

A.2.2 Transmit Levels

When a host controller device transmits to a Serial ATA 1.0a device over some length of FR4 PCB, the Serial ATA 1.0a specification requires that the signal levels at the device connector must be between 325mV and 600mV differential. As seen in Figure 66, the host-side transmitter will need to have a minimum output signal level of 500mV differential in order to achieve the required 325mV differential receive levels for 1.5Gbps at the device.

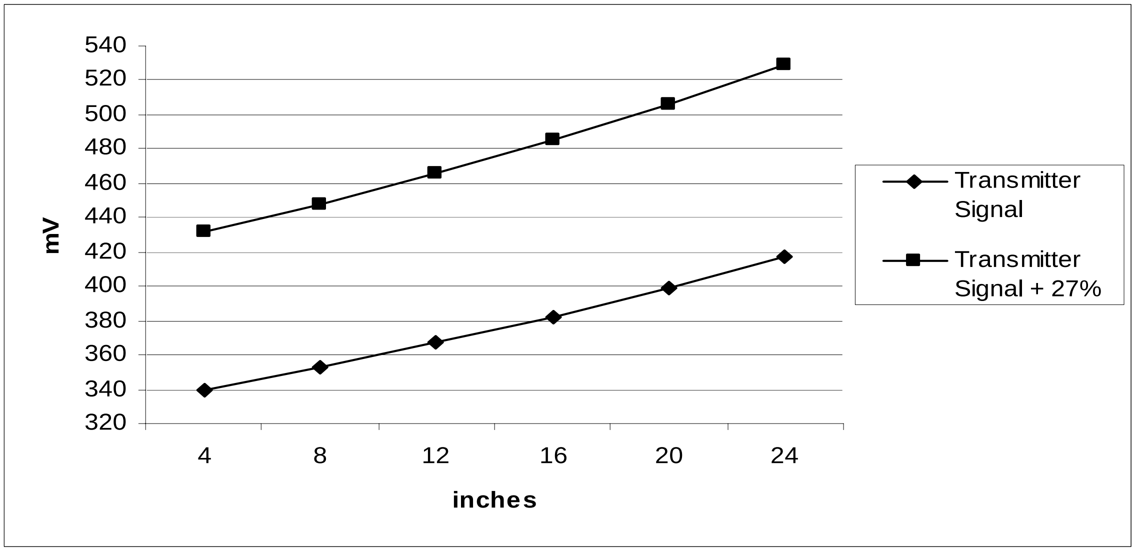


Figure 66: Minimum host-side transmit level versus trace length to meet minimum SATA 1.0a receive level

Appendix B. Sample Native Command Queuing Transaction Sequences (Informative)

B.1 Sample Sequences (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.1.1 Queued Commands with Out of Order Completion (Informative)

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 de-asserts BSY by transmitting a Register FIS to the host.
Host issues Read Command Tag=5 (if BSY not yet 0, host must 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 de-asserts BSY.
		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 I bit set 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 I bit set. 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 I bit set 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 I bit set. 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.1.2 Interrupt Aggregation (Informative)

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 de-asserts BSY by transmitting a Register FIS to the host.
Host issues Read Command Tag=5 (if BSY not yet 0, host must 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 de-asserts BSY.
		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 I bit set 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 I bit set. 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		
		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 rate for this example.		
		Device sends Set Device Bits FIS with I bit set 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 I bit set. 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.1.1		
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