

SATA Storage Architecture

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute



Legal Stuff

The material in this presentation is copyrighted and is intended for the use of students who have taken this MindShare course. It must not be copied or distributed without specific permission from MindShare, Inc.

Table of Contents

Part I: SATA Overview

- Evolution of Parallel ATA
- Motivation for SATA
- SATA Overview
- Intro to FIS Transfers

Part II: FIS Transmission Protocols

- FIS Types & Formats
- Transport & Link Protocol Details
- FIS Retry (Transport Layer)
- Data Flow Control
- Physical Layer Functions
- Error Detection & Handling

Part III: Cmd & Control Protocols

- The Command Protocol
- Control Protocol

Part IV: SATA II

- SATA II Features
- Native Command Queuing
- Port Multipliers
- Port Selectors
- Enclosure Services

Part V: Physical Layer Details

- SATA Initialization
- PHY Electrical Characteristics
- Cables/Connectors
- Hot Plug
- Link Power Management
- BIST

Appendix: AHCI

Part 1

SATA Overview



Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

Evolution of Parallel ATA

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute



Introduction

- Serial ATA 1.0 (SATA) provides a high-speed (1.5 Gb/s) serial connection between the HBA and Mass Storage Devices
- Intended as a replacement for Parallel ATA (PATA)
- Provides software compatibility to the PATA environment

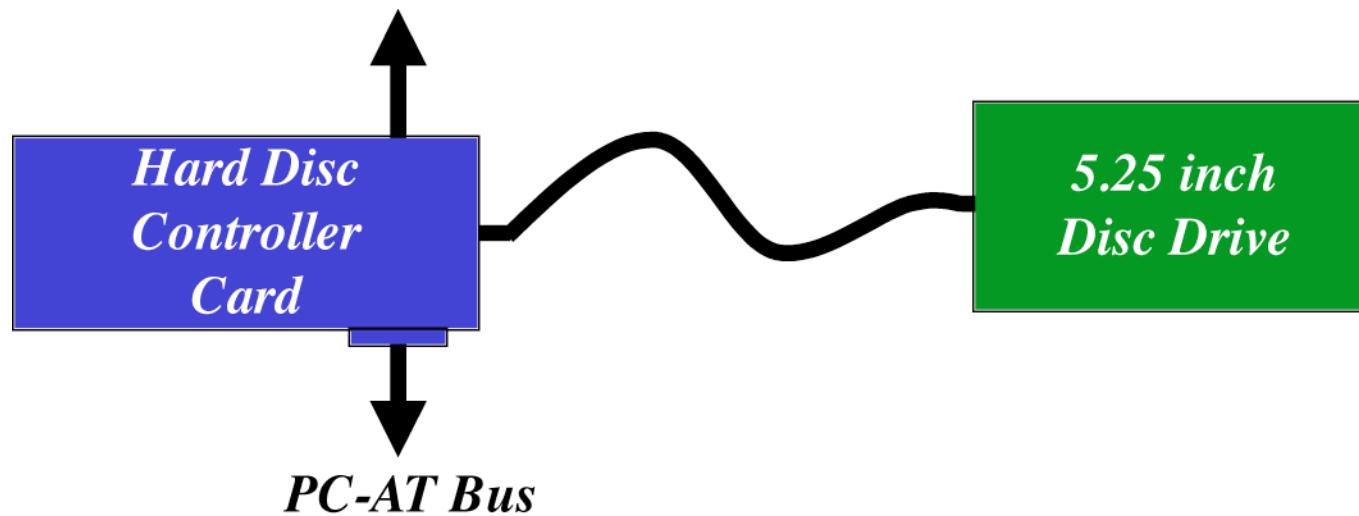
Why Transition to Serial Interface?

- Cost reduction through reduced pin count
- Lower voltages for smaller silicon sizes
- Improved speed between the drive and Host Adapter
- Enhanced reliability
- Improved cables/connector
- Aggressive positioning of ATA for the server environment

Details regarding these issues are covered in
the next section

Origins of ATA

IBM PCs with hard drives included a Hard Disc Controller Card connected to a 5.25 inch Disc Drive mounted in a drive bay.



Origins of ATA, continued

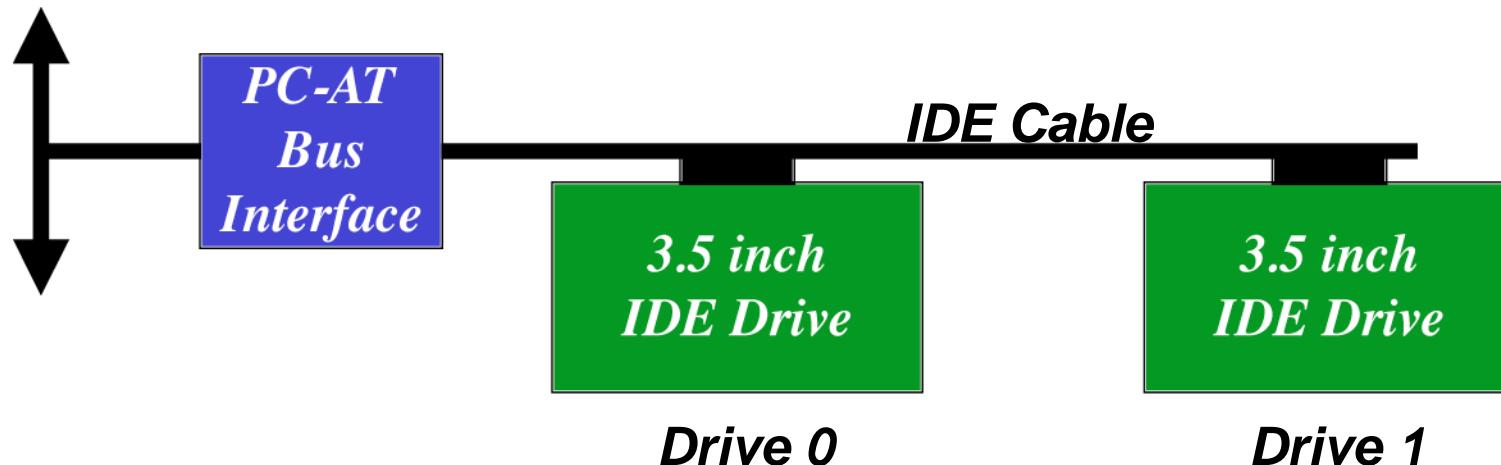
Compaq Computer sought to eliminate expansion slots within their portable computers. The result -- an integrated drive that combined the controller card and Disc Drive. This proprietary solution included a simple bus interface to which the integrated drive cable was attached.



Origins of ATA, continued

Soon, other manufacturer's implemented integrated drive solutions. These drives were typically known as IDE drives but were slightly different from the Compaq's proprietary implementation. Still no standard definition existed and compatibility problems were typical.

Note that the integrated drives required configuration via jumpers or switches to select the device address.



The ATA Standard

Eventually, a group was formed to define a standard for integrated drives. Since these drives were used in the PC-AT (ISA) systems, the specification labeled them as ATA (AT Attachment) drives.

The first ATA specification was finally published as:
ANSI X3.221-1994

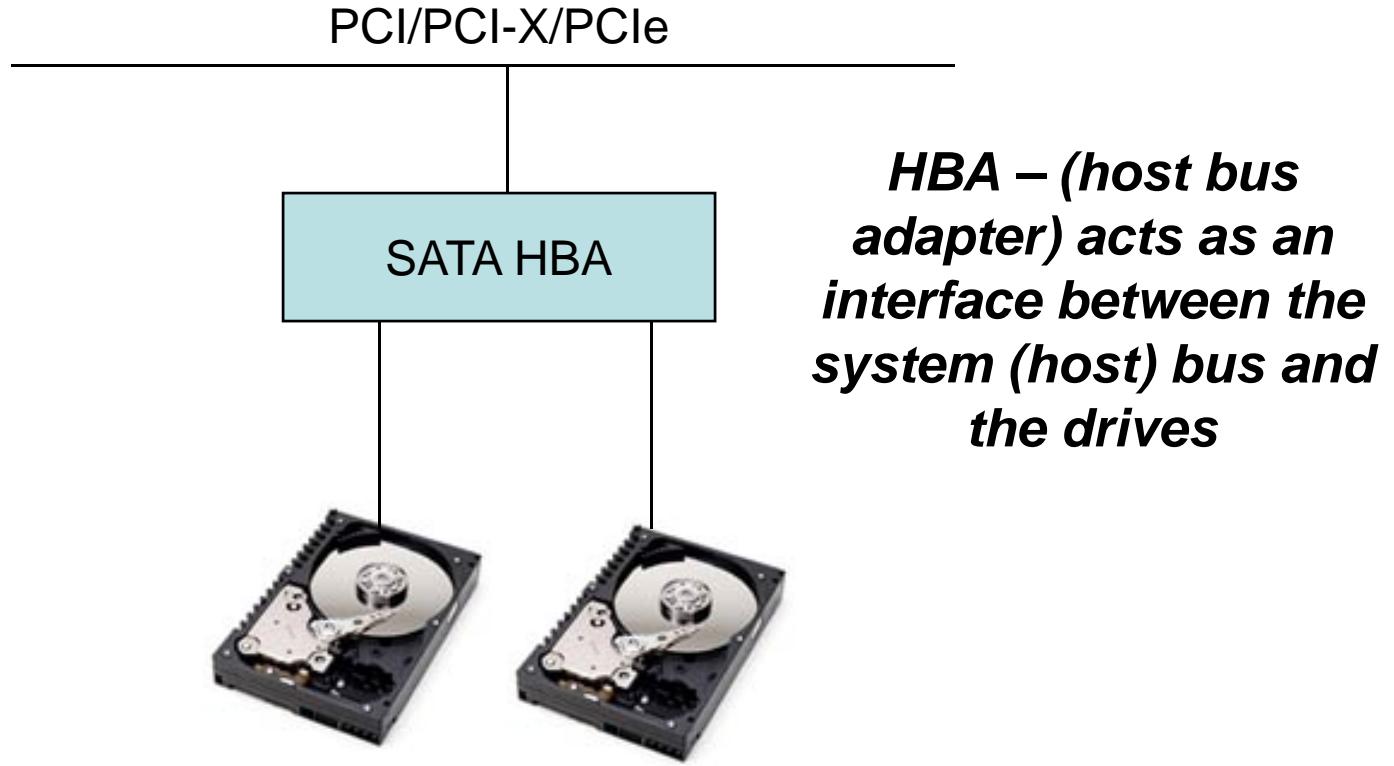
Today the T13 working committee defines the direction of ATA and is chartered by:

INCITS - Technical Committee for the InterNational Committee on Information Technology Standards

ATA Releases

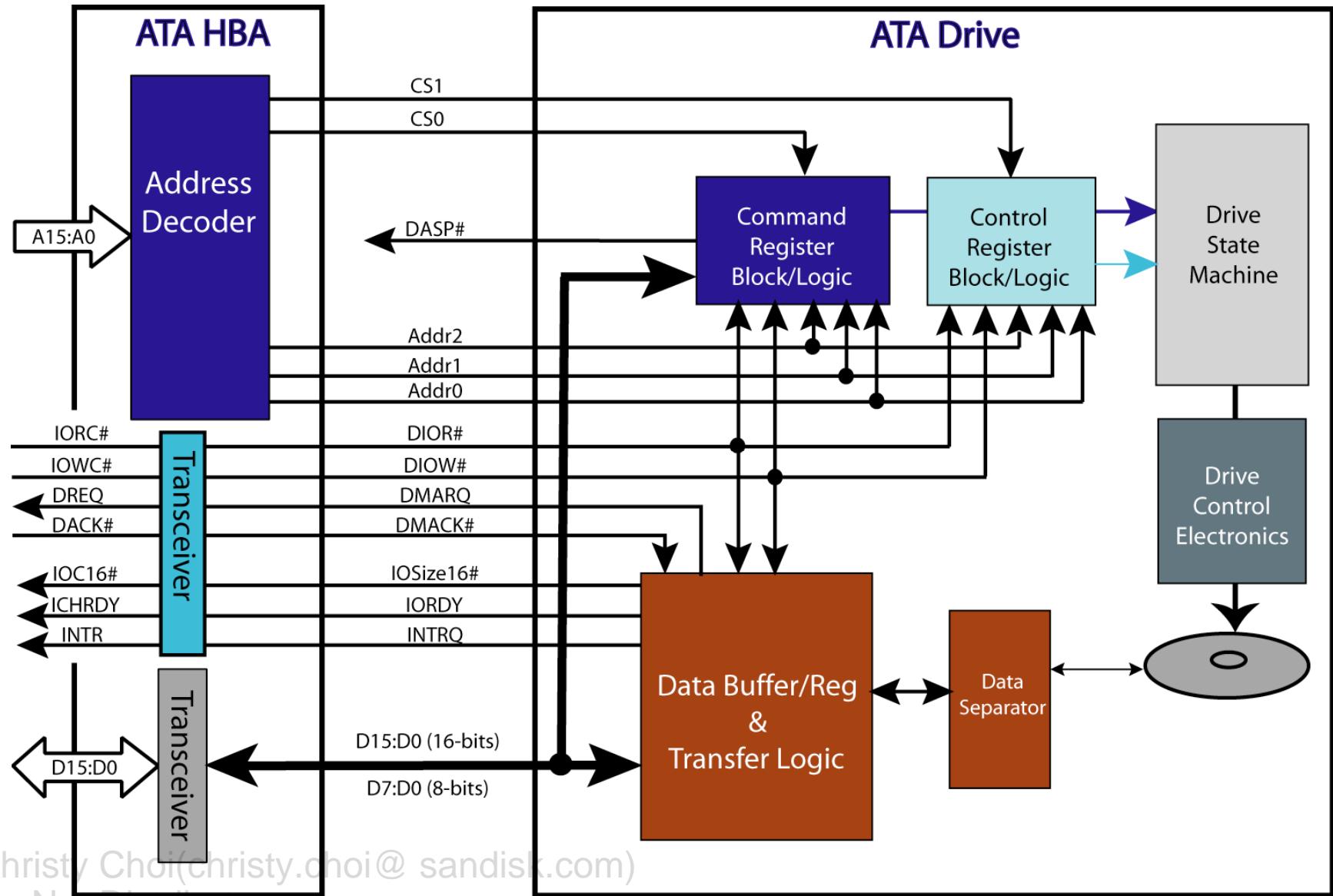
Standard	Year	Speed	Transfer Modes
ATA	1986		Pre-standard
ATA-2	1994	5MB/s	PIO modes 0-2 Multiword DMA mode 0
ATA-3	1996	16MB/s	PIO modes 3-4
ATA-4	1997		Multiword MDA modes 1-2
ATA-5	1998	33MB/s	Ultra DMA modes 0-2
ATA-6	2000	66MB/s	Ultra DMA modes 3-4
ATA-7	2002	100MB/s	Ultra DMA mode 5
ATA-8	2003	133MB/s	Ultra DMA mode 6

SATA Topology (Legacy Type Implementation)



Parallel ATA Block Diagram

IBM PC-AT Bus (ISA)



Parallel ATA Pinout

Vendor-Specific	A	B	Vendor-Specific
Vendor-Specific	C	D	Vendor-Specific
N.C. (Coding Pin)	E	F	N.C. (Coding Pin)
RESET#	1	2	Grnd
Data7	3	4	Data8
Data6	5	6	Data9
Data5	7	8	Data10
Data4	9	10	Data11
Data3	11	12	Data12
Data2	13	14	Data13
Data1	15	16	Data14
Data0	17	18	Data15
Grnd	19	20	N.C.
DMARQ	21	22	Grnd
IOW#	23	24	Grnd
IOR#	25	26	Grnd
IORDY	27	28	SPSync/CSEL
DMACK#	29	30	Grnd
INTRQ	31	32	IOCS16
ADDR1	33	34	PDIAG
ADDR0	35	36	ADDR2
CS0#	37	38	CS1#
DASP	39	40	Grnd
+5V (Logic)	41	42	+5V (Motor)
Grnd	43	44	Type#

**See Table 1-1 for
signal descriptions
(page 14-15)**

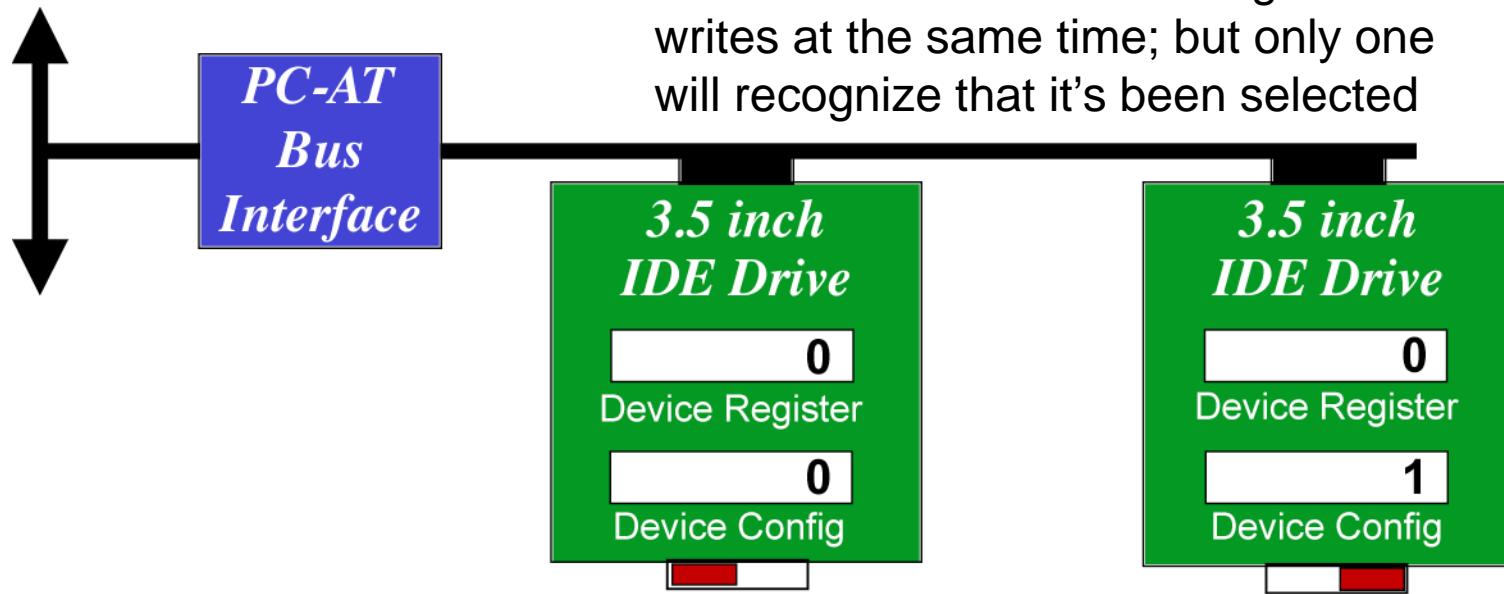
Legacy Programming Interface

[ATA Registers (Task Files)]

Cmd Reg	Reads		Writes		Notes
Address	7	0	7	0	
01F0	Data		Data		16-bit accesses
01F1	Error		Feature		8-bit access only
01F2	Sector Count		Sector Count		8-bit access only
01F3	Sector #		Sector #		8-bit access only
01F4	Cylinder Low		Cylinder Low		8-bit access only
01F5	Cylinder High		Cylinder High		8-bit access only
01F6	Device	Head	Device	Head	8-bit access only
01F7	Status		Command		8-bit access only
Ctrl Reg					
03F6	Alternate Status		Device Control		8-bit access only

Device Register

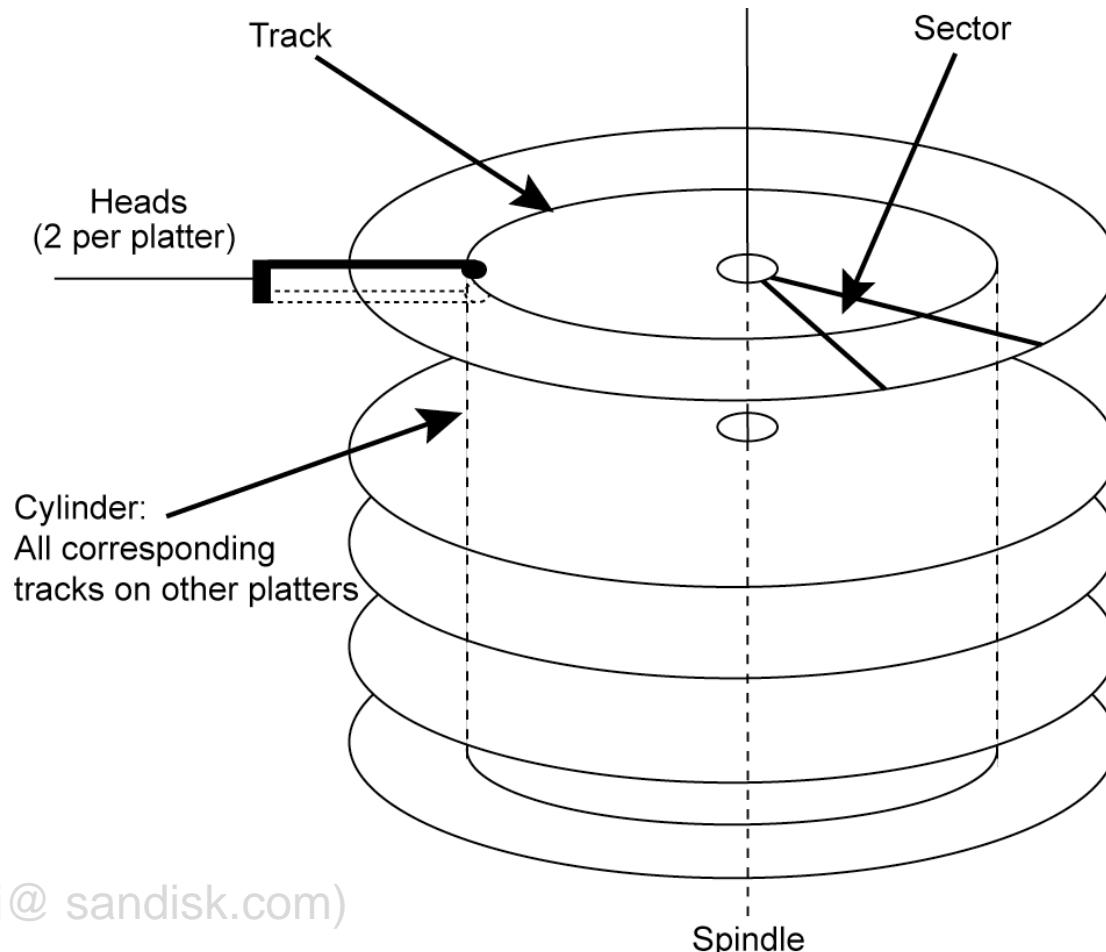
- Device is configured as drive 0 or 1 via switch
- Software selects target device by writing either 0 or 1 into the Device register. In this example device 0 is selected.



Start Sector Address (Legacy)

CHS = Cylinder/Head/Sector

Physical location on a drive is specified in three dimensions. The old style required the software to use this physical address.



Start Sector Address (Legacy)

CHS = Cylinder/Head/Sector

Cmd Reg	Reads		Writes		Notes
Address	7	0	7	0	
01F0	Data				16-bit accesses
01F1	Error		Feature		8-bit access only
01F2	Sector Count				8-bit access only
01F3	Sector Number				8-bit access only
01F4	Cylinder Low				8-bit access only
01F5	Cylinder High				8-bit access only
01F6	Device	Head	Device	Head	8-bit access only
01F7	Status		Command		8-bit access only
Ctrl Reg					
03F6	Alternate Status		Device Control		8-bit access only

Start Sector Address (LBA)

LBA = Logical Block Address (28 bits)

Cmd Reg	Reads		Writes		Notes
Address	7	0	7	0	
01F0	Data				16-bit accesses
01F1	Error		Feature		8-bit access only
01F2	Sector Count				8-bit access only
01F3	LBA Low (7:0)				8-bit access only
01F4	LBA Middle (15:8)				8-bit access only
01F5	LBA High (23:16)				8-bit access only
01F6	Device	LBA (27:24)	Device	LBA (27:24)	8-bit access only
01F7	Status		Command		8-bit access only
Ctrl Reg					
03F6	Alternate Status		Device Control		8-bit access only

Start Sector Address (LBA)

LBA = Logical Block Address (48 bits)

Cmd Reg	Reads	Writes	Notes
Address	7	0	7
01F0		Data	16-bit accesses
01F1	Error	Feature	Two 8-bit accesses
01F2		Sector Count	Two 8-bit accesses
01F3		LBA Low (31:24 then 7:0)	Two 8-bit accesses
01F4		LBA Middle (39:32 then 15:8)	Two 8-bit accesses
01F5		LBA High (47:40 then 23:16)	Two 8-bit accesses
01F6	Device	Device	8-bit access only
01F7	Status	Command	8-bit access only
Ctrl Reg			
03F6	Alternate Status	Device Control	8-bit access only

Sector Count Register

- The sector count defines the transfer size by specifying one or more sectors that are to be accessed.
- Note that some commands use the sector count register for other information (e.g., “tag” value during DMA Queued commands)

Feature Register

- The feature register is typically used to specify a variable associated with a given command.
- Example 1: Set Feature commands specify the feature to be set.
- Example 2: DMA Queued commands use the Feature register to specify the sector count.

Command Register

- Writing to the command register triggers execution of the selected command in the drive.
- Software writes to the ATA registers have the following effects:
 - The Status register's BSY (Busy) bit is set indicating that the drive is executing the command and has taken control of the registers.
 - While BSY is set, no host software writes are permitted to the Feature, Sector Count, LBA Low, LBA Mid, LBA High, or Device registers.
 - When the command completes the Status/Error registers are updated to report completion status and the BSY bit is cleared, informing host software that the registers can be written to again.

Data Register

The data register is accessed by software during execution of Programmed IO (PIO) commands to transfer data to or from the drive, usually 16 bits at a time.

Status Register

7	6	5	4	3	2	1	0
BSY	DRDY	DF/SE	#	DRQ	obsolete	obsolete	ERR/ CHK

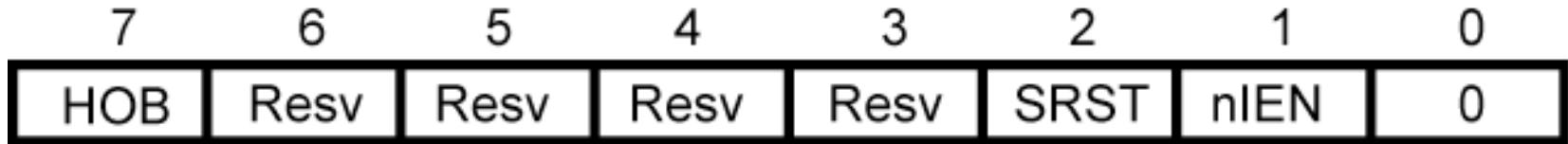
See book page 27 for definition and description of each bit.

Error Register

The ABRT (Abort) bit indicates the command could not be executed due to command corrupt, command not supported, etc.

7	6	5	4	3	2	1	0
#	#	#	#	#	ABRT	#	#

Control Register



- HOB (High Order Byte) — Some register contents require back-to-back byte-sized writes to the same address location. Setting the HOB bit enables software to read the previous written byte from the selected 8-bit register.
- SRST (Soft Reset) — Software sets this bit (writes a 1) to reset the drive.
- nIEN (Interrupt Enable, negative logic) when nIEN is cleared (0), interrupt request generation is enabled within the device, while setting the bit disables interrupt generation.

Alternate Status Register

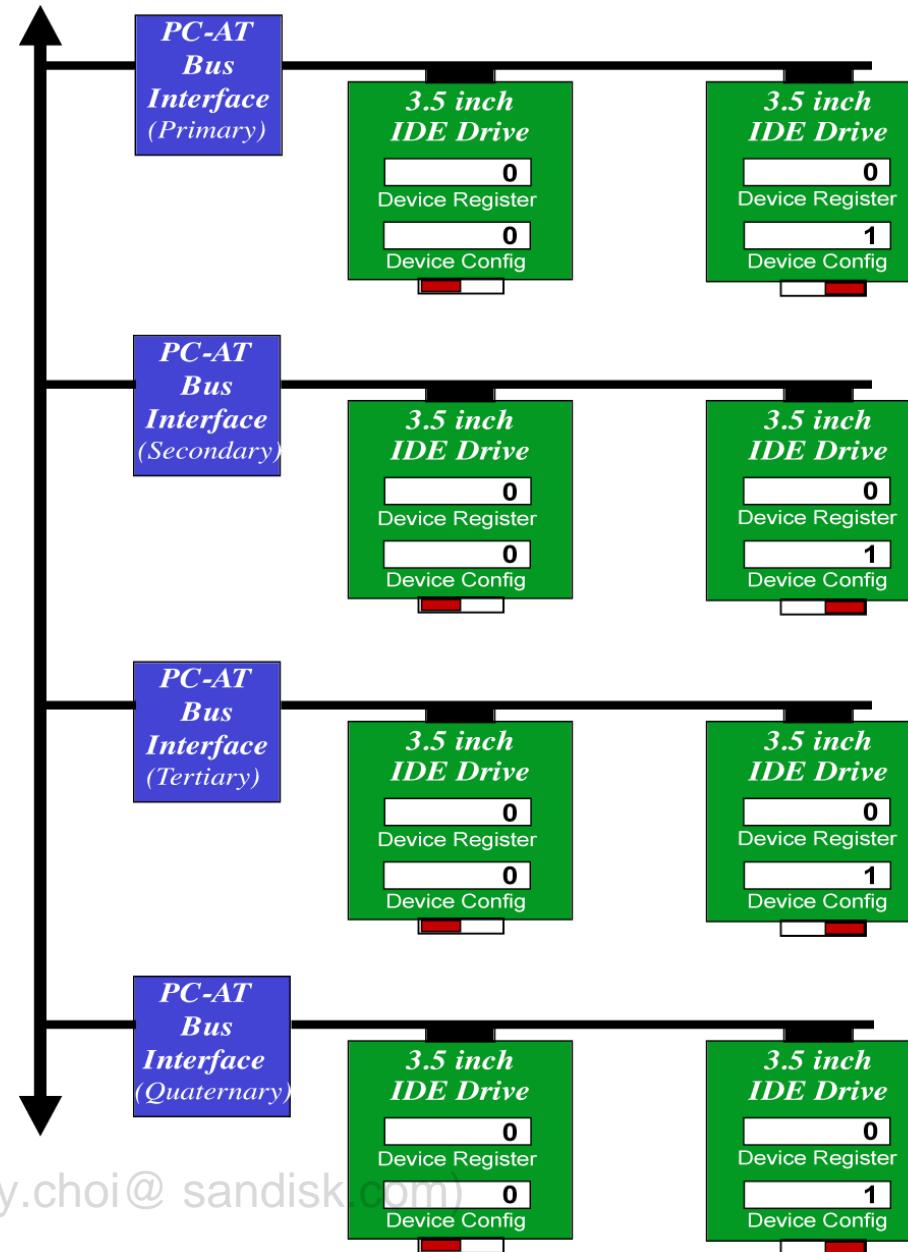
- The Alternate Status register contains the same bit designations as the Status register.
- The reason for having an alternate is that reading this one won't clear a pending interrupt, but reading the regular Status register will.

Multiple ATA Interfaces

- Registers historically mapped to x86 I/O space
- Four Channels supported

Channel	Command block	Control block	Interrupt Request
Primary	01F0 - 01F7h	03F6h	IRQ 14
Secondary	0170 - 0177h	0376h	IRQ 15
Tertiary	01E8 - 01EFh	03EEh	IRQ11
Quaternary	0168 - 016Fh	036Fh	IRQ 10

Multiple ATA Interfaces



Device Signature

Upon completion of drive reset and initialization, drives load the ATA registers with the following information:

- Device signature — indicating the device as either an ATA device or an ATAPI device
- Diagnostic results — indicating whether the drive passed the diagnostics

Device Signatures

ATA Devices

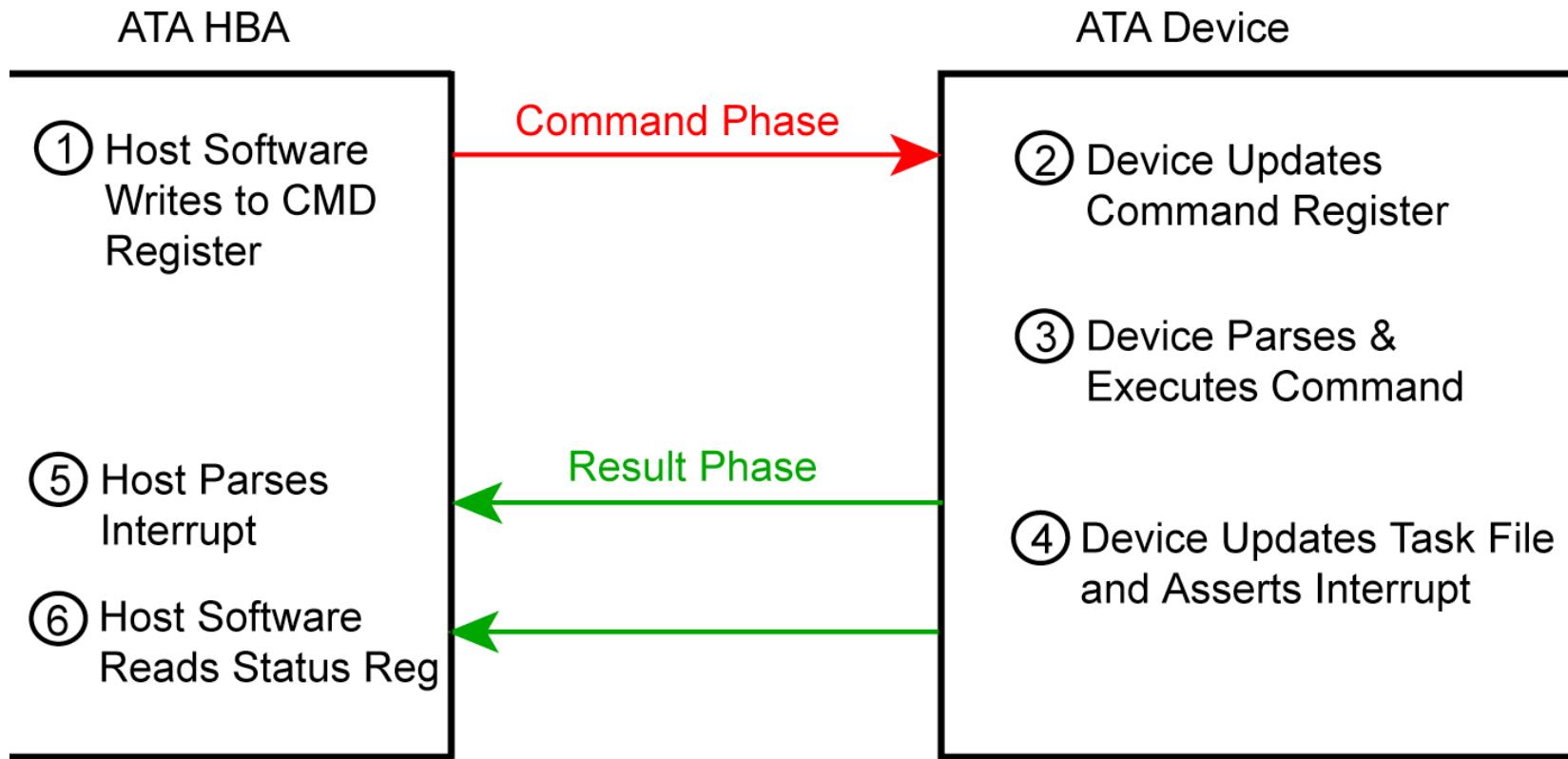
Data	
Error	01h
Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device	00h
Status	00h-70h

ATAPI Devices

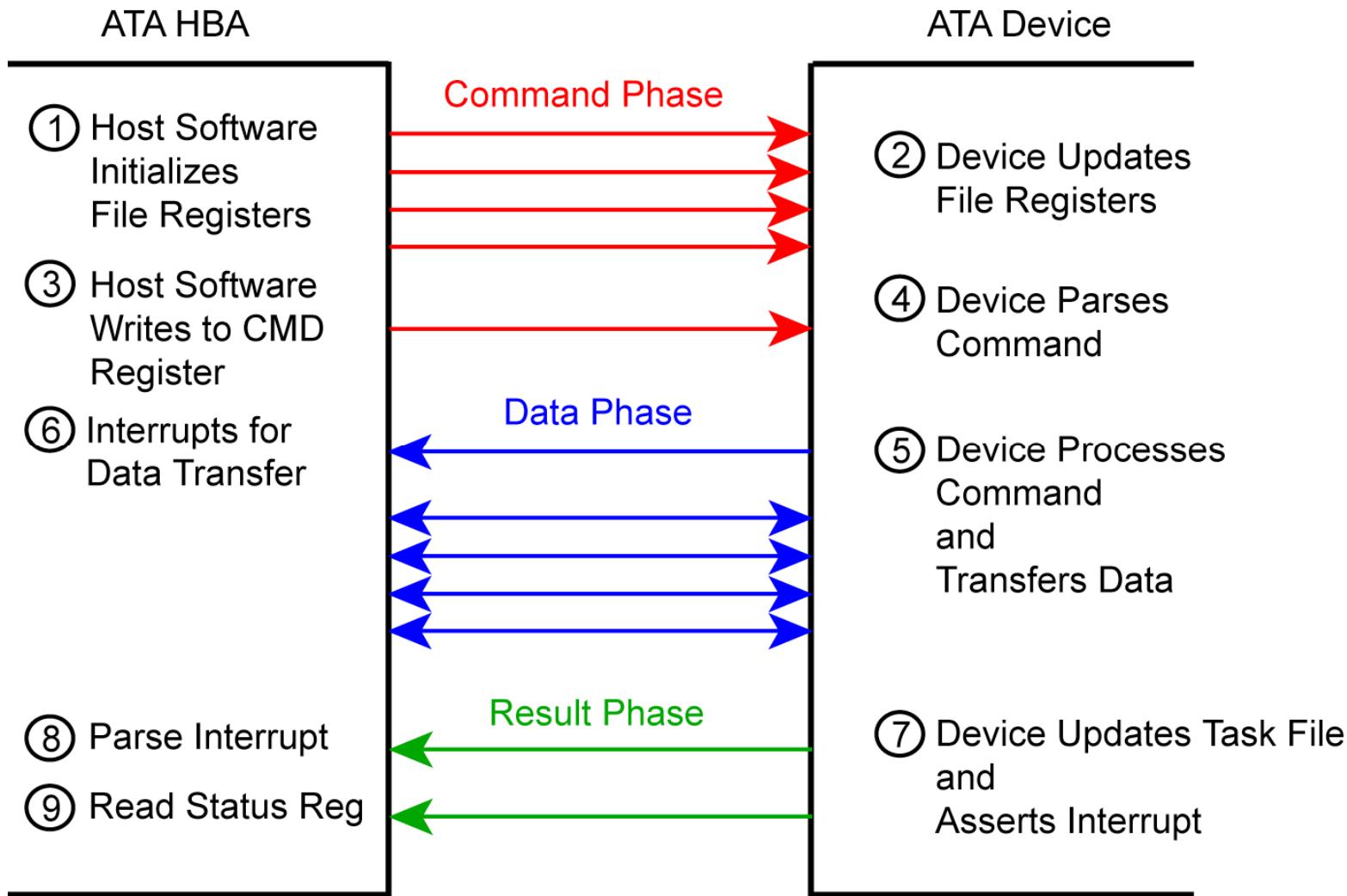
Data	
Error	01h
Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device	00h
Status	00h

Diagnostic results are written into the Error register; basically just pass/fail indication

Command Execution (Non-Data Commands)



ATA Data Command Sequence



Overlap Feature

- Overlap provides a method for drives to relinquish ownership of the shared ATA interface.
- Queued commands take advantage of overlap to request another command be delivered to the drive.

Device Identify Command

- Software detects a Drive's capabilities by reading its configuration information via the Device Identify command.
- The definition, format, and organization of the capability information is defined by the ATA standard.

History of Parallel ATA

Generation	Standard	Year	Speed	Key features
IDE		1986		Pre-standard
	ATA	1994		PIO modes 0-2, multiword DMA 0
EIDE	ATA-2	1996	16 MB/sec	PIO modes 3-4, multiword DMA modes 1-2, LBAs
	ATA-3	1997	16 MB/sec	SMART
	ATA/ATAPI-4	1998	33 MB/sec	Ultra DMA modes 0- 2, CRC, overlap, queuing, 80-wire
Ultra DMA 66	ATA/ATAPI-5	2000	66 MB/sec	Ultra DMA mode 3-4
Ultra DMA 100	ATA/ATAPI-6	2002	100 MB/sec	Ultra DMA mode 5, 48-bit LBA
Ultra DMA 133	ATA/ATAPI-7	2003	133 MB/sec	Ultra DMA mode 6

The Motivation for SATA



Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

Motivation & Design Goals

- Higher performance
- Lower pin count
- Simple drive configuration
- Better cables/connectors
- Greater reliability
- Low voltage support
- Easier migration to server market

PATA Pin Count Problems

- Large and costly chips needed for the high-pin-count bus
- Board space taken up by large bus
- Large 40-pin connectors take up space, are cumbersome to route, and inhibit air flow through a system

Interface Performance

- Maximum PATA transfer rate - 133MB/s
- SATA transfers rates:
 - 150MB/s (Generation 1)
 - 300MB/s (Generation 2)
 - 600MB/s (Generation 3)

No Drive Config Required

- PATA drives have jumpers/switches to configure drive as device 0 or 1.
- SATA drives are implemented as point-to-point interconnects (not shared) and don't need device selection.

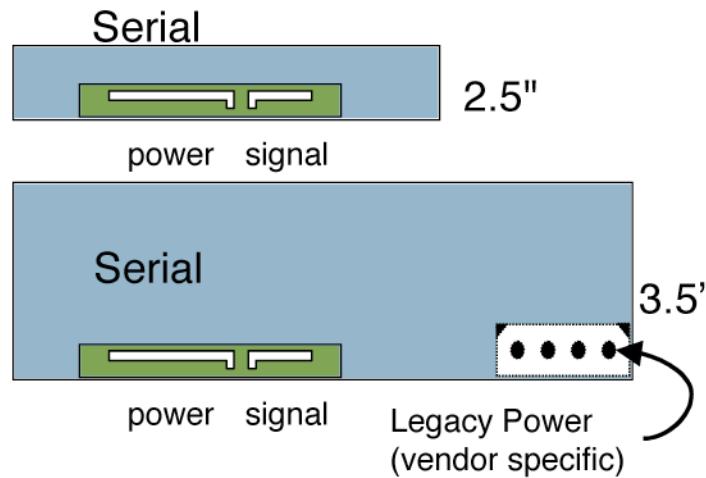
Cables & Connectors

Improved Connectors and Cabling

- Smaller and thinner
- Easier to manage
- Improves airflow through the box, reducing the number of fans needed
- Blind mating capability and use of contacts rather than pins eliminates bent pins
- Cable lengths up to 1 meter
- Hot Pluggable connectors supported

Cables & Connectors, continued

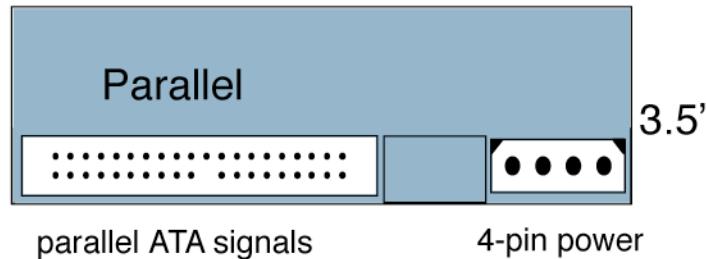
Device connector sizes and locations



(5.25" form factor also defined for devices like tape drives and DVDs)

Some Platforms do not support the new SATA power cable & use the legacy power connect instead.

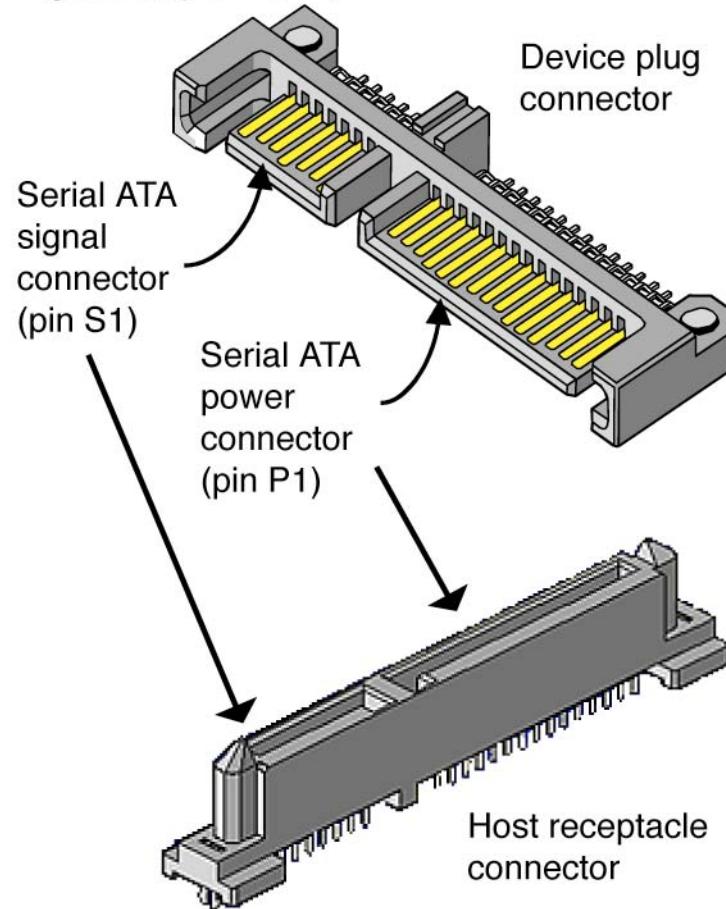
New drives don't use legacy power, just include an adapter from the old to the new cable.



SATA Drive Plug/Host Receptacle

Appearance of Serial ATA Connectors

(Drawing courtesy of Molex)



Reliability

- PATA bus supports CRC error detection for Ultra DMA data transfers only
- SATA integrates CRC on all packets, improving reliability
- CRC is commonly used in serial transports and detects all single- and double-bit errors, resulting in detection of 99.99% of all errors

Lower Voltages

Smaller silicon manufacturing process sizes require smaller voltages.

- ATA 5 volt signaling is not suitable
- SATA typically uses 0.4v to 0.7 volts, permitting smaller drive interface chips and reducing power.

Migration to Servers

A number of server-related features were introduced with the SATA II specification:

- Higher speed: 3.0Gb/sec (not part of first SATA II definition, added with later versions)
- Port Multipliers: allow a single HBA port to support up to 15 drives
- Port Selectors: provide fail-over support
- Hot Docking support
- Native Command Queuing: improves performance
- Support for Enclosure Services and Management

PATA Software Compatibility

- Major design goal of SATA was compatibility with legacy firmware and software
 - Primarily, this meant the PATA programming interface (Task File registers) had to remain visible to software so legacy code would work without changes
 - Extremely important for encouraging migration from PATA (already cheap and well established) to the new SATA model

SATA Overview

Christy Choi(christy.choi@sandisk.com)
Do Not Distribute



SATA Specification

- Developed by:
 - Serial ATA International Organization
 - Member companies can download the specification from www.sata-io.org
 - Current version = 2.6

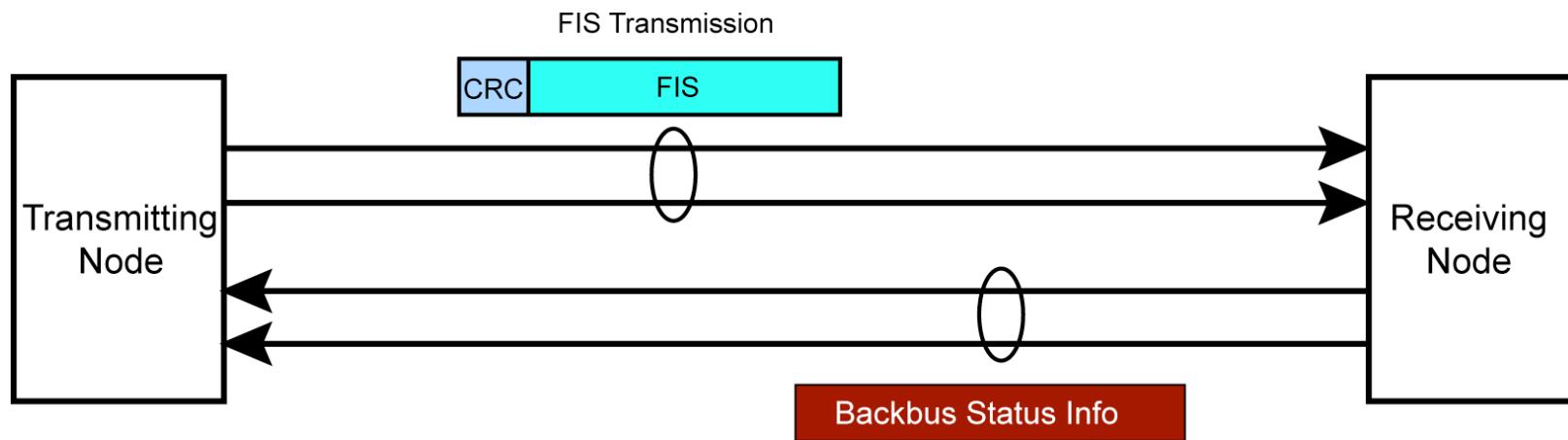
Summary of SATA Features

- High-speed, serial interconnect
- Low voltage, Differential signaling
- Point-to-point connections
- Improved cables and connectors
- Error detection and retry
- Link power management

(Complete list – book page 44)

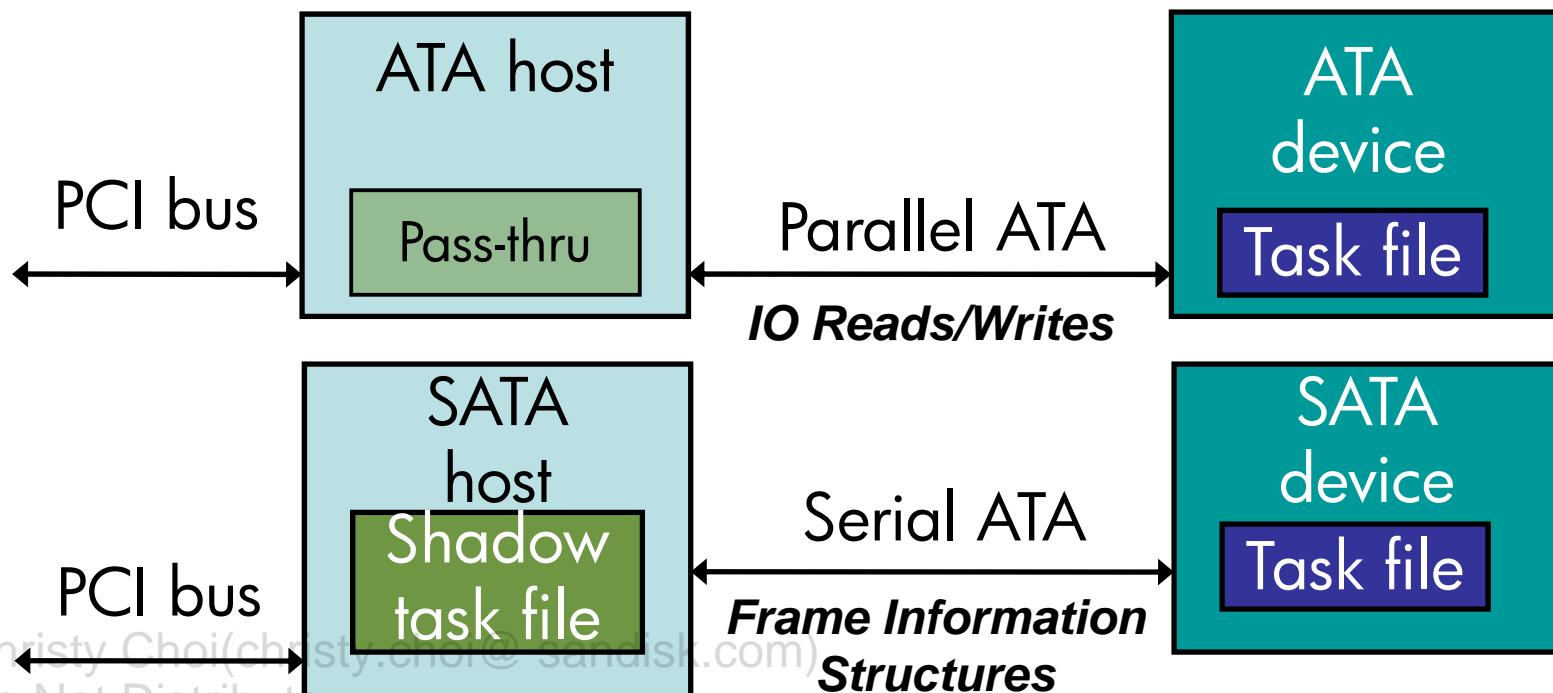
Serial Interconnect

SATA employs two differential pairs that are used in a half-duplex arrangement.



The SATA Legacy Approach

- The task file is in the ATA device
 - In parallel ATA, accesses to these registers resulted in several transfers across the bus, one for each register
- In serial ATA, a Shadow Task file register bank is managed by the host to mirror the ATA device's task file. This allows register accesses to be grouped into one serial packet, rather than many smaller transfers.



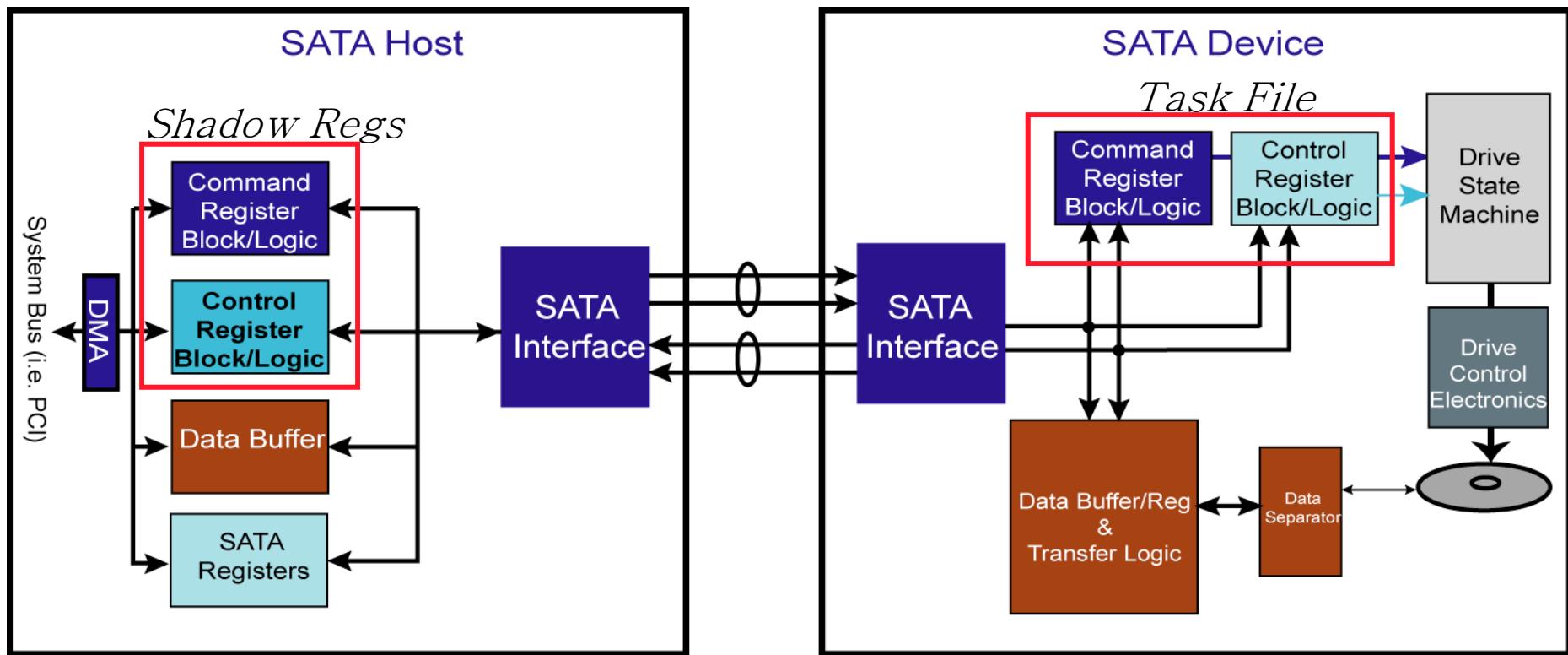
SATA Improvements

- Ability to add more interfaces per HBA
- Link power management support

SATA Programming Interfaces

- Legacy Register Set
 - Software writes to Control and Command register blocks
 - Software programs DMA registers to move data
 - HBA generates interrupt to tell software that data is ready to move into or out of memory
 - HBA generates interrupt to notify software that transfer has completed and status is available
- Advanced Host Controller Interface (AHCI)
 - Software creates command table in memory that points to a data structure containing the commands to be performed
 - Data structure also contains scatter/gather lists that specify the target memory locations

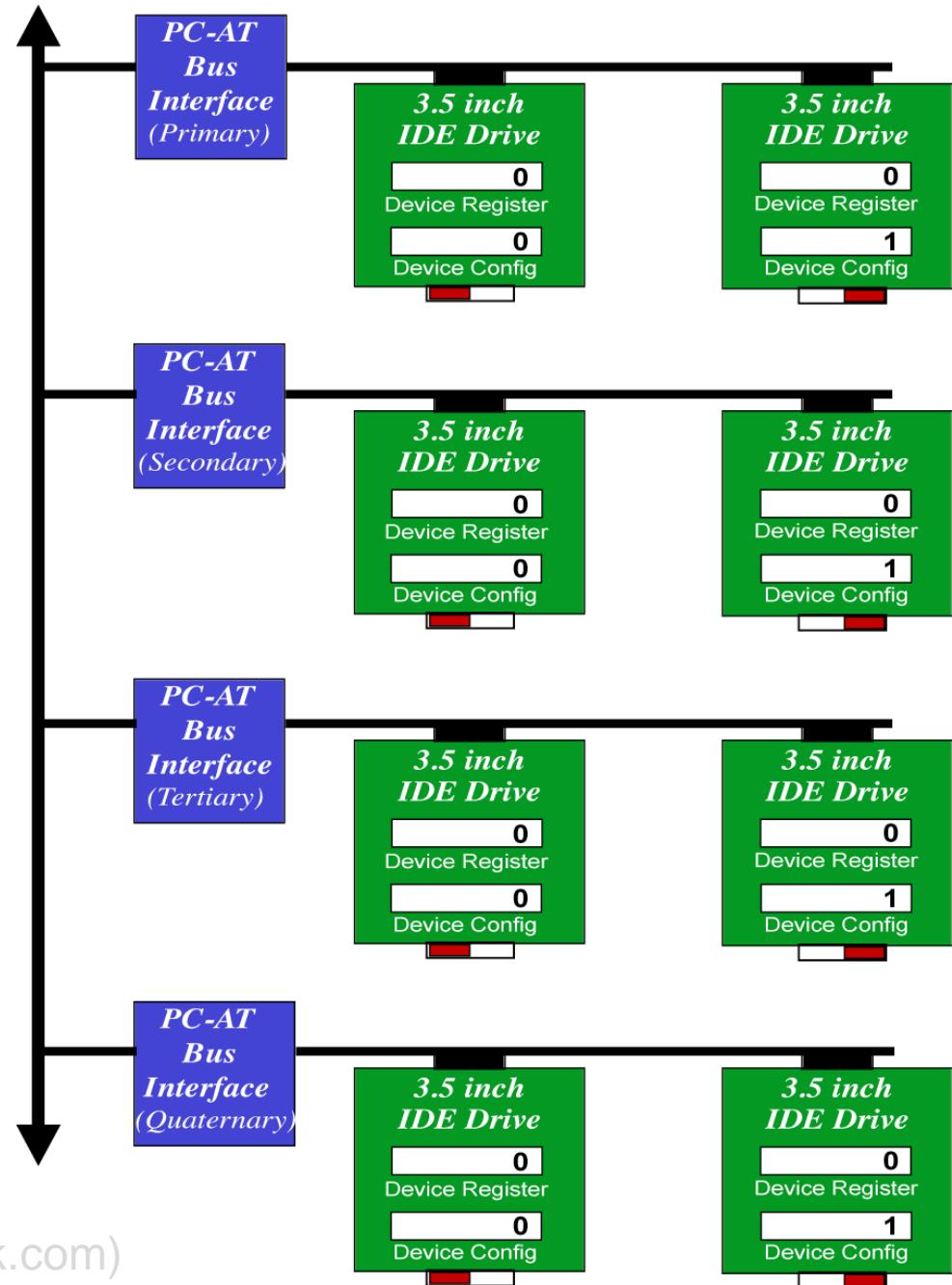
Legacy Register Set Interface



Legacy Systems

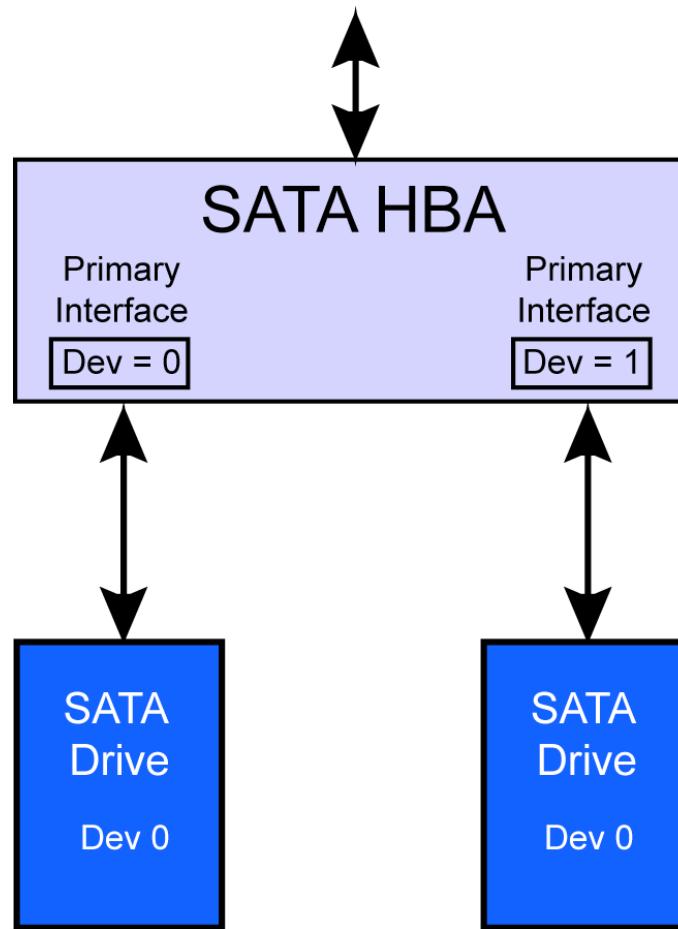
(up to 8 Drives)

- Four HBAs
- Each drive supports up to two drives



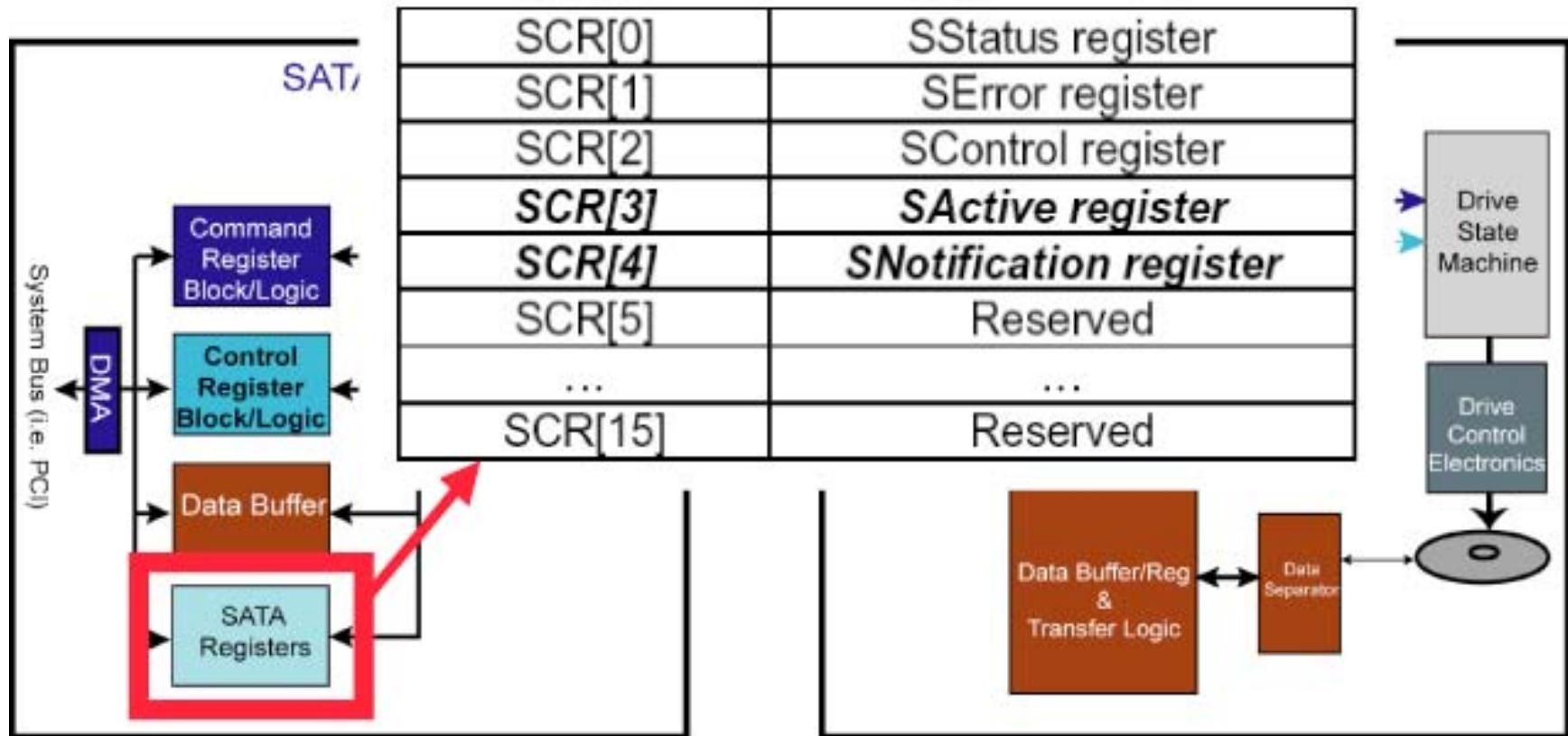
Master/Slave Emulation

- HBAs may support Master/Slave emulation.



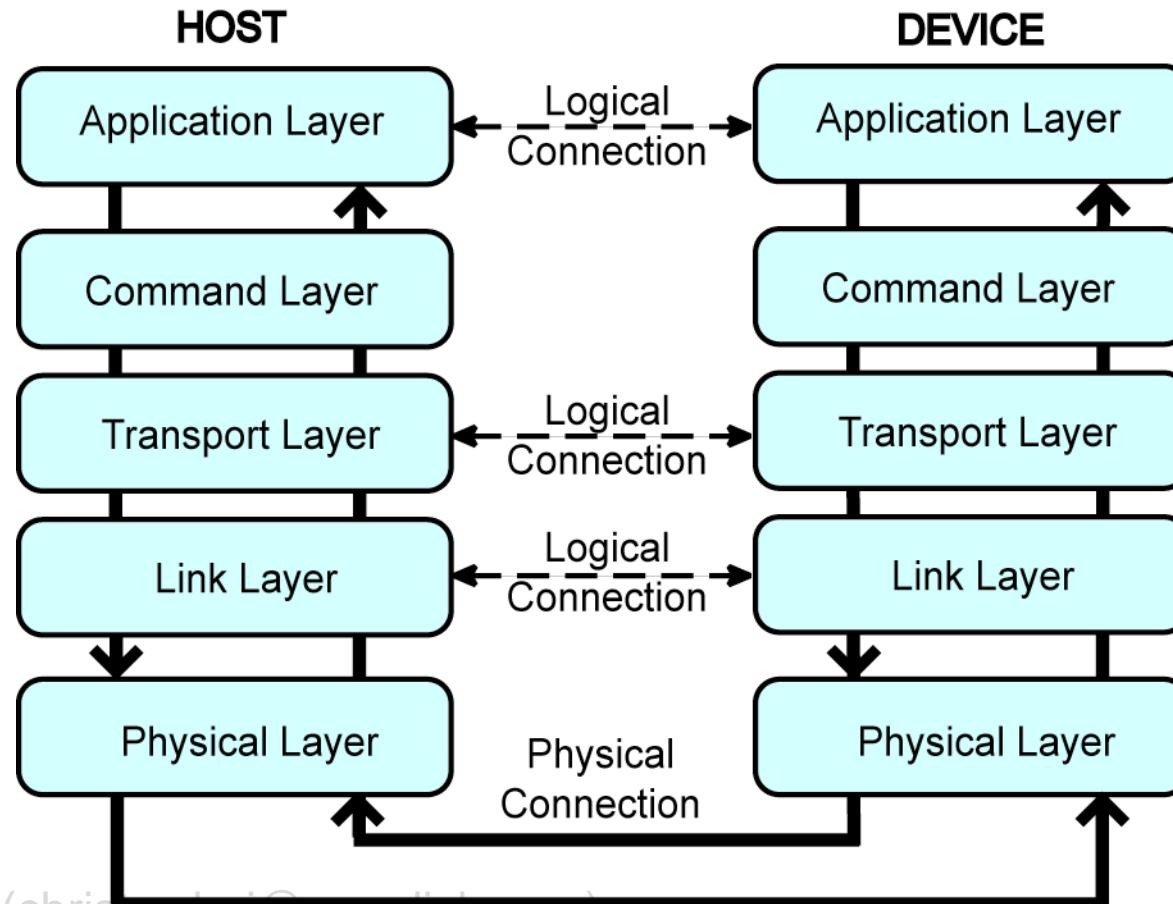
SATA-Specific Registers

See page 51 for description of registers

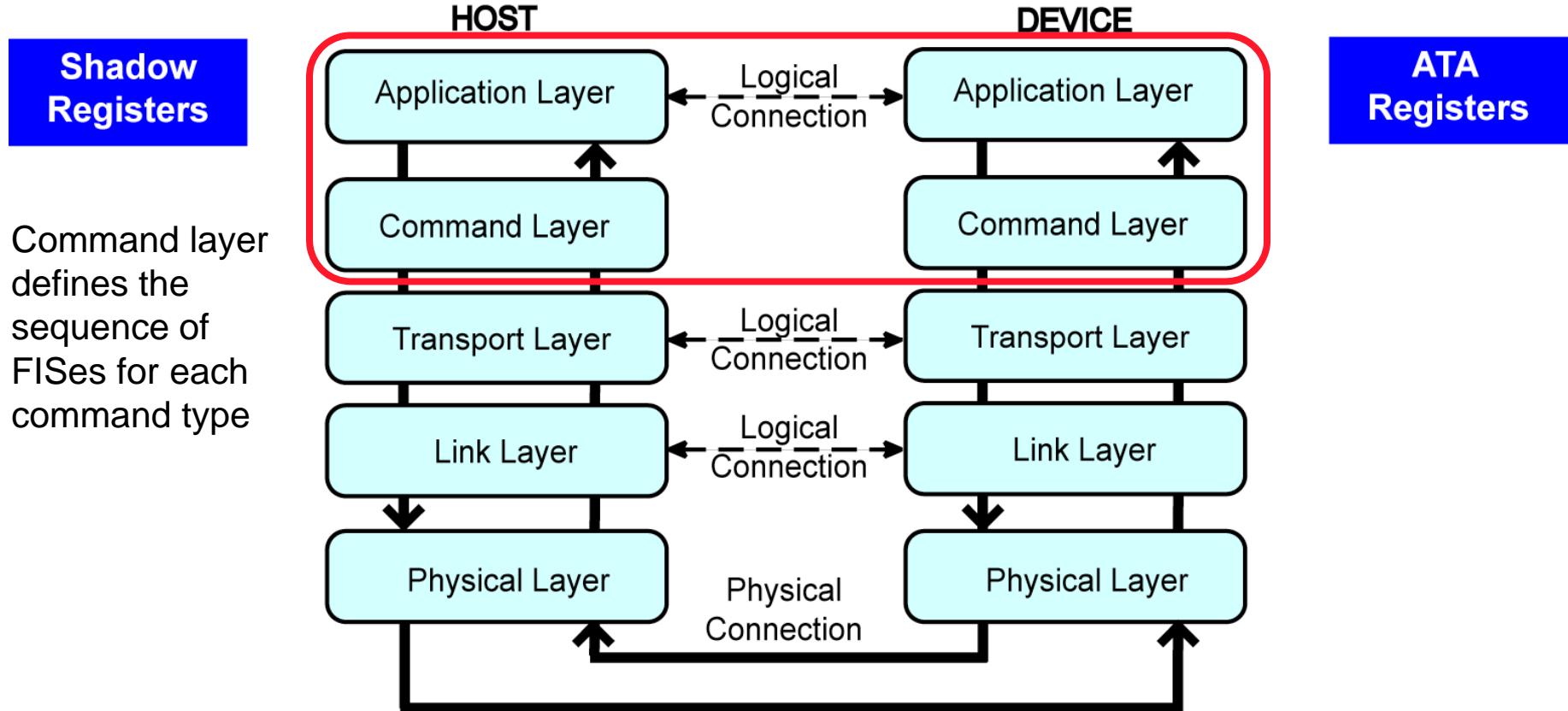


SATA Protocol Layers

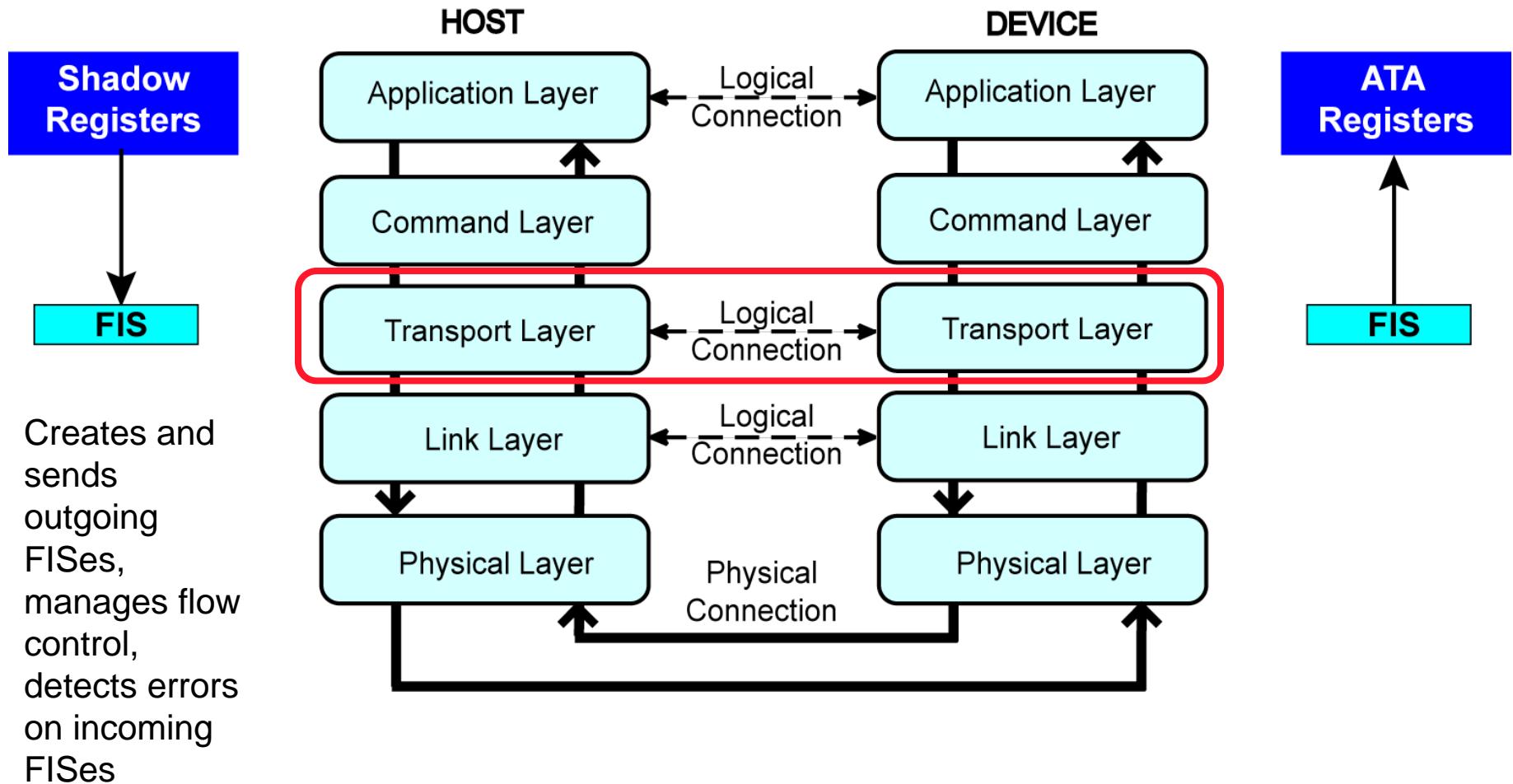
See page 52 for general description of the layers



Application/Command Layers



Transport Layer



Register FIS – Device to Host

The transport layer builds a compliant FIS in response to a command issued by software.

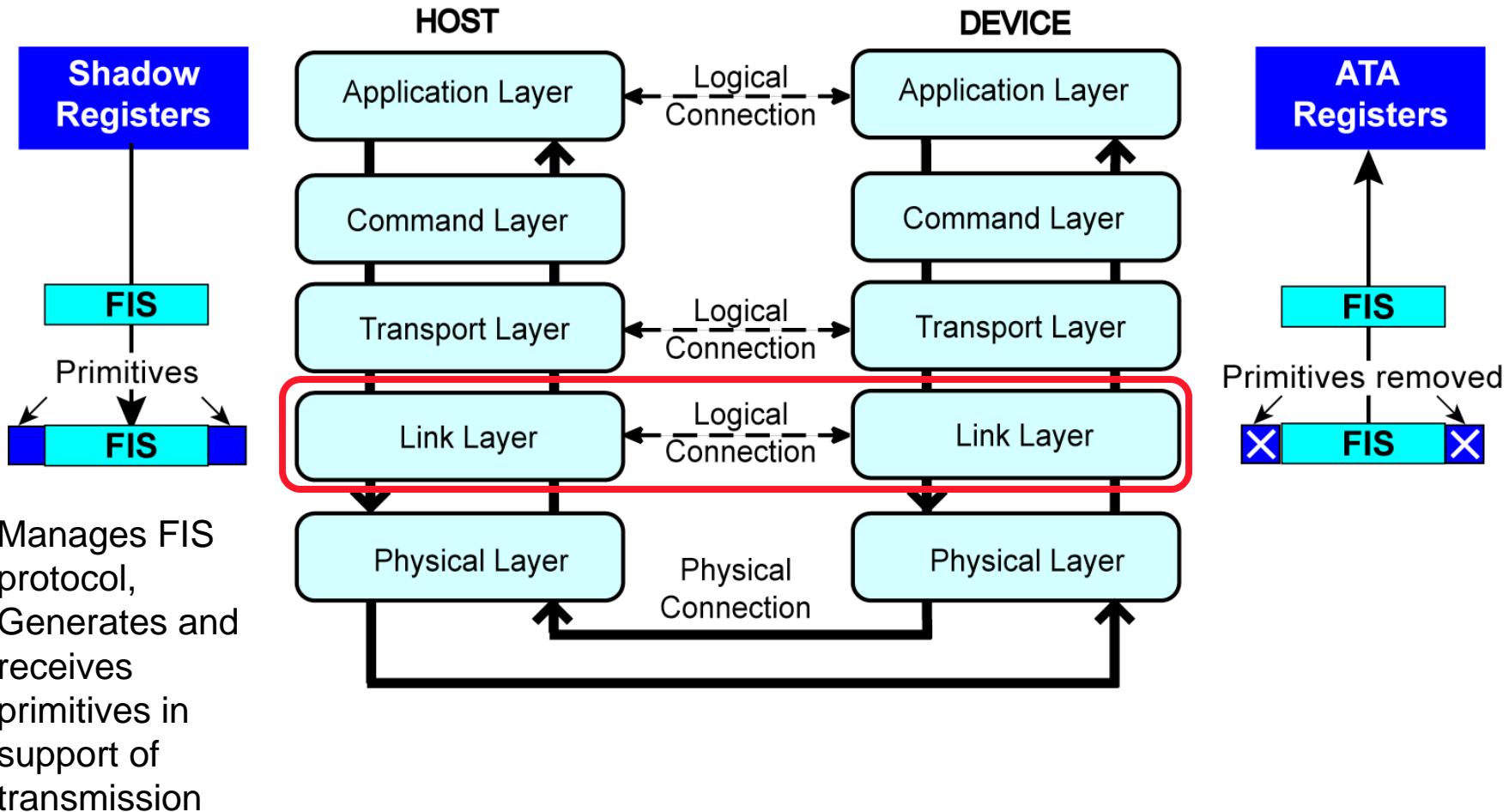
	+3	+2	+1	+0
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
DW 0	Error	Status	R I R Reserved	FIS Type (34h)
DW 1	Dev/Head	LBA High	LBA Middle	LBA Low
DW 2	Reserved (0)	LBA High (exp)	LBA Mid (exp)	LBA Low (exp)
DW 3	Reserved (0)	Reserved (0)	Sec Count (exp)	Sector Count
DW 4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

Register FIS - Host to Device

This Register FIS illustrates the LBA addressing.

	+3	+2	+1	+0
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
DW 0	Features	Command	C R R Reserved	FIS Type (27h)
DW 1	Device	LBA High	LBA Middle	LBA Low
DW 2	Features (exp)	LBA High (exp)	LBA Mid (exp)	LBA Low (exp)
DW 3	Control	Reserved (0)	Sec Count (exp)	Sector Count
DW 4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

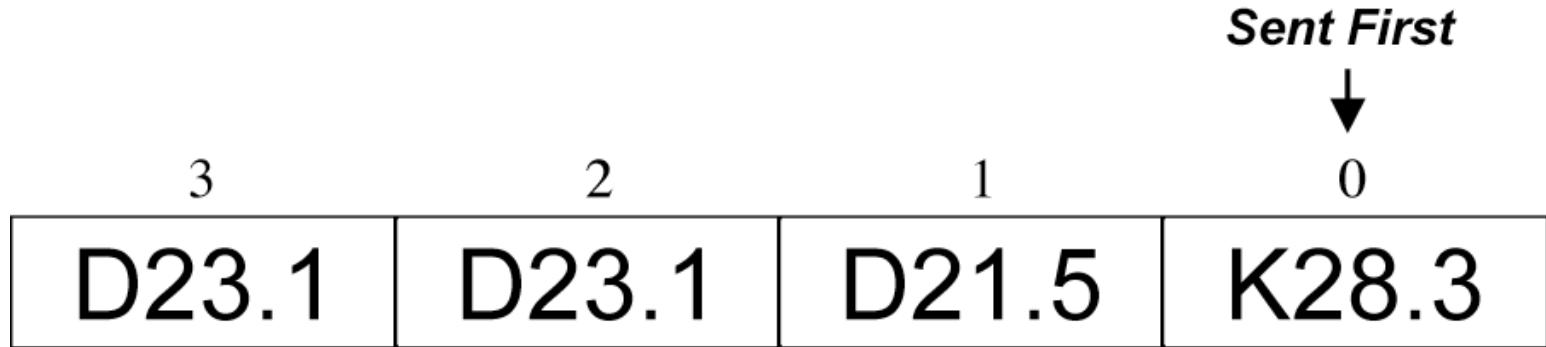
Link Layer



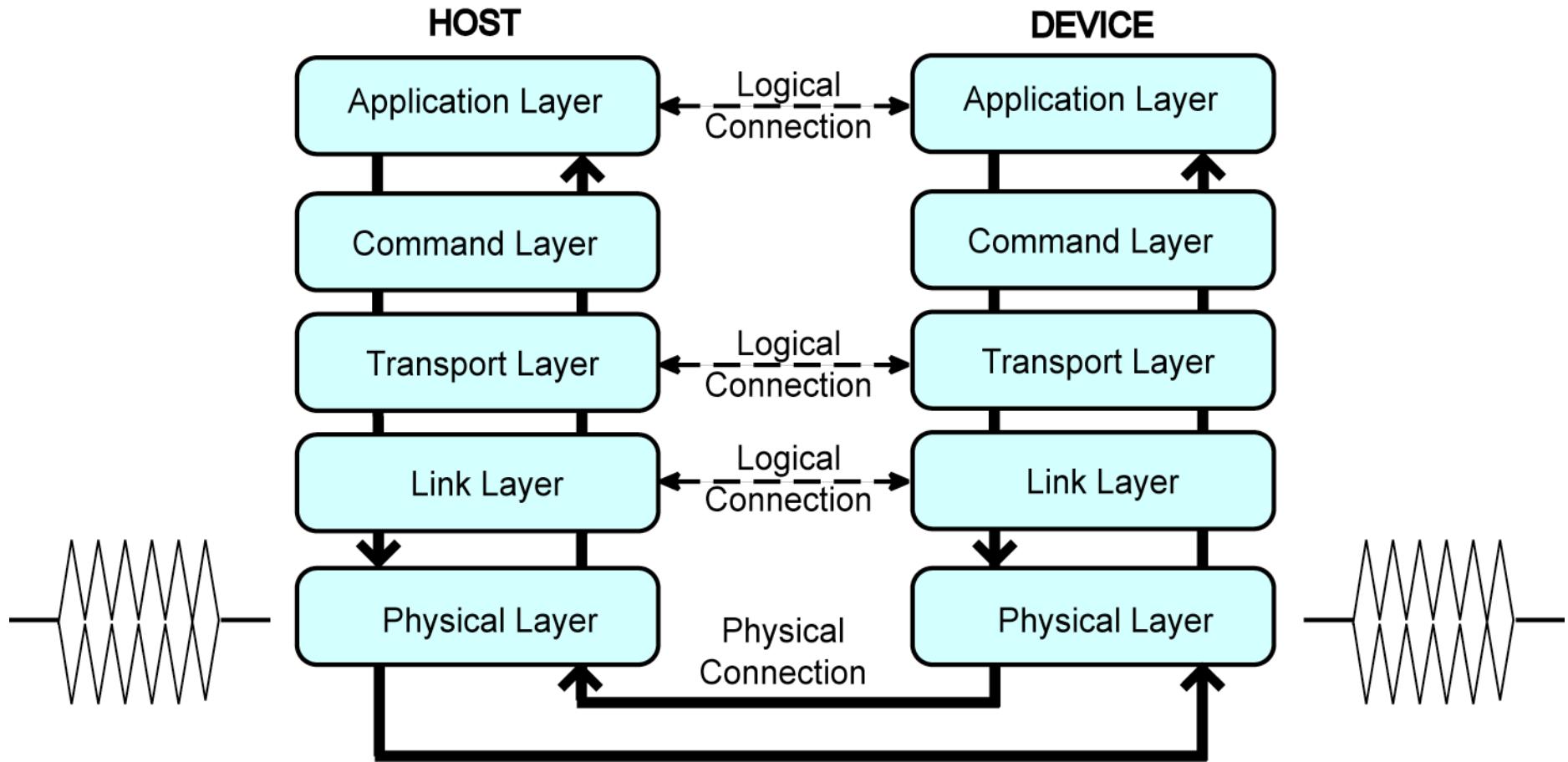
Link Layer

- The Link Layer essentially manages the FIS transmission protocol.
- This involves generating and decoding small packets called primitives that are involved in many functions:
 - Indicate beginning and end of each FIS
 - Bus arbitration
 - FIS transmission status
 - Flow control
 - Link power management
 - Clock compensation

Link Layer (Example Primitive)



Physical Layer



Physical Layer Functions

- Establishing link communications following reset (via Out of Band, or OOB signaling)
- Transmit and receive FIS and primitive traffic

Physical Layer Functions

(OOB Signaling)

- Begins during or just after the Hardware Reset.
- Consists of an exchange of signaling bursts between the HBA and drive.
- Uses a sequence of six cycles of burst and idle on the bus. The idle times vary in duration to communicate very simple information.

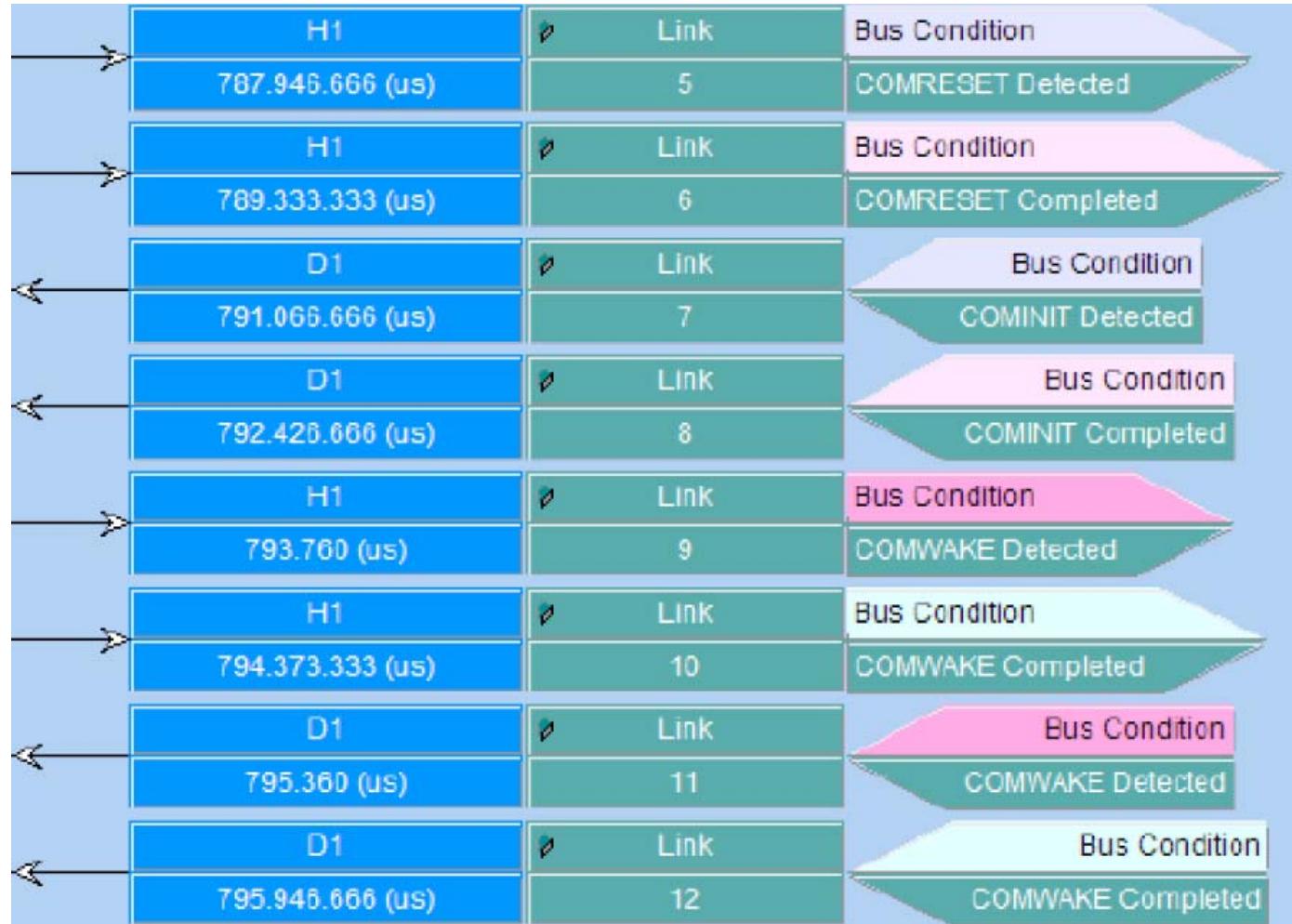
Physical Layer Functions

(OOB Sequence)

Example OOB Sequence:

- HBA signals COMRESET; drive detects it starting
- Drive detects completion of COMRESET
- Drive signals COMINIT; HBA detects it starting
- HBA detects completion of COMINIT
- HBA signals COMWAKE; drive detects start
- Drive detects completion of COMWAKE
- Drive signals COMWAKE; HBA detects start
- HBA detects completion of COMWAKE and enters Link initialization

Physical Layer Functions (OOB Sequence Capture)



Christy Choi(christy.choi@sandisk.com)

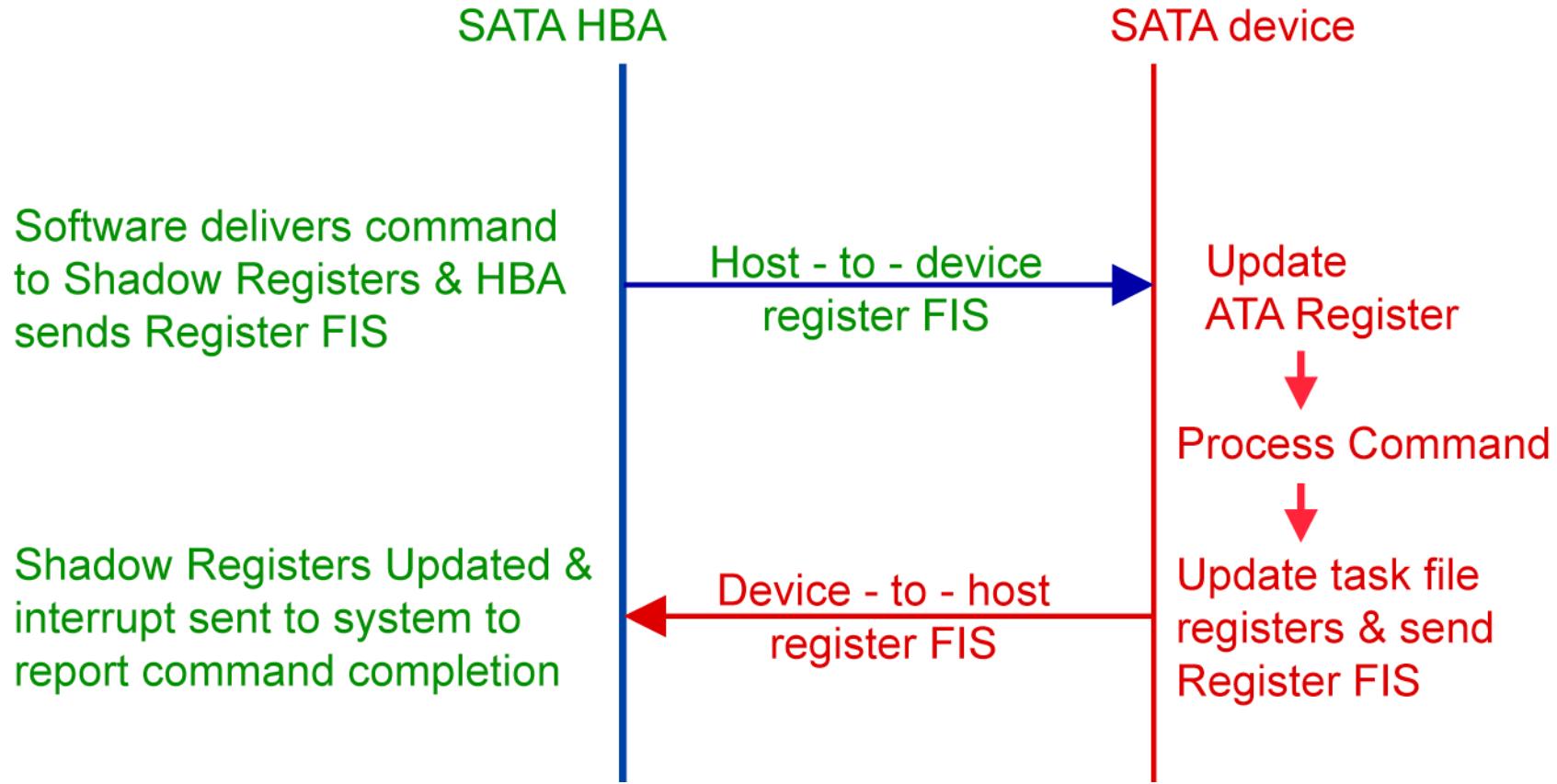
Do Not Distribute

Copyright Mindshare Inc, 2009

Link Initialization

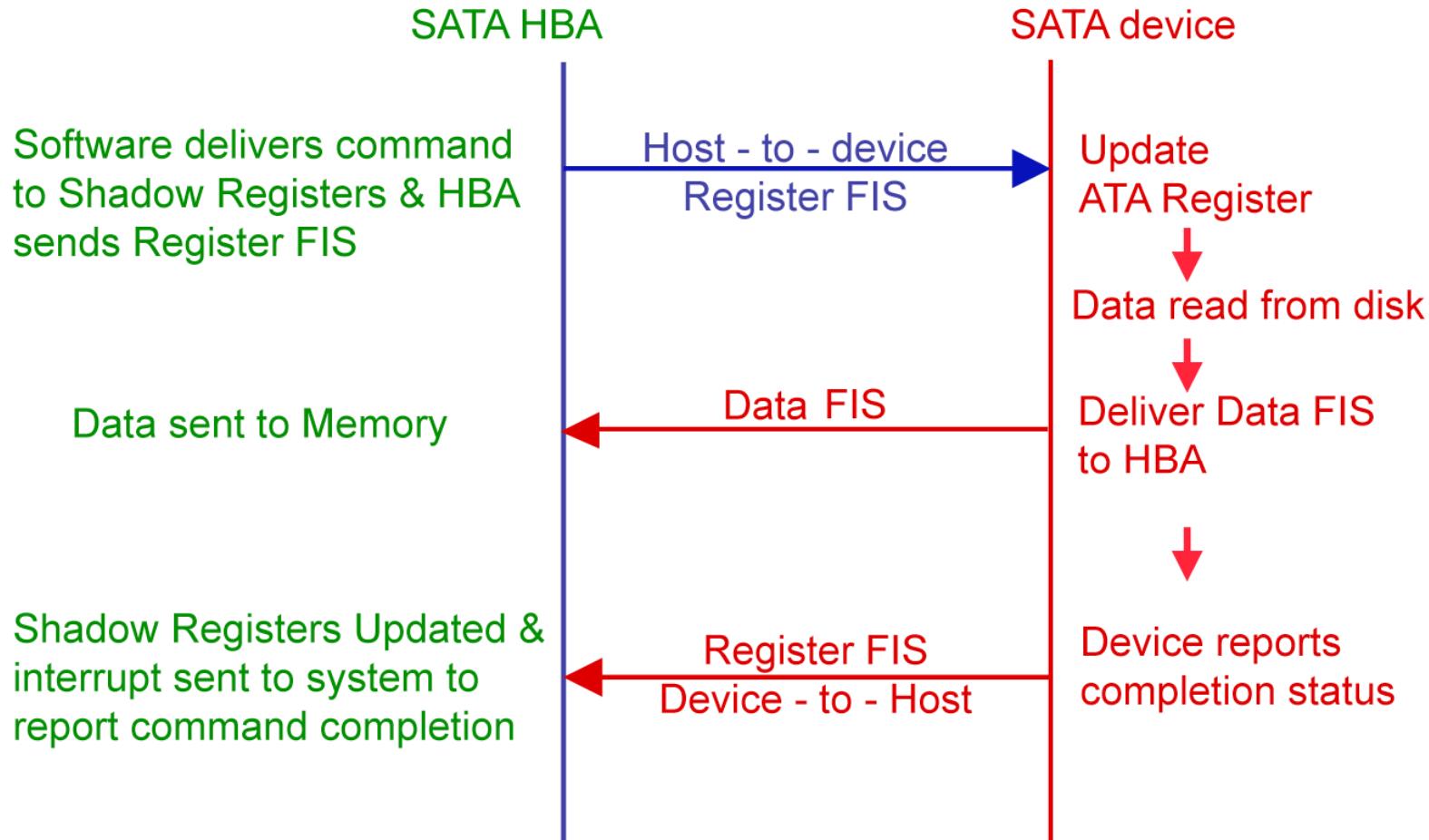
- Following the OOB sequence link initialization begins. This includes:
 - Speed negotiation
 - Establishing bit lock
 - Achieving byte alignment
- Details regarding OOB and initialization covered later.

SATA Command Protocol (Example non-data command)



See next slide for additional information

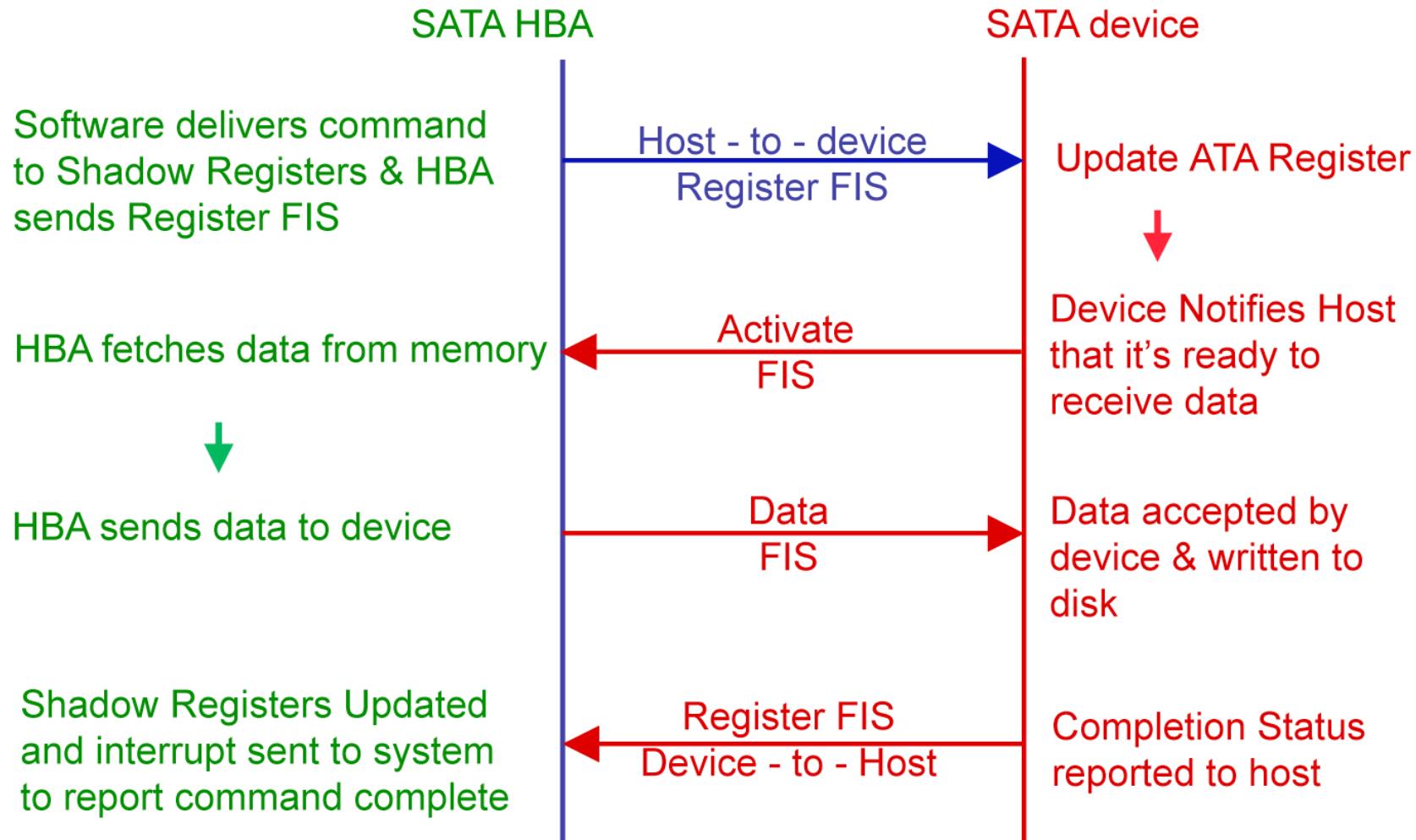
SATA Command Protocol (Example data read command)



See next slide for additional information

SATA Command Protocol

(Example data Write command)



See next slide for additional information

Features of SATA II

- Compatible with SATA 1.0a
- Scalable performance:
 - 150MB/s -- Generation 1
 - 300MB/s -- Generation 2
- New features implemented as extensions to SATA 1.0a
- Support for Native Command Queuing (NCQ)

Features of SATA II

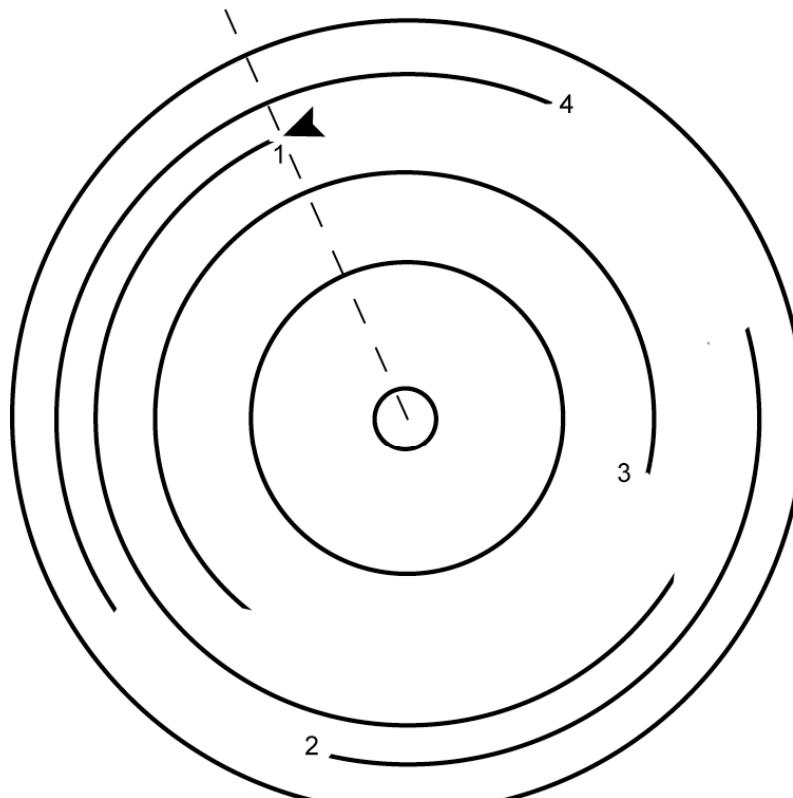
Support for:

- Enclosure services
 - SEMB (SATA Enclosure Management Bridge)
 - SEP (Storage Enclosure Processor)
- Device management via storage subsystem
- Redundant hosts (Port Selector)
- Adding more ports (Port Multiplier)

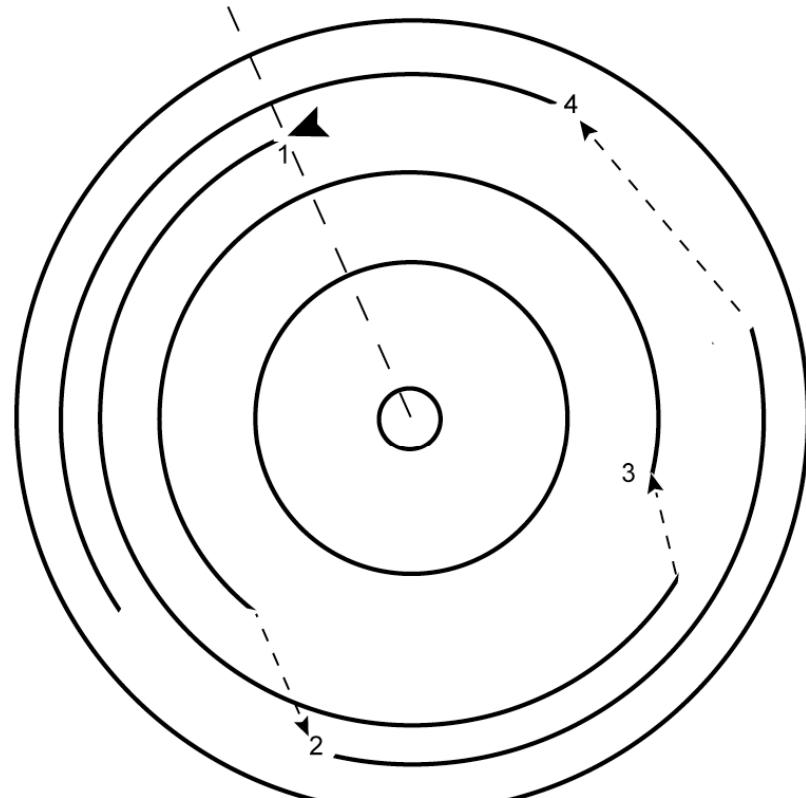
Migrating SATA to the High End

NCQ supports up to 32 queued commands that can complete out-of-order to reduce access latency

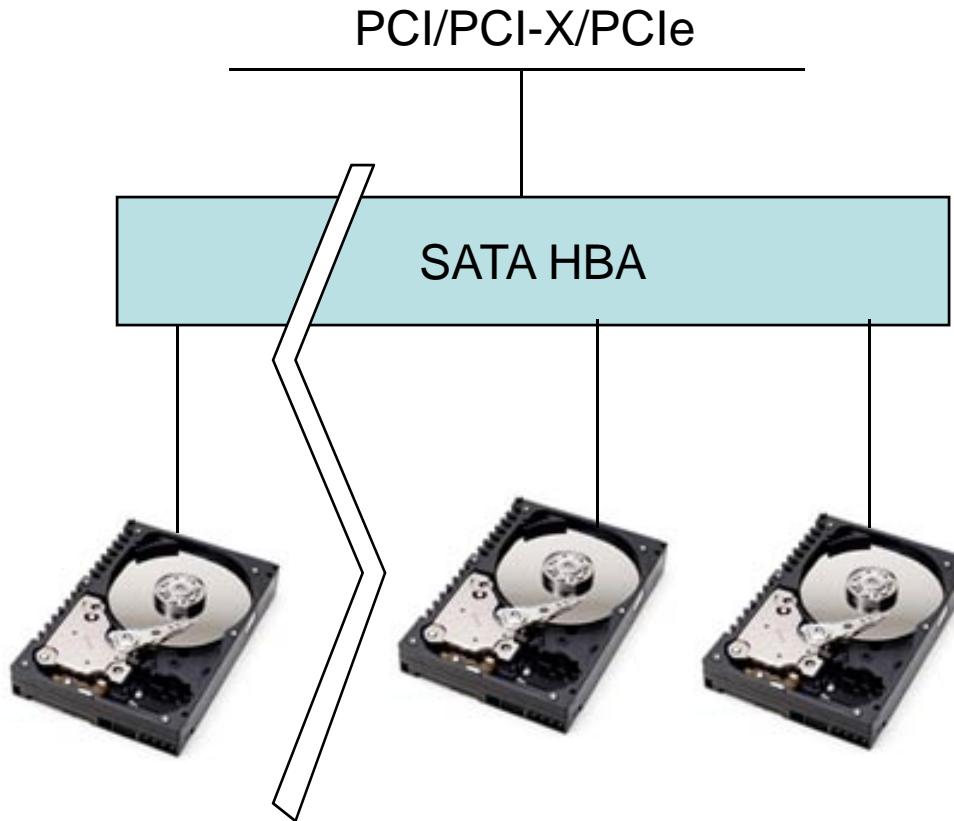
Linear Command Execution
~ 4.25 revolutions



Out-of-Order Command Execution
~ 2.25 revolutions



Migrating SATA to the High End



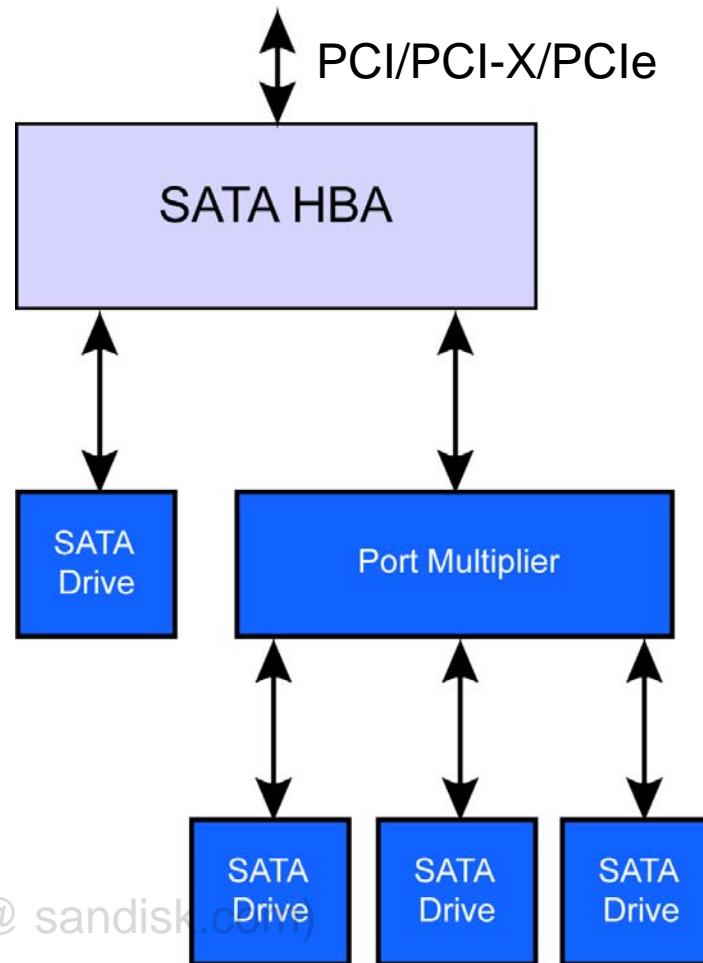
Christy Choi(christy.choi@ sandisk.com)

Do Not Distribute

Copyright Mindshare Inc, 2009

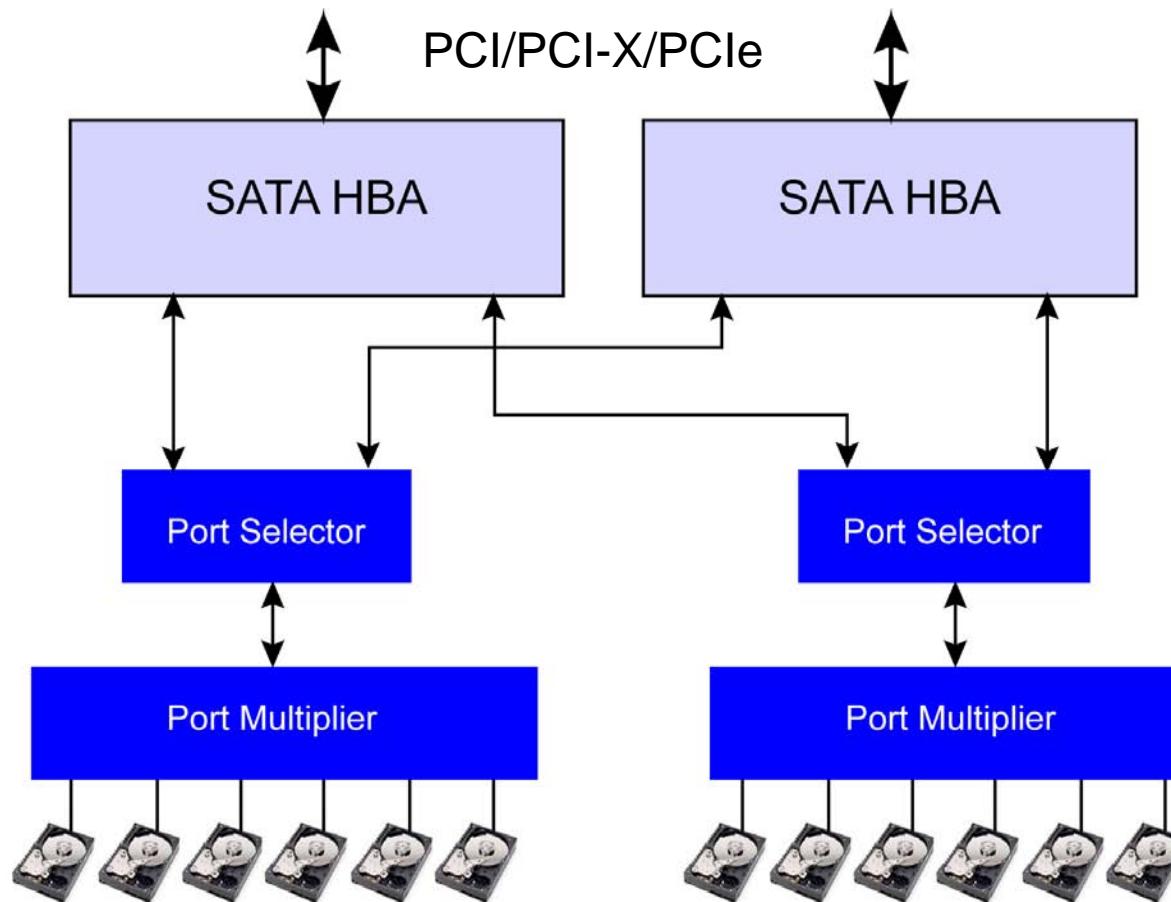
Migrating SATA to the High End

Port multipliers increase the number
of devices per HBA port



Migrating SATA to the High End

Port selectors give HBA failover protection



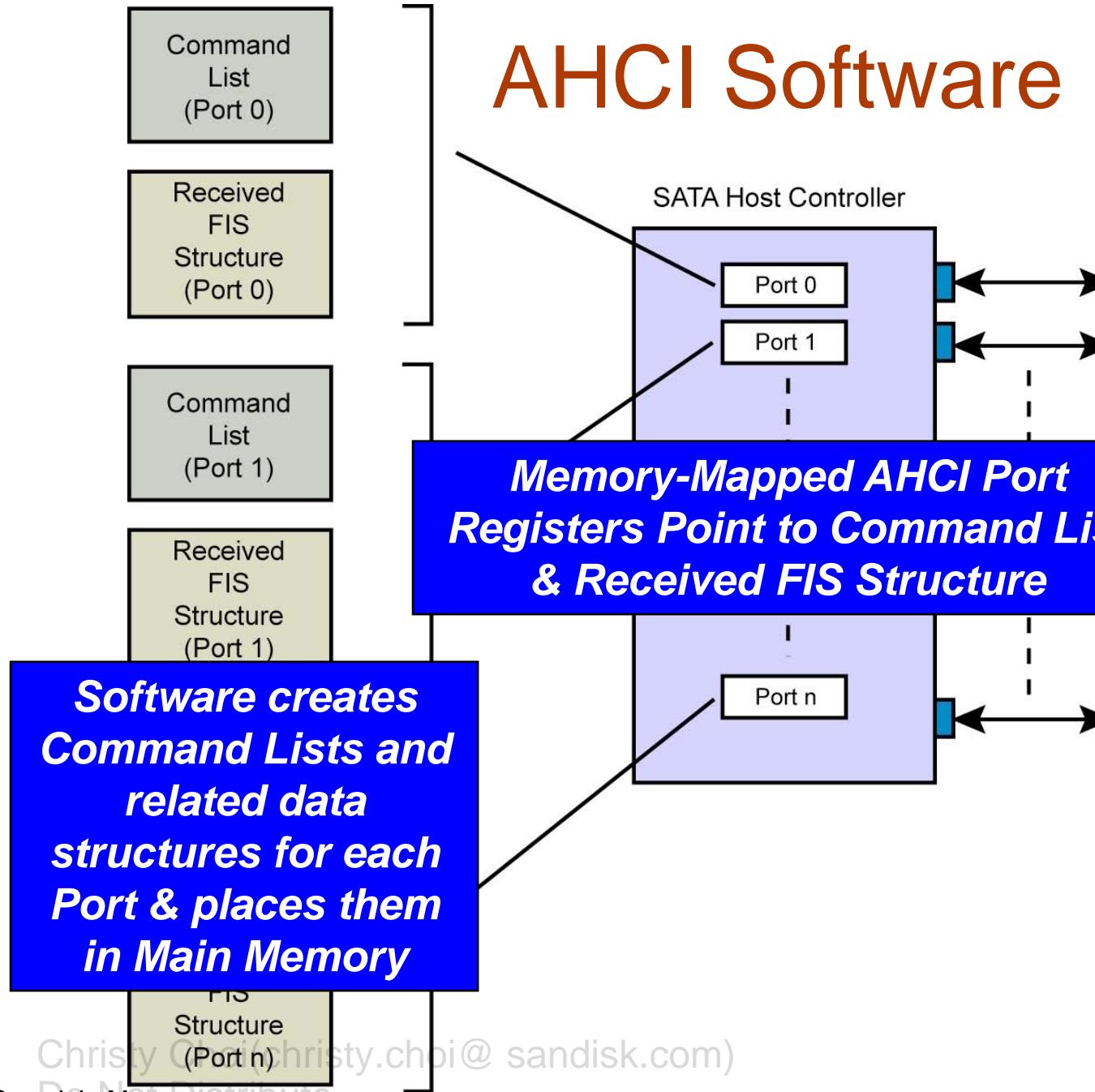
High-End SATA - Transport

- Advantages: simple, low cost, high volume due to consumer market use
- Disadvantages: half-duplex bus and lower performance compared to other enterprise-class transports like SCSI or Fibre Channel
 - Use of RAID can mitigate performance and reliability problems, but adds some cost

High-End SATA - Drives

- Advantages: high capacity at low cost
 - Cost example: drives can get by with small buffers – on writes they typically write data to disk without waiting to verify CRC (SATA packets can be up to 8K, so waiting for the whole packet would be slow)
- Disadvantages: not dual ported, lower performance, lower MTBF
 - Use of RAID can mitigate performance and reliability problems. Creates viable near-line storage option for SATA.

AHCI Software Interface



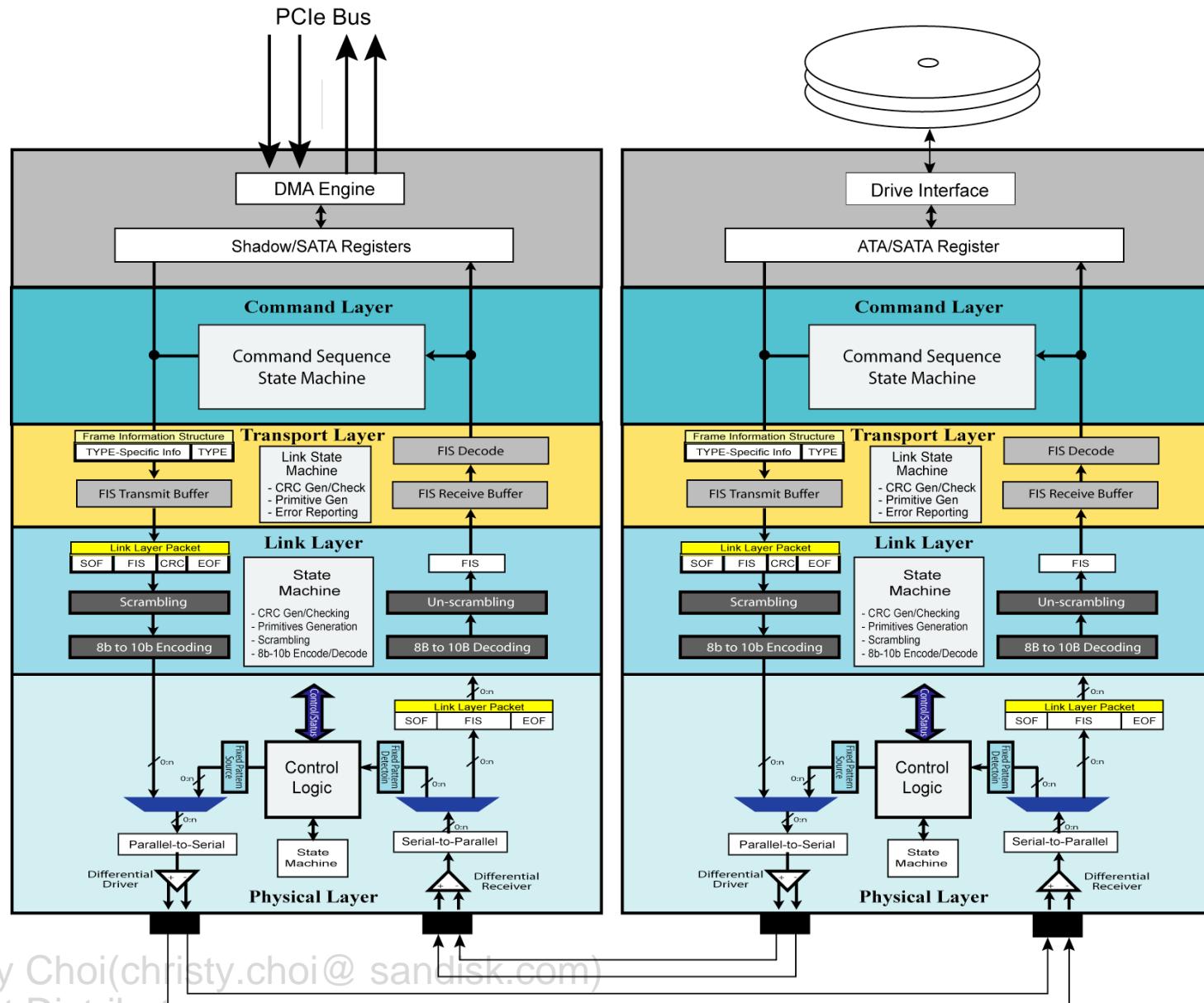
Introduction to FIS Transfers

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

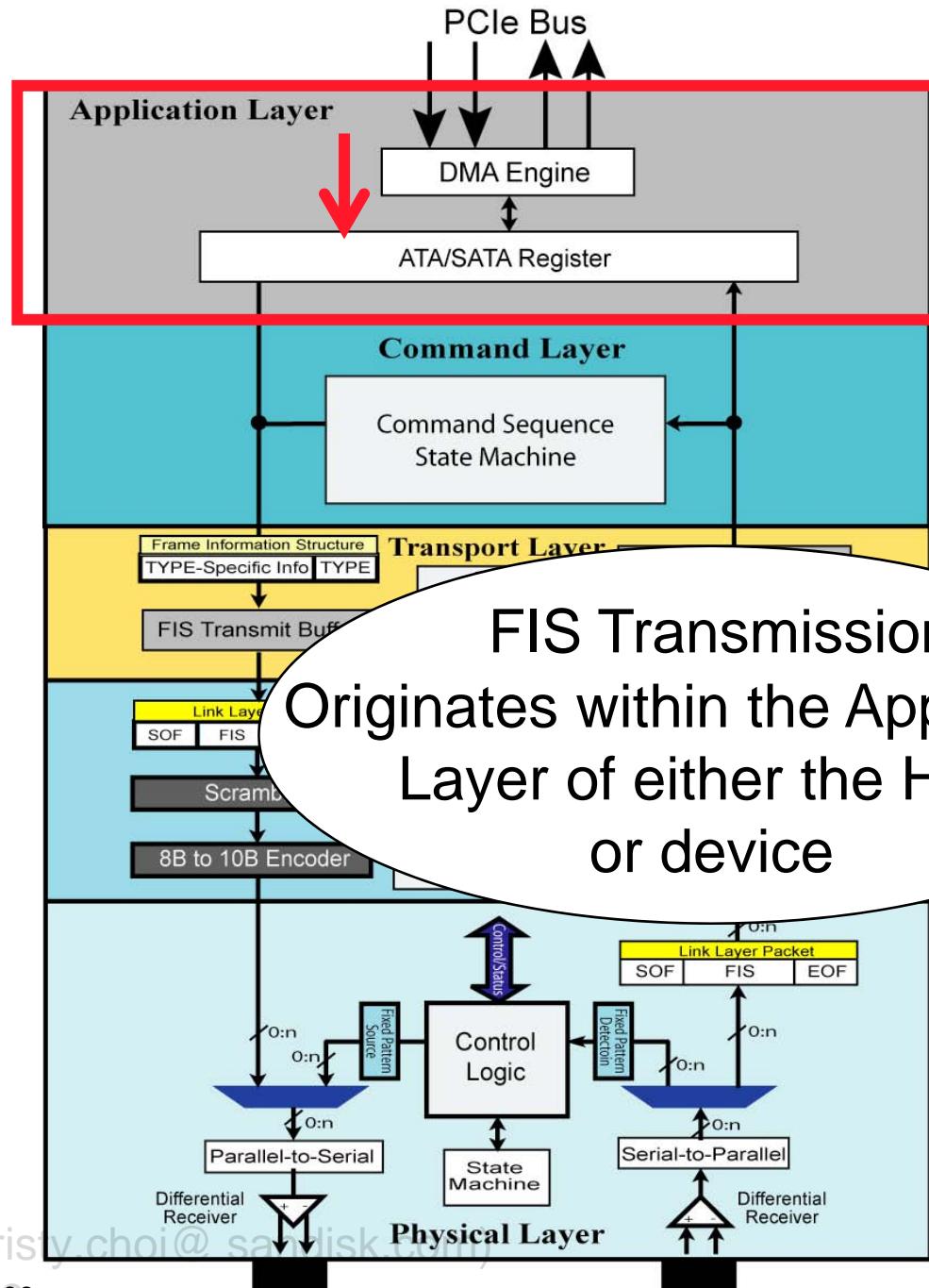


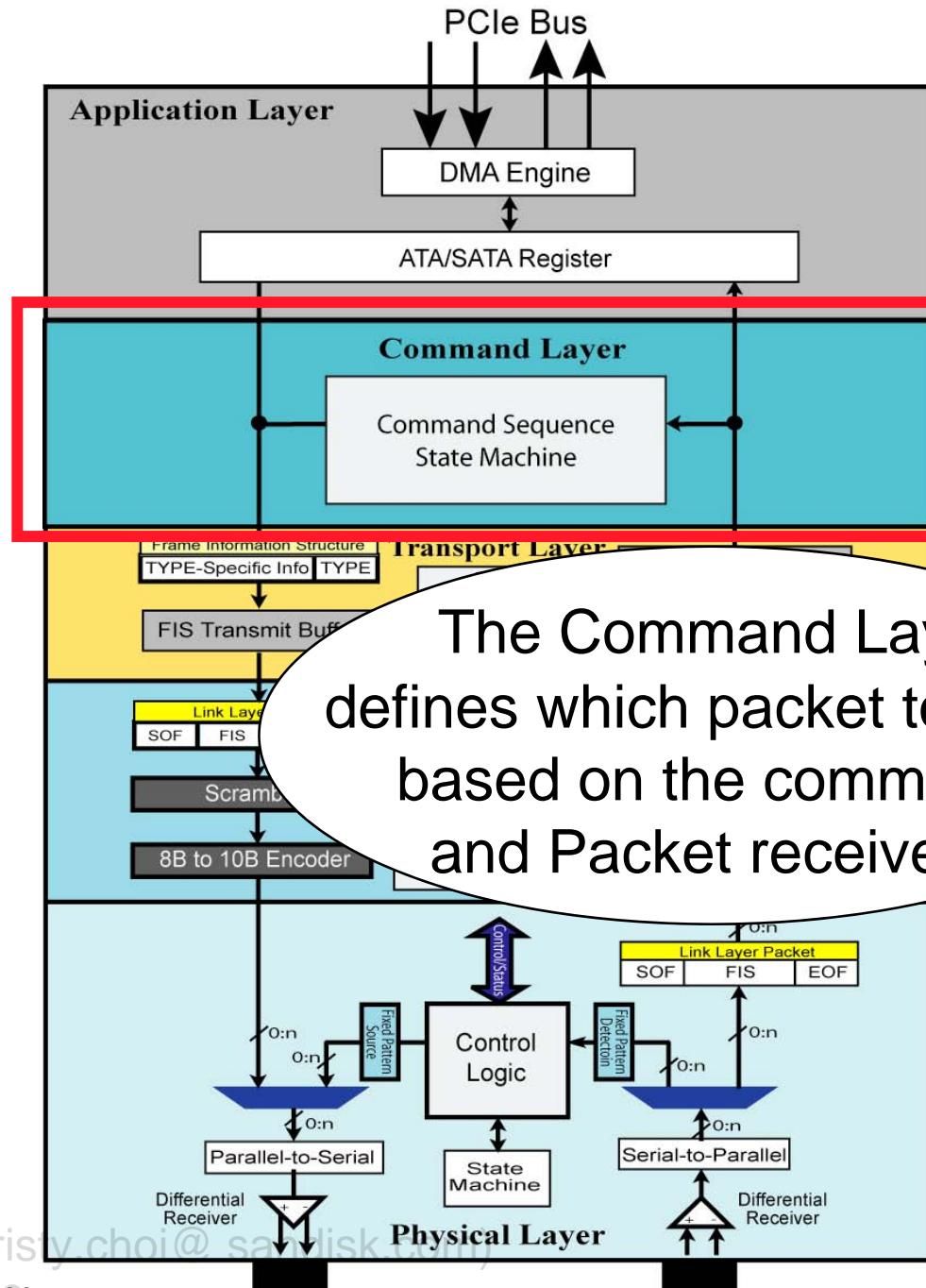
FIS Transmission Protocol

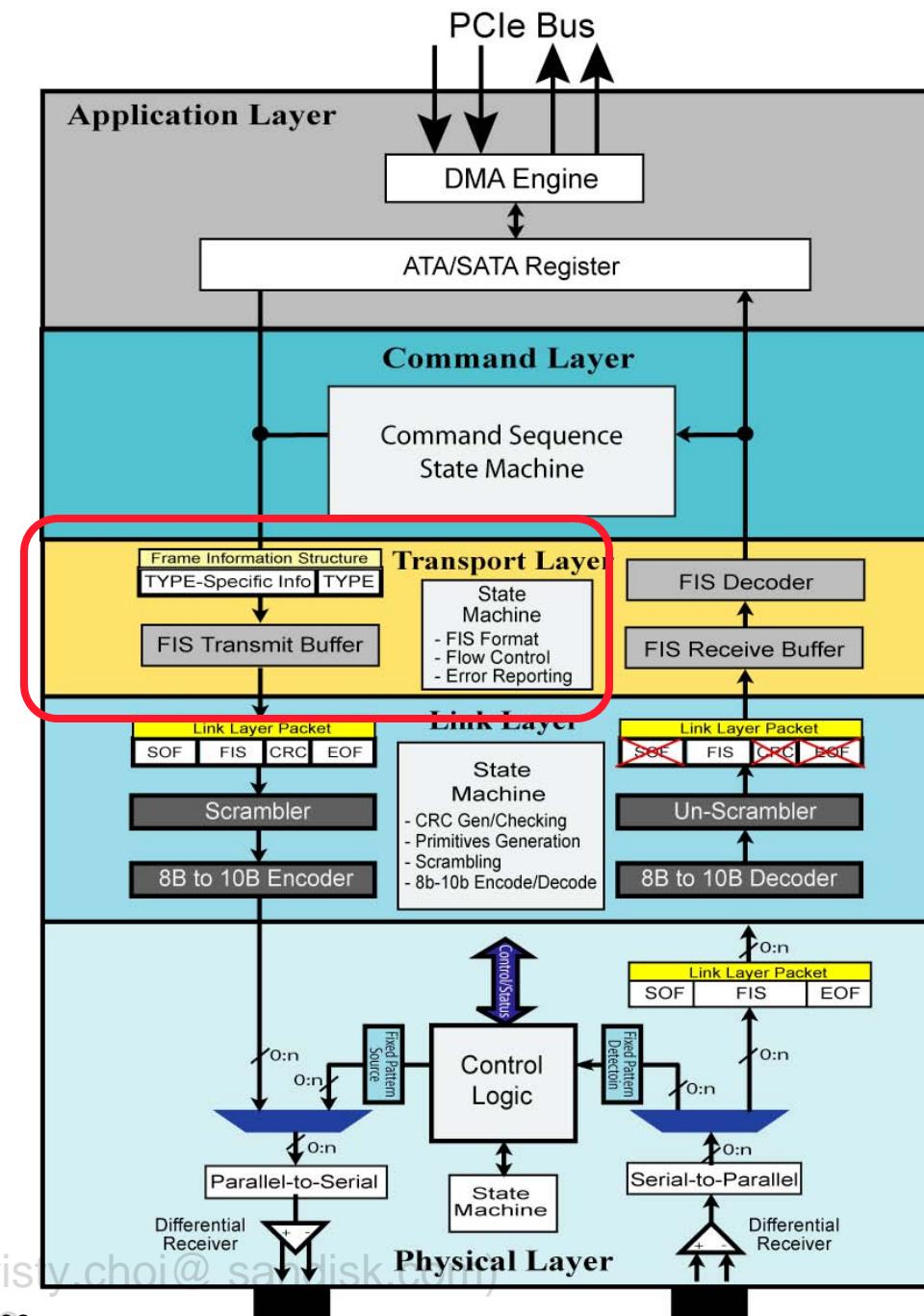


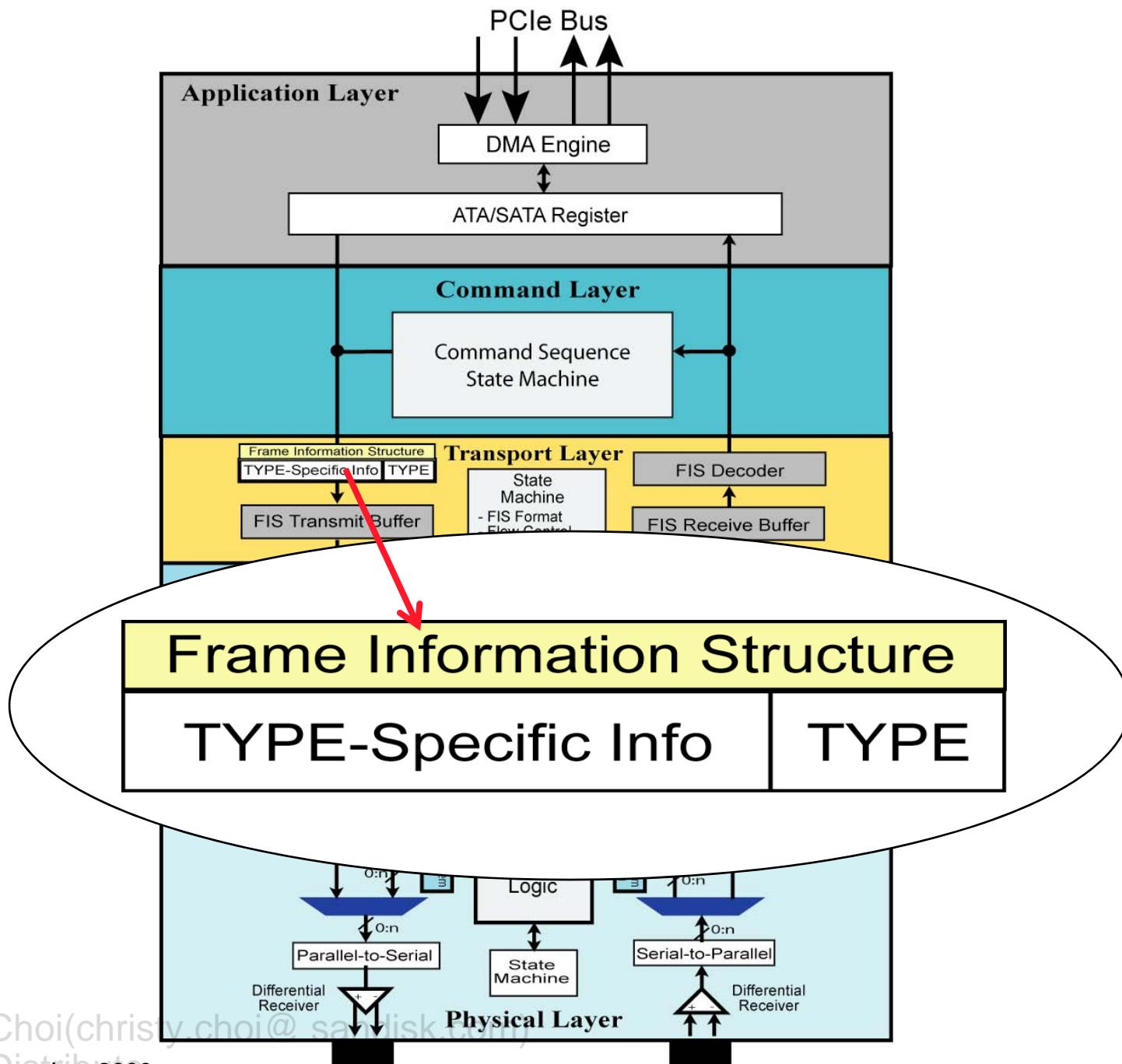
Transmit Functions

The following slides define the transmit functions associated with each layer.





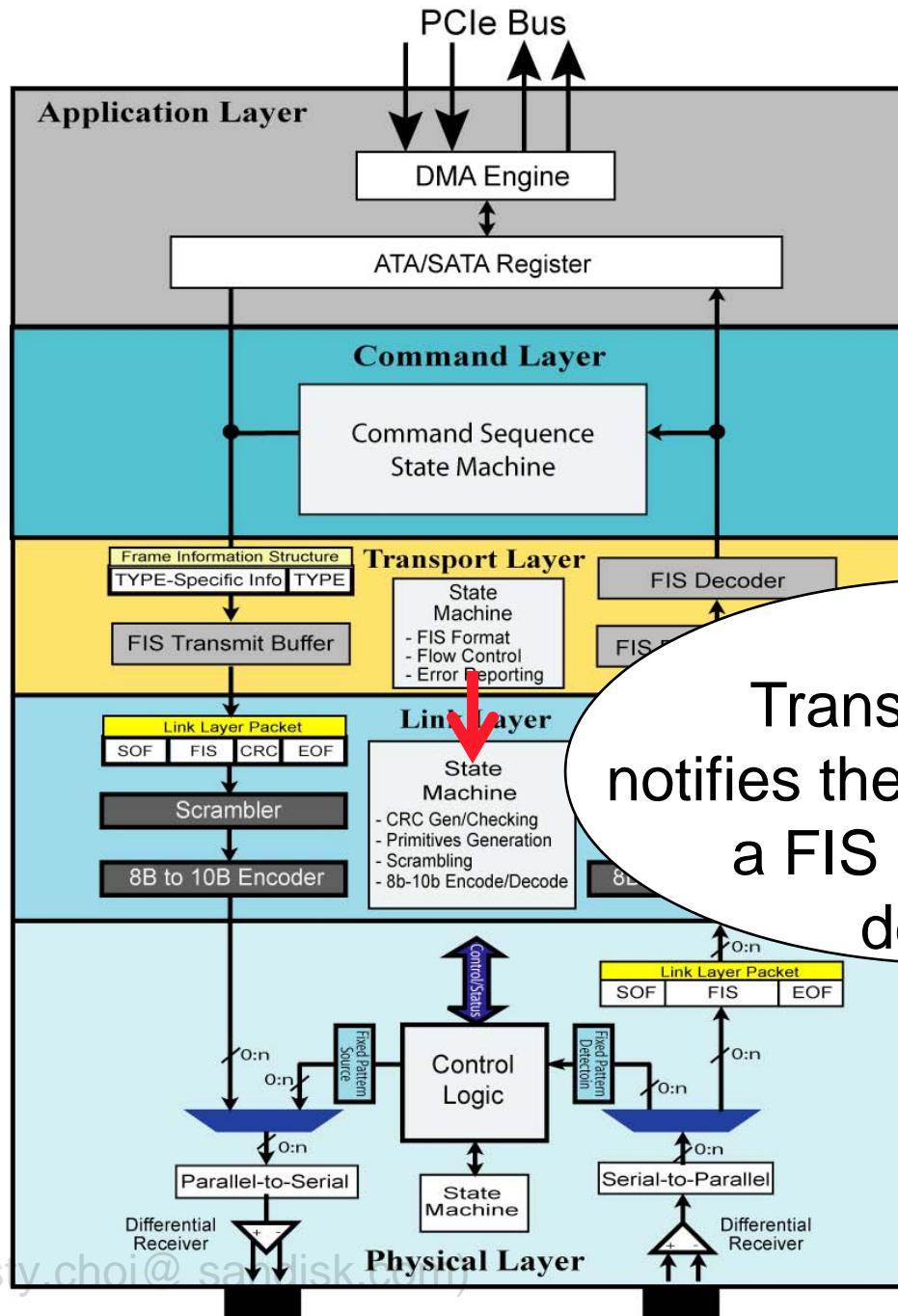




Frame Information Structures

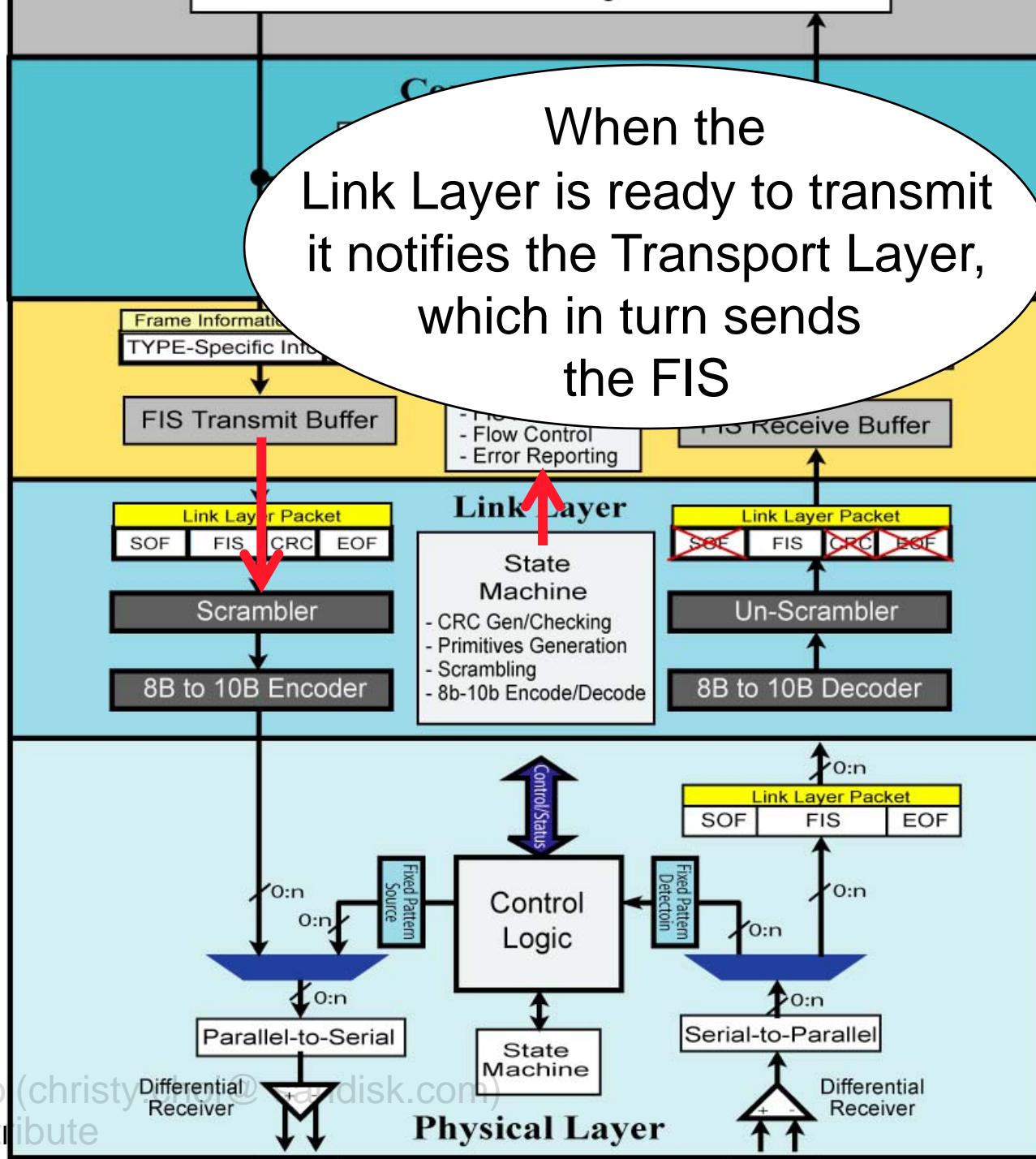
Different FIS's are used for different types of commands

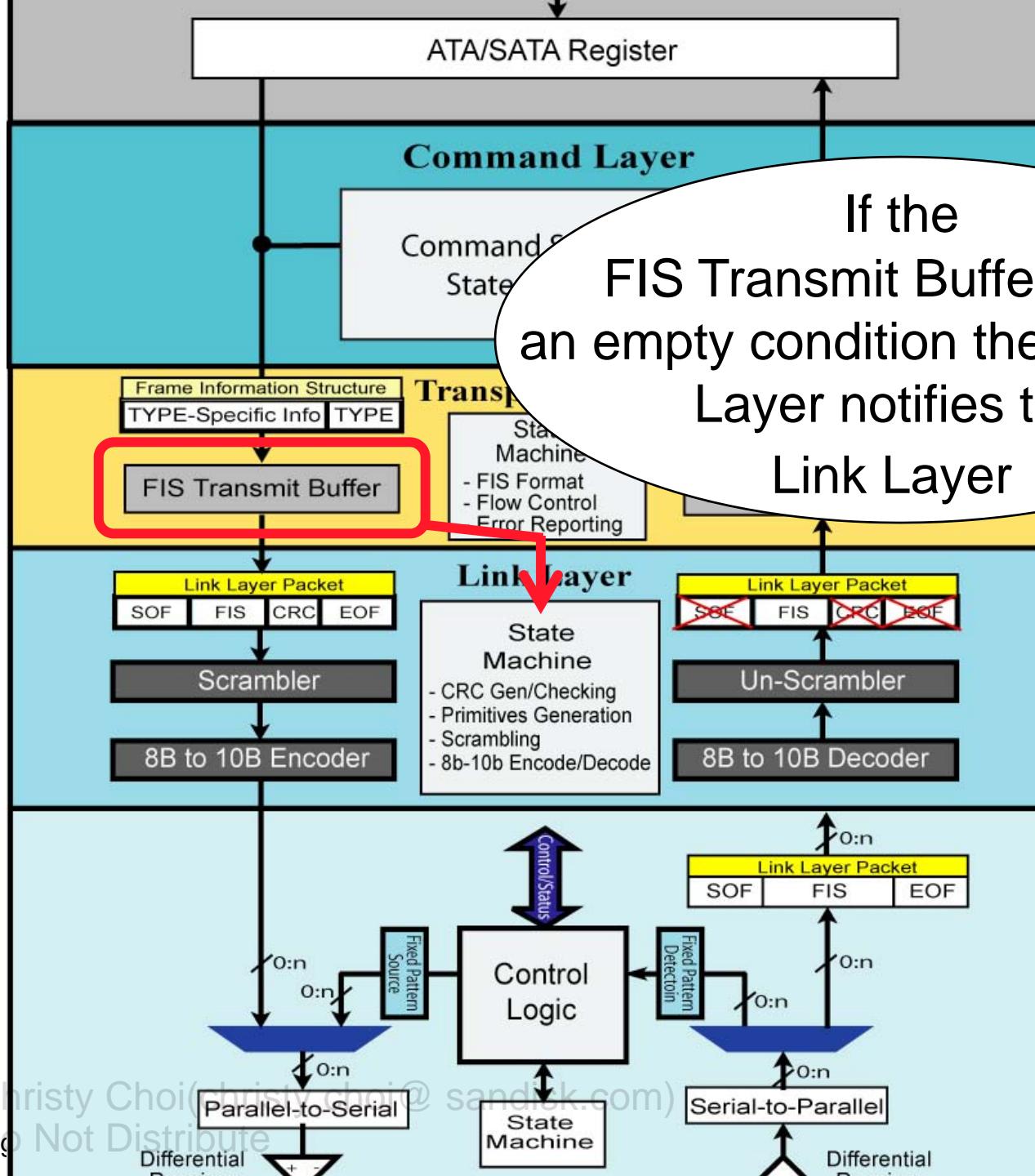
FIS Type	Size	FIS name	Commands
27h	24	Register – Host to Device	N, PIO, DMA, FPDMA
34h	24	Register – Device to Host	N, PIO, DMA, FPDMA
A1h	12	Set Device Bits – Device to Host	DMAQ
5Fh	20	PIO Setup – Device to Host	PIO, ATAPI
39h	8	DMA Activate – Device to Host	DMA
41h	32	DMA Setup – Device to Host	FPDMA
46h	8 to 8196	Data – Device to Host Data – Host to Device	PIO, DMA, FPDMA
58h	16	BIST Activate – Device to Host BIST Activate – Host to Device	BIST



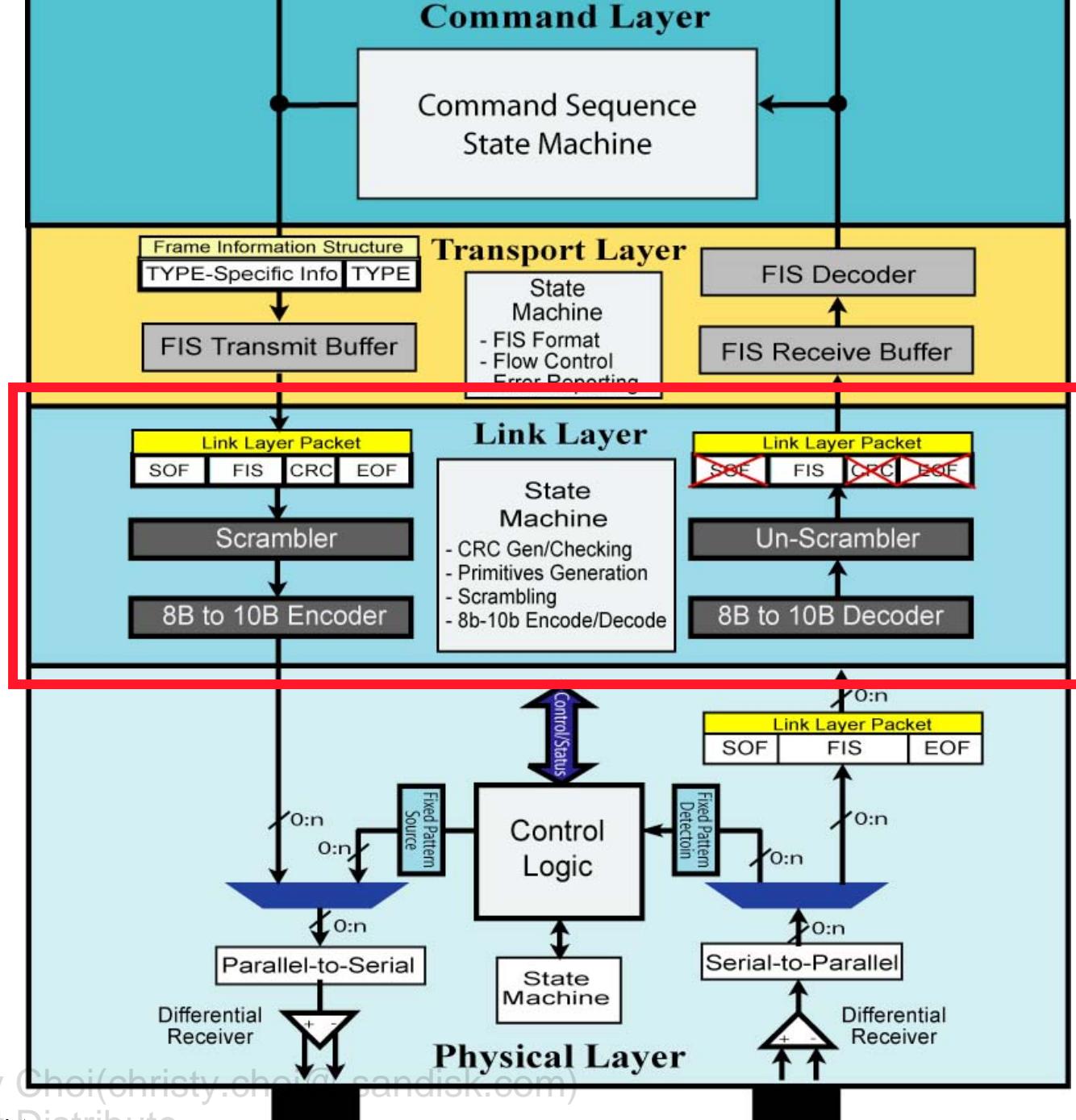
The Transport Layer notifies the Link Layer that a FIS is ready for delivery

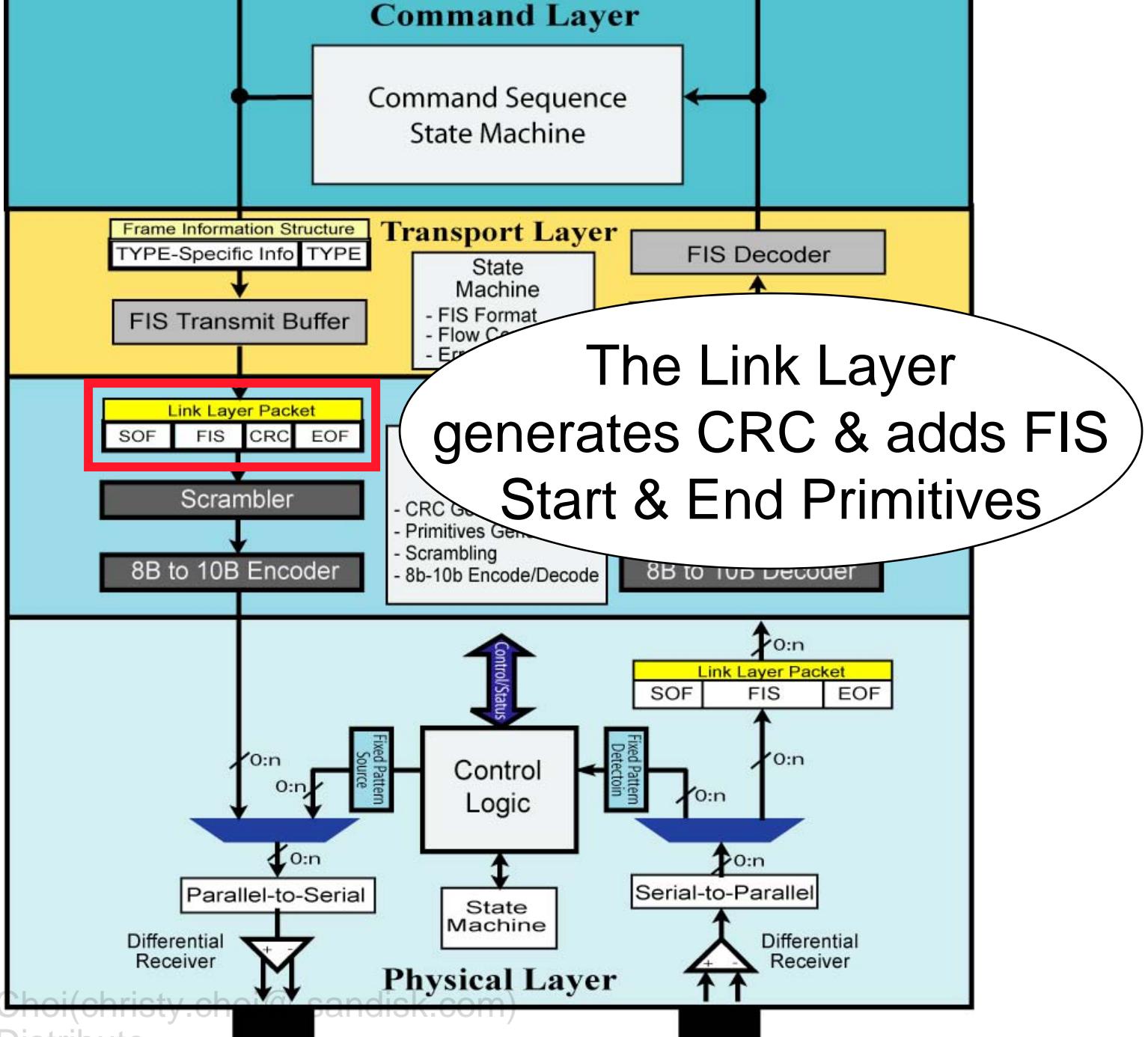
When the Link Layer is ready to transmit it notifies the Transport Layer, which in turn sends the FIS





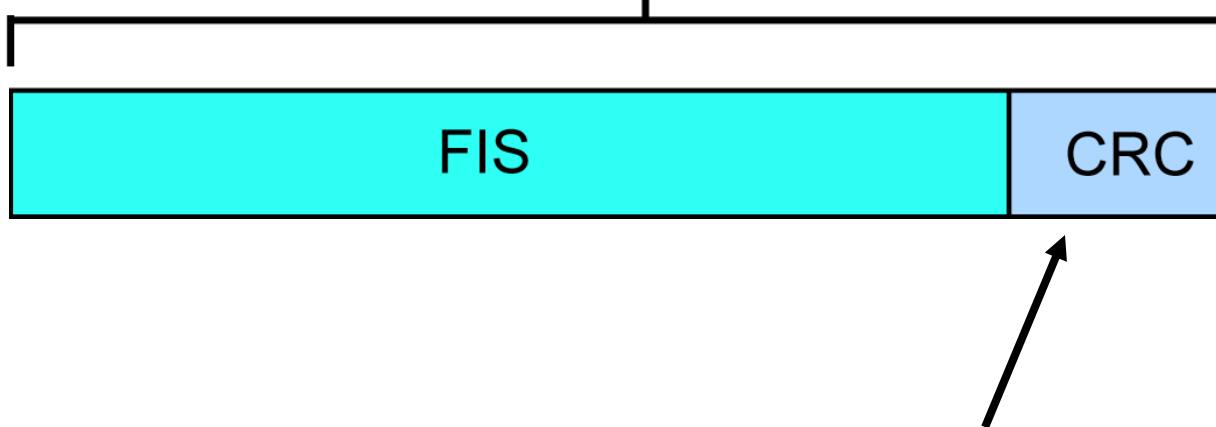
If the FIS Transmit Buffer nears an empty condition the Transport Layer notifies the Link Layer





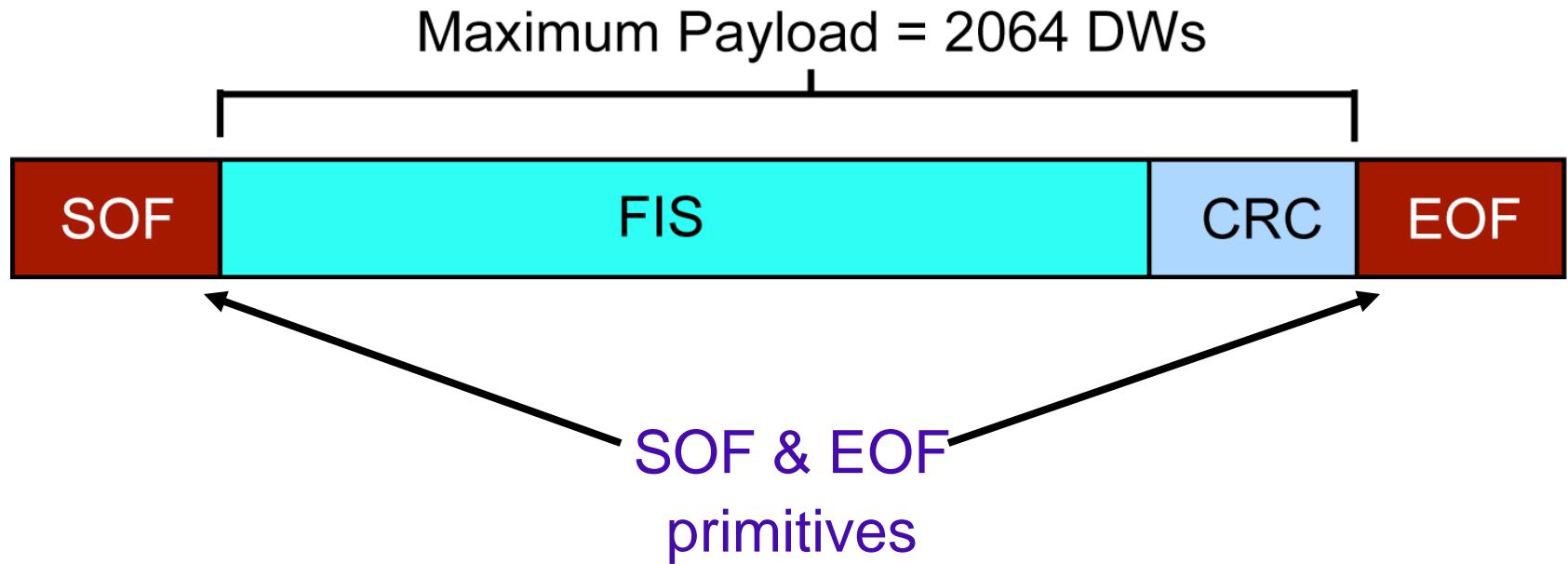
CRC Generation

Maximum Payload = 2064 DWs

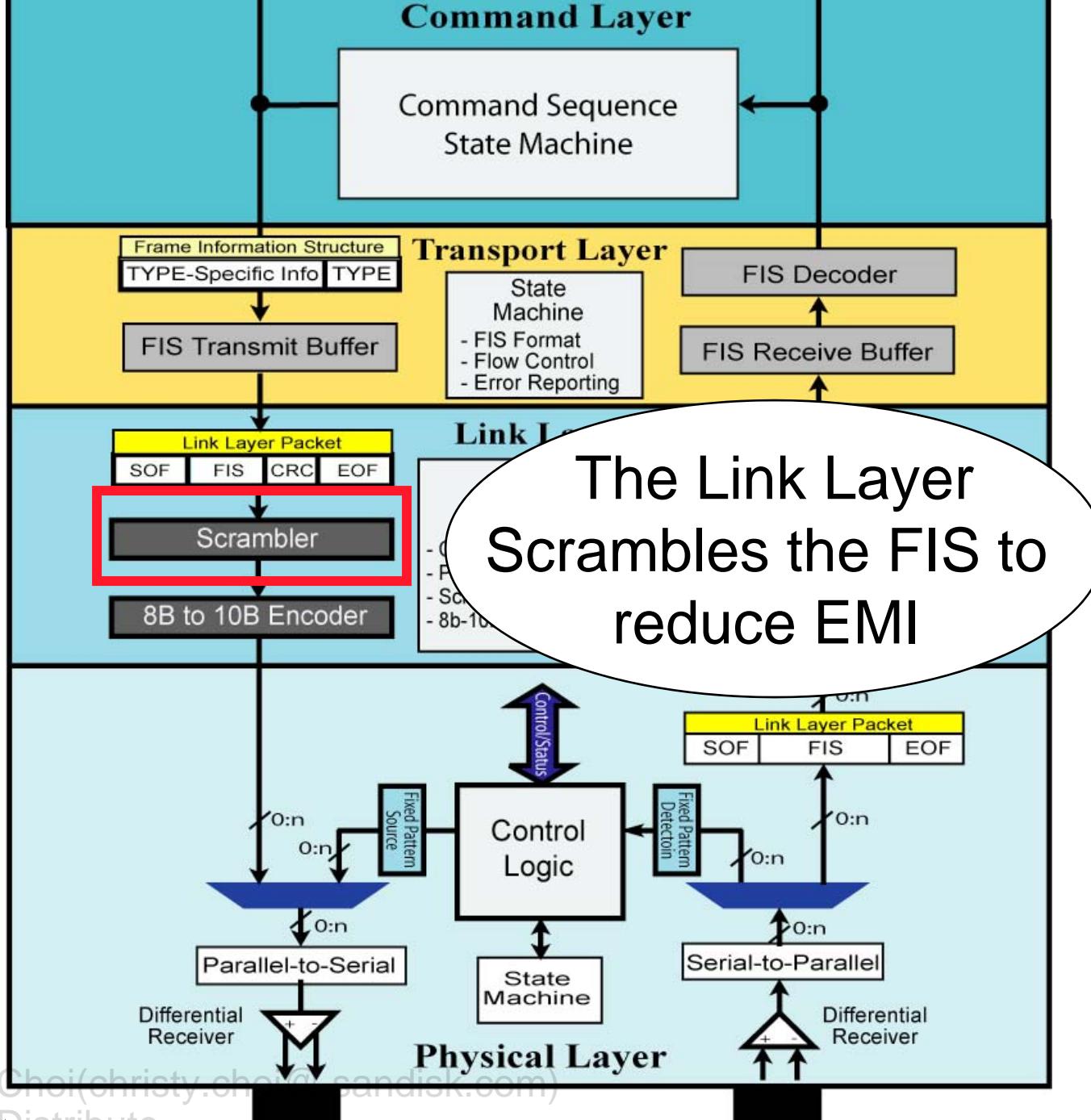


32-bit CRC protects the FIS contents

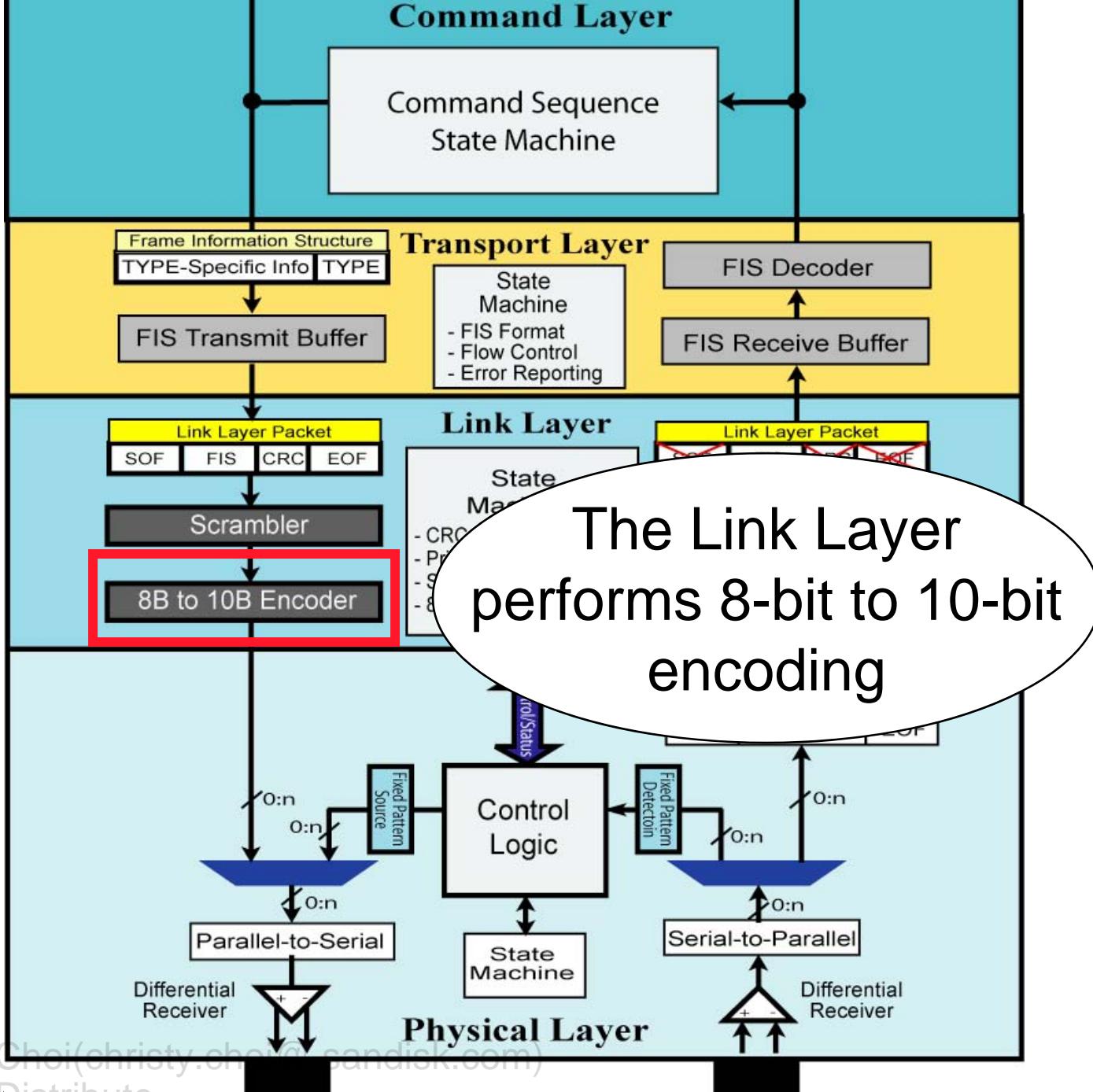
SOF/EOF



Note: Each Primitive consist of a sequence of 4 bytes



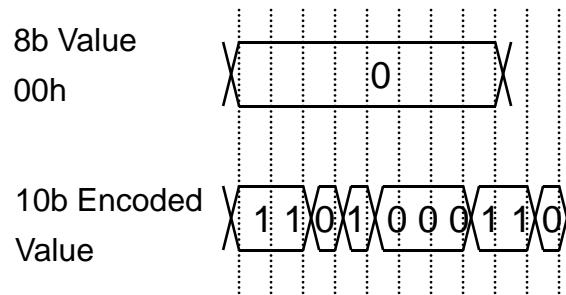
The Link Layer Scrambles the FIS to reduce EMI

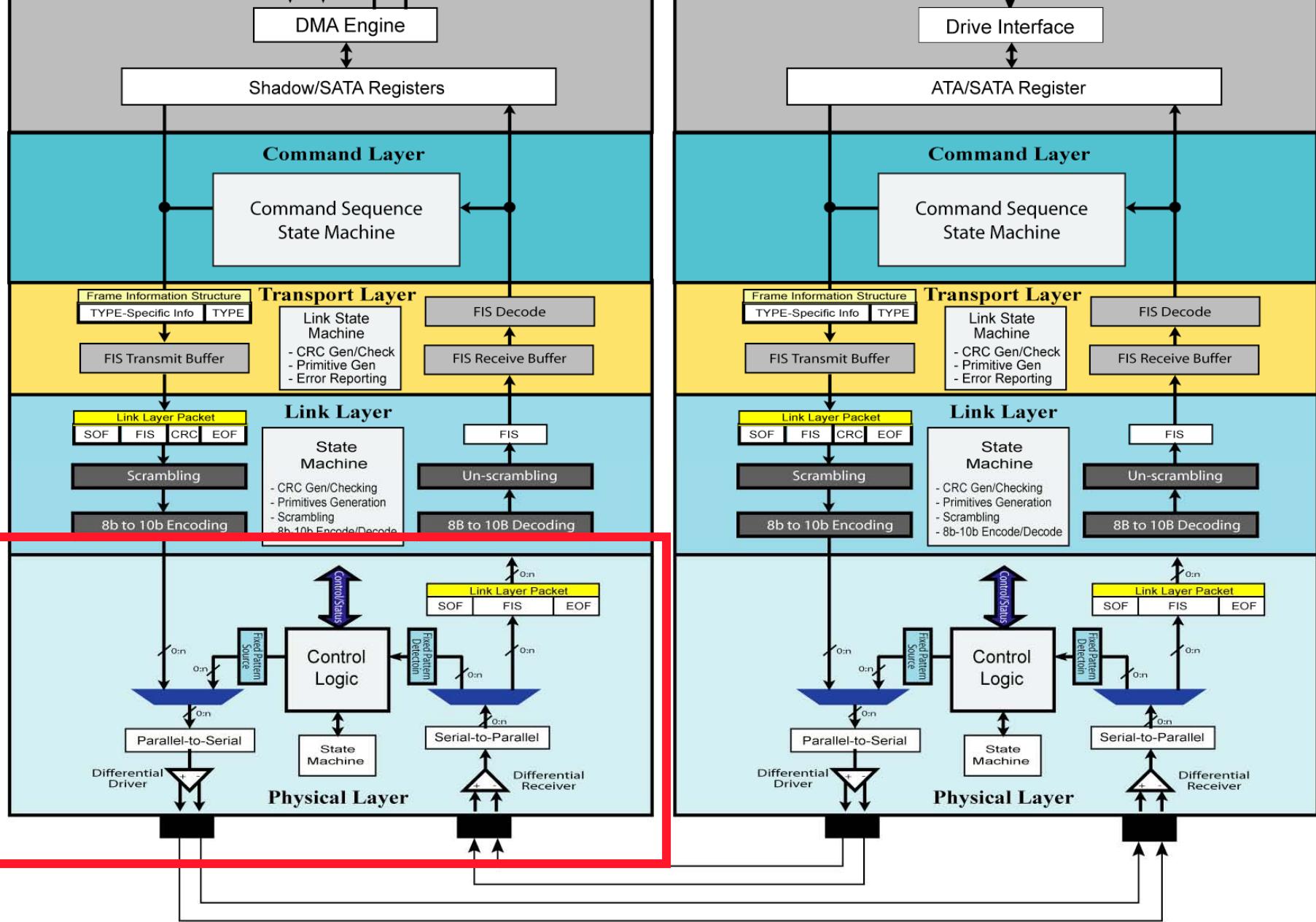


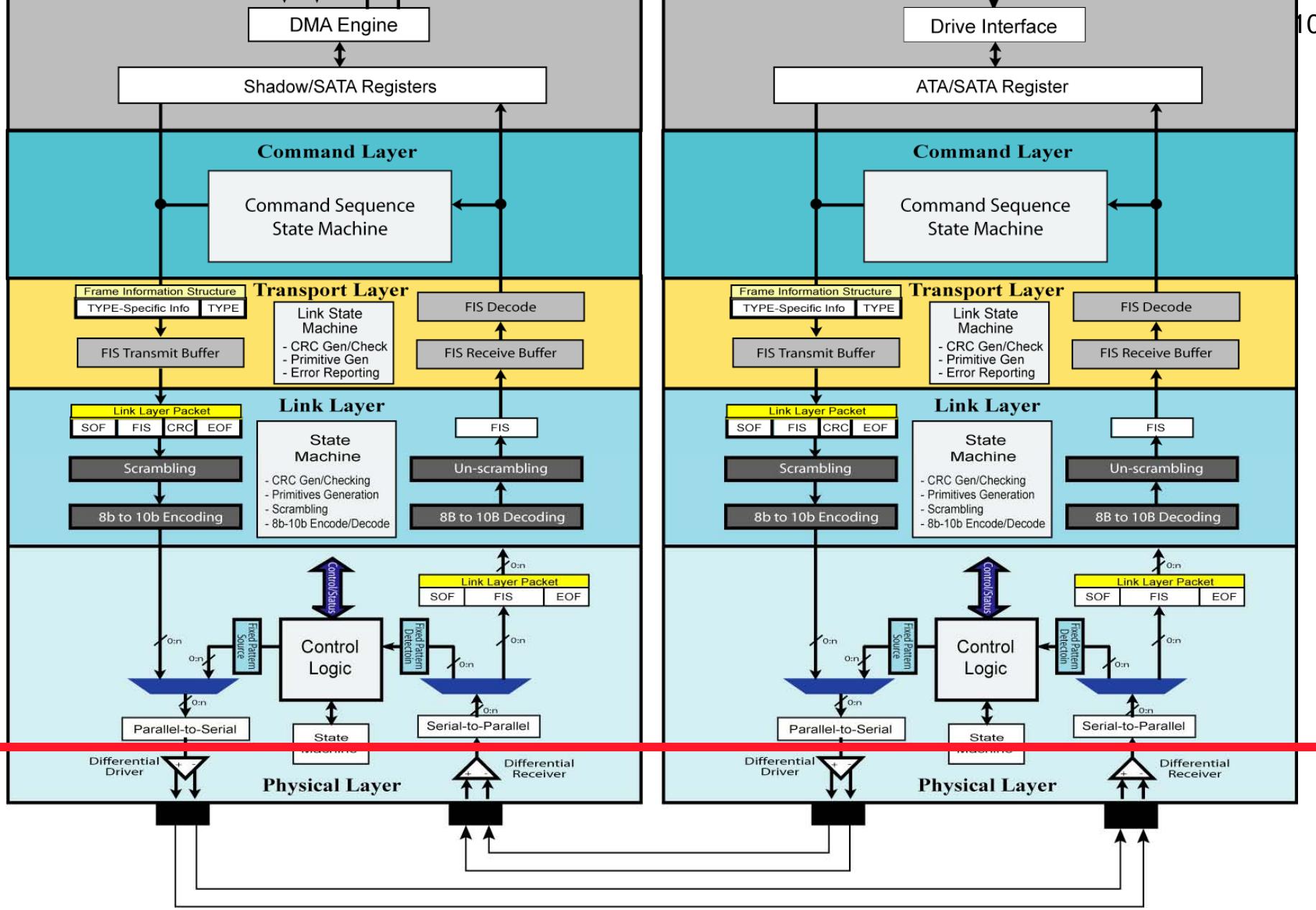
8b/10b Encoding Example

Each byte of the packet is converted to a 10-bit value prior to being sent to the physical layer for transmission. This encoding is used by many serial transports and has several advantages:

- Ensures sufficient transition density to embed a clock into the data stream
- Provides DC balance over time
- Facilitates detection of transmission errors

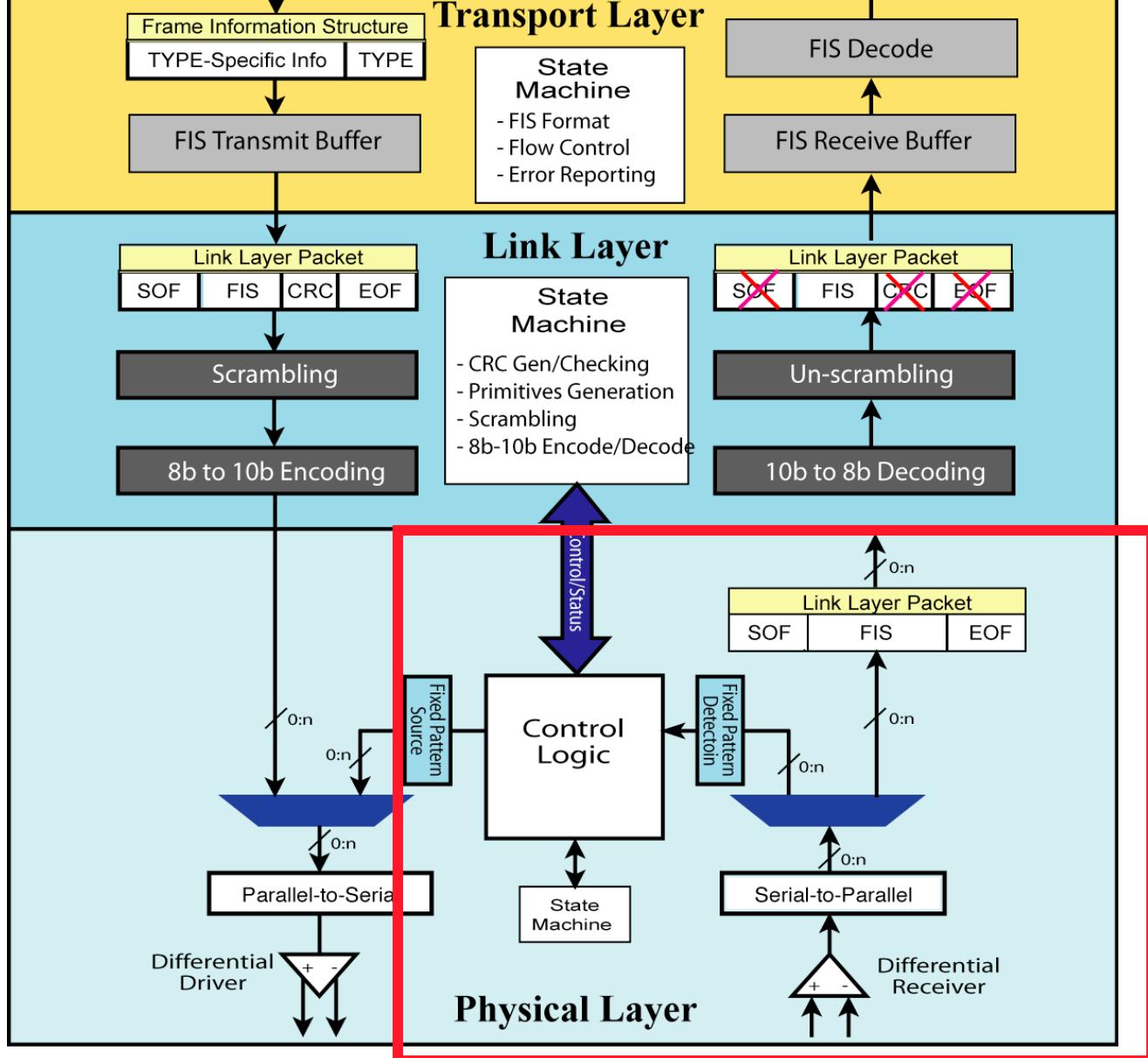




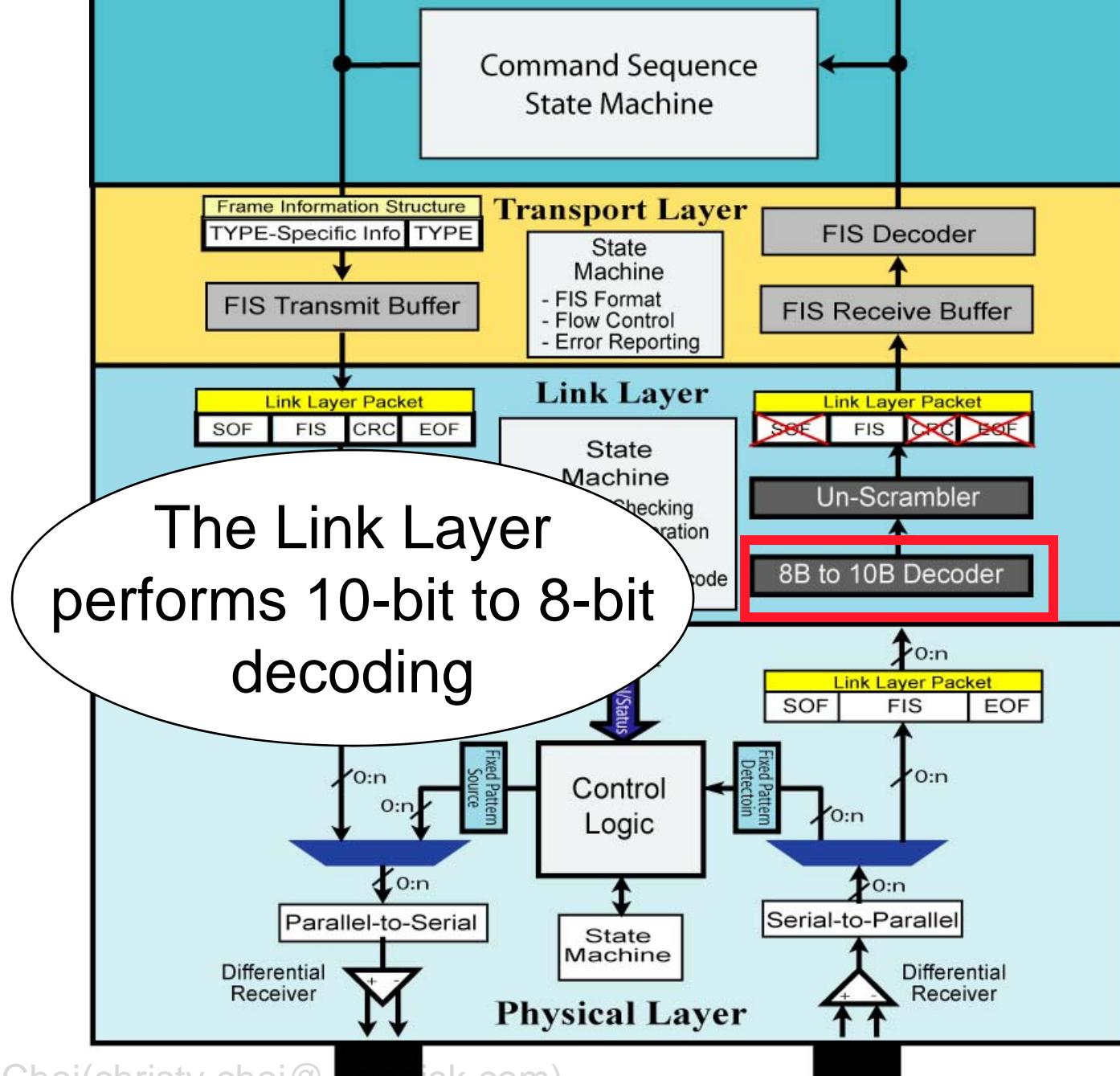


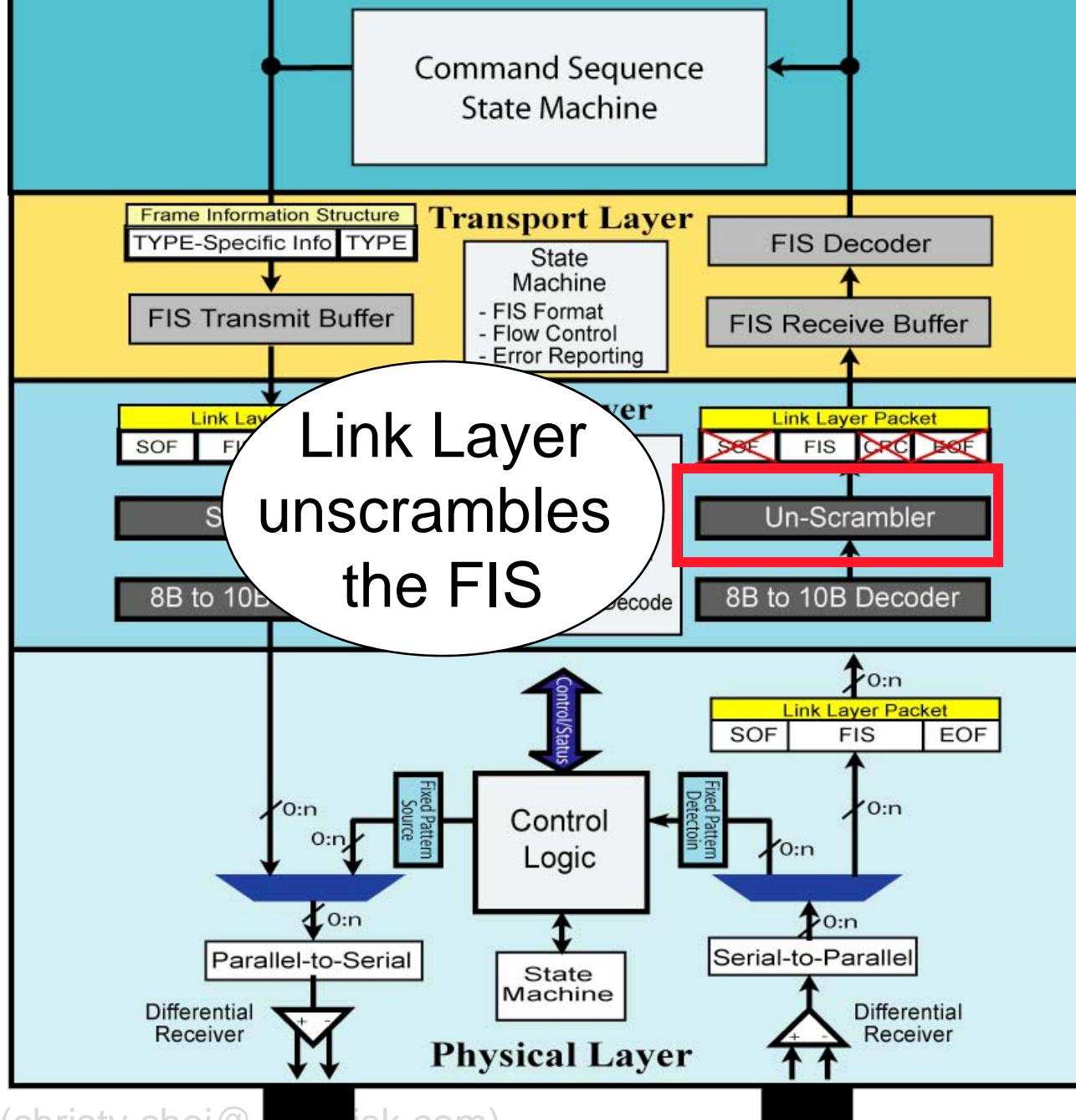
Receive Functions

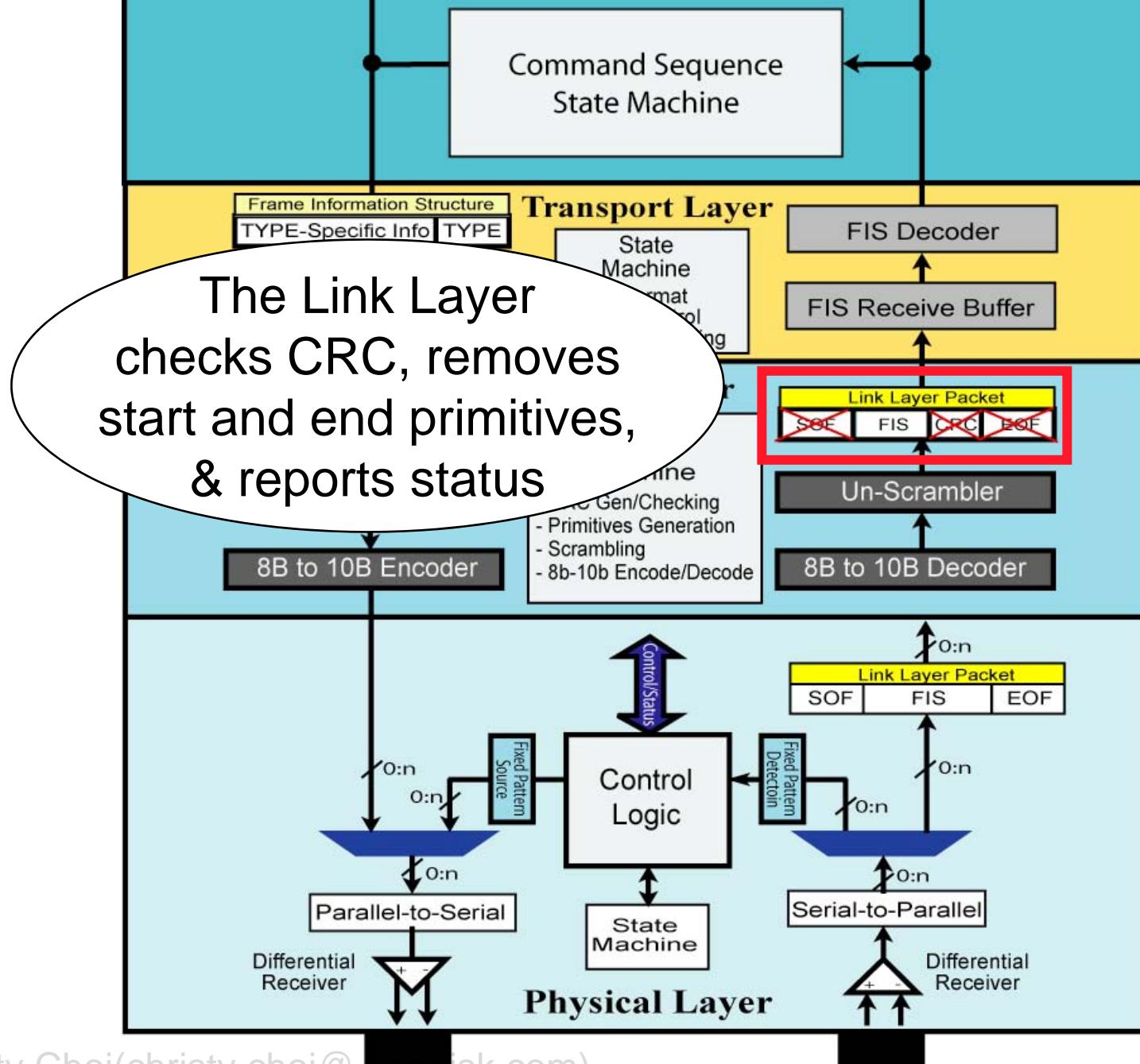
The following slides define the receive functions associated with each layer.



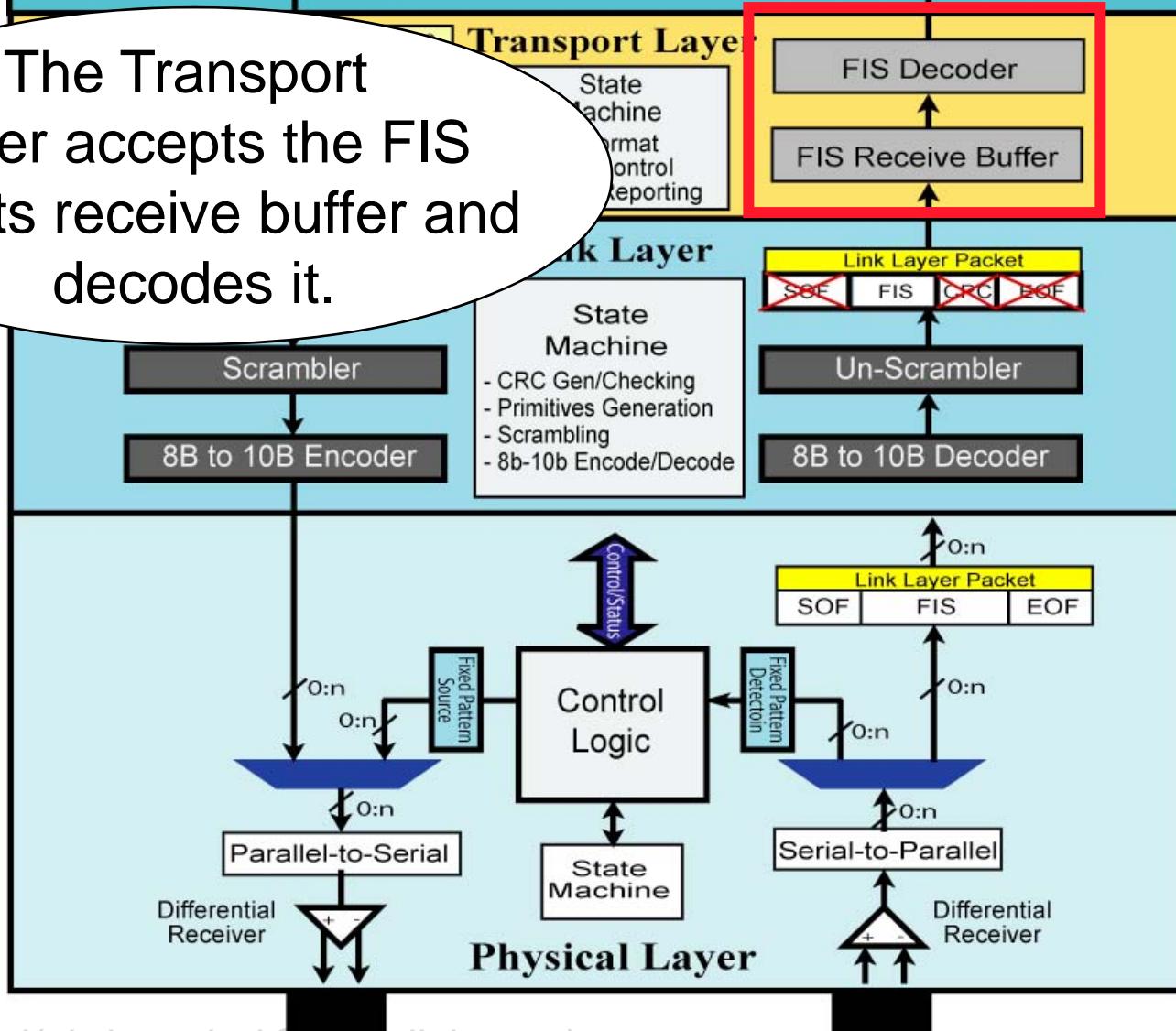
Changing now to the receiver side







The Transport Layer accepts the FIS into its receive buffer and decodes it.

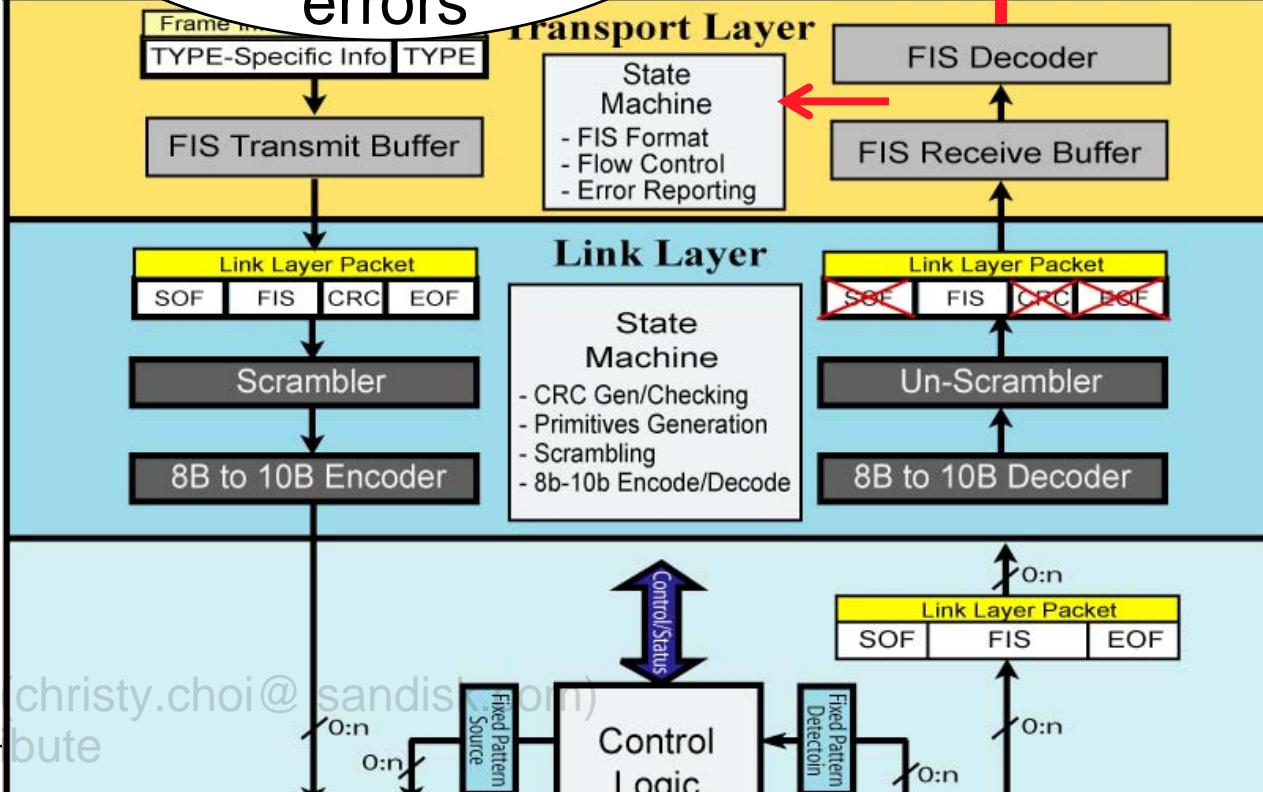


Application Layer

Drive Interface

SATA Register

The Transport Layer forwards the FIS to the Cmd/App Layers and reports any errors



Part 2

FIS Transmission

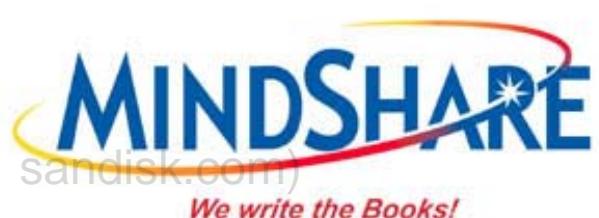
Protocols

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute



FIS Types and Format



Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

Register FIS - Host to Device

	+3	+2	+1	+0
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
DW 0	Features	Command	C R R Reserved	FIS Type (27h)
DW 1	Dev/Head	Cyl High	Cyl Low	Sector Number
DW 2	Features (exp)	Cyl High (exp)	Cyl Low (exp)	Sec Num (exp)
DW 3	Control	Reserved (0)	Sec Count (exp)	Sector Count
DW 4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

See Table 5-2 (page 88-89) for field descriptions

C = 0 write to Control Register

C = 1 write to Command Register

Register Write Values

Cmd Reg	Writes	Notes
Address	7	0
01F0	Data	16-bit accesses
01F1	Feature	Two 8-bit accesses
01F2	Sector Count	Two 8-bit accesses
01F3	LBA Low (31:24 then 7:0)	Two 8-bit accesses
01F4	LBA Middle (39:32 then 15:8)	Two 8-bit accesses
01F5	LBA High (47:40 then 23:16)	Two 8-bit accesses
01F6	Device	8-bit access only
01F7	Command	8-bit access only
Ctrl Reg		
03F6	Device Control	8-bit access only

Register FIS - Dev to Host

	+3	+2	+1	+0
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
DW 0	Error	Status	R I R Reserved	FIS Type (34h)
DW 1	Dev/Head	Cyl High	Cyl Low	Sector Number
DW 2	Reserved (0)	Cyl High (exp)	Cyl Low (exp)	Sec Num (exp)
DW 3	Reserved (0)	Reserved (0)	Sec Count (exp)	Sector Count
DW 4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

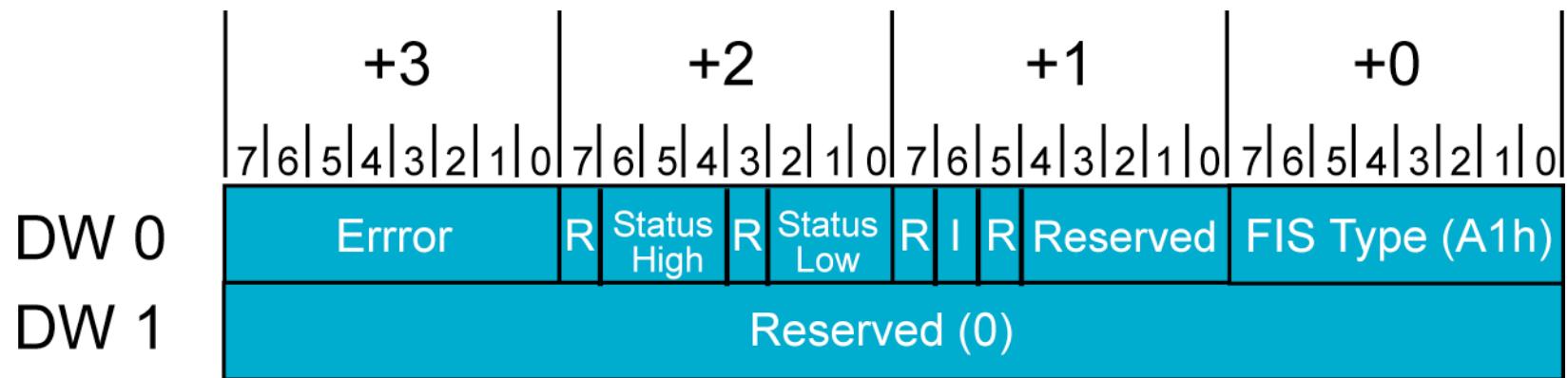
See Table 5-3 (page 91-92) for field descriptions.

I – represents the state of the interrupt line within the device.

Drive's ATA Reg Content when Read

Cmd Reg	Reads	Notes
Address	7	0
01F0	Data	Two 8-bit accesses
01F1	Error	8-bit access
01F2	Sector Count	Two 8-bit accesses
01F3	LBA Low (24:31 then 7:0)	Two 8-bit accesses
01F4	LBA Middle (32-39 then 15:8)	Two 8-bit accesses
01F5	LBA High (40-47 then 23:16)	Two 8-bit accesses
01F6	Device	8-bit access
01F7	Status	8-bit access
Ctrl Reg		
03F6	Alternate Status	8-bit access

Set Device Bits FIS - Dev to Host

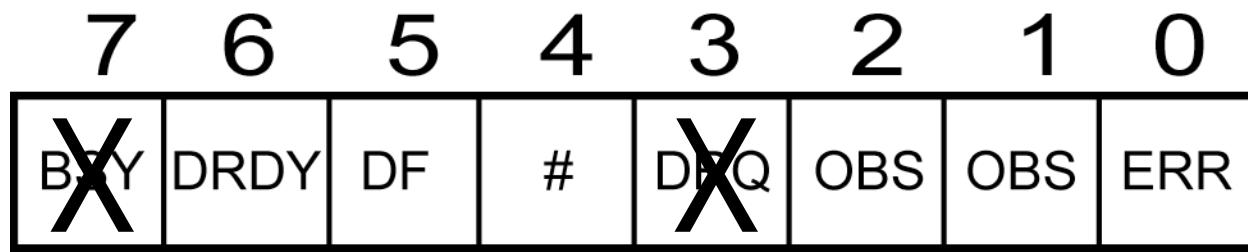


R = Reserved (0)

I = Interrupt Bit

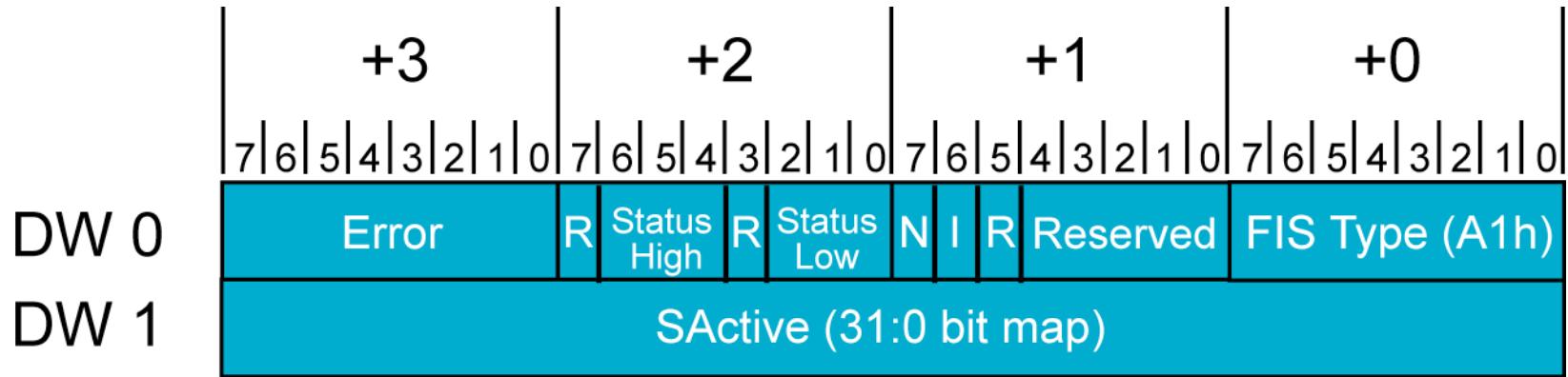
Device has exclusive access to all 8 bits of Error register and 6 bits of Status register. This FIS is used to change those bits regardless of the BSY setting.

ATA Status Register Bits Modified

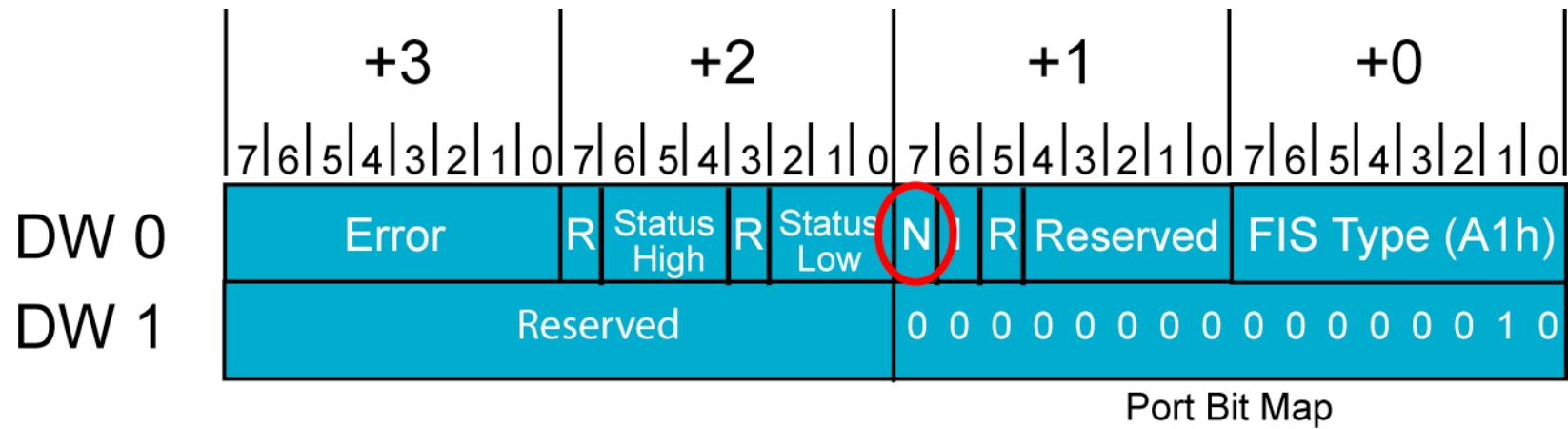


OBS = obsolete

Set Device Bits with Active Field



Set Device Bits with Event Notification



PIO Setup FIS - Dev to Host

	+3	+2	+1	+0
DW 0	Error	Status	R I D Reserved	FIS Type (5Fh)
DW 1	Dev/Head	Cyl High	Cyl Low	Sector Number
DW 2	Reserved (0)	Cyl High (exp)	Cyl Low (exp)	Sec Num (exp)
DW 3	E_Status	Reserved (0)	Sec Count (exp)	Sector Count
DW 4	Reserved (0)		Transfer Count	

R = Reserved (0)

I = Interrupt Bit

D = Direction of Transfer 0=Read from Host Memory

DMA_Setup FIS

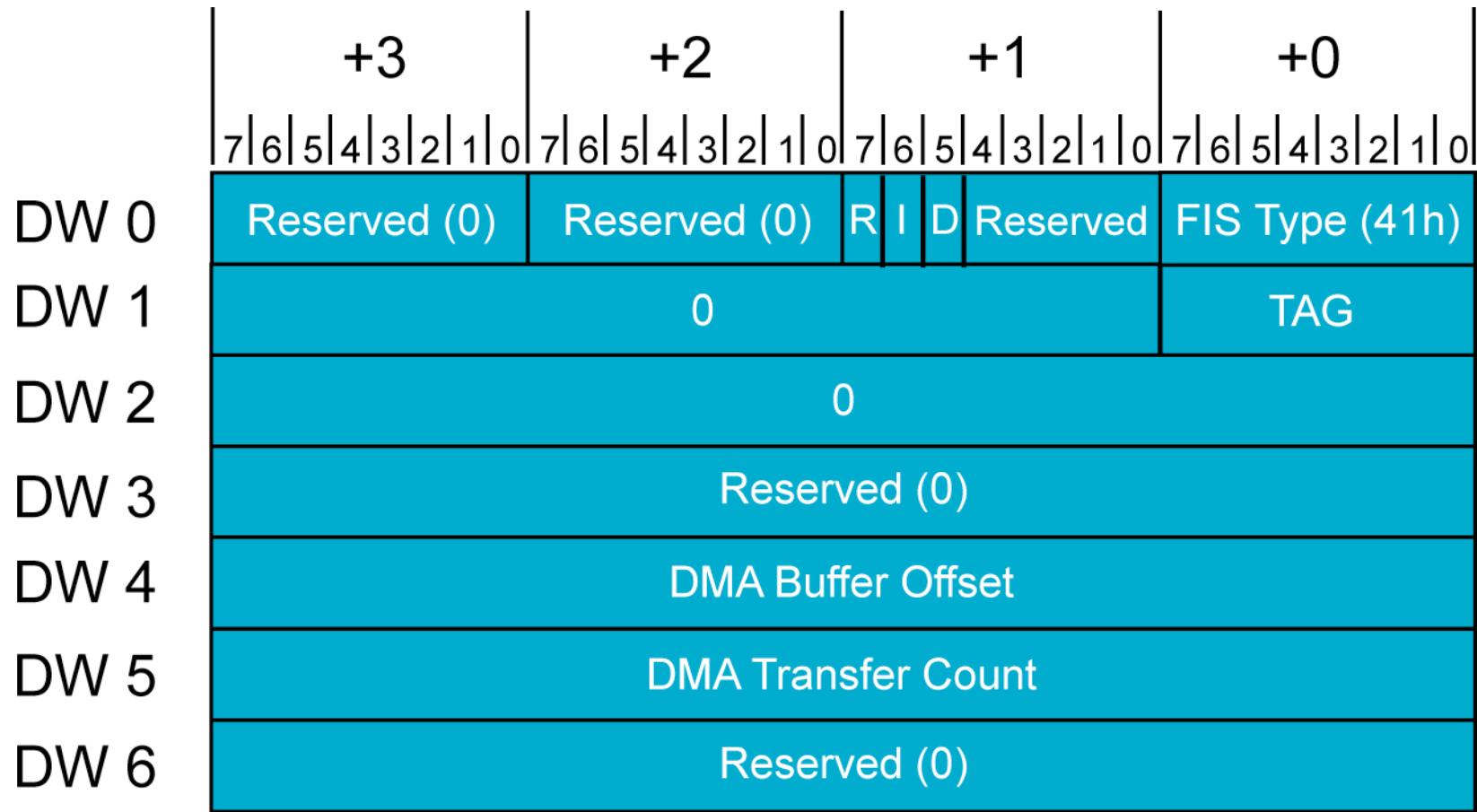
	+3	+2	+1	+0
DW 0	Reserved (0)	Reserved (0)	R I D Reserved	FIS Type (41h)
DW 1	DMA Buffer Identifier Low			
DW 2	DMA Buffer Identifier High			
DW 3	Reserved (0)			
DW 4	DMA Buffer Offset			
DW 5	DMA Transfer Count			
DW 6	Reserved (0)			

R = Reserved (0)

I = Interrupt Bit

D = Direction of Transfer 0=Read from Host Memory

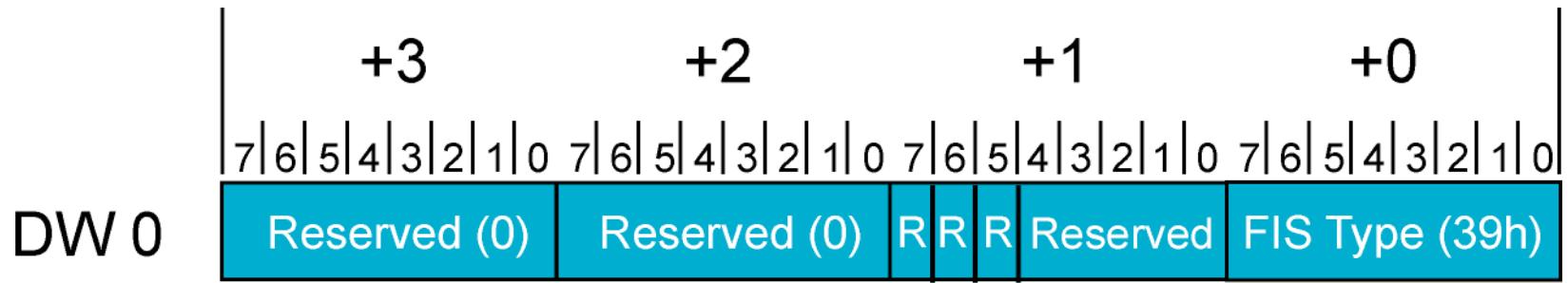
DMA Setup FIS 1st Party DMA



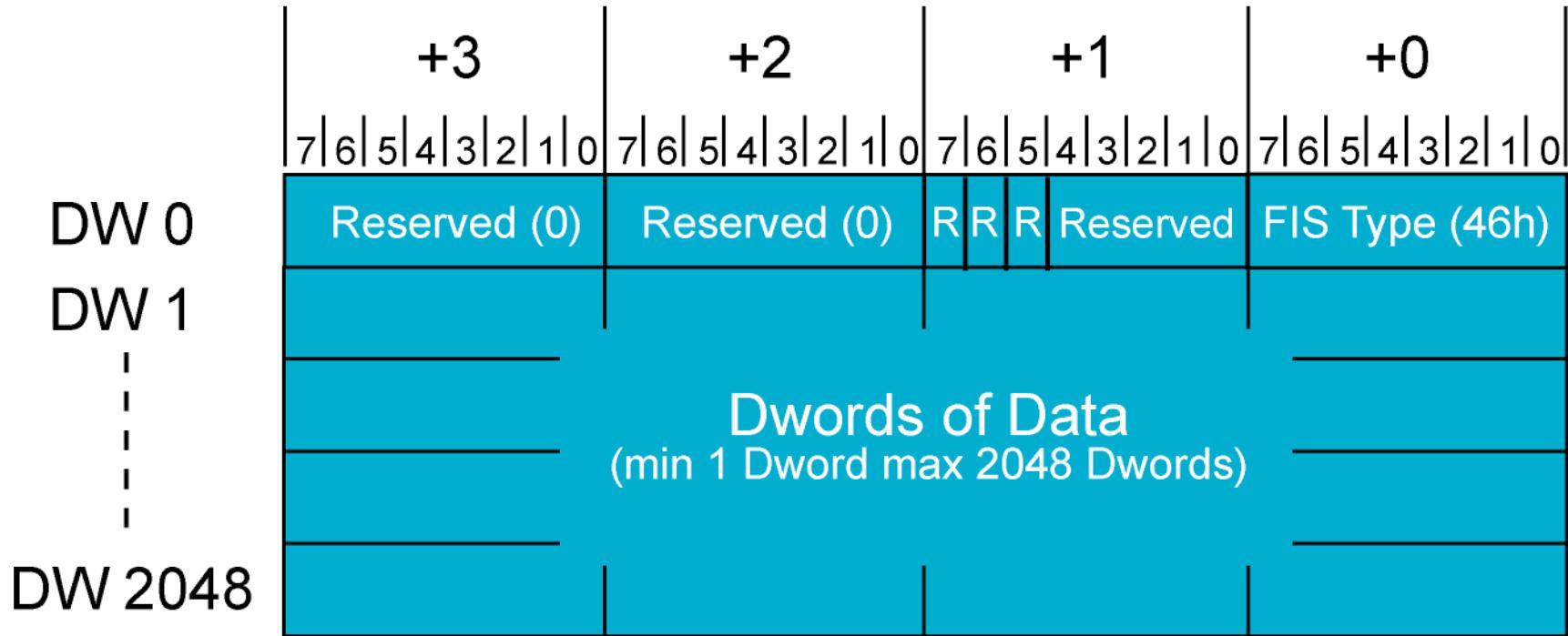
DMA Setup with Auto Activate

	+3	+2	+1	+0
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
DW 0	Reserved (0)	Reserved (0)	A I D Reserved	FIS Type (41h)
DW 1	0			TAG
DW 2	0			
DW 3	Reserved (0)			
DW 4	DMA Buffer Offset			
DW 5	DMA Transfer Count			
DW 6	Reserved (0)			

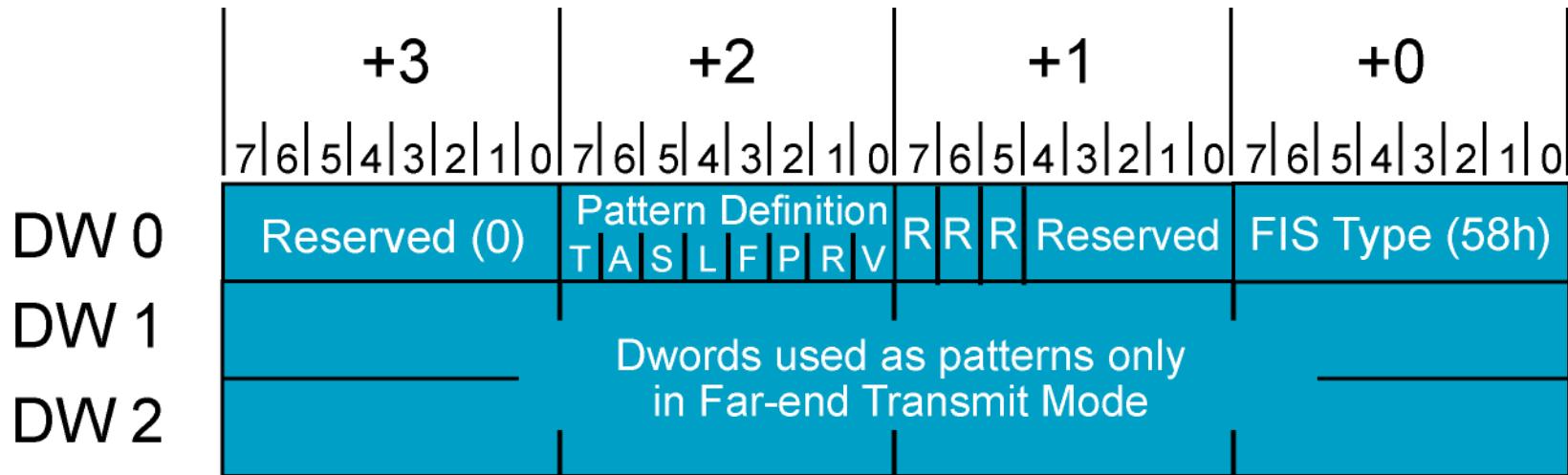
DMA Activate FIS Device to Host



Data FIS (bidirectional)



BIST_Activate (bidirectional) FIS



T = Far end transmit only mode

A = ALIGN Bypass (only when T=1)

S = Bypass Scrambling (only when T=1)

L = Far End Retimed

F = Far End Analog Loopback

P = Primitive bit (only when T=1)

R = Reserved (0)

V = Vendor Unique Test Mode

Transport & Link Protocols

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute



FIS Transfer Protocols

- This section describes FIS transmission and reception in more detail. Note that the protocol described is principally the same for FIS transmission in both directions.
- The discussion assumes that:
 - sufficient buffer space is available within the transmitter (Transport Layer) so that no delays are incurred
 - the receiver (Transport Layer) has sufficient buffer space to accept the entire FIS
 - no errors occur during the transmission
- Later sections deal with Flow Control, Error handling, FIS retry, and other aspects of the protocol.

The Application Layer Triggers FIS Transfer Protocol

- Host software writes to the HBA shadow registers, triggering the transport layer to send a FIS; or the HBA fetches a FIS from memory, triggering the transport layer to send it.
- The application layer of SATA devices responds to host initiated FIS's by requesting a FIS be sent back to the host.

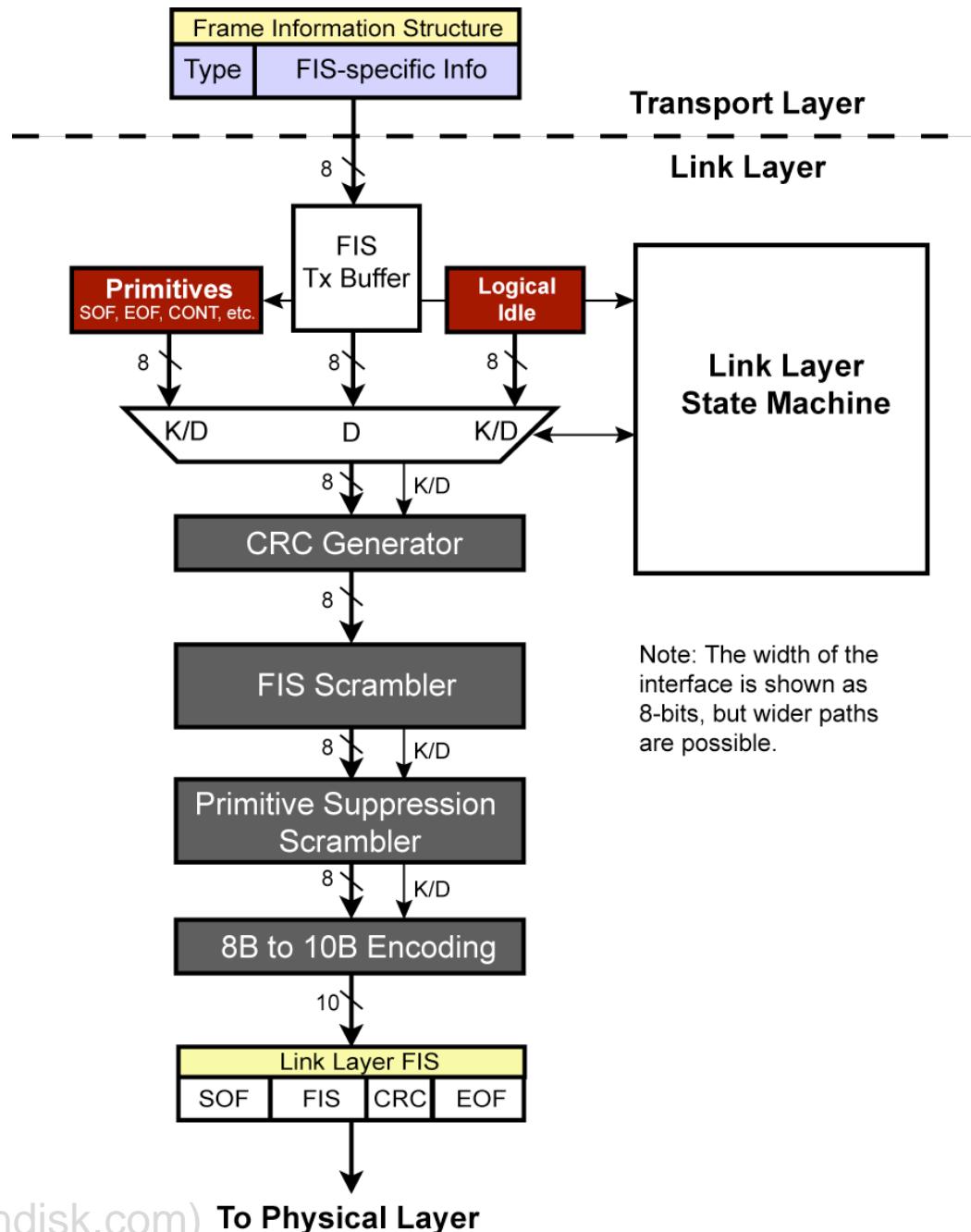
Transport Layer FIS Transmission

The following types of FISs can be sent by the Transport Layer:

- Register FIS (Host or Device)
- Set Device Bits in Shadow Regs (Device)
- PIO Setup FIS (Device)
- Data (Host & Device)
- 1st Party DMA Setup (Host or Device)
- DMA Activate (Device)
- BIST Activate FIS (Host or Device)

FIS/Primitive Transmission (Link Layer)

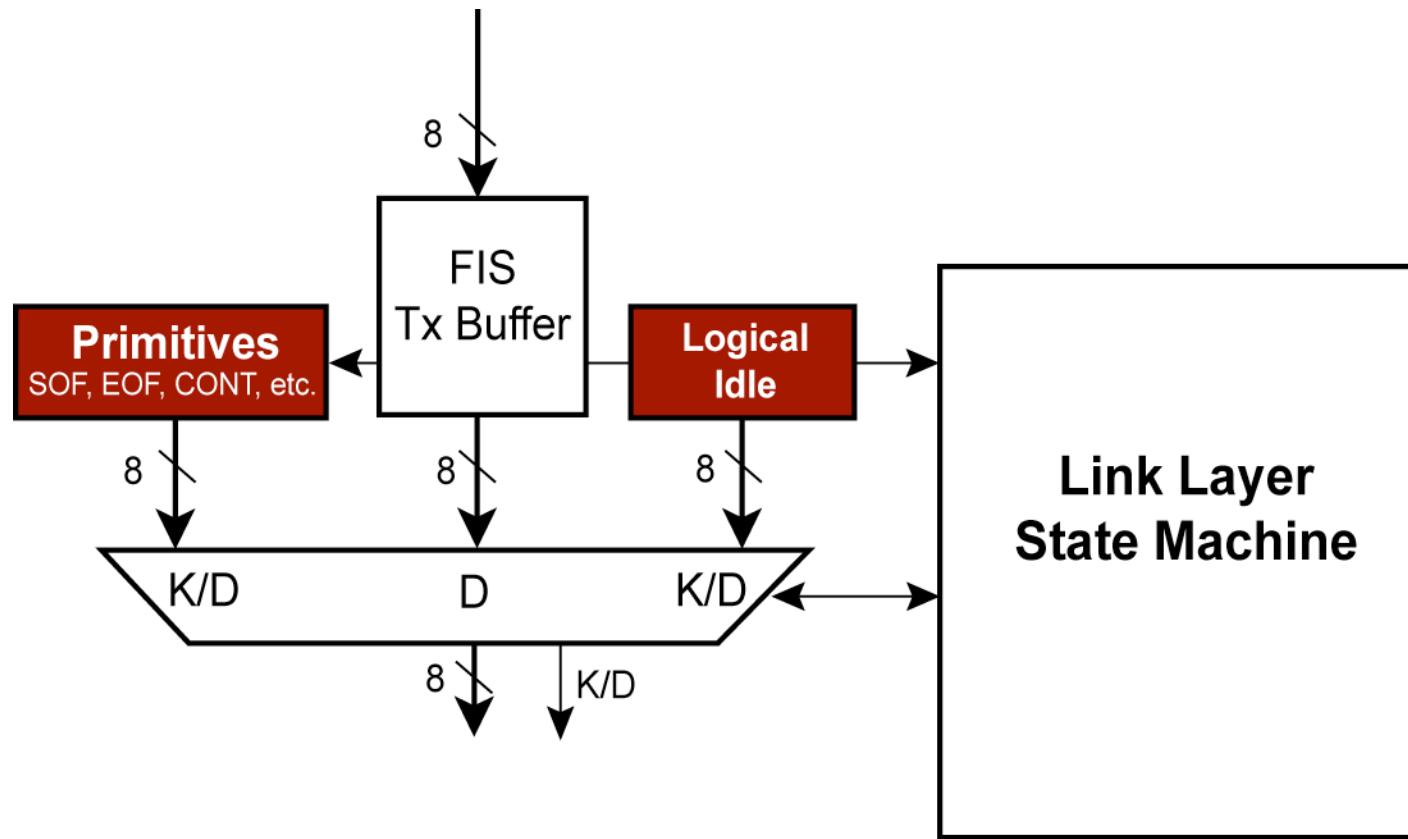
- Link Layer manages event sequences associated with sending a FIS



FIS Transfers & Related Primitives

Primitive	Name	Purpose
SYNC	Synchronization	Used to indicate a logical bus idle condition
X_RDY	Transmitter Ready	Host or Device ready to transmit data
R_RDY	Receiver Ready	Host or Device ready to receive
SOF	Start of Frame	Beginning of frame data payload and CRC to follow
EOF	End of Frame	EOF marks the end of a frame and that the previous non-primitive Dword is the frame's CRC
HOLD	Hold data transmission	HOLD is transmitted in place of payload data within a frame when the transmitter does not have the next payload data ready for transmission. HOLD is also transmitted on the backchannel when a receiver is not ready to receive additional payload data.
HOLDA	Hold Acknowledge	This primitive is sent by a transmitter as long the HOLD primitive is received by its companion receiver.
DMAT	DMA Terminate	This primitive is sent as a request to the transmitter to terminate a DMA data transmission early by computing a CRC on the data sent and ending with a EOF primitive.
WTRM	Wait for Frame Termination	After transmission of any of the EOF, the transmitter will transmit WTRM while waiting for reception status from receiver.
R_IP	Reception in progress	Host or device is receiving payload
R_OK	Reception OK	No error detected during payload reception
R_ERR	Reception Error	Error(s) detected during payload reception
CONT	Continue	The CONT primitive allows long strings of repeated primitives to be eliminated. The CONT primitive implies that the previously received primitive be repeated as long as another primitive is not received.

Primitive Generation



Primitive Generation

- 18 primitives defined
- Link Layer protocol uses primitives in a variety of circumstances, including:
 - Indicating intent to send an FIS
 - Indicating beginning and end of each FIS
 - Flow Control signaling in response to Transport Layer Buffer state
 - Reporting Transmission Status

List of Primitives

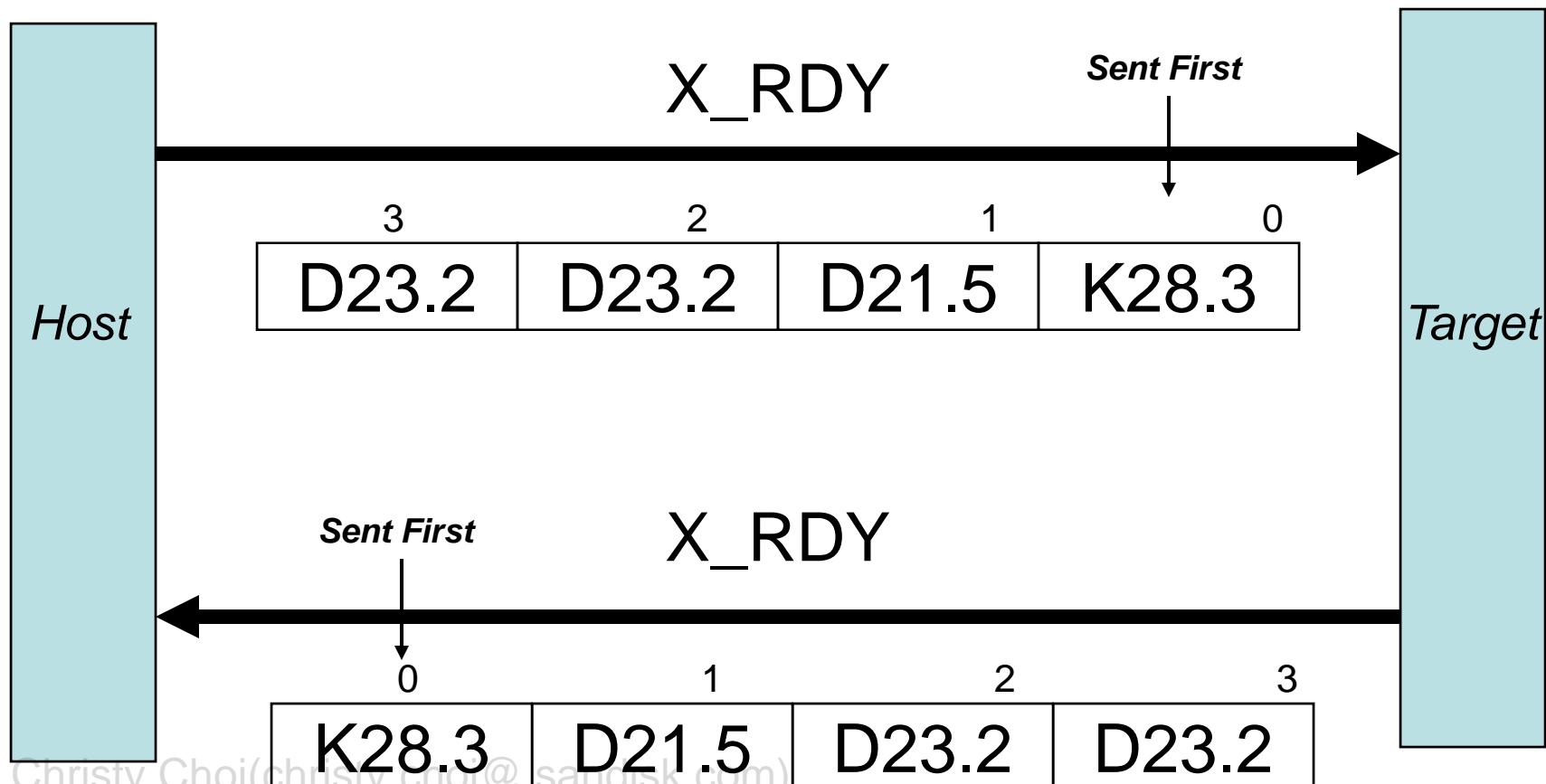
Primitive Name
ALIGN
CONT
DMAT
EOF
HOLD
HOLDA
PMACK
PMNAK
PMREQ_P
PMREQ_S
R_ERR
R_IP
R_OK
R_RDY
SOF
SYNC
WTRM
X_RDY

Beginning the FIS Transfer

- When the Link is ready to transmit, sends the X_RDY Primitive to advertise intent to send a FIS
- Target indicates readiness to receive the FIS by returning R_RDY primitives
- Upon receipt of R_RDY, the requesting device delivers the FIS

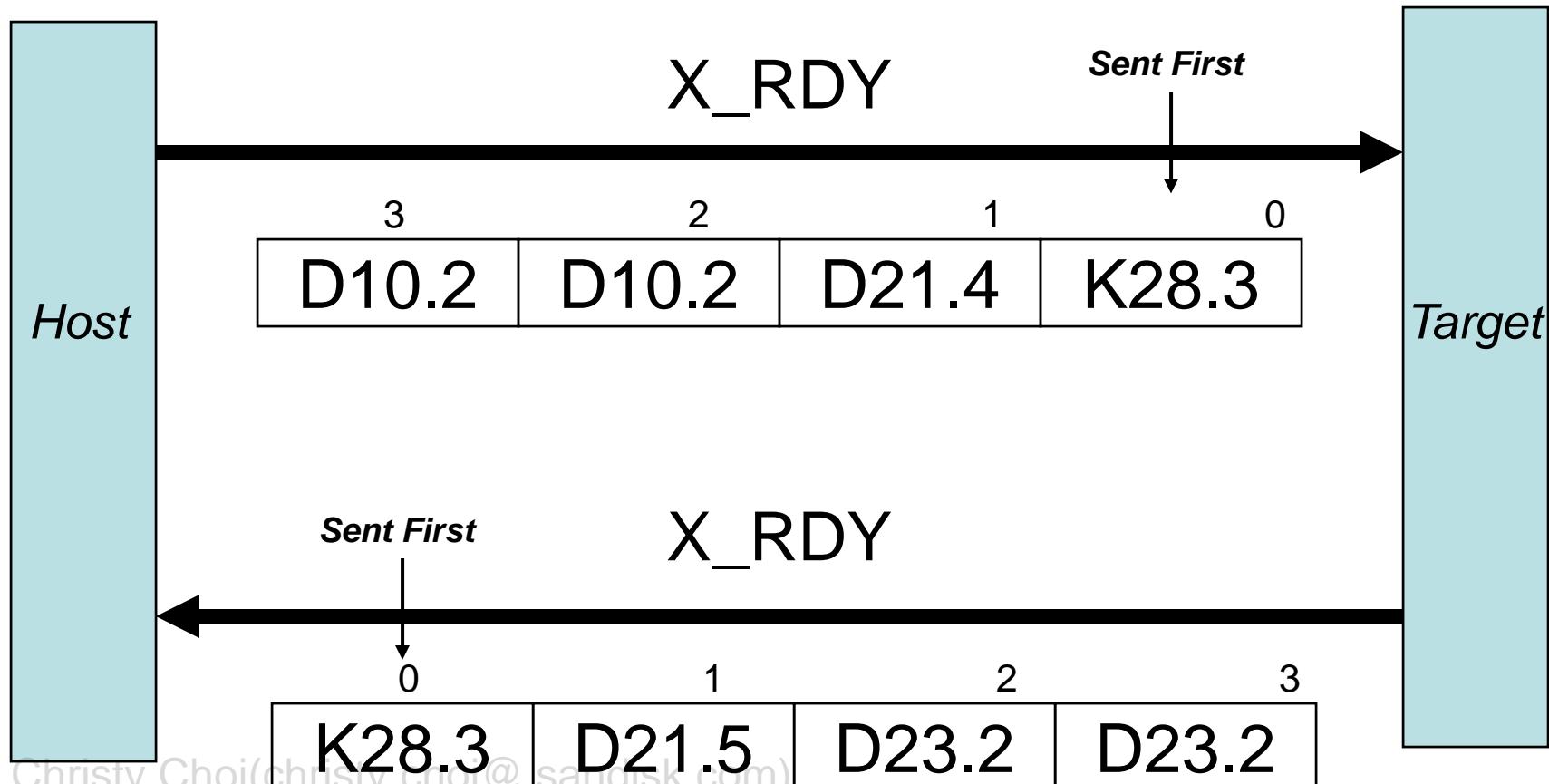
Arbitration Problem

Problem: Both Host and Device send X_RDY indicating they want to send a FIS.



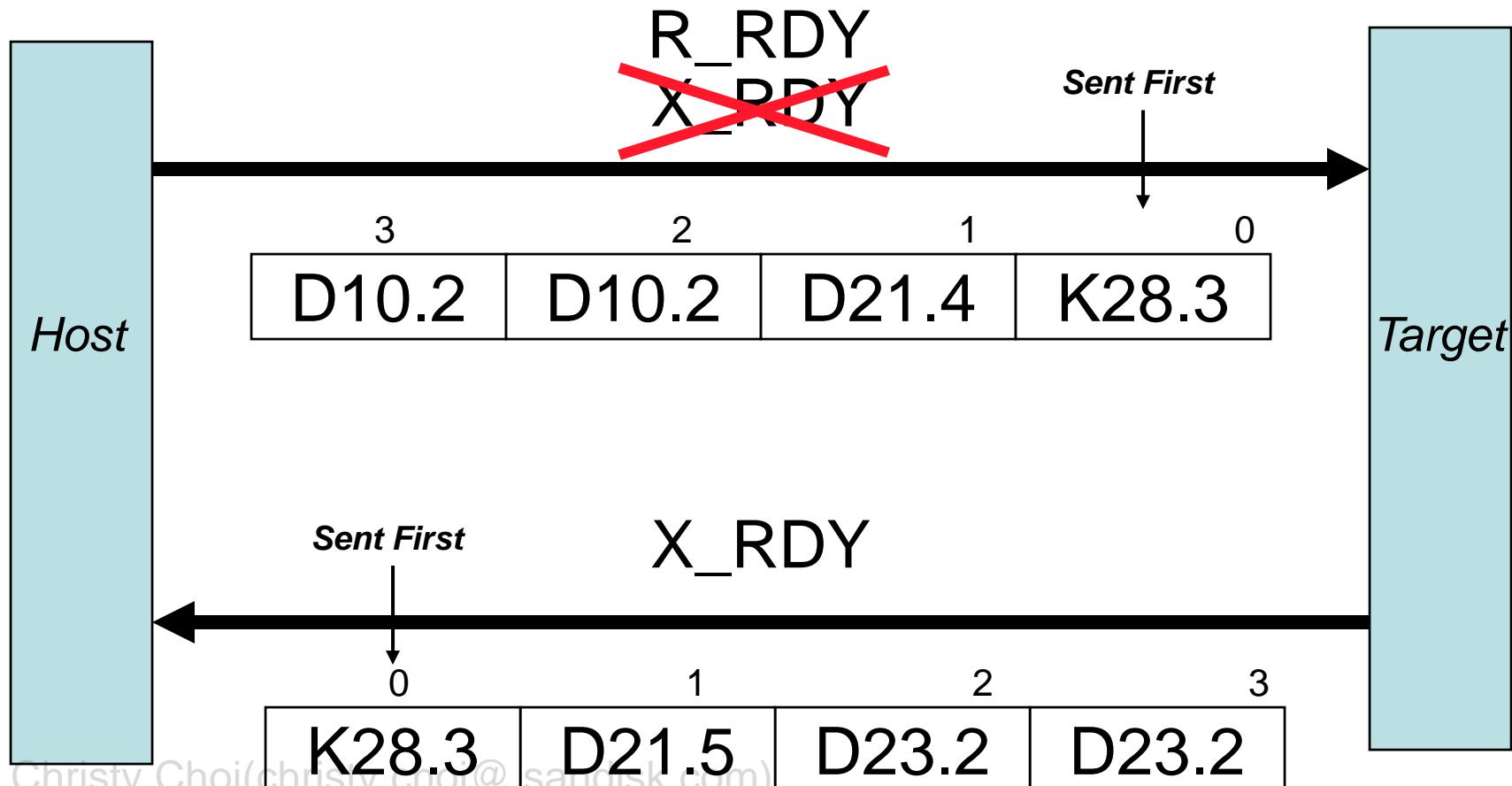
Arbitration Problem

If the Device receives a request from its Transport Layer to send a frame and receives X_RDY from the host it ignores the host request and waits on the host to deliver R_RDY so it can perform the transfer.

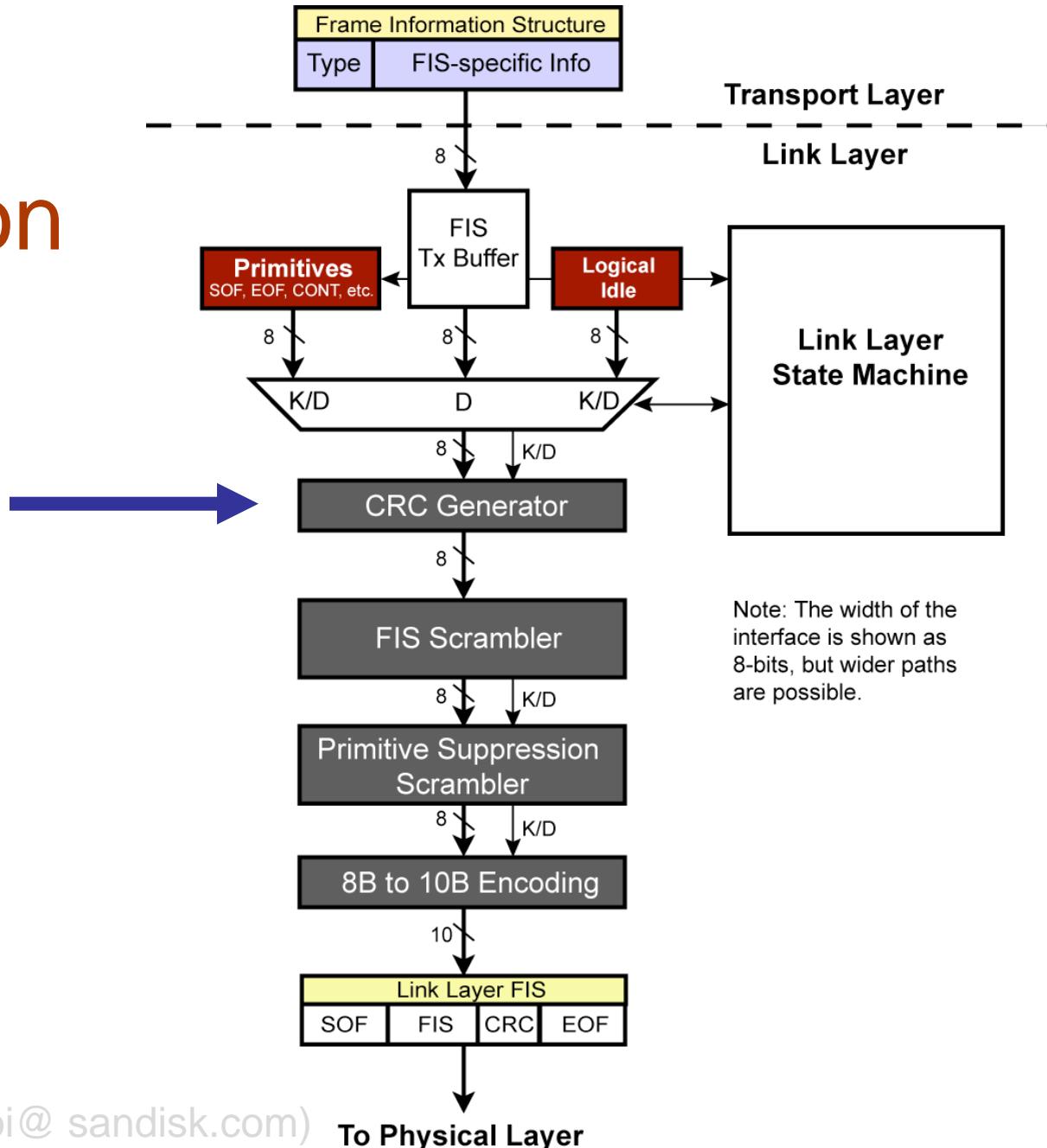


Arbitration Problem

If the Host is sending R_RDY and receives X_RDY from the device, the host must back off and send R_RDY when it is ready to receive the FIS.

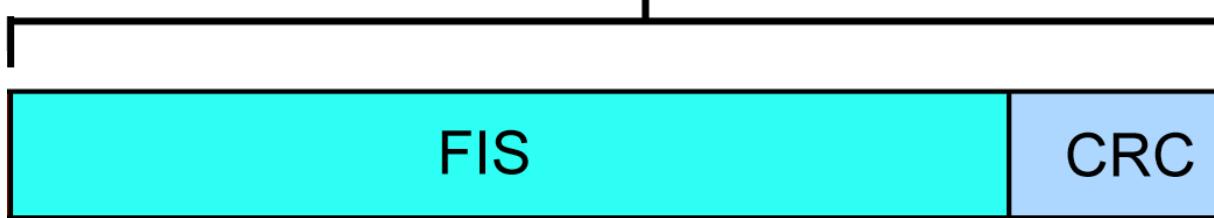


CRC Generation



CRC Generation

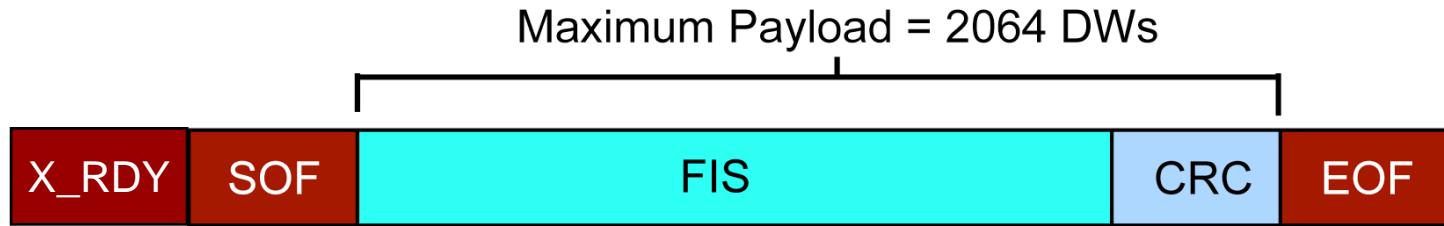
Maximum Payload = 2064 DWs



- 32-bit CRC is calculated for the FIS contents
 - Note that CRC generation is performed on Dword quantities
 - If FIS does not contain 4 byte multiples (DWs) the FIS is padded with zeros to yield a FIS size that is a multiple of 4 bytes
 - The CRC Generator Polynomial is long:

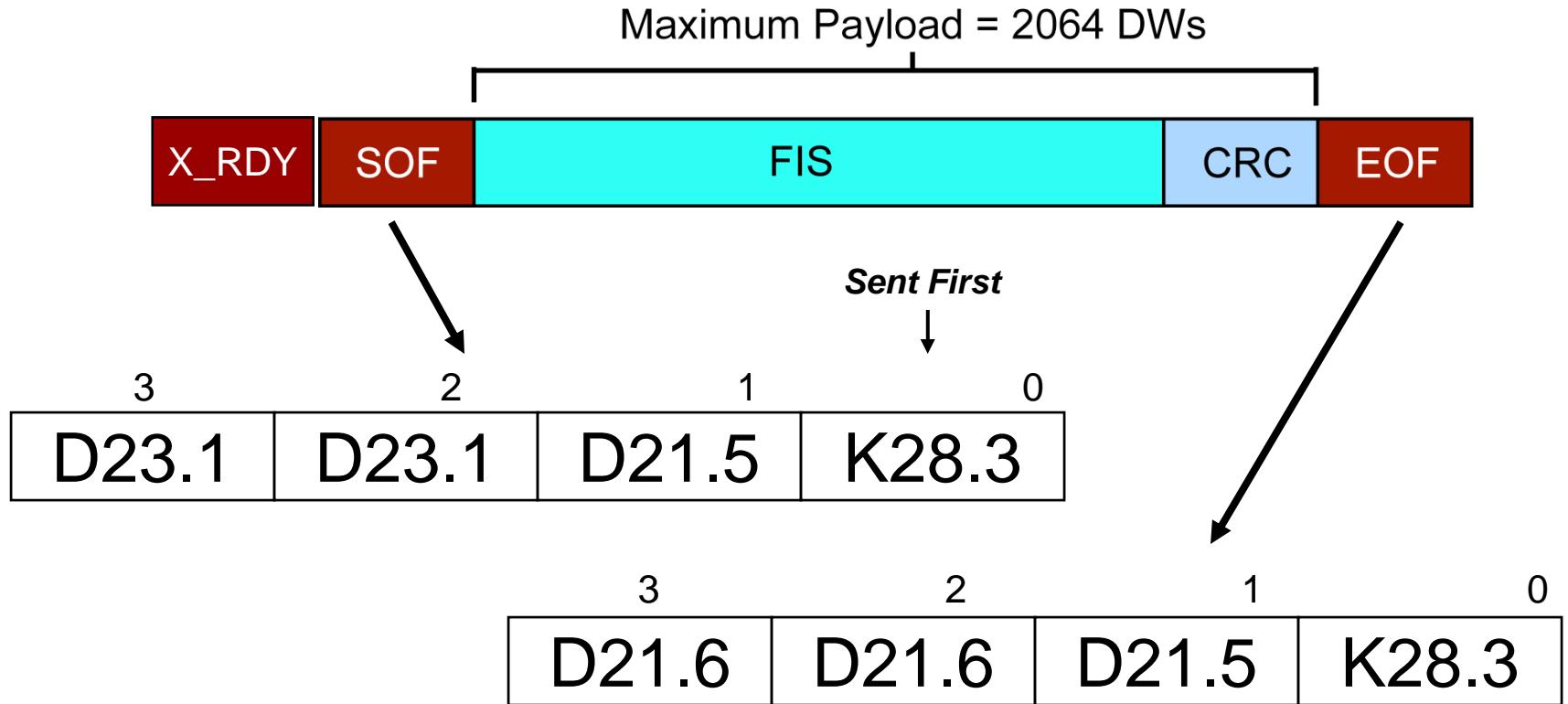
$$G(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

SOF/EOF

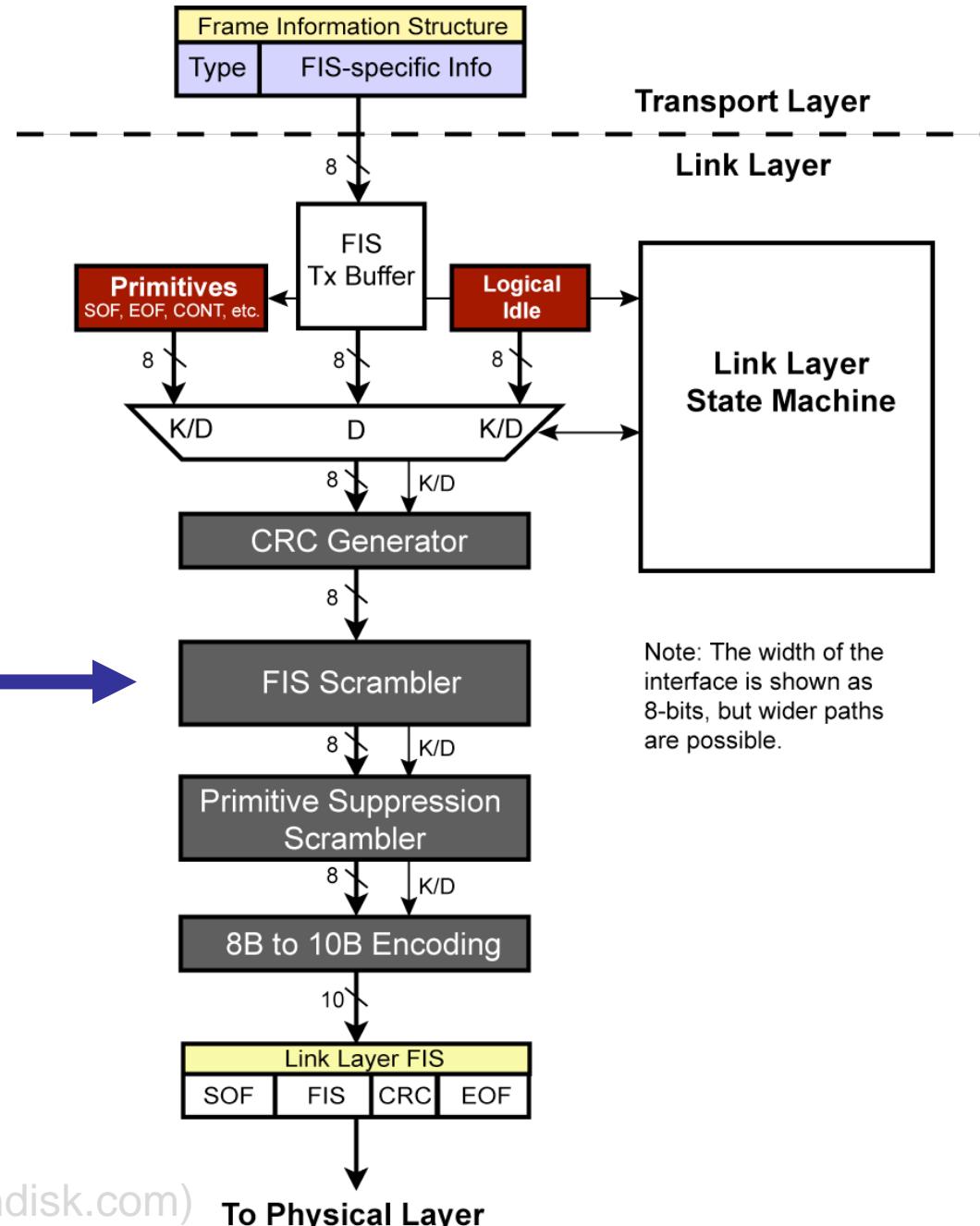


- SOF primitive is added to the front of the FIS
- EOF primitive is placed at the end of the FIS

SOF/EOF Symbols



FIS Scrambling



FIS Scrambling

- Each DW of the FIS and CRC is scrambled to reduce repeated patterns
- Primitives (including SOF and EOF) are not scrambled, making them easier for receiver to recognize if descrambler should get out of synch
- The Scrambler functionality may be represented as a Linear Feedback Shift Register (LFSR)
- The 32-bit scrambling polynomial is:

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

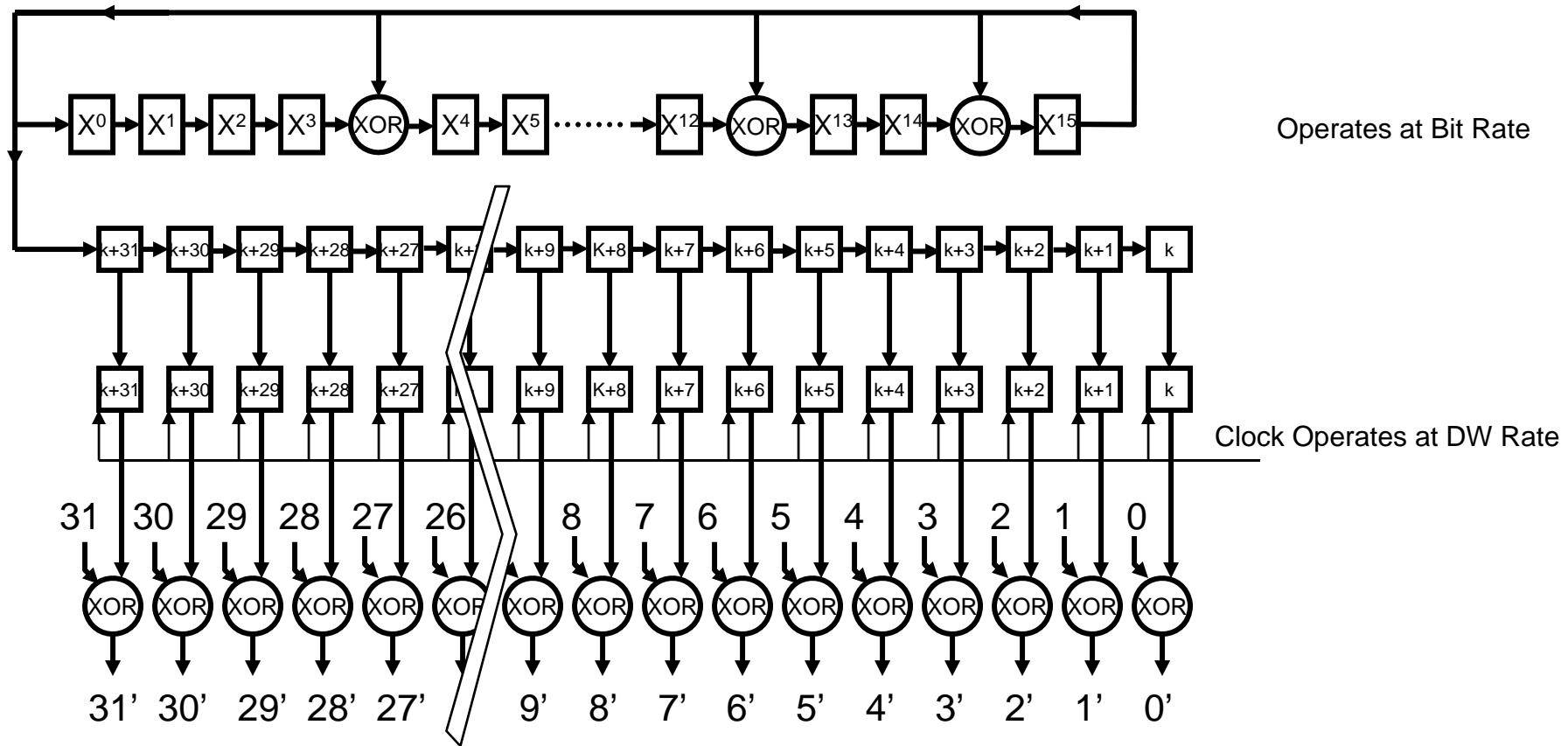
FIS Scrambling

- LFSR initializes to all Fs whenever an SOF primitive appears
- LFSR rolls over at 2048 DWs
- LFSR is not incremented during delivery of primitives

FIS Scrambler

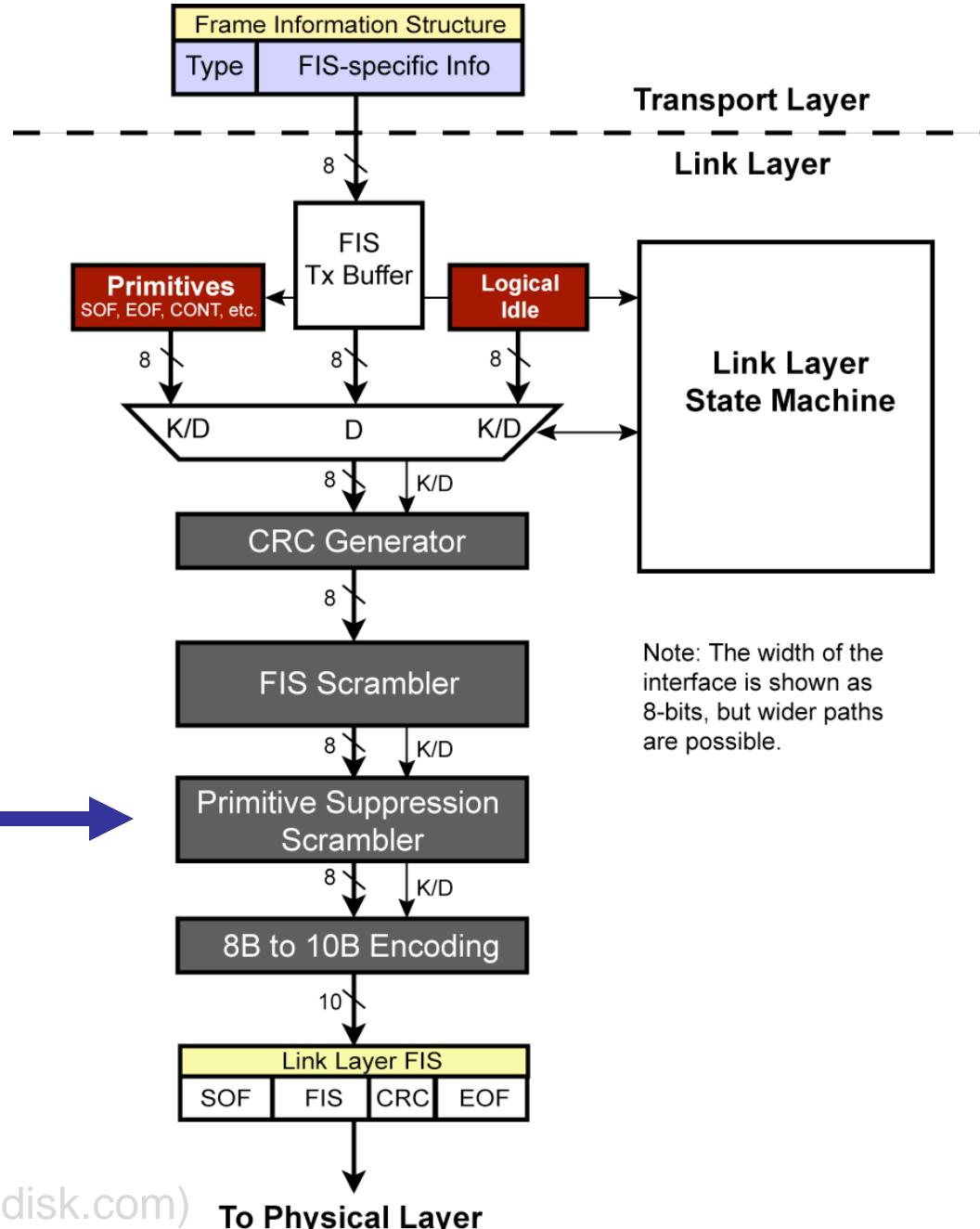
$$G(x) = X^{16} + X^{15} + X^{13} + X^4 + 1$$

- LFSR loaded with F_s
- Rolls over @ 2048 DWs
- Resets after each primitive



$$\begin{aligned}[31': 0'] &= [31:0] \text{ XOR} \\ &[Scr(k+31:k)]\end{aligned}$$

Primitive Suppression & Scrambling



Repeated Primitives and EMI

- Primitives that are send repeatedly as illustrated below can cause noise

Example of repeated primitives:

- Host sends X_RDY continuously until R_RDY is received
- SYNC (logical idle) is signaled when no transaction is being delivered
- Device sends R_RDY continuously until SOF is received

<i>Host</i>	<i>Device</i>
X_RDY	SYNC
X_RDY	R_RDY
X_RDY	R_RDY
SOF	R_RDY
FIS Data	R_RDY

CONT Primitive

Solution: The CONT (continue) primitive indicates that a primitive should be understood as being repeated even though the transmitter won't keep sending it. This understanding remains in place at the receiver until another primitive is received.

Repeated Primitive Suppression

- Primitives are repeated 2 times followed by the CONT primitive
- Scrambled Data (XXXX) is sent after the CONT primitive until a different primitive is sent
- Uses the same scrambling algorithm employed for FIS scrambling
- Transmission of the CONT primitive is optional, but proper recognition and interpretation of the CONT primitive is required

Host (Tx) Device (Tx)

X_RDY	XXXX
X_RDY	XXXX
CONT	XXXX
XXXX	R_RDY
XXXX	R_RDY
SOF	CONT
FIS Data	XXXX

Limitations of Continue Primitive

The CONT primitive may only follow these types of primitive:

- HOLD
- HOLDA
- PMREQ_P
- PMREQ_S
- R_ERR
- R_IP
- R_OK
- R_RDY
- SYNC
- WTRM
- X_RDY

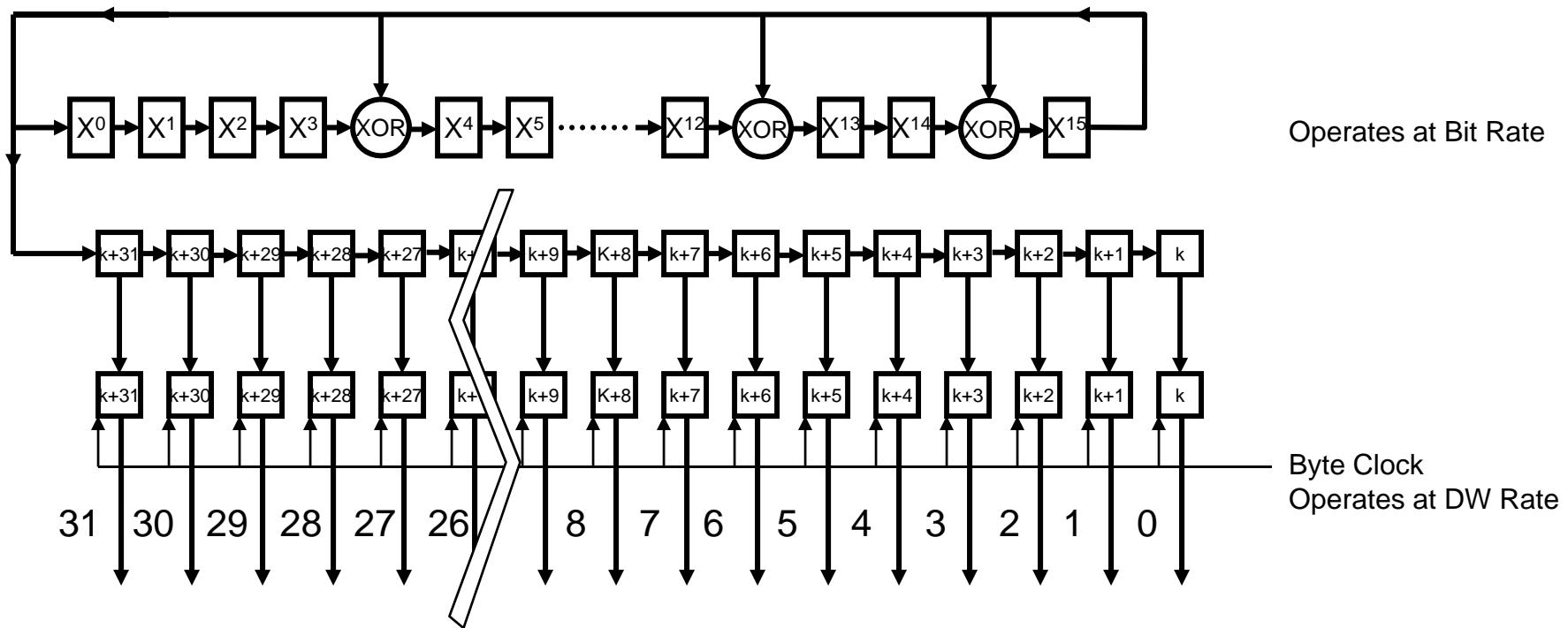
Host (Tx) Device (Tx)

X_RDY	XXXX
X_RDY	XXXX
CONT	XXXX
XXXX	R_RDY
XXXX	R_RDY
SOF	CONT
FIS Data	XXXX

Repeated Primitive Scrambler

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

- LFSR loaded with F_s
- Resets after each
 - COMRESET (Host)
 - COMINIT (Device)



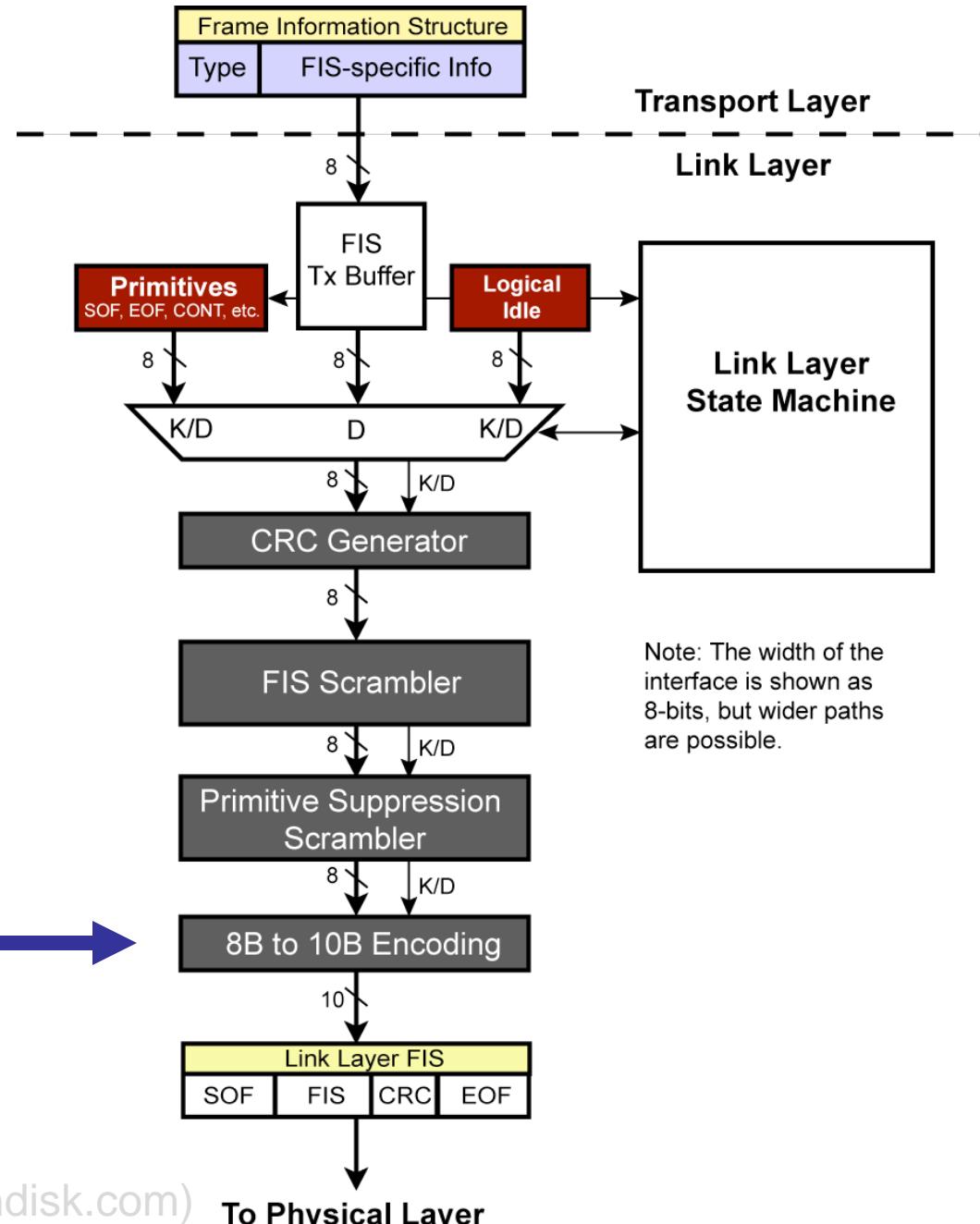
Receiver Discards Data Following CONT

Suppressed Primitive Data (XXXX) is discarded by the Link Layer when received

Host (Tx) Device (Tx)

X_RDY	XXXX
X_RDY	XXXX
CONT	XXXX
XXXX	R_RDY
XXXX	R_RDY
SOF	CONT
FIS Data	XXXX

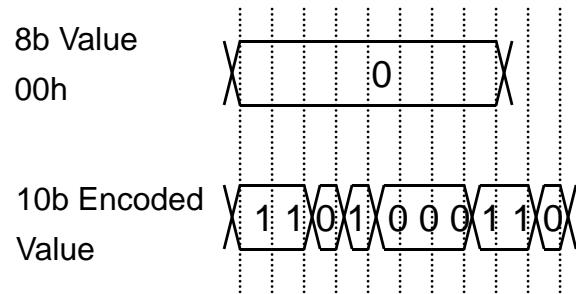
8b/10b Encoding



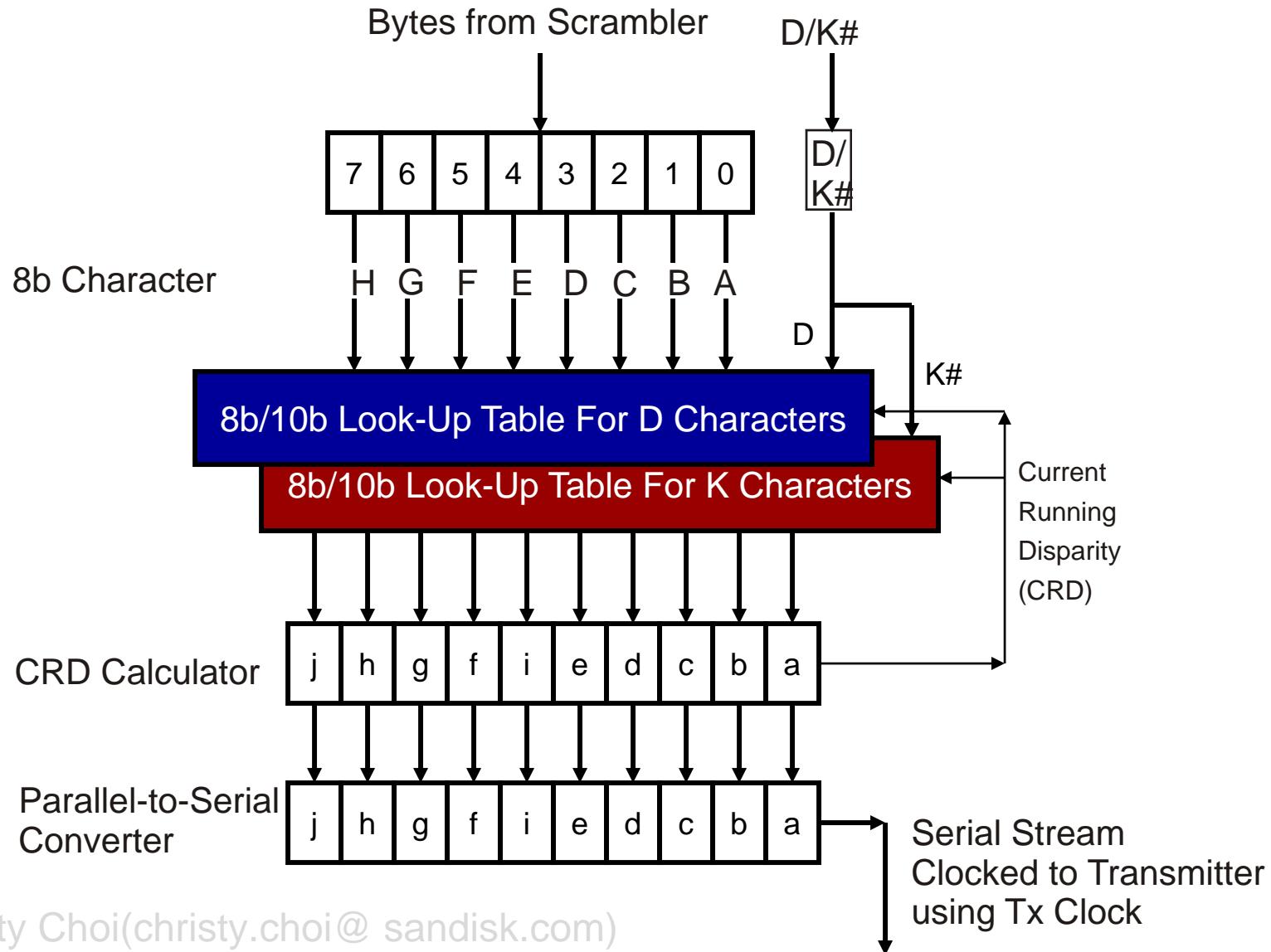
8b-to-10b Encoding

- All information transferred across the link must first be converted from 8-bit values (bytes) into 10 bit values (symbols)
- Standard encoding method invented by IBM and also used in the following standards:
 - Ethernet, Fibre Channel, ServerNet, IBA, SAS, PCIe
- Reasons for encoding
 - Integrates clock into data stream
 - Creates sufficient transition density to limit “Run Length”
 - Balance number of 1’s and 0’s to maintain “DC Balance”
 - Ability to encode for special control characters (K)
 - Facilitate detection of most transmission errors
- Transmission performance degraded by 20%

8b/10b Encoding Example



8b-to-10b Encoding



Data Encoding

- Two encodings for each 8-bit value
- Encoding based on current running disparity
- Example encoding:

Name	Byte	abcdei fghj output	
		Current rd-	Current rd+
D0.0	00h	100111 0100	011000 1011
D1.0	01h	011101 0100	100010 1011
D0.1	20h	100111 1001	011000 1001
D1.1	21h	011101 1001	100010 1001

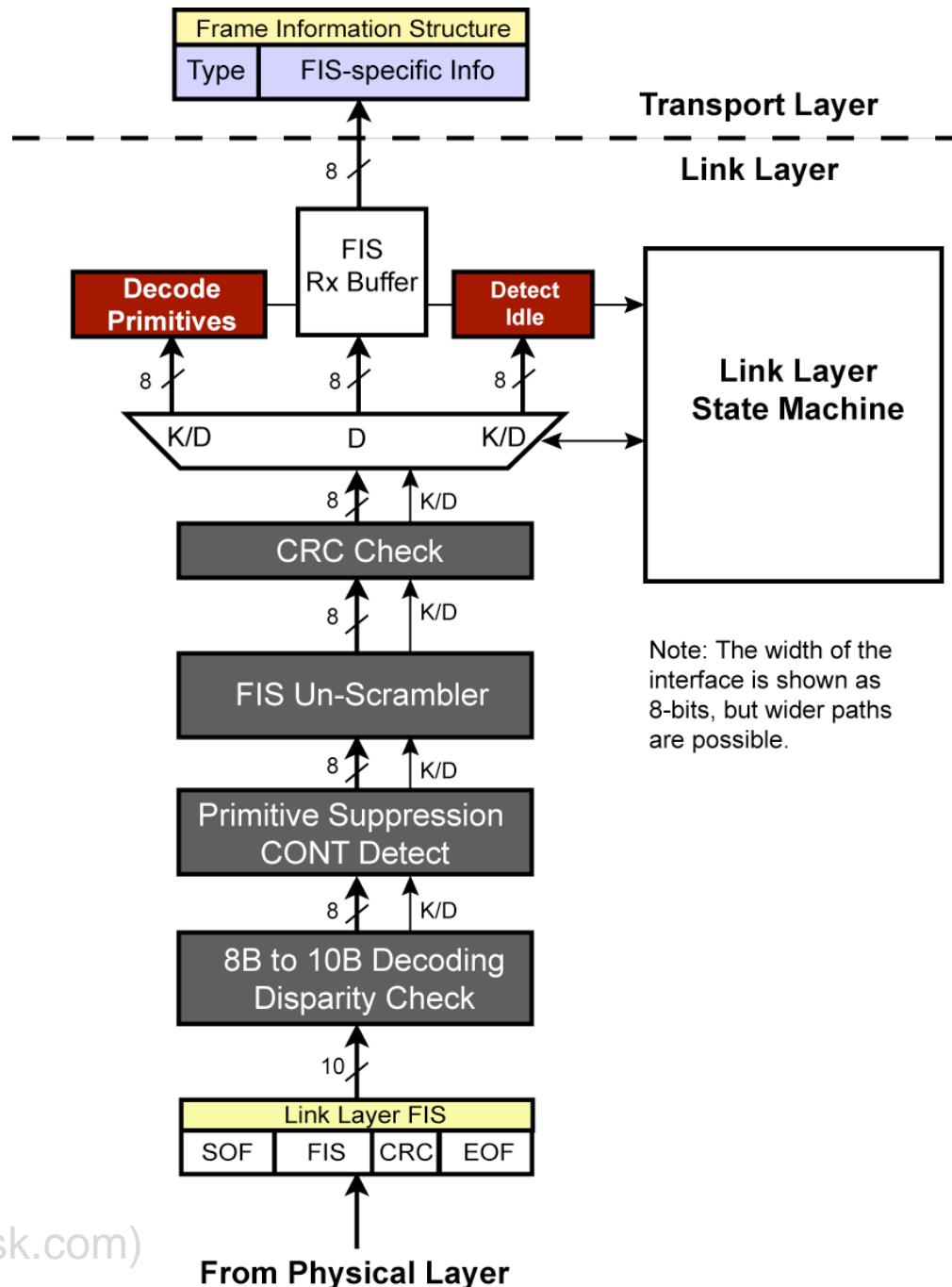
Control Charter Encoding

- SATA uses only 2 control characters:

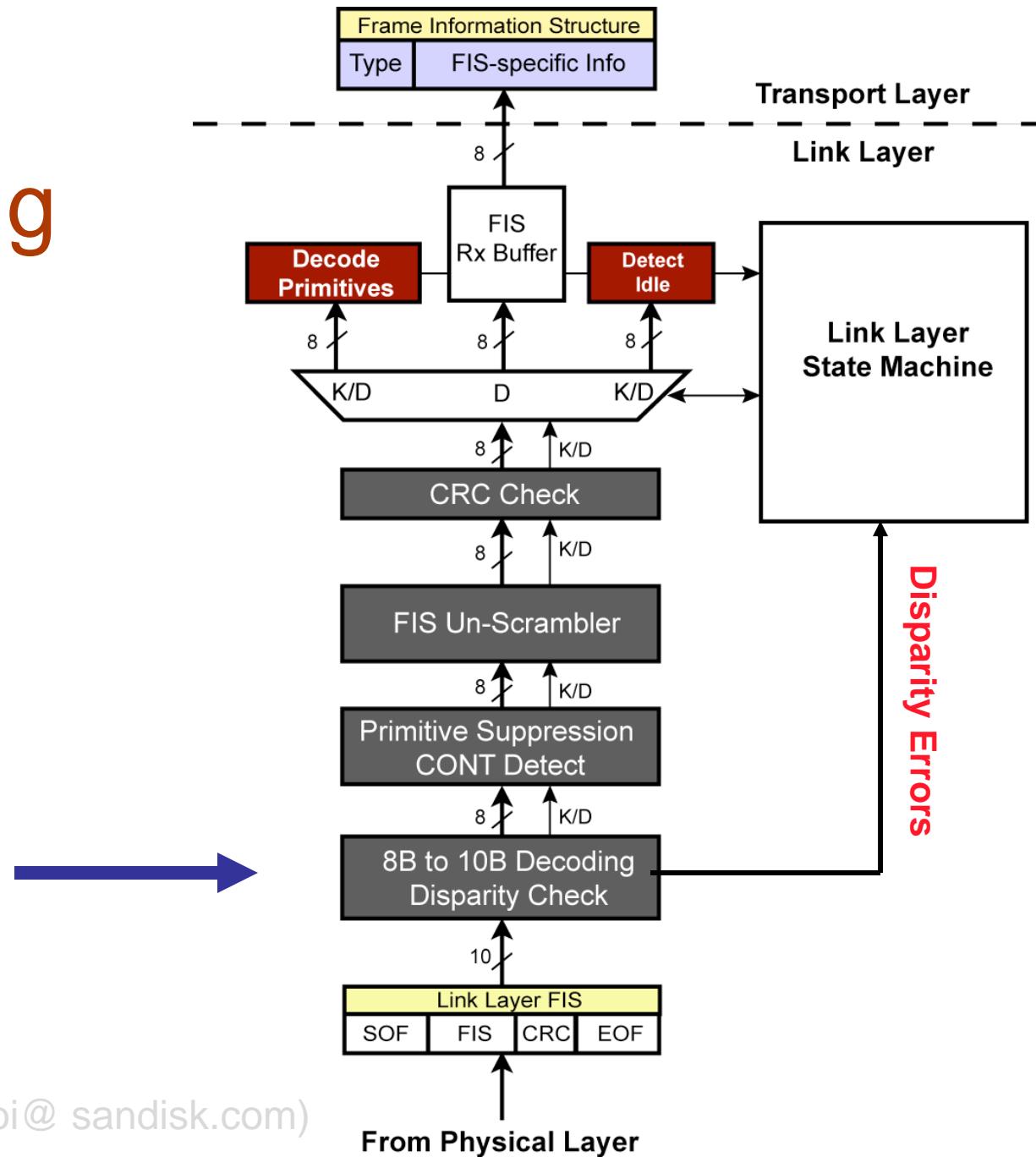
Name	abcdei fghj output		Description
	Current rd-	Current rd+	
K28.3	001111 0011	110000 1100	Occurs only at byte 0 of all primitives except for the ALIGN primitive
K28.5	001111 1010	110000 0101	Occurs only at byte 0 of the ALIGN primitive

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

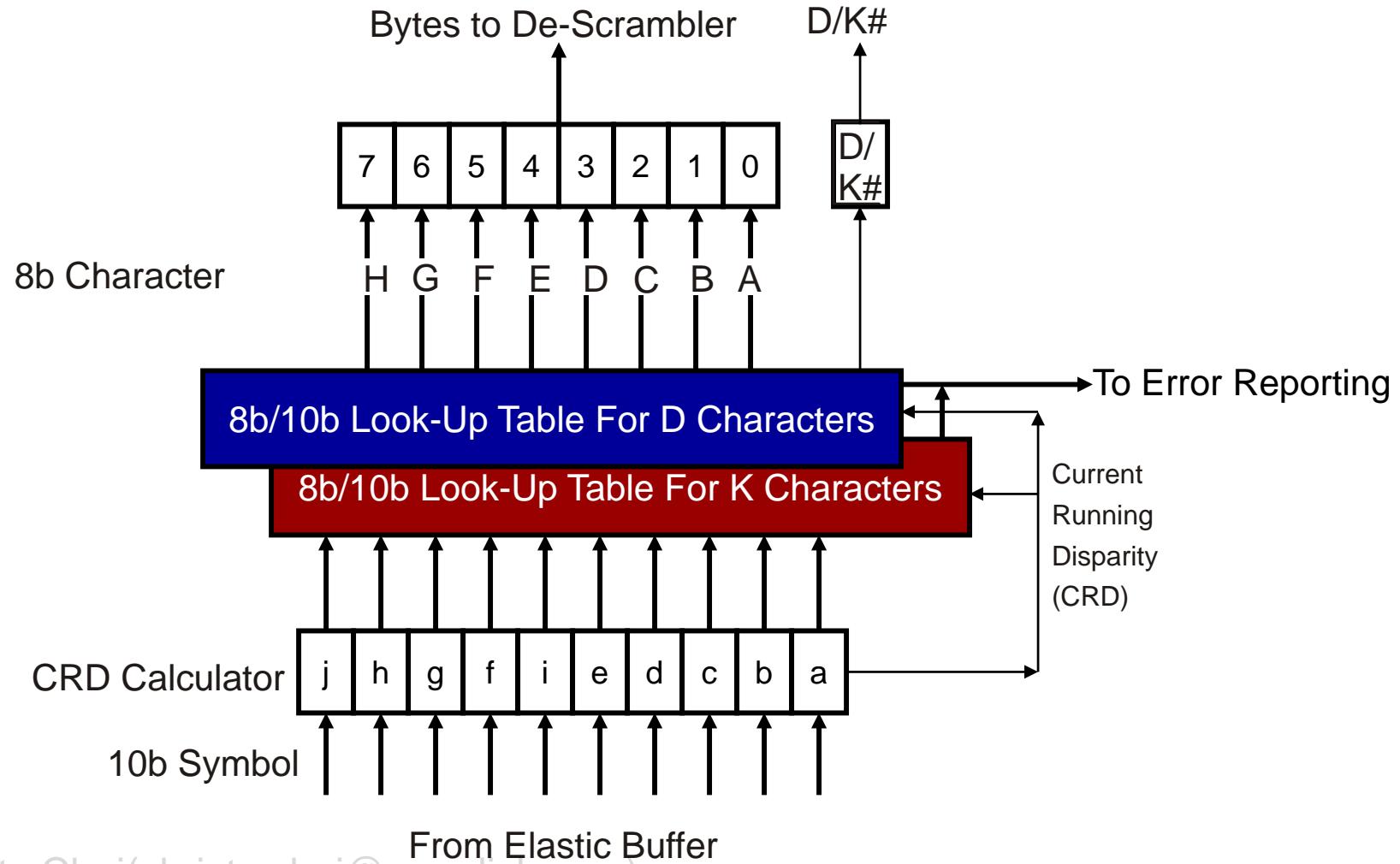
FIS Reception



8b/10b Decoding



Decoding/Disparity Error Check



8b-10b Decode (Disparity Errors)

Dword boundary

Last byte of previous Dword First byte of next Dword

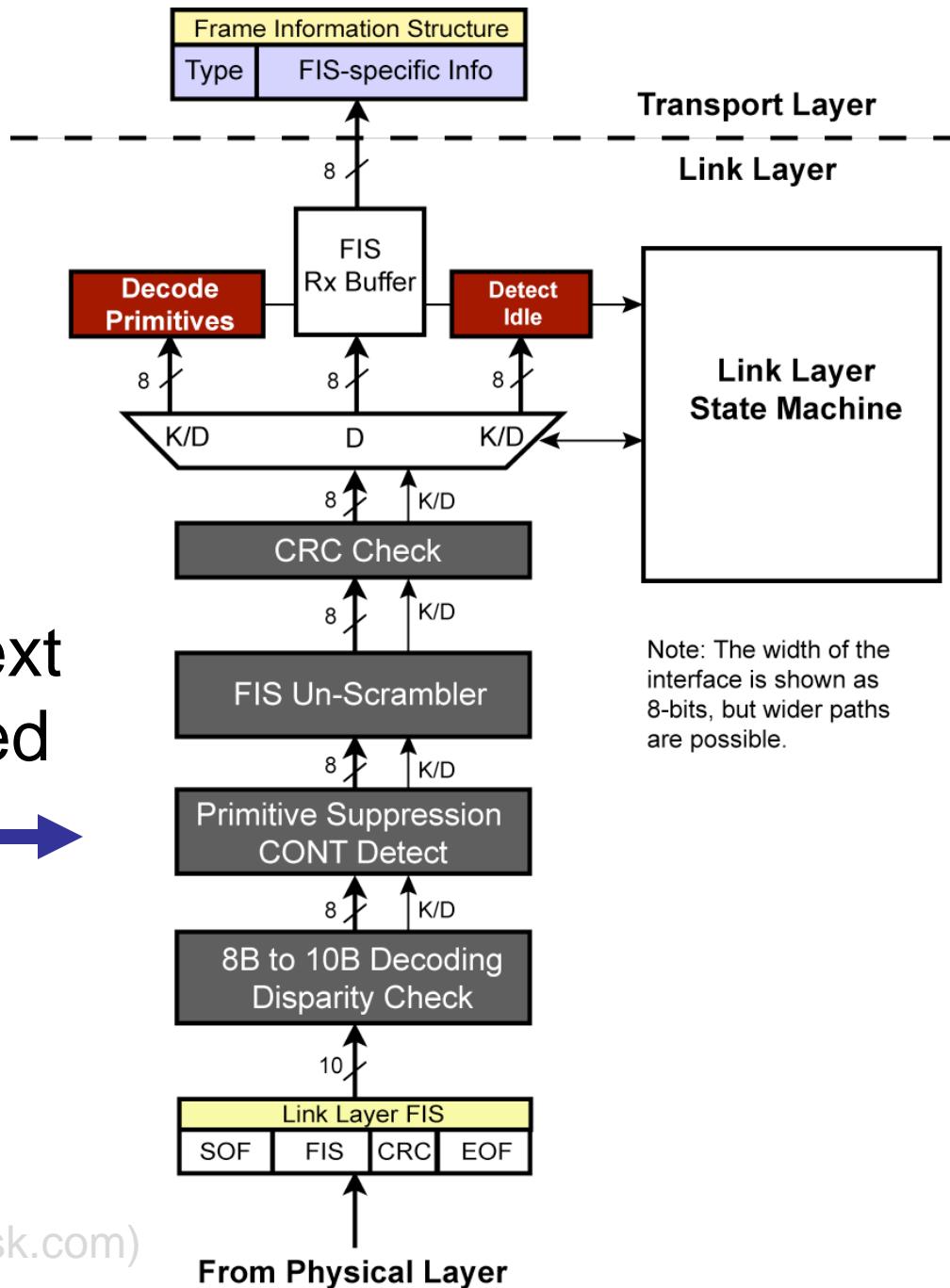
	CRD	Character	CRD	Character	CRD	Character	CRD
Transmitted Character Stream	-	D21.1	-	D10.2	-	D23.5	+
Transmitted Bit Stream	-	101010 1001	-	010101 0101	-	111010 1010	+
Bit Stream After Error	-	101010 10 <u>11</u>	+	010101 0101	+	111010 1010	+
Decoded Character Stream	-	D21.0	+	D10.2	+	Invalid	+

Error occurs here

Error detected here
This dword labeled as invalid

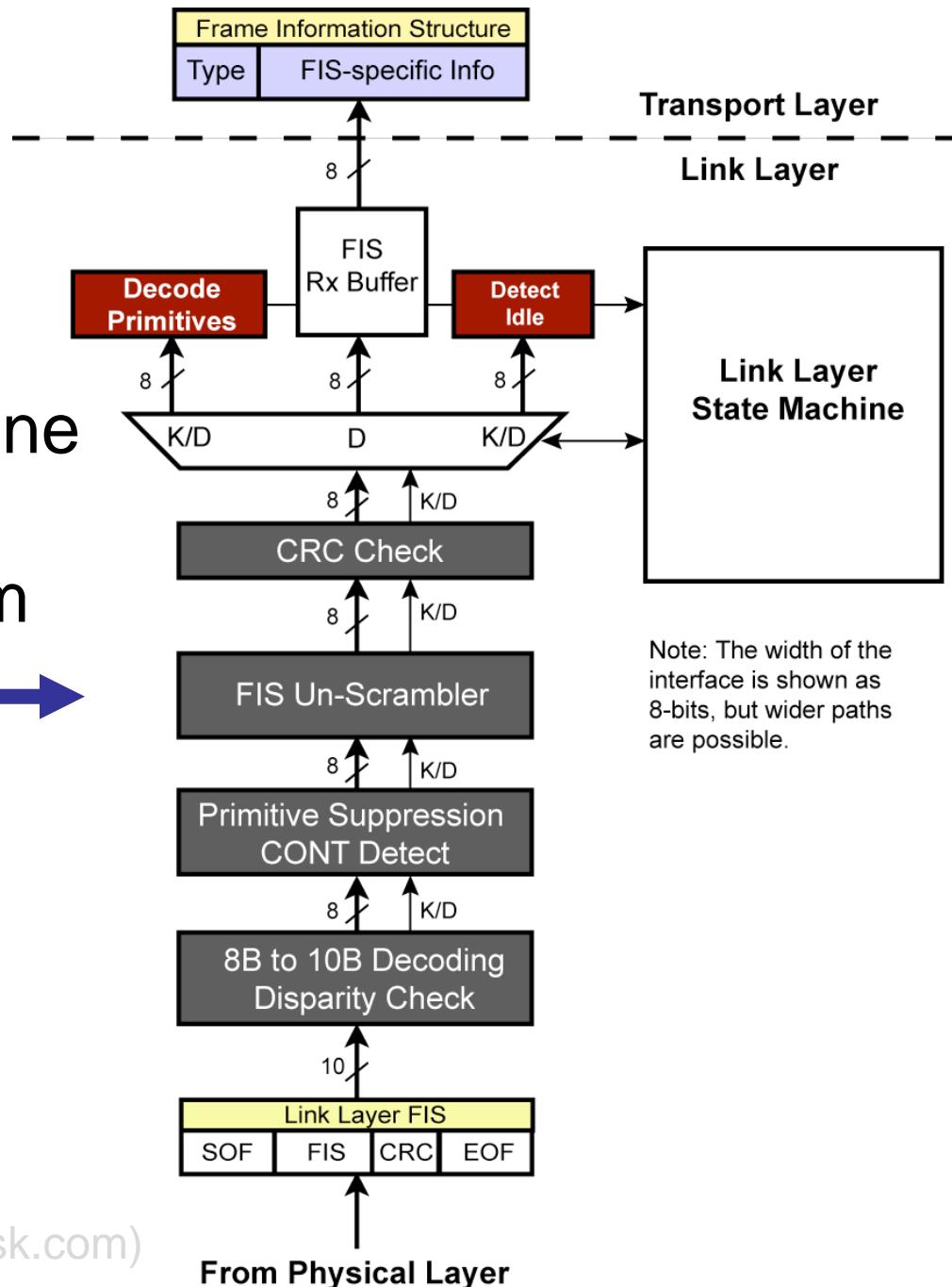
Repeated Primitive Suppression

All data received
Between a CONT
primitive and the next
primitive is discarded

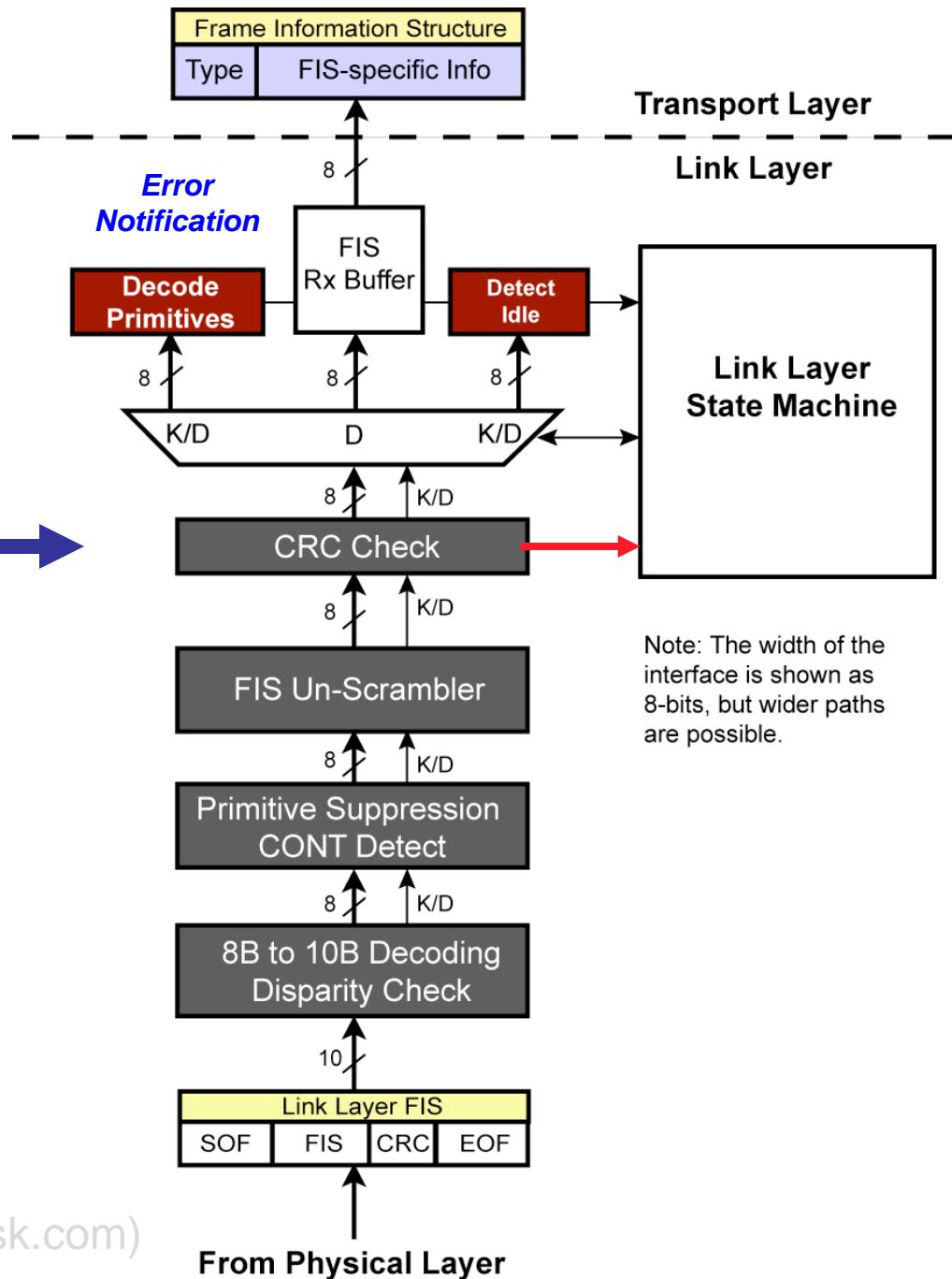


Un-Scrambler

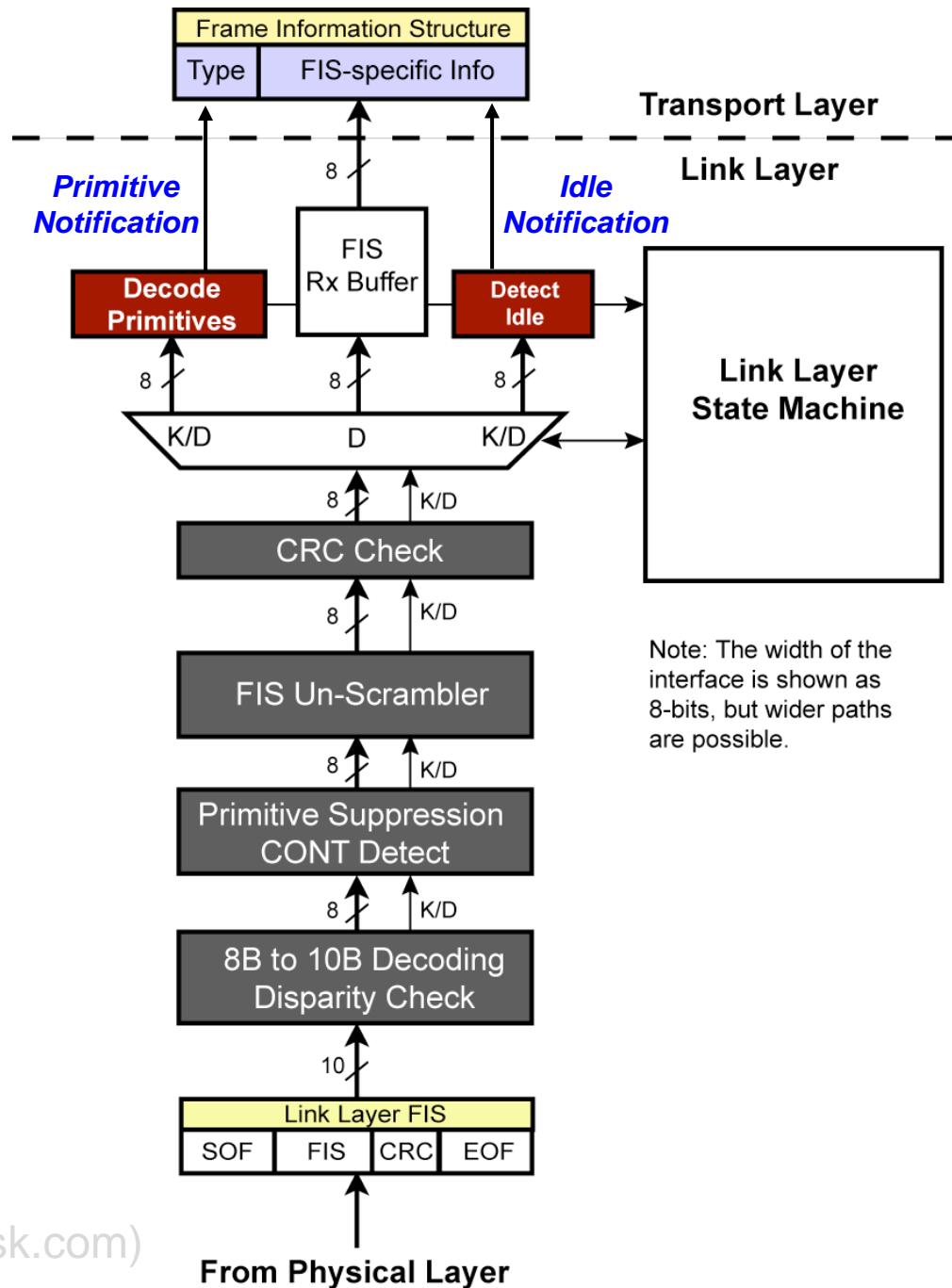
Un-scrambling is done using the same FIS scrambling algorithm



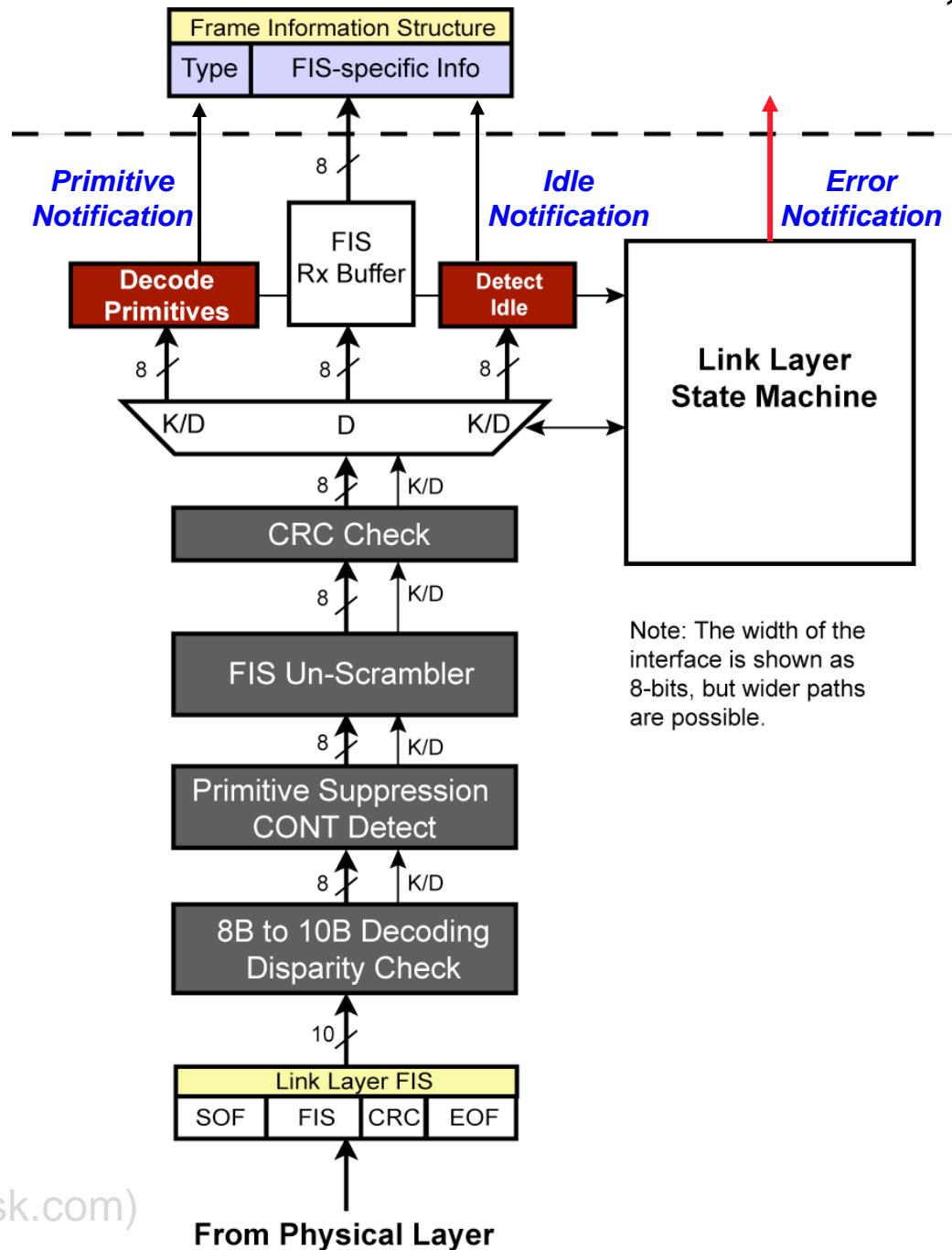
CRC Check



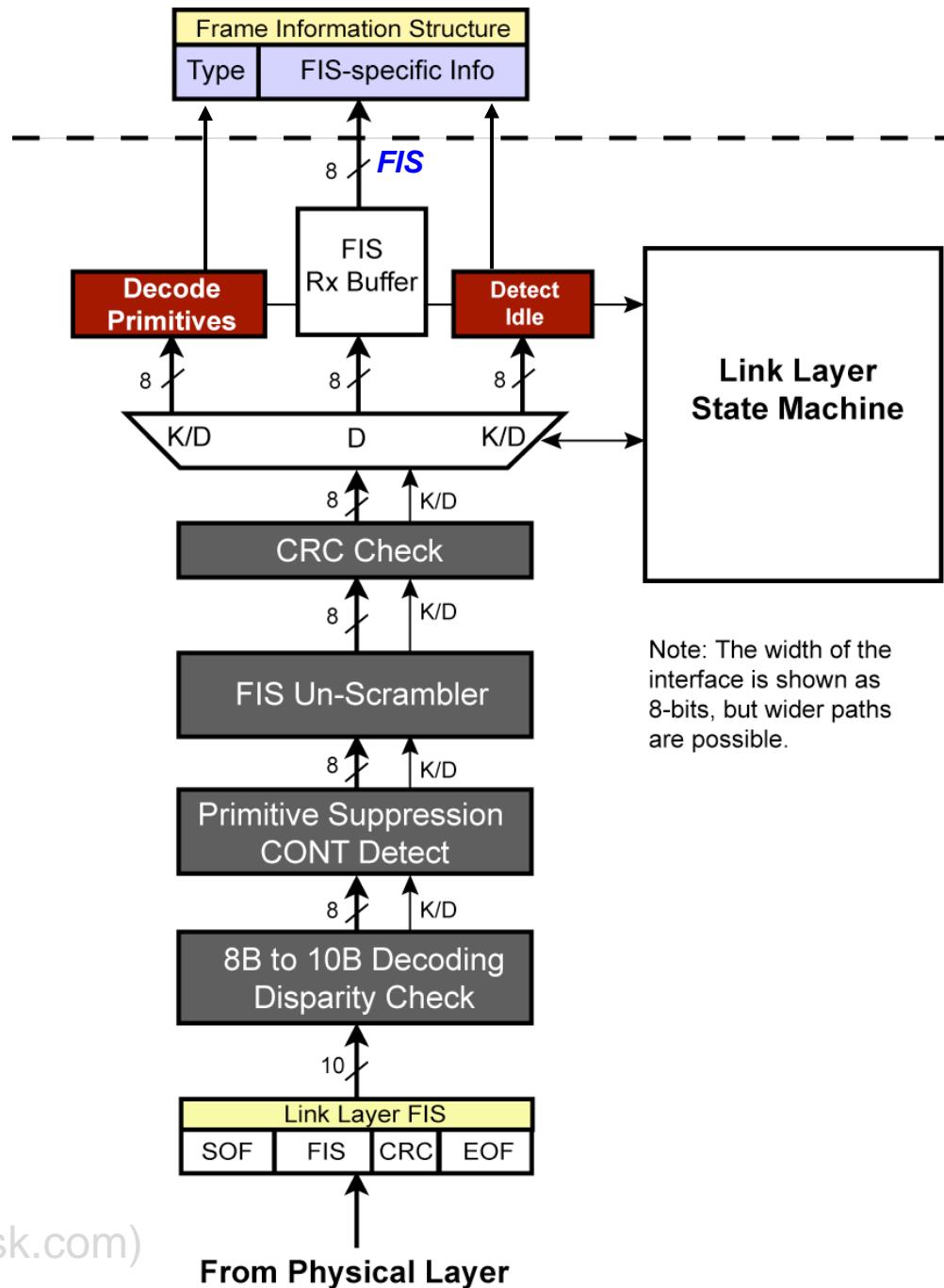
Primitive Decoding



Completion Status & Error Reporting



Link Layer Forwards FIS to Transport Layer



Transport Layer FIS Reception

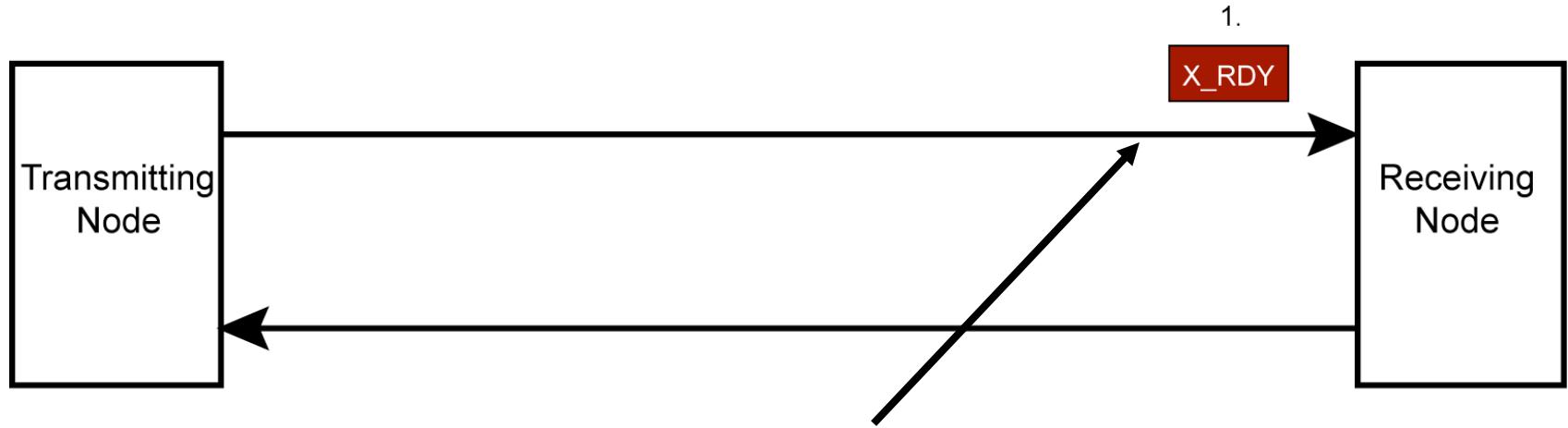
- The Transport Layer receives FISs into its buffer and decodes the packet
- Several FIS types may be detected:
 - Register (Host or Device)
 - Set Device Bits (Host)
 - DMA Activate (Host)
 - PIO Setup (Host)
 - 1st Party DMA Setup (Host or Device)
 - BIST (Host or Device)
 - Data (Host or Device)
 - Unrecognizable FIS (Host or Device)

FIS Transmission Summary

- The slides that follow summarize the transmission protocol between the host and device during FIS transmission
- The example illustrates a Host to Device transfer

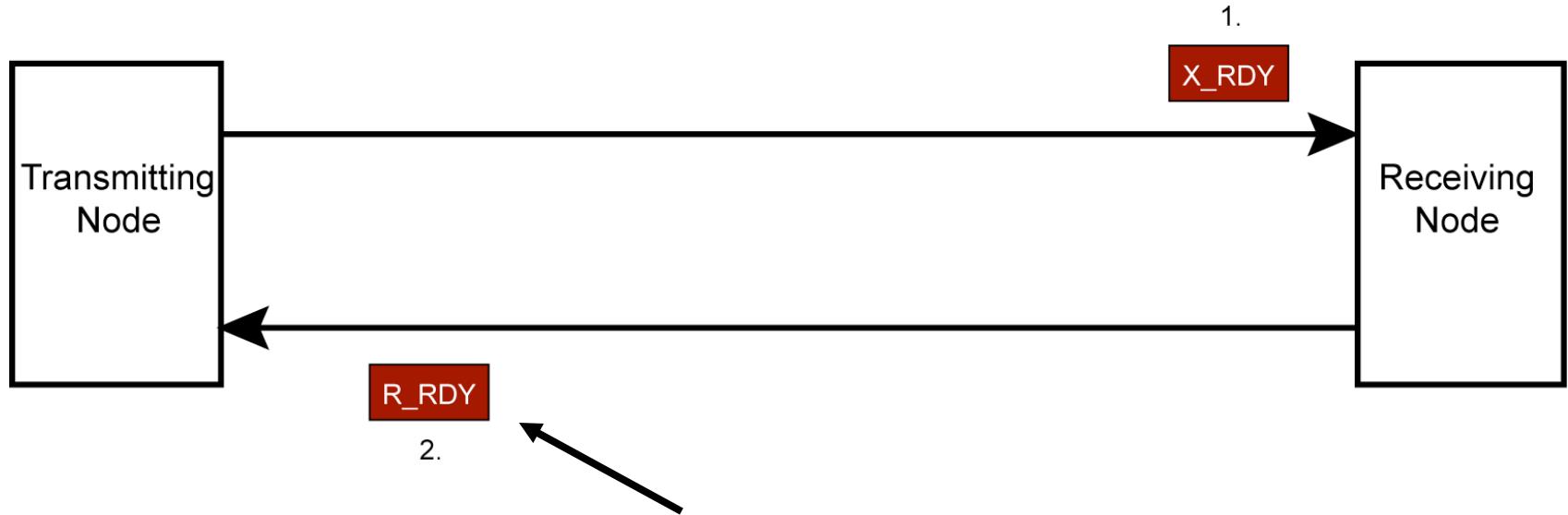
FIS Transmission Example

Example Link Protocol Sequence:



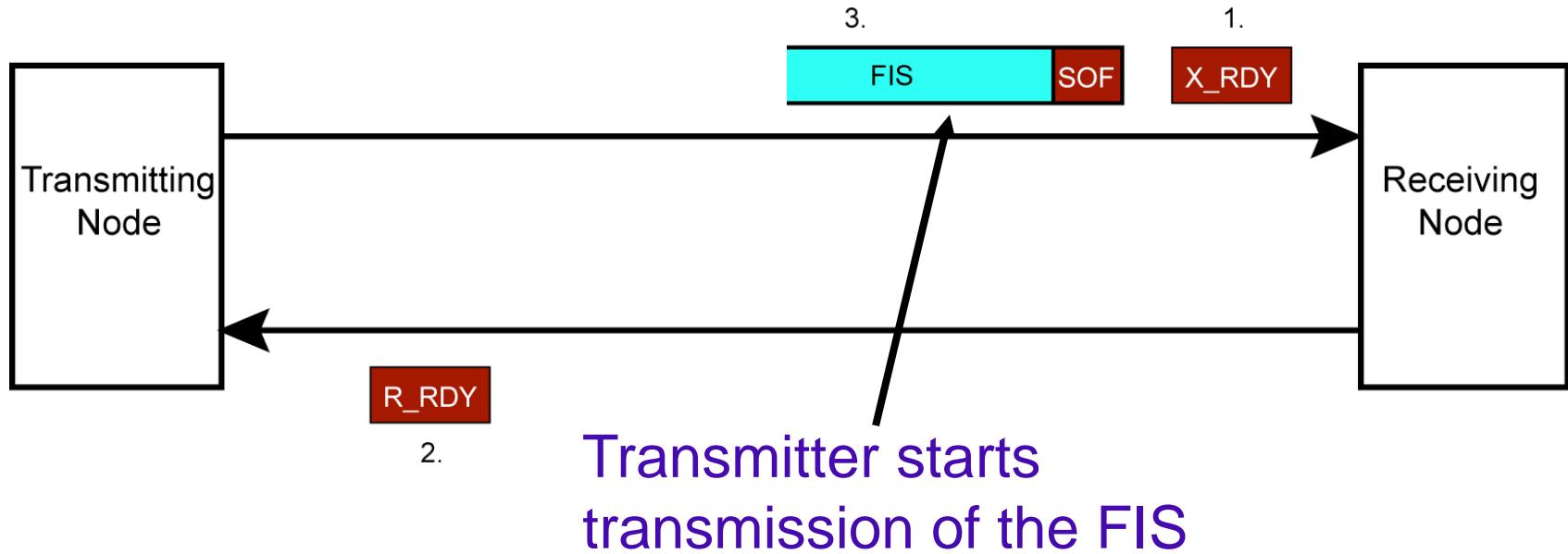
Transmitter sends an X_RDY primitive to notify receiver that it's ready to send a FIS

FIS Transmission Example



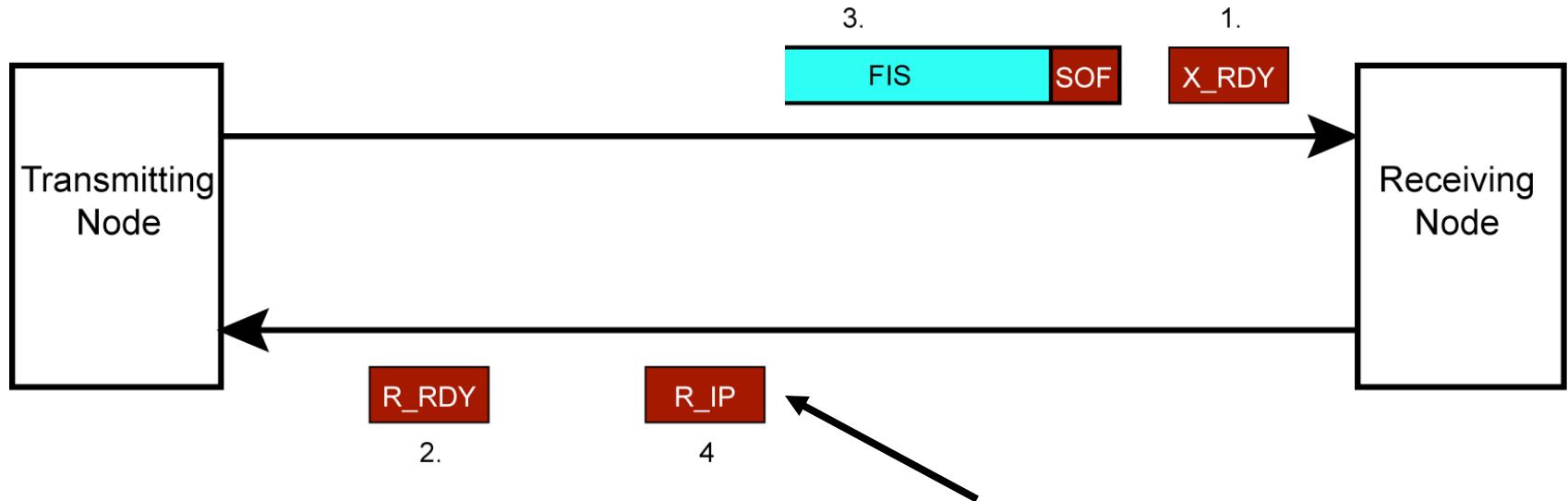
Upon detecting X_RDY, the receiver indicates it's ready to receive the FIS via R_RDY
Primitives

FIS Transmission Example



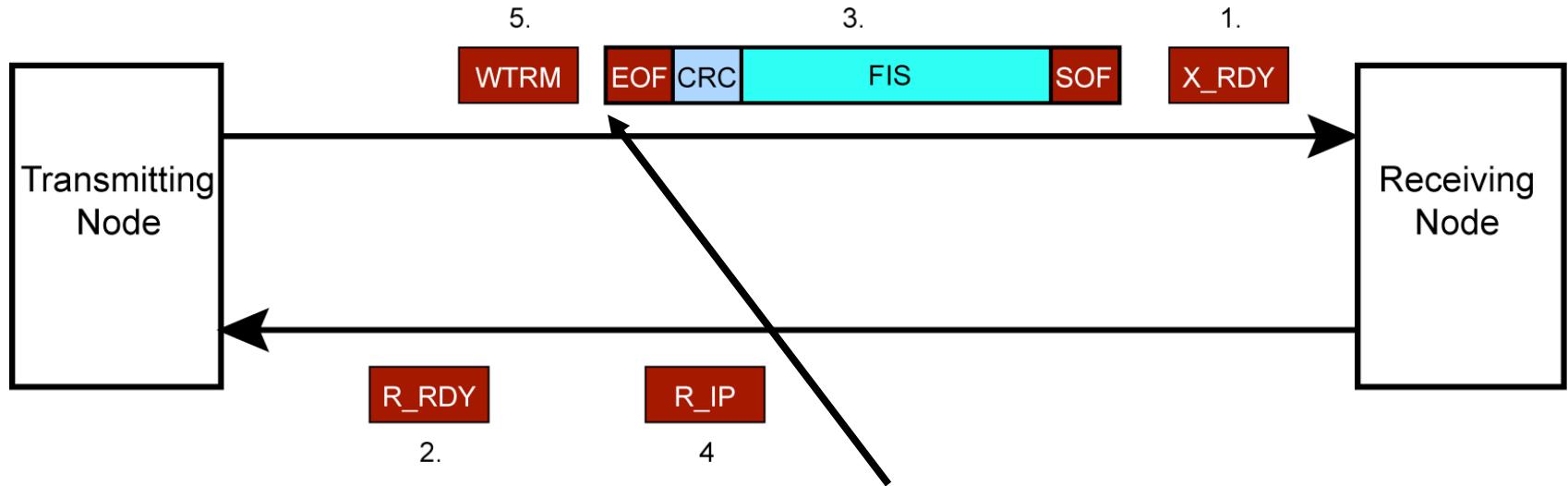
Primitives

FIS Transmission Example



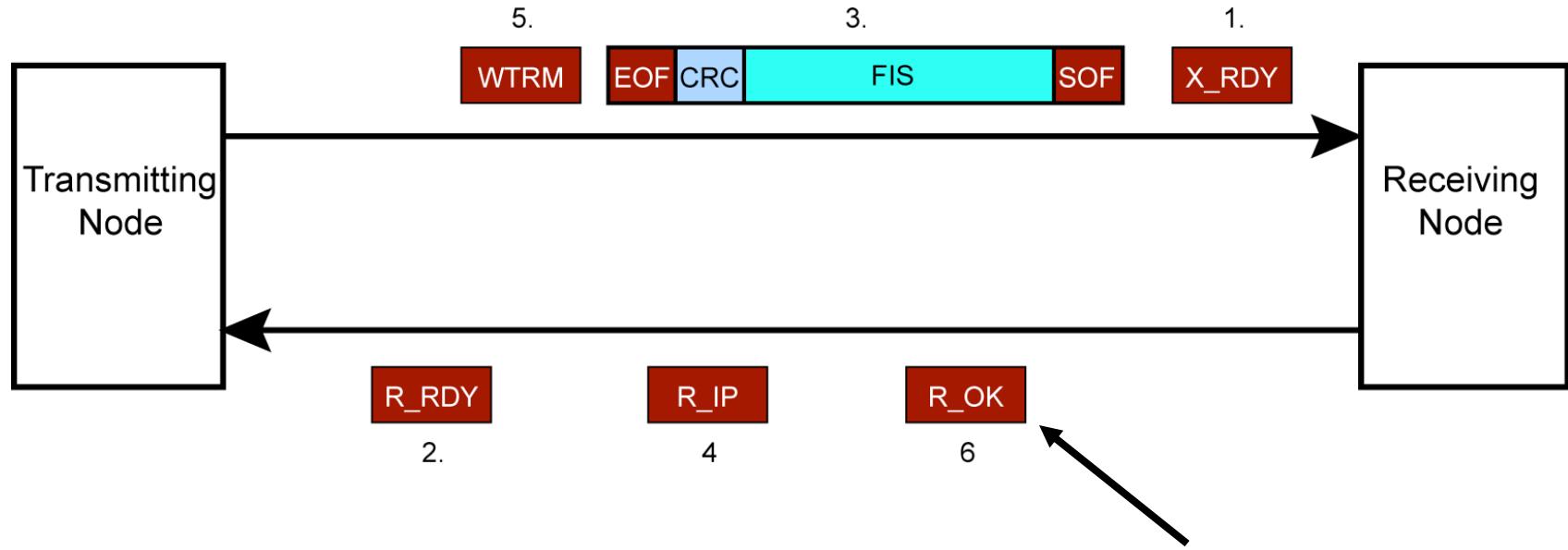
Receiver reports that FIS
reception is In Progress

FIS Transmission Example



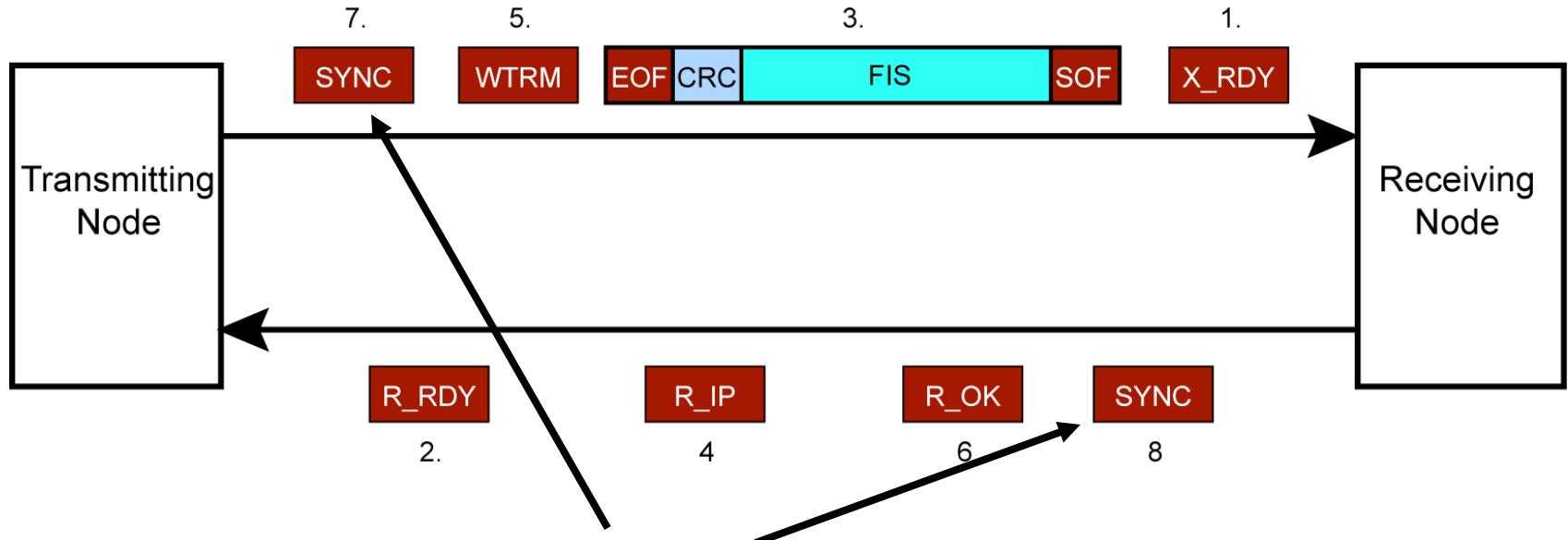
Transmitter completes FIS transfer & signals WTRM (wait for termination), indicating that it's waiting for the receiver to report completion and thus terminate the transmission

FIS Transmission Example



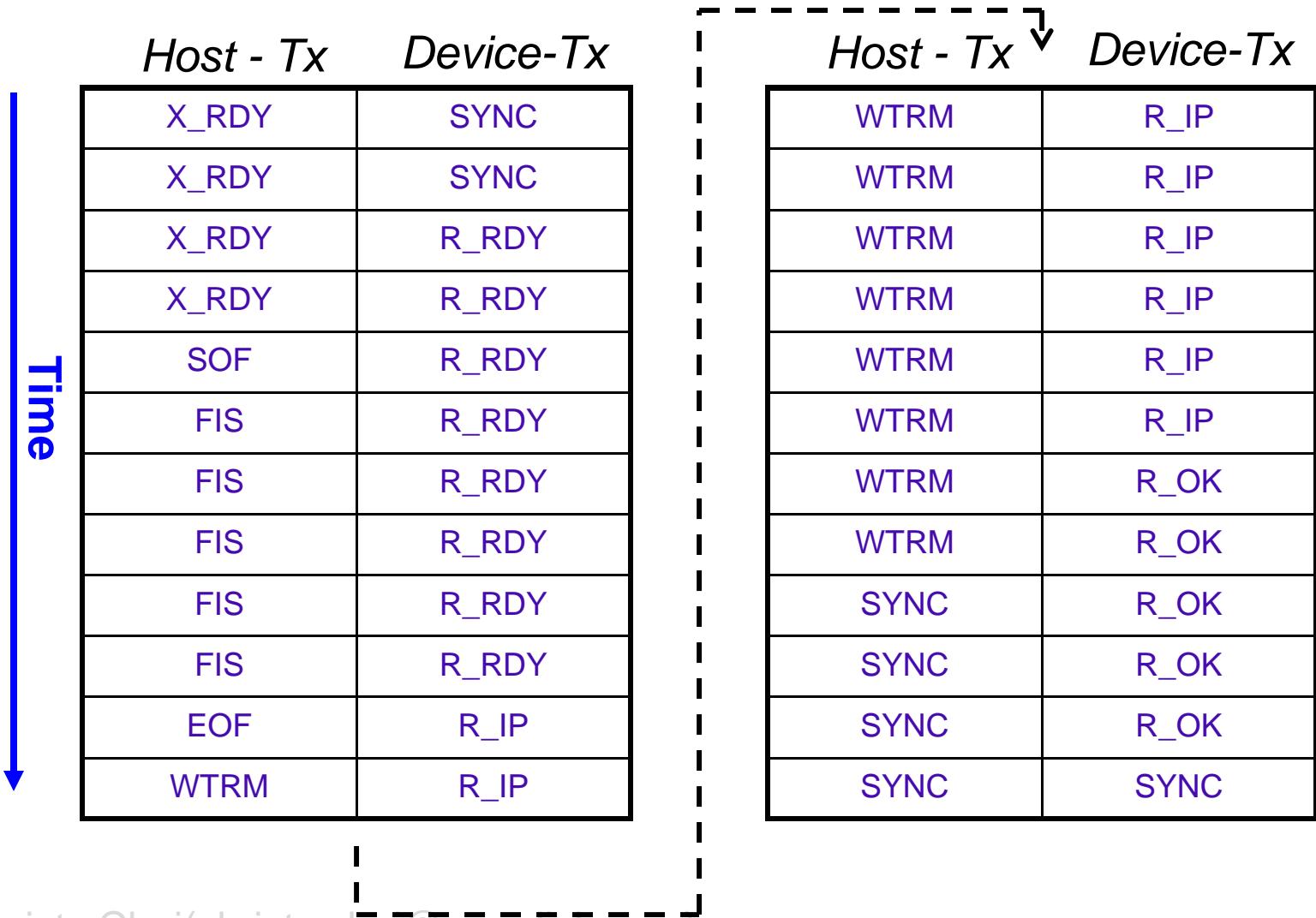
Receiver reports that
Primitive FIS was received
without errors

FIS Transmission Example



Upon detecting R_OK, transmitter sends SYNC (which is understood as logical idle). When receiver detects idle, it also signals idle.

Column Presentation of FIS Transfer



Transport Layer

FIS Retry

Christy Choi(christy.choi@sandisk.com)
Do Not Distribute



We write the Books!

General

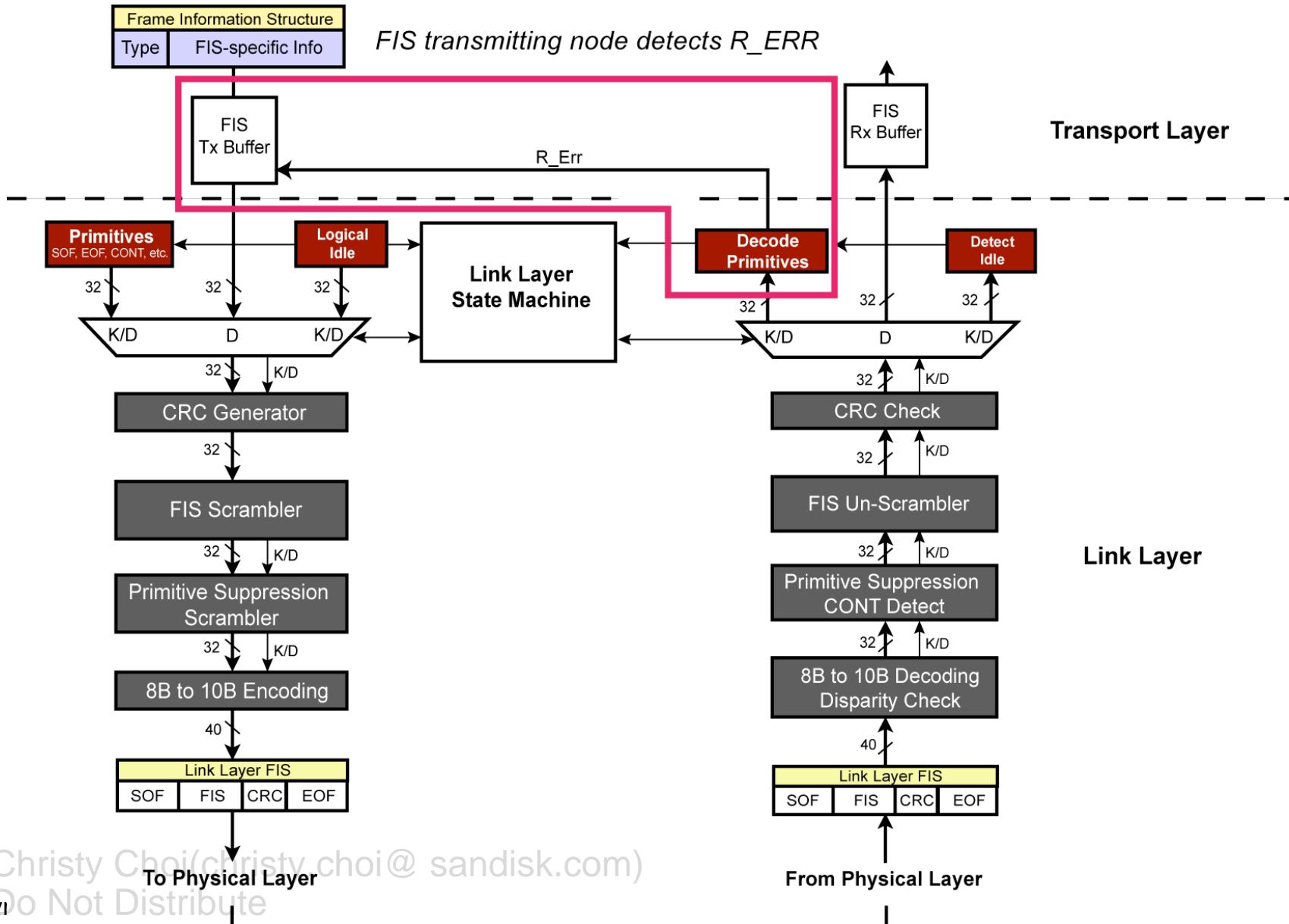
- FIS packets other than the DataFIS packets can be retried (resent) in the event of some types of transmission failure.
- A FIS retry is handled by SATA hardware (HBA and drives) without notifying software.
- The retry is possible because a copy of FISes that can be retried is kept in the Transport Layer buffer until confirmation is received that the FIS has transferred successfully.

FISes That Can be Retried

- FIS packets other than the DataFIS packets and BIST Activate can be retried (resent) in the event of some types of transmission failure.

FIS Type	Size	Retry?
Data	2049 DWs	No
First Party DMA Setup	7 DWs	Yes
Register FIS	5 DWs	Yes
PIO Setup	5 DWs	Yes
BIST Activate	3 DWs	No
Set Device Bits	2 DWs	Yes
DMA Activate	1 DWs	Yes

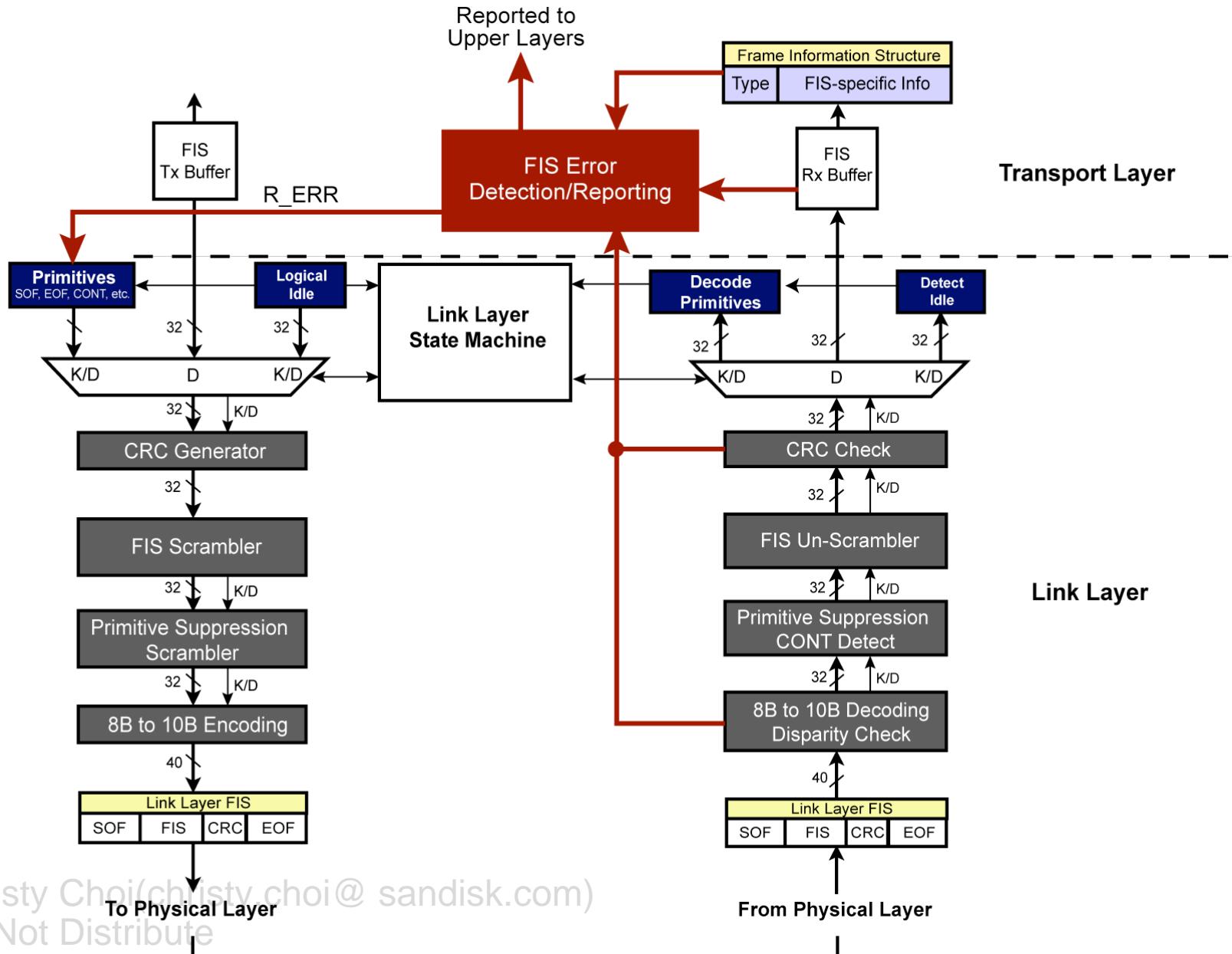
Transmitting Node Detects Error



What Errors Can Be Retried

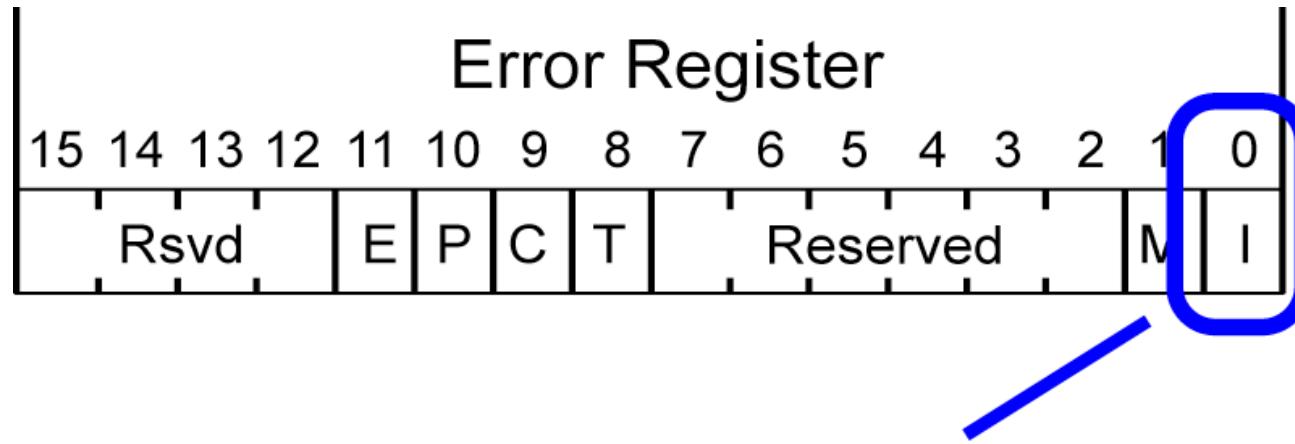
- Errors that are transient typically qualify for retry.
- Errors are reported via two mechanisms:
 - Delivery of R_ERR back to FIS transmitter
 - Notifying the upper layers to set appropriate SATA Error (SError) register bits.

FIS Receiver Detects & Reports Error



Recovered Error Reporting

Error Register field with SError register



Integrity error (data recovered) - a transient error solved with FIS retry. Software is not involved in solving this problem.

Transport Layer Non-Transient Errors

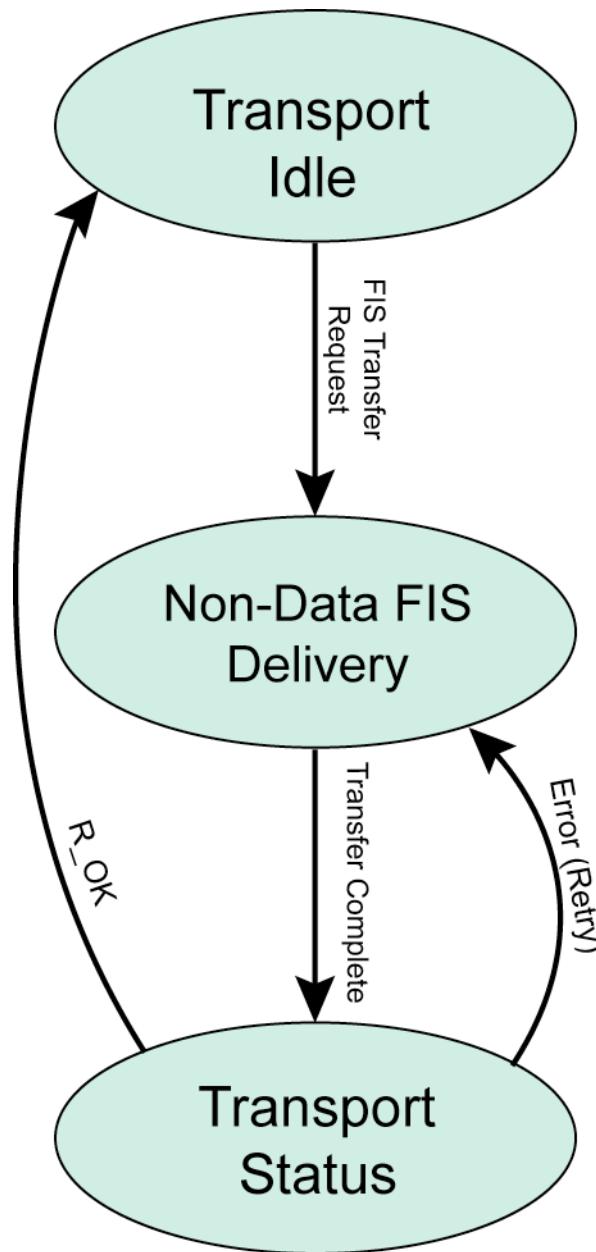
Non-Transient Errors, not retried:

- Internal Transport Errors — these errors are detected within the Transport layer and may include buffer overflow errors, etc.
- Frame Errors — these errors occur when a problem exists with the FIS itself, such as a malformed header. Some of these errors may be considered transient and others may be persistent.
- FIS Transmission Protocol Errors — when the FIS transmission protocol is violated, such as events occurring out of order, or when the data payload size disagrees with the requested size

RegFis Transport to Device

- Reg FIS transmission error results in Transport Layer retry
- Other results cause return to idle:
 - Successful transfer
 - illegal transitions
 - detection of FIS sent from device
 - Transmission error with Reset asserted

This is a simple example showing the state machine interaction.



Data Flow Control

Christy Choi(christy.choi@sandisk.com)
Do Not Distribute



SATA Flow Control

- Goal of flow control:
 - Allow designers to use receive buffers that are much smaller than the max frame size (2064 DWs):
 - Lower cost, almost “buffer-less” design
 - If the buffer begins to fill up, pause transmission until condition resolved

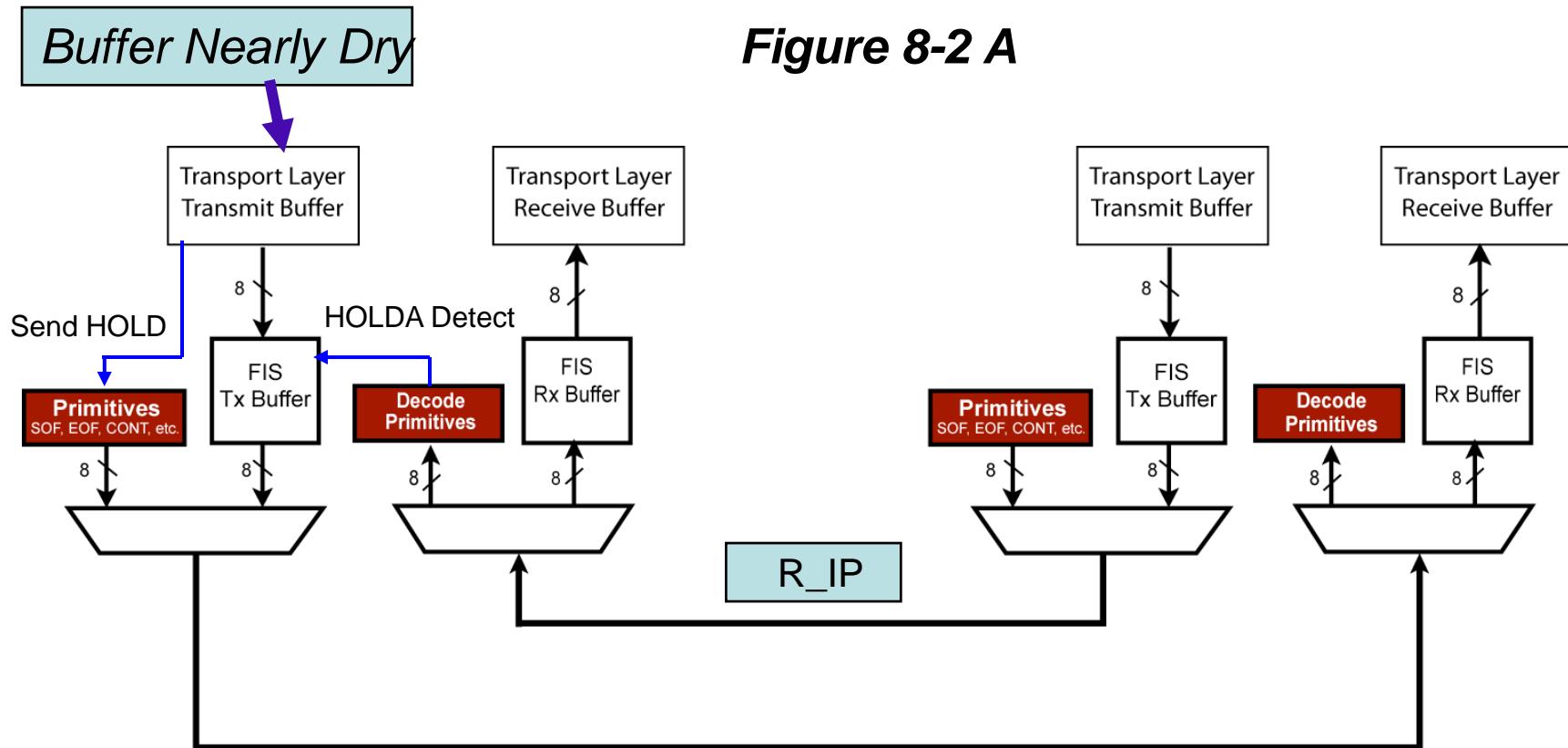
Flow Control Overview

See *Figure 8-1 on page 135*

- Transmitter (Tx) Flow Control
 - Sender can assert HOLD to indicate buffer dry condition
 - Receiver responds with HOLDA, but data flow was already stopped
- Receiver (Rx) Flow Control
 - Receiver can pause the flow of data by asserting HOLD
 - Sender is required to stop transmission and send HOLDA in response

Transmitter Flow Control

If Tx approaches buffer-dry & doesn't have enough data to end the frame, it simply begins sending HOLD instead of the rest of the FIS

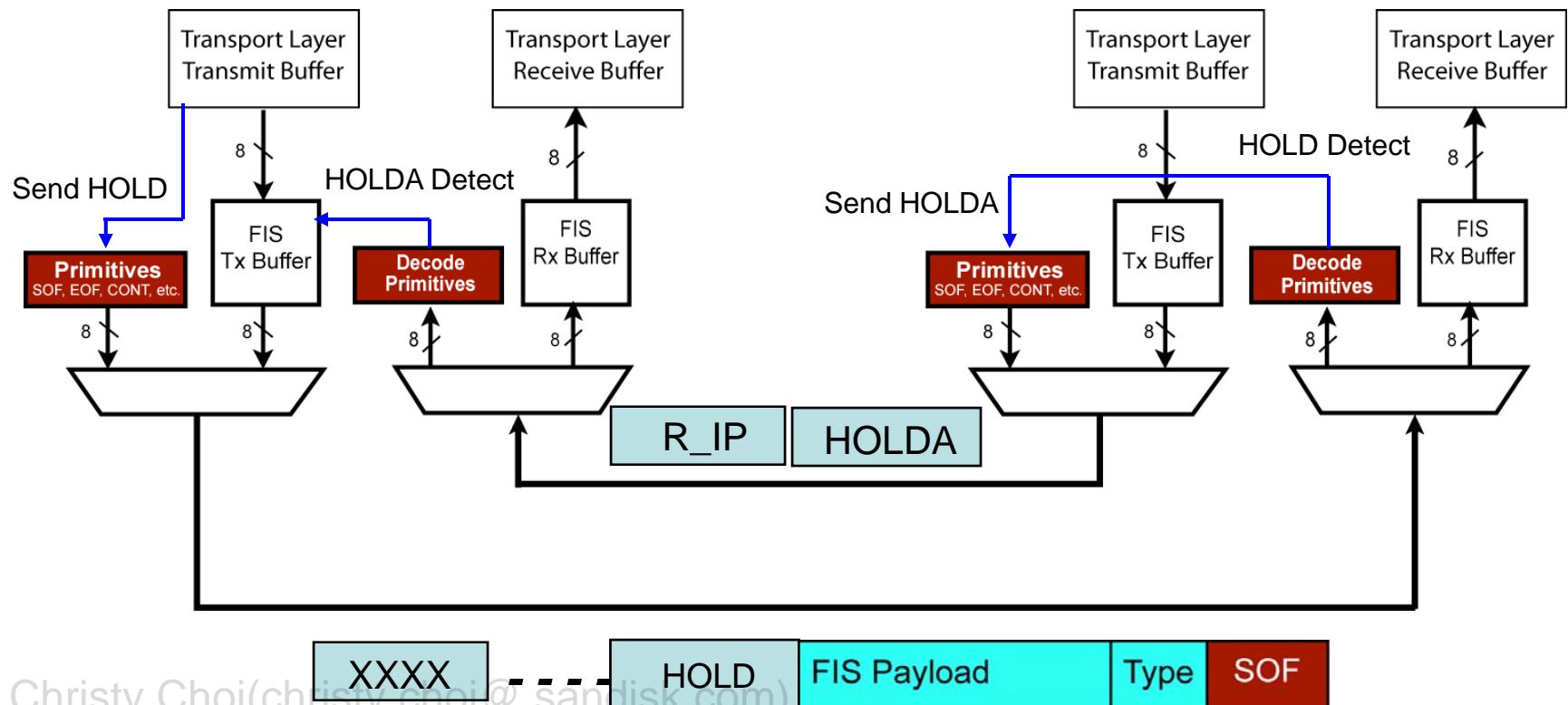


Transmitter Flow Control

SATA target responds to HOLD by returning HOLDA

Note: It doesn't need to keep track of buffer space since the transmitter has already stopped

Figure 8-2 B



Flow Control Example

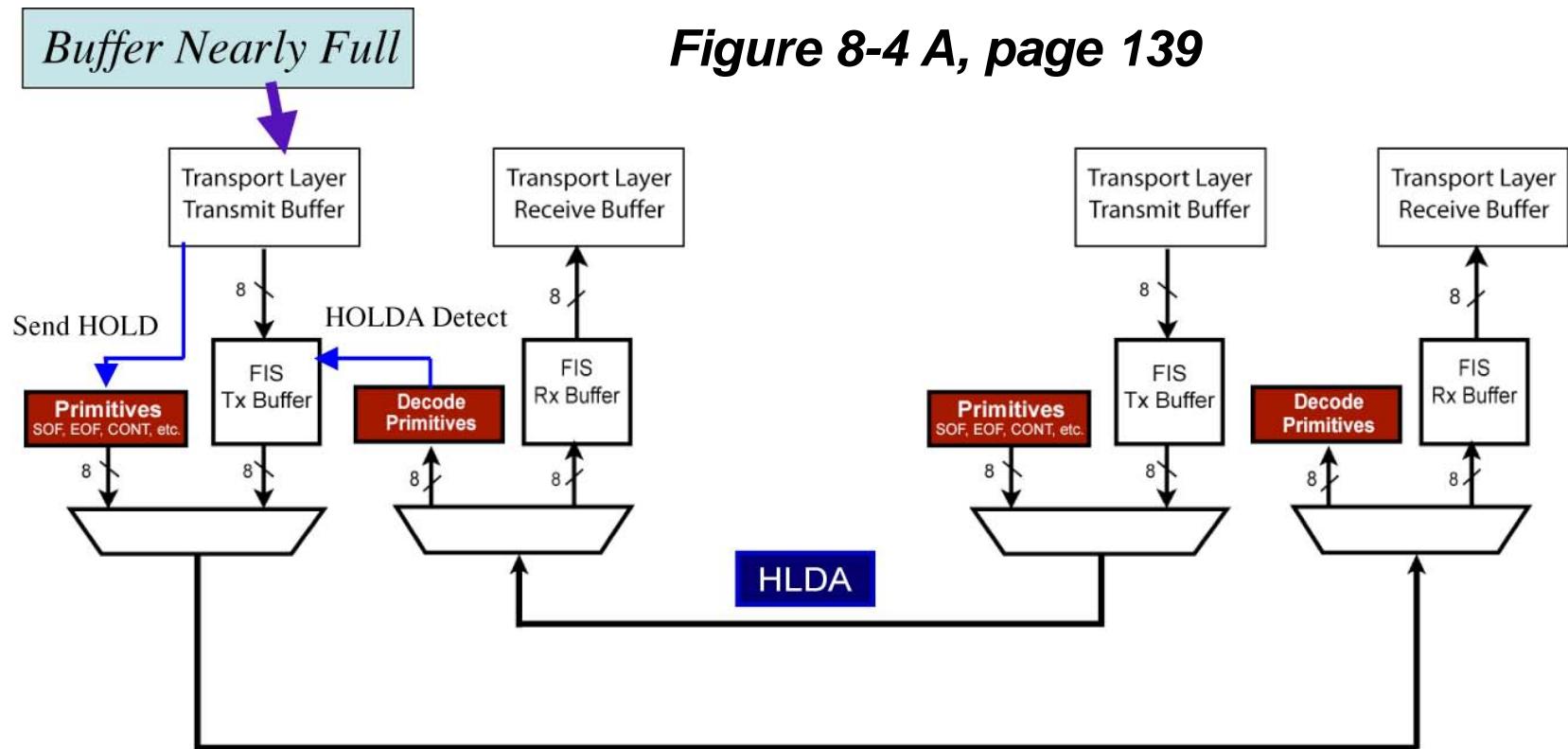
Figure 8-3, page 138

Note: this example shows the case in which the device is sending data but is approaching a buffer-empty condition, so it tells the receiver (HBA) that the data will be delayed.

H2	D2
XXXX (R_RDY)	XXXX (X_RDY)
XXXX (R_RDY)	XXXX (X_RDY)
XXXX (R_RDY)	SOF
XXXX (R_RDY)	00000046
XXXX (R_RDY)	HOLD
XXXX (R_RDY)	HOLD
XXXX (R_RDY)	CONT
XXXX (R_RDY)	XXXX (HOLD)
R_IP	HOLD
R_IP	HOLD
HOLDA	CONT
HOLDA	XXXX (HOLD)
CONT	XXXX (HOLD)
XXXX (HOLDA)	XXXX (HOLD)
R_IP	XXXX (HOLD)
HOLDA	XXXX (HOLD)
HOLDA	XXXX (HOLD)

Hold Released & Transfer Resumed

When Tx is ready again, it stops sending HOLD & resumes sending Data

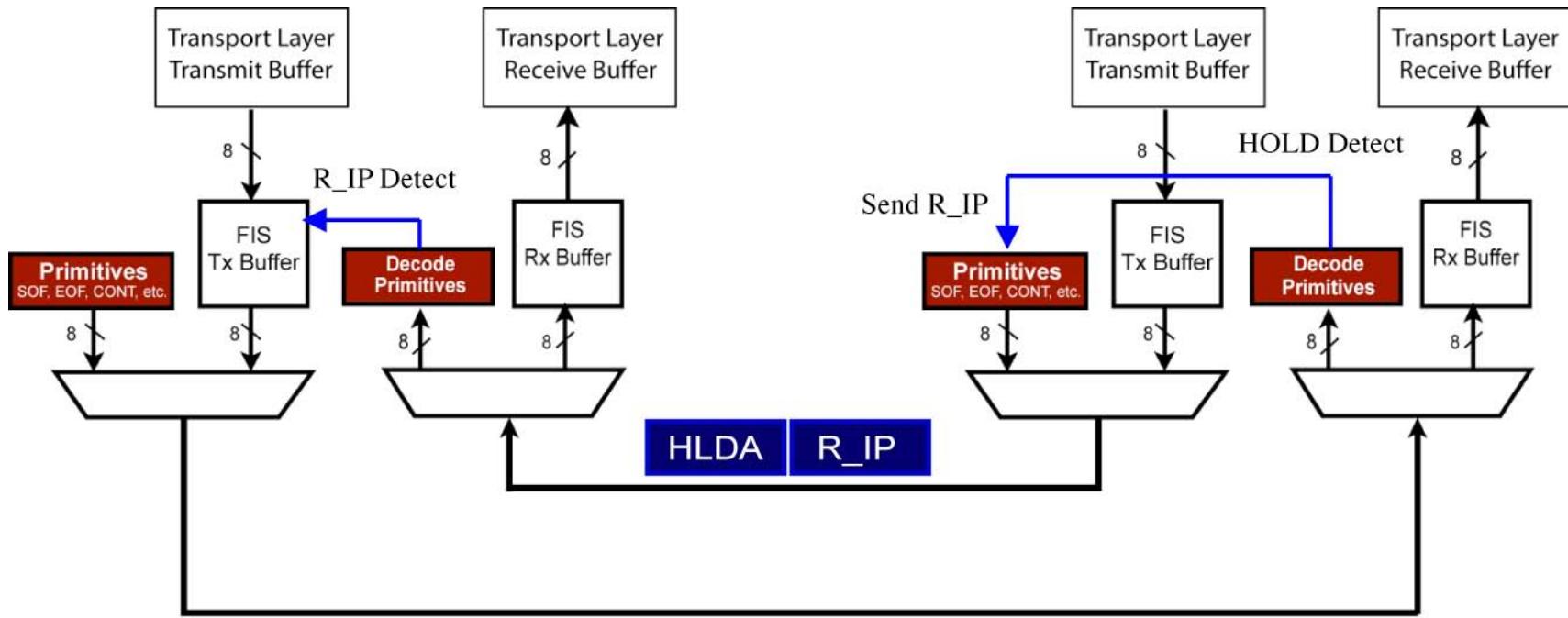


Christy Cho (ChristyCho@disk.com)
Copyright Not Distribute

Transmitter Flow Control

When Rx sees data resumed, it stops sending HLDA and resumes sending R_IPs

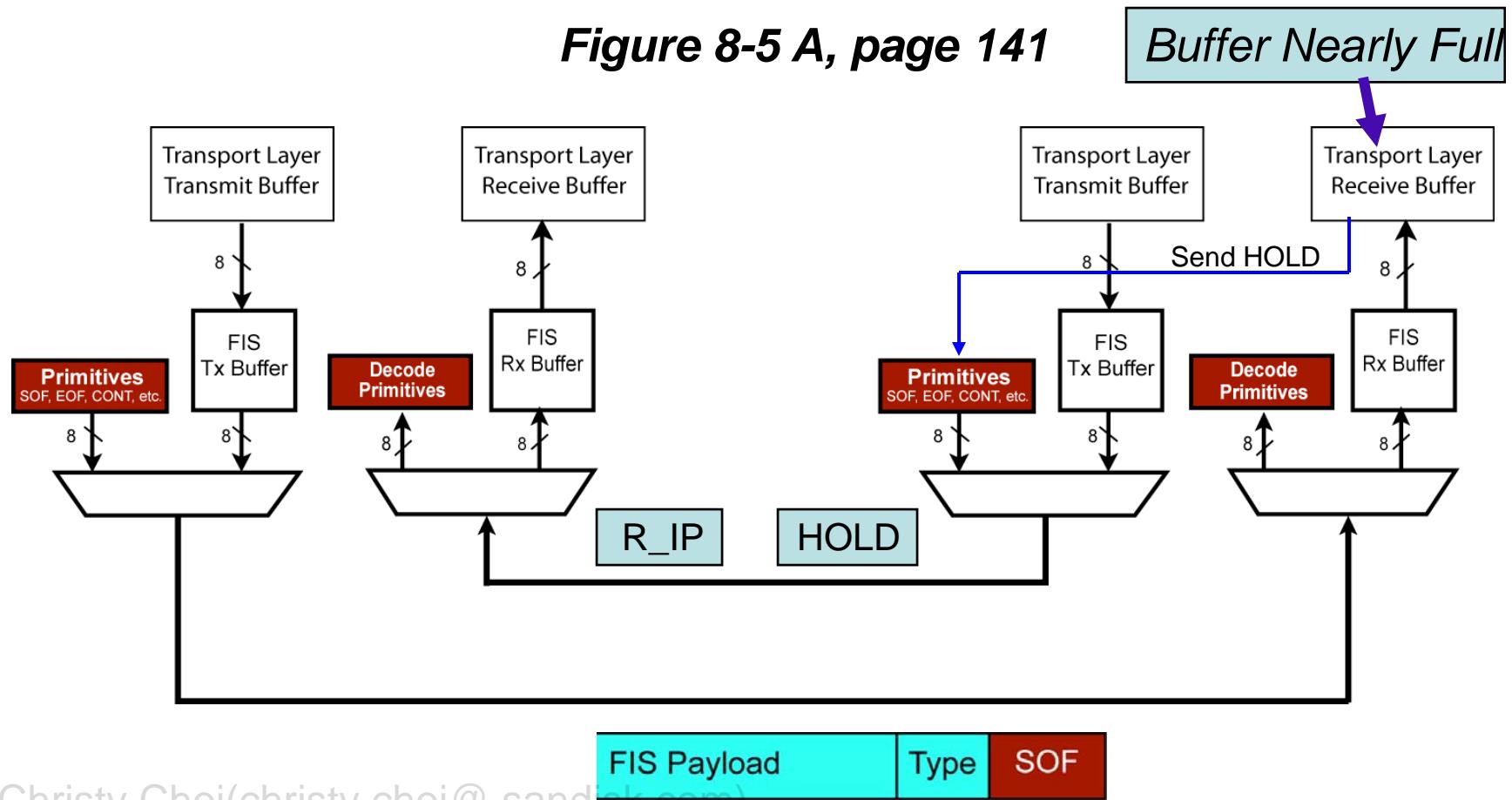
Figure 8-4 B, page 139



Receiver Flow Control

If SATA receiver approaches buffer full condition, it stops sending R_IP and begins sending HOLD instead.

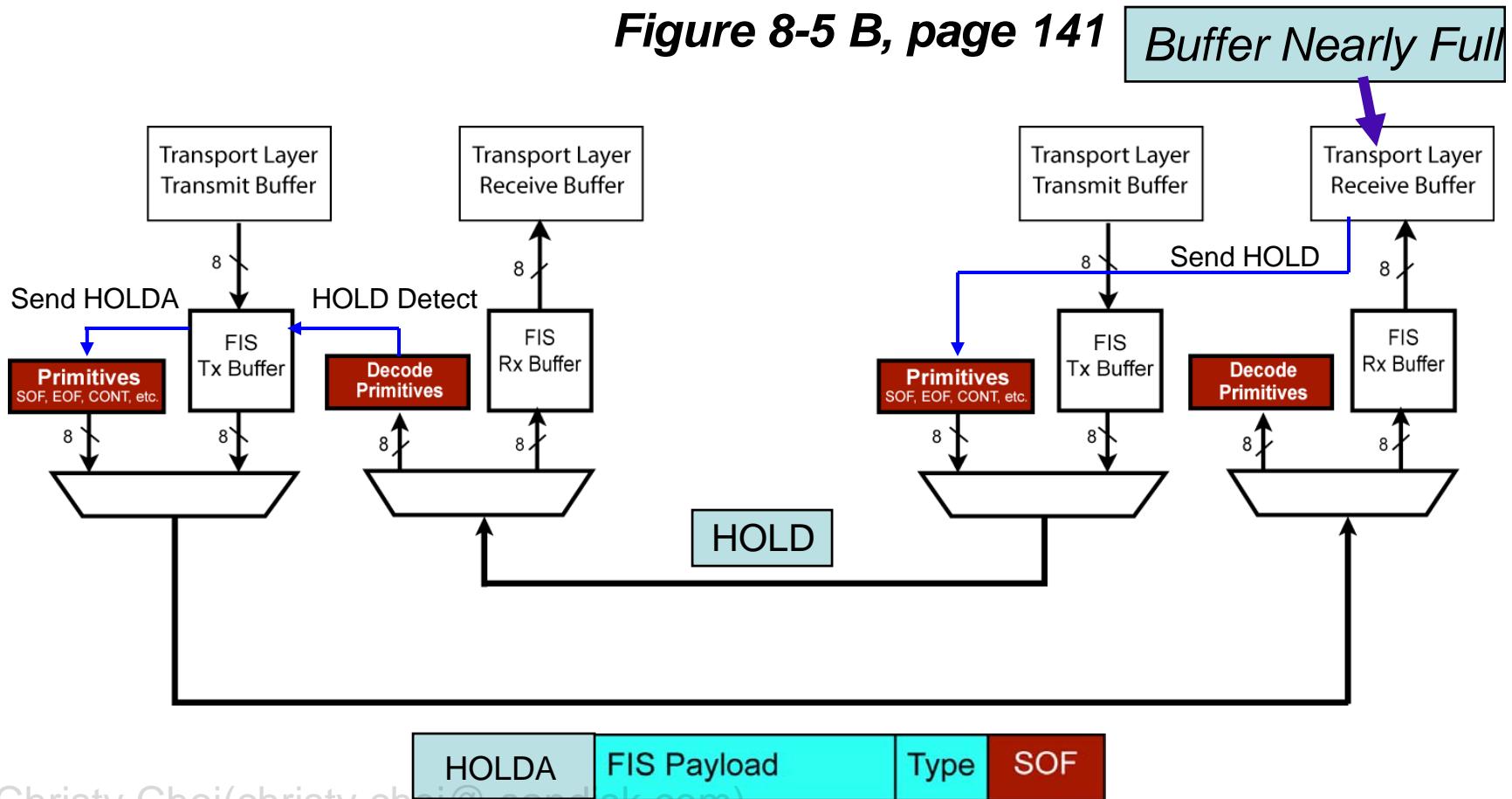
Figure 8-5 A, page 141



Receiver Flow Control

When SATA transmitter sees HOLD it must stop sending packets and respond with HOLDA.

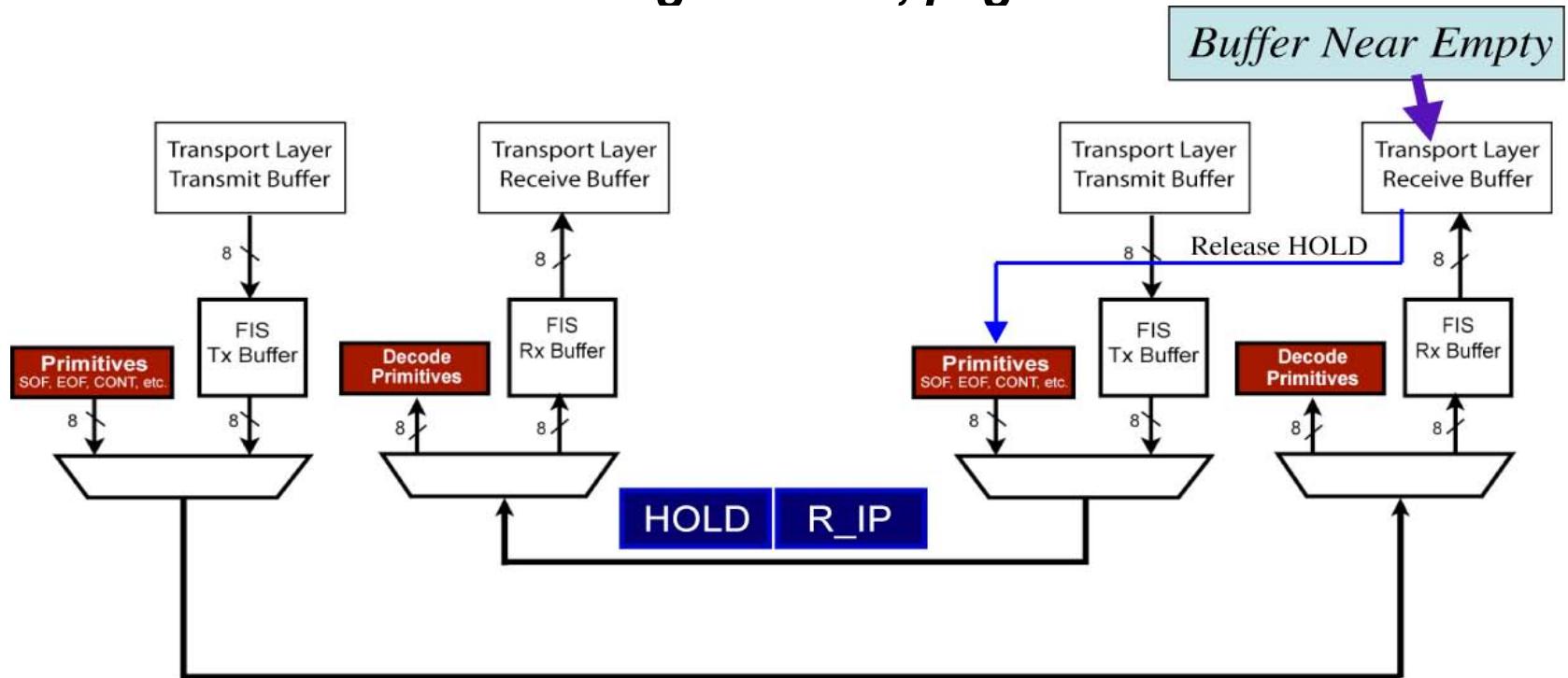
Figure 8-5 B, page 141



Receiver Flow Control

When receiver is ready again, it resumes sending SATA_R_IP

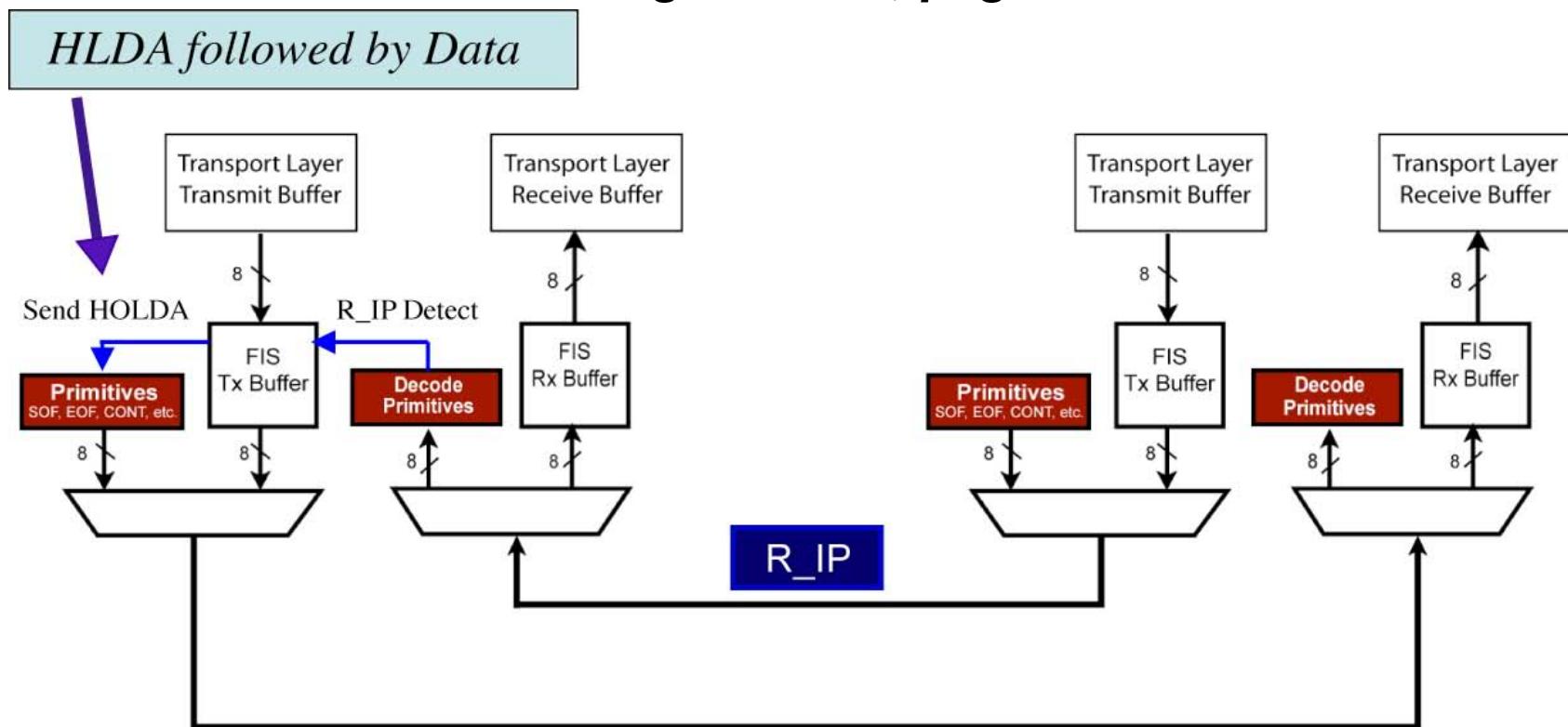
Figure 8-6 A, page 143



Receiver Flow Control

Transmitter responds by sending HLDA again (to break out of the CONT mode) and then resumes data transmission

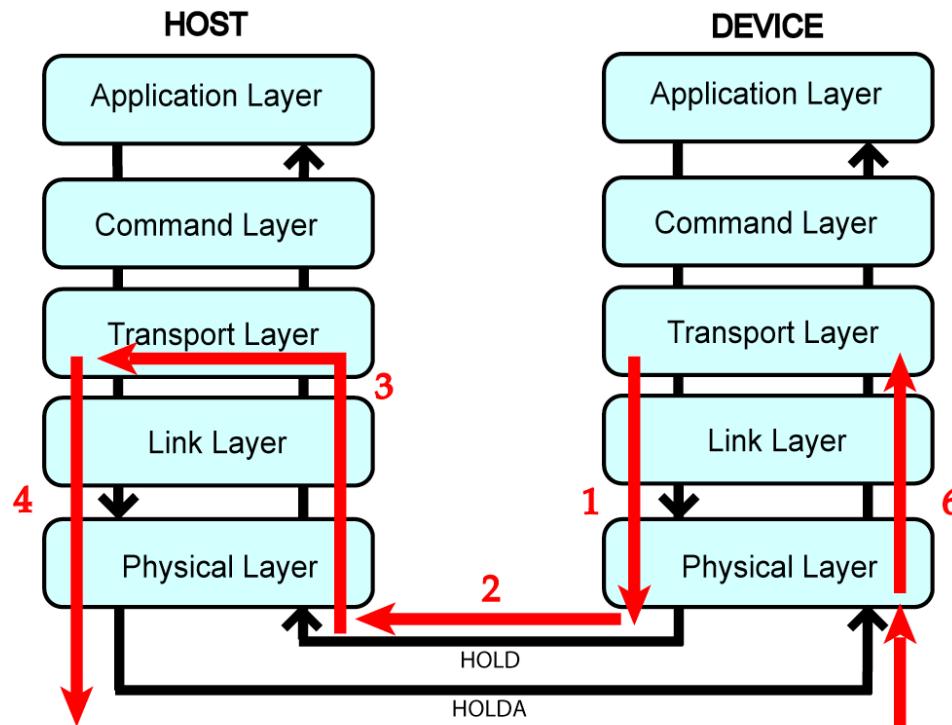
Figure 8-6 B, page 143



HOLD/HOLDA Propagation Delay

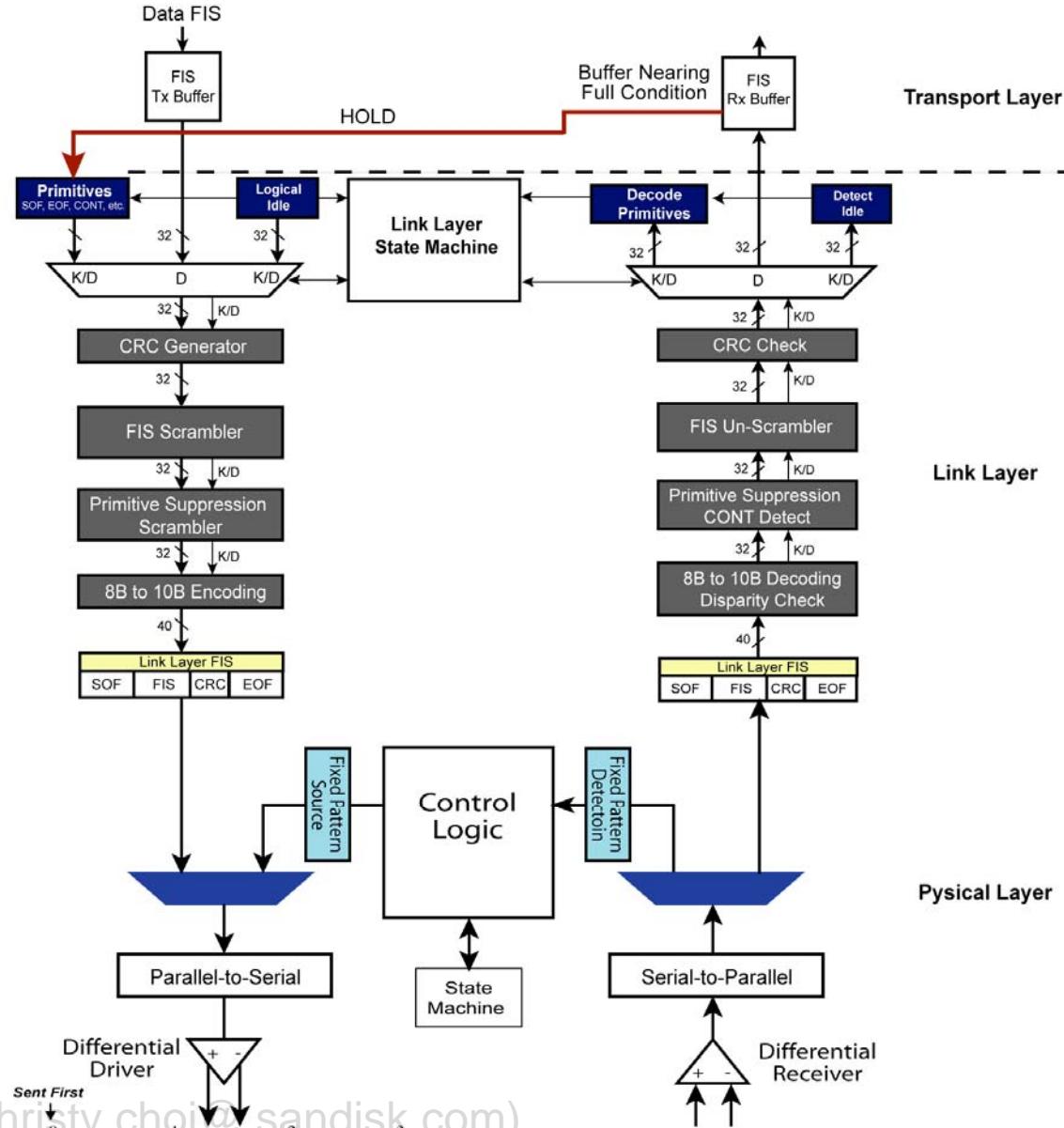
The prop delay from SATA transmitter to receiver is small but nonzero. This delay consists of six elements.

Figure 8-7, page 144



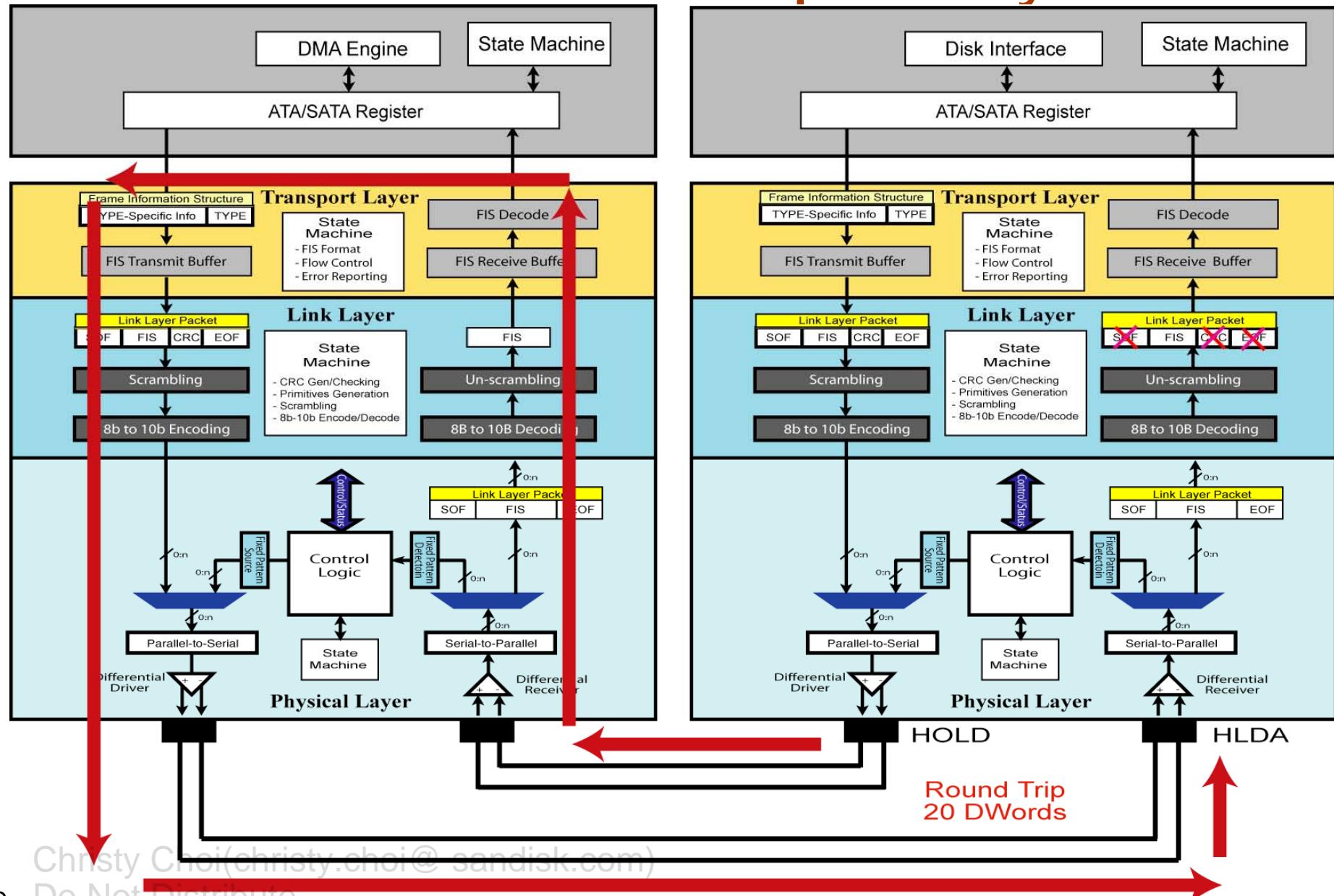
1. Receiving Node HOLD Transmit Delay
2. Link Propagation Delay
3. Transmitting Node HOLD Receive Delay
4. Transmitting Node HOLDA Transmit Delay
5. Link Propagation Delay
6. Receiving Node HOLDA Reception Delay

Internal HOLD Transmission Delay



Christy Choi(christy.choi@sandisk.com)

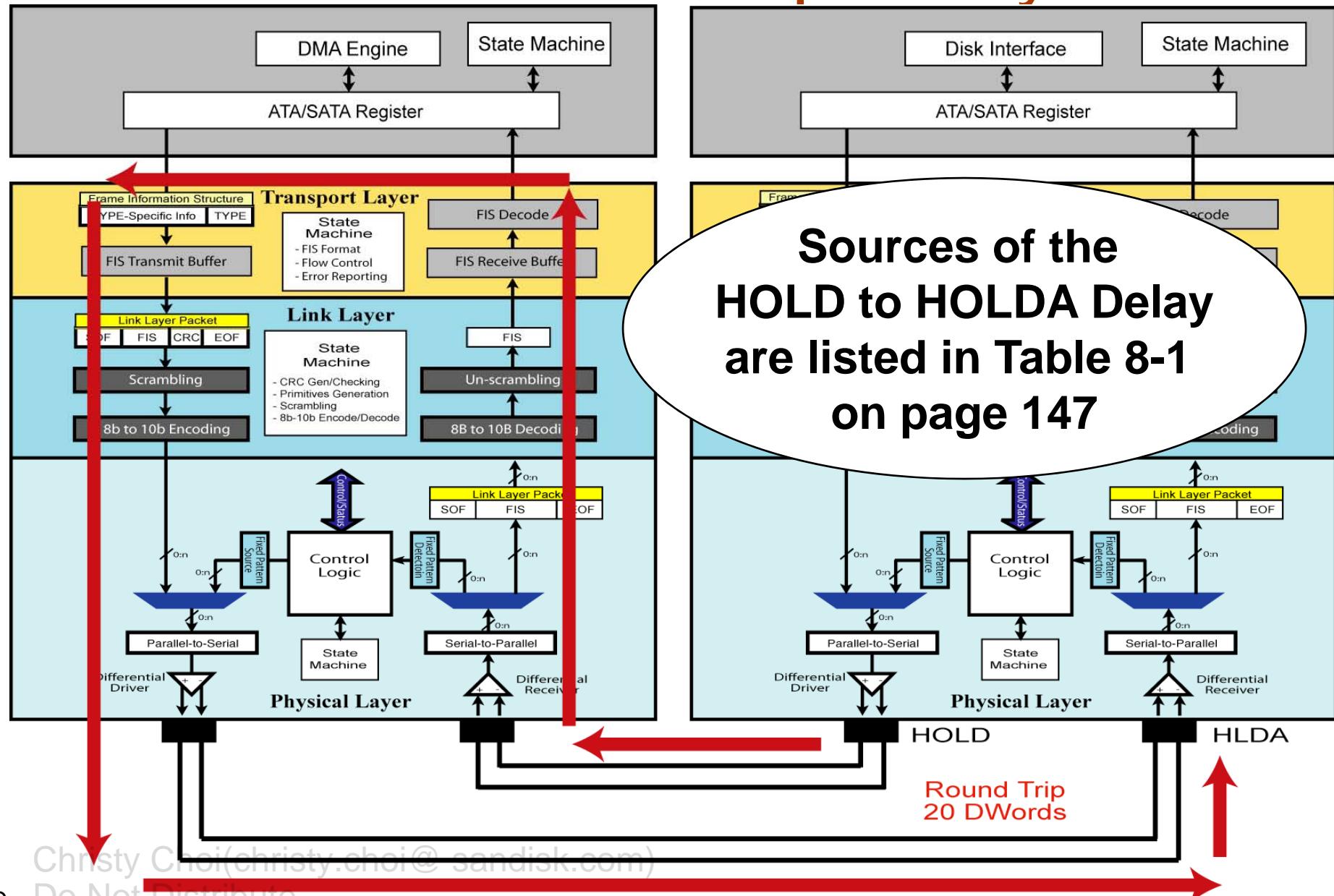
Total Round Trip Delay



Definition of 20 Dword Delay

- Maximum allowed latency defined as: Time from MSB of the HOLD primitive on the wire until the MSB of the HOLDA is on the wire
- Cannot be more than 20 Dword symbol times, and receiver must be able to accept 20 more Dwords after it transmits HOLD
- Transmitter must respond with HOLDA within 20 Dword symbol times.

Total Round Trip Delay



Receiver Flow Control Capture

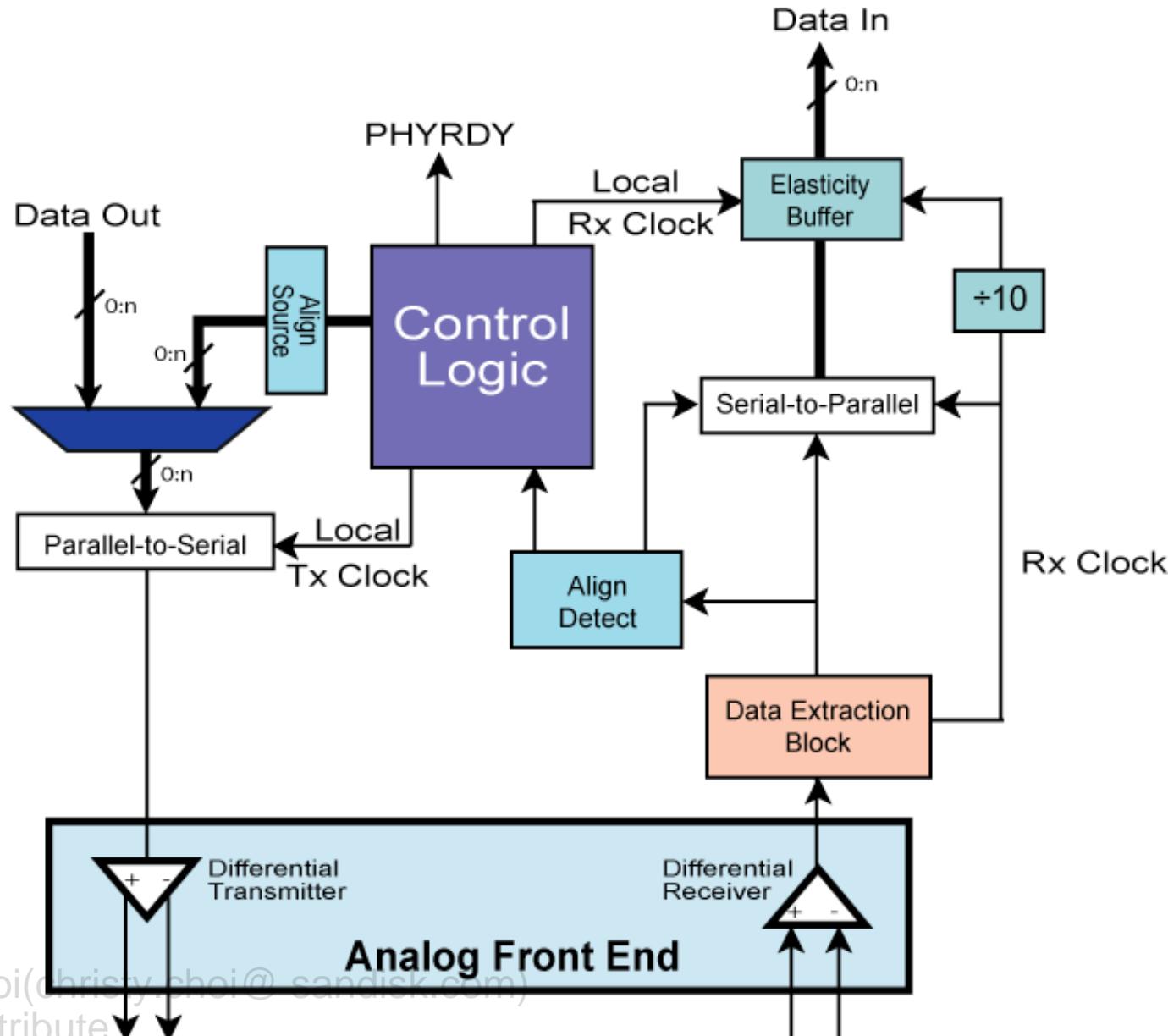
H1	D1
00000000	HOLD
00000000	HOLD
00000000	CONT
00000000	XXXX
HOLDA	XXXX
HOLDA	XXXX
CONT	XXXX
XXXX	HOLD
XXXX	R_IP
XXXX	R_IP
XXXX	CONT
XXXX	XXXX

Logical PHY Functions

Christy Choi(christy.choi@sandisk.com)
Do Not Distribute



Physical Layer Functional Blocks



Physical Layer Functions

FIS Transmission:

- Parallel to serial conversion
- Differential transmission at 1.5 or 3.0Gb/s
- Align primitive insertion for clock compensation
- Spread-spectrum clocking optional

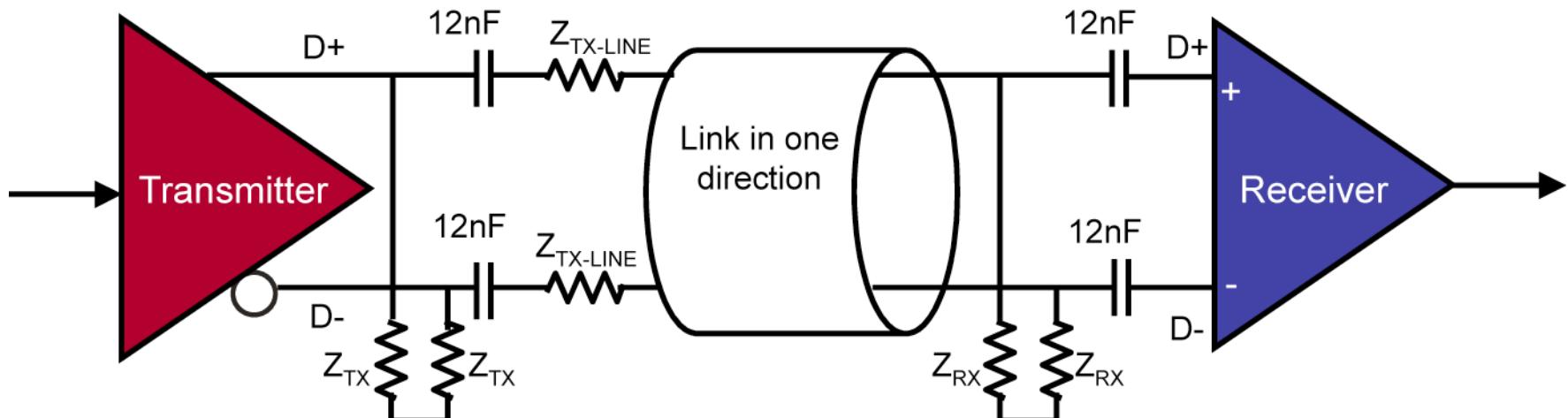
Physical Layer Functions

FIS Reception:

- Differential reception
- Data Extraction
- Serial to parallel conversion
- Use ALIGN primitives for clock compensation in the Elasticity buffer

Differential Transmitter/Receiver

SATA may or may not have Caps on either end of the cable. Typical drive implementations frequently include the capacitors



$$Z_{TX} = Z_{RX} = 50 \text{ Ohms}$$

Coupling capacitor = 12 nF max

The DC common mode impedance is typically 50 ohms, differential impedance is 100 ohms.
The coupling capacitor is between 10 and 12 nF.

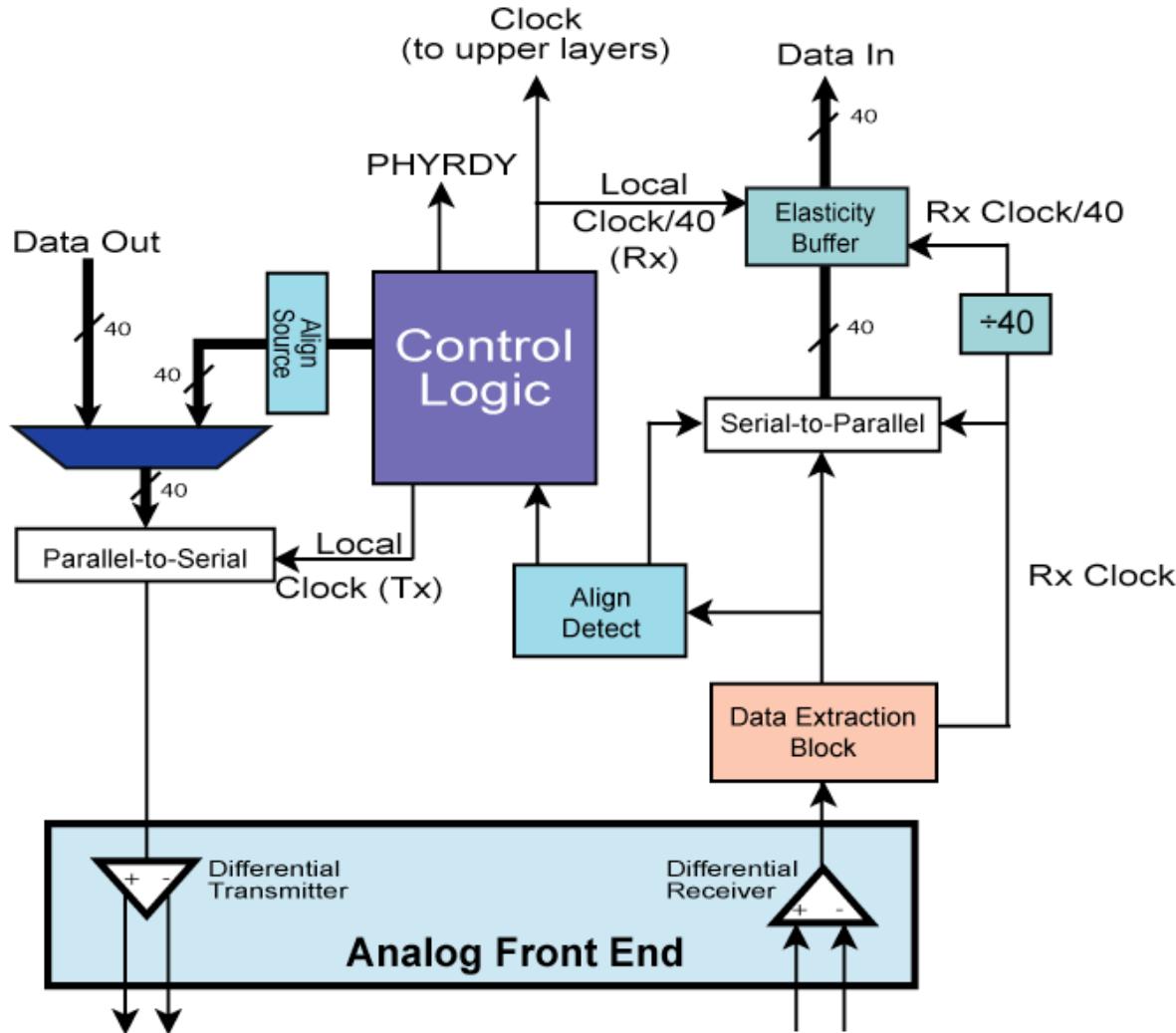
Min/Max Transmit Voltages

Application	Gen 1	Gen2
Internal (i)	400 - 600mV	400-700mV
Medium (m)	500-600mV	500-700mV
External (x)	800-1600mV	800-1600mV

Min/Max Receive Voltages

Application	Gen 1	Gen2
Internal (i)	325 - 600mV	275 - 750mV
Medium (m)	240 - 600mV	240 - 750mV
External (x)	275 - 1600mV	275 - 1600mV

Local Clock Distribution



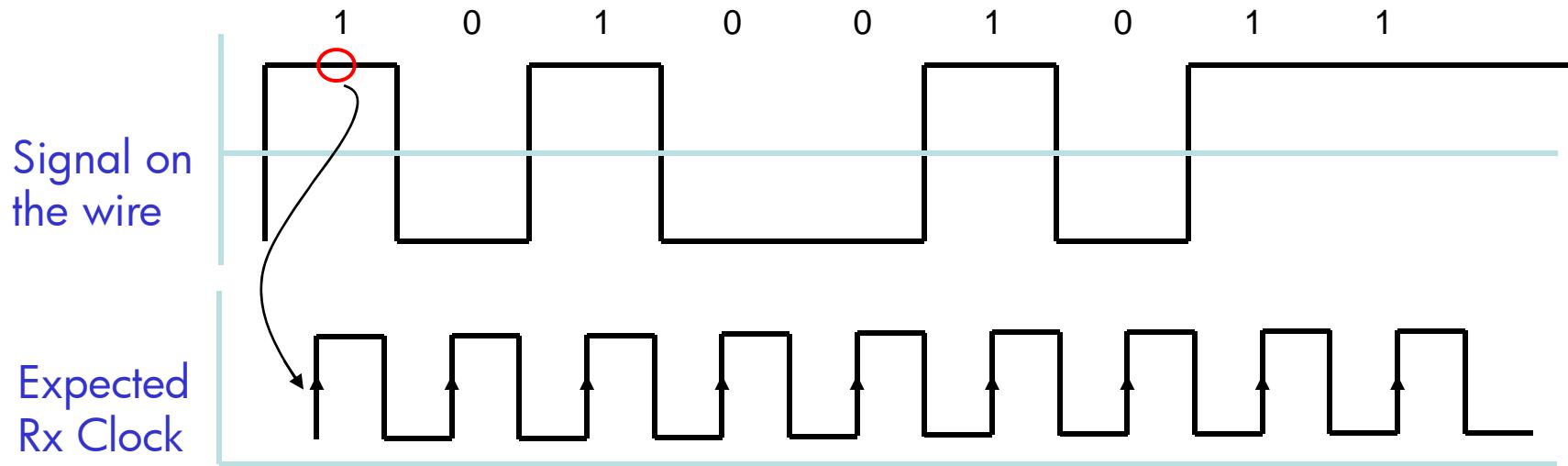
Tx Clock

- The serial output is clocked out to the differential driver at 1.5 or 3.0 Gbps
- Accuracy:
 - +350/-350ppm for SATA.
 - SSC (Spread Spectrum Clocking) is permitted, resulting in large apparent jitter.
SSC accuracy +0/-5000ppm
 - Clock accuracy with SSC is +350/-5350
- Note that Tx Clock is different from the Rx local clock even though both run at same rate. They are in different clock domains.

Rx Clock

- Using PLL or other means, receiver circuit generates a Rx Clock from transitions in the input data
- Rx Clock has same frequency as the Tx Clock at the transmitter
- 1.5 GHz or 3.0 GHz clock recovered
- Rx Clock latches incoming packets into register and elastic buffer
- “Rx Clock” is different from “Rx Local Clock”, which is used to clock data out of the elastic buffer (+350/-5350ppm difference)

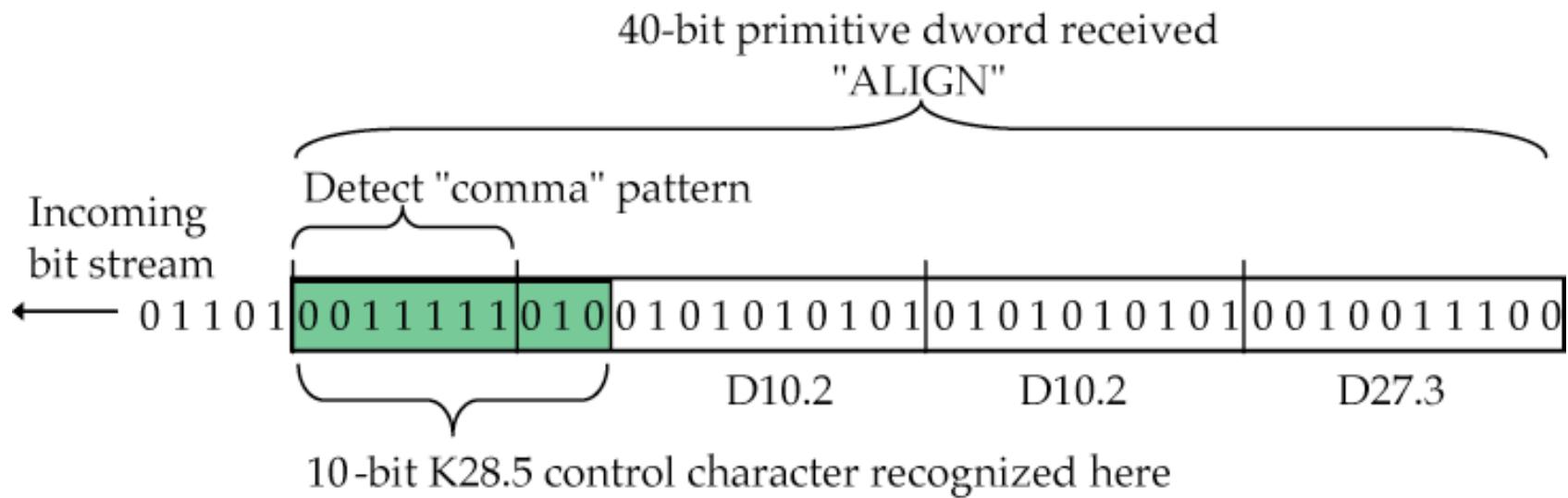
Clock Recovery



Receiver PLL knows what the negotiated frequency is, only needs to align expected clock rising edges away from signal transitions to reliably latch incoming bits.

Dword Detection

- Once receiver clock is recovered, detect Dword boundaries
 - Search incoming bits for the 7-bit COM pattern
 - When pattern is found, the next 3 bits will finish the COM symbol. The following 30 bits should then complete a 40-bit Dword ALIGN primitive, so dword bit position is known.



Clock Compensation

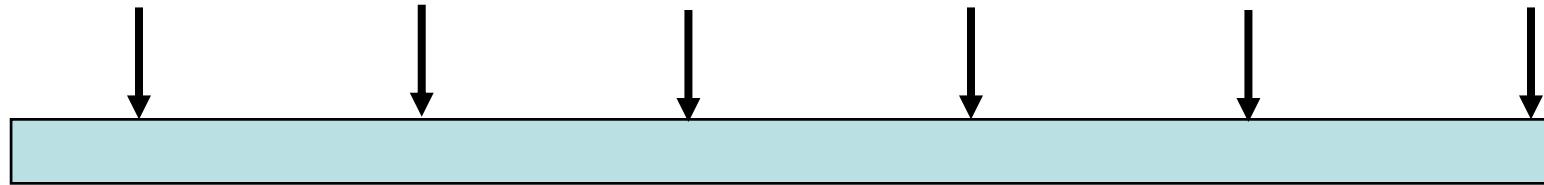
- Differences between the transmit clock (recovered clock from the received bit stream) and the local receive clock must be managed.
- +350/-5350ppm clock accuracy

Align Primitive

- Link Layer has a counter that rolls over every 1024 characters (256 Dwords). It sends two consecutive ALIGN primitives that are included in the 256 Dword count.
- After communications have been established, the first and second words sent from the Link Layer are the two ALIGN primitives, followed by at most 254 non-ALIGN Dwords. The cycle repeats during SATA transmissions.
- Regular arrival of ALIGNs allows the elastic buffer to add or remove them as needed to compensate for differences in the clocks without affecting the Dword stream.

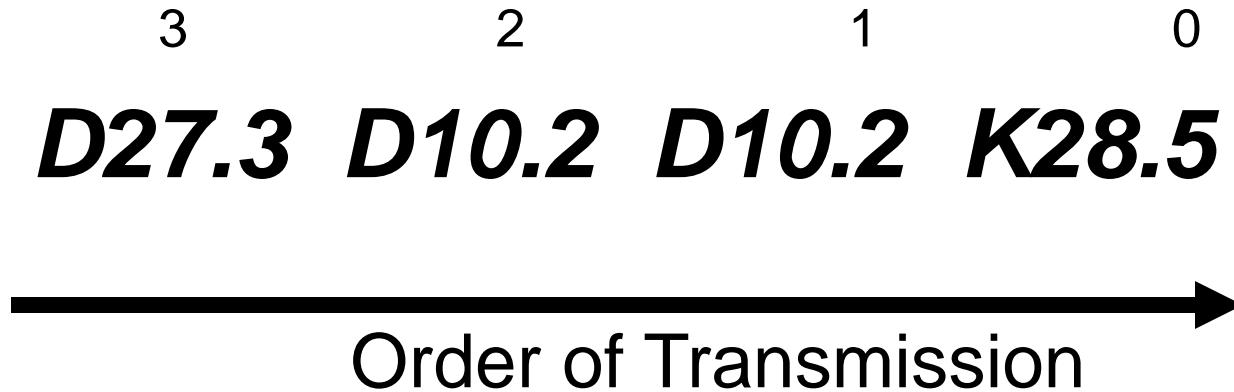
Periodic Align Primitives

Align primitives inserted every 256 DWs and
are included in the 256 DW count



— *Bit Stream* — →

Align Primitive



(rd+)	(rd-)		
1100000101	001111010	Align1	(K28.5)
0101010101	0101010101	Align2	(D10.2)
0101010101	0101010101	Align3	(D10.2)
1101100011	0010011100	Align4	(D27.3)

Error Detection/Handling

Christy Choi(christy.choi@sandisk.com)
Do Not Distribute



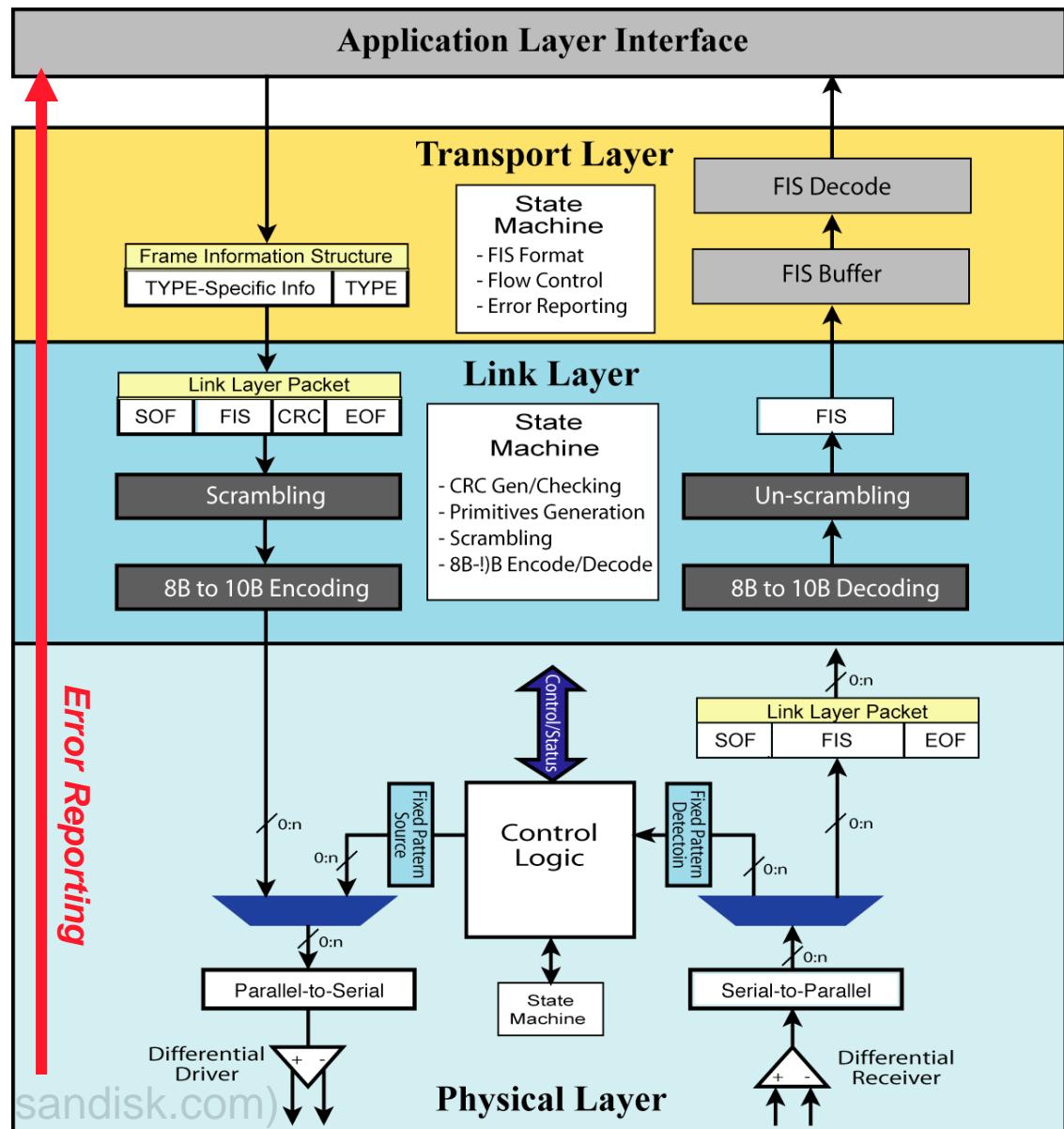
Layered Architecture - Errors

Command Errors

Framing/Protocol Errors

Link-Related Errors

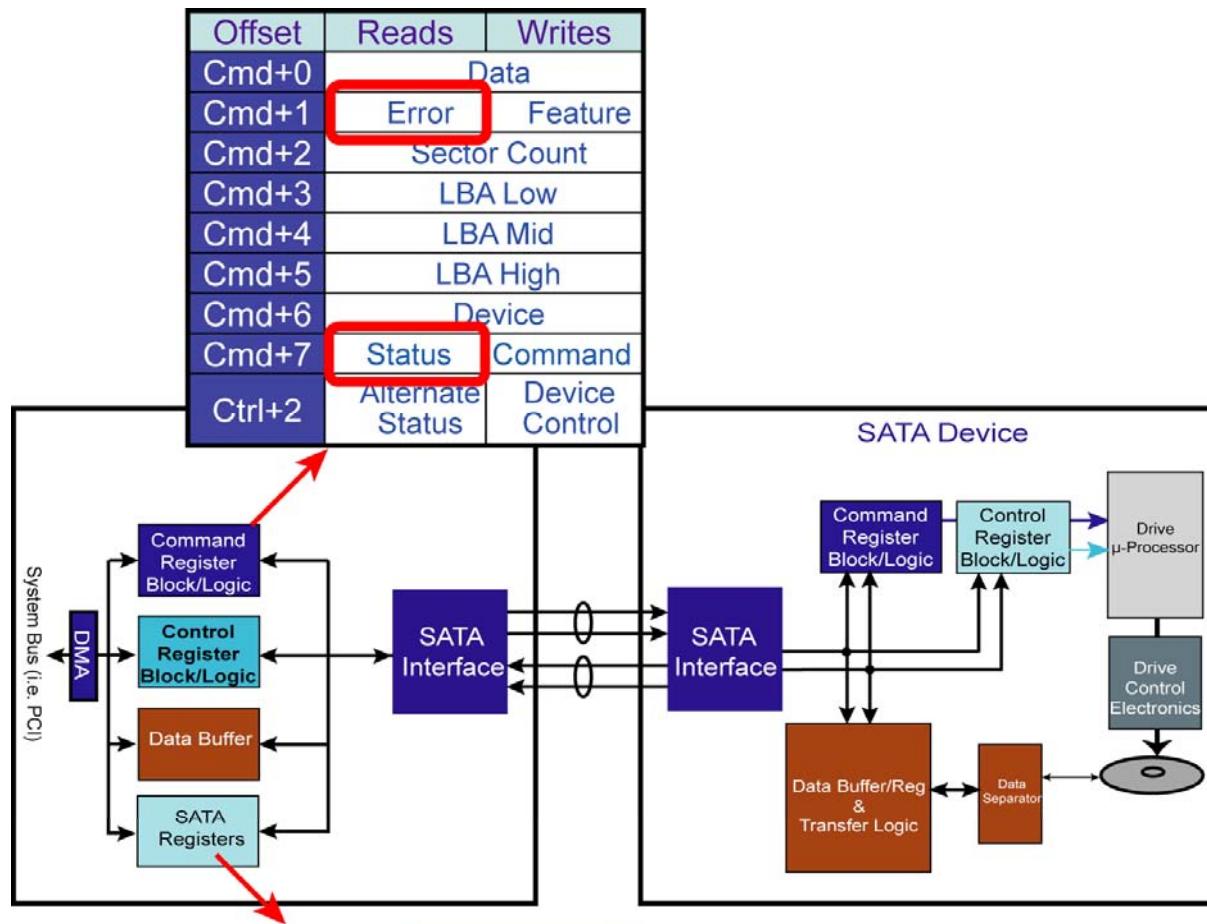
Signaling Errors



Scope of SATA Error Checking

- Commands failing to complete properly — reported in the ATA completion status and error registers
- FIS transfer Protocol Errors — reported in the SATA-specific error register
- SATA Link Transfer Errors — reported in the SATA-specific error register
- HBA Errors — reported in the SATA-specific error register and/or in IO bus-specific registers (e.g., in PCI configuration status registers)

Location of Error-Reporting Registers



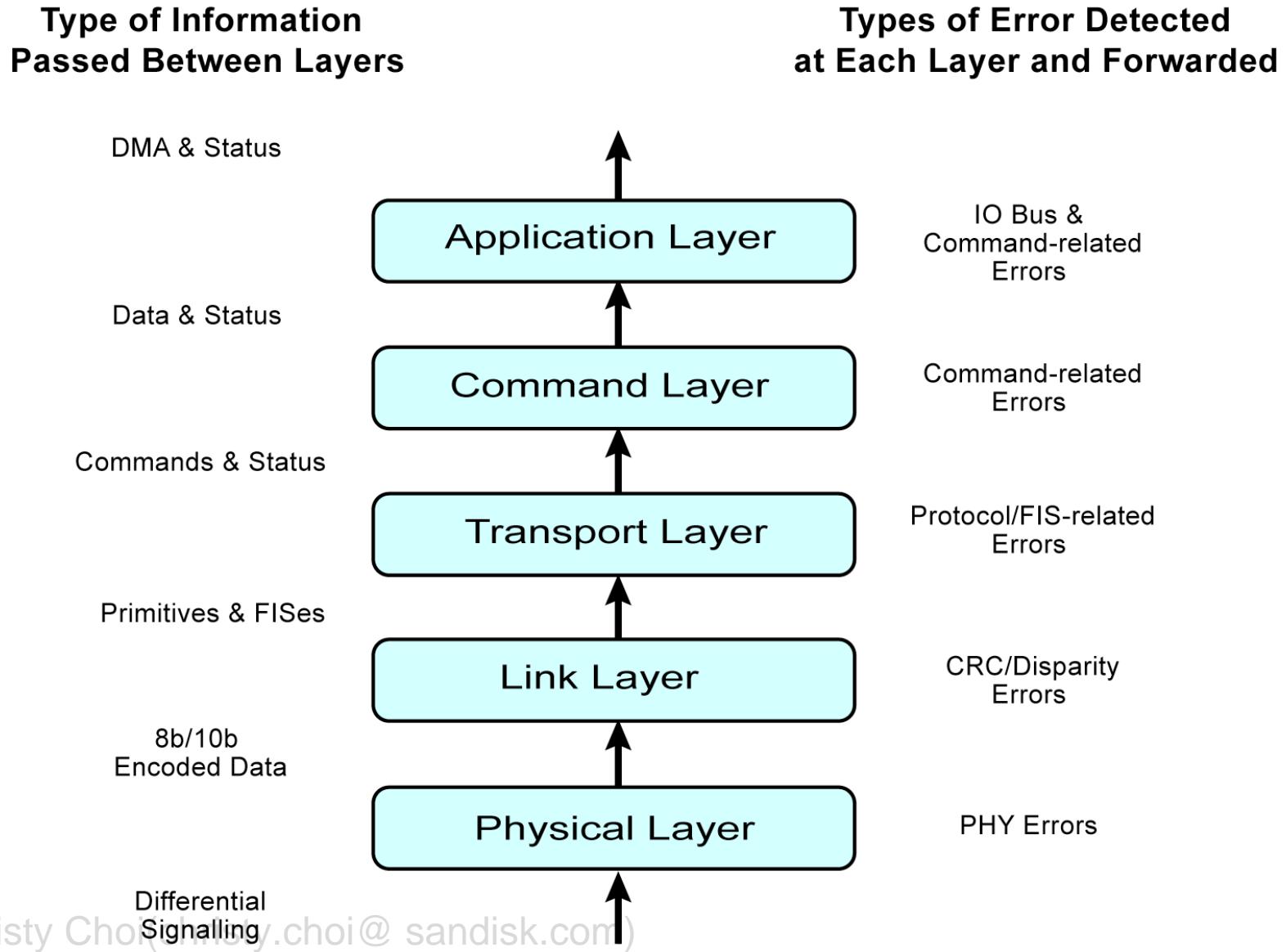
SCR [0]	SStatus
SCR [1]	SError
SCR [2]	SControl
SCR [3]	SActive
SCR [4]	SNotification

HBA vs SATA Drive Error Reporting

234

- Drives update the ATA Status and Error registers during command execution and forward results to the HBA via the Register FIS - Device to Host.
- Drives report FIS transmission errors via the FIS transfer protocol handshake (R_ERR). In this case, the HBA has no visibility regarding the nature of the error.

Error Reporting Methodology



Error Response Classifications

- **Ignore or Track** - action taken when error is recoverable
- **Retry** - used when error is considered to be transient and has not affected the integrity or state of the system
- **Abort** - result if error is considered persistent and typically would result in host software being notified
- **Freeze** - results from severe error that is not recoverable and typically requires reset to clear the error

ATA Status Register

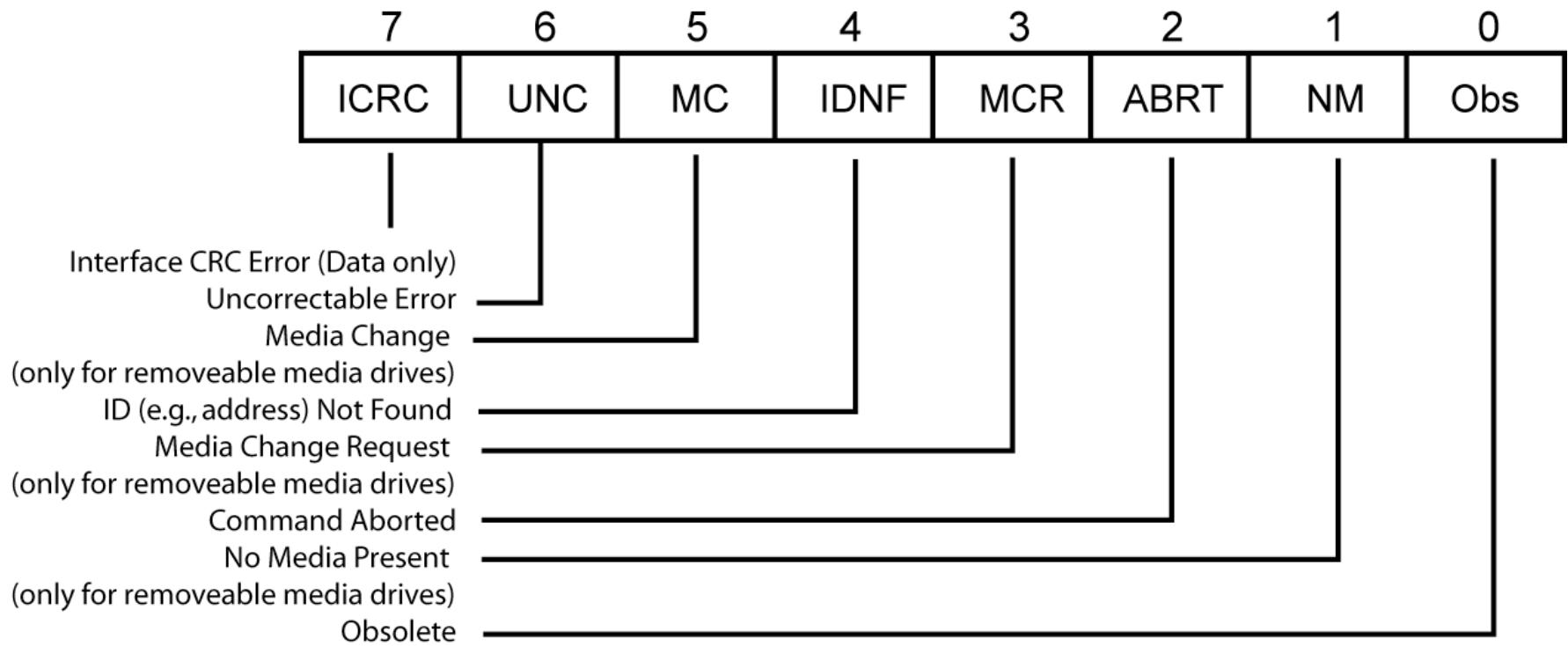
The ATA Status register located within the drive indicates when the drive has detected an error. The ERR bit being set indicates that the Error register has set one or more bits that define the error.

Figure 10-3, page 163



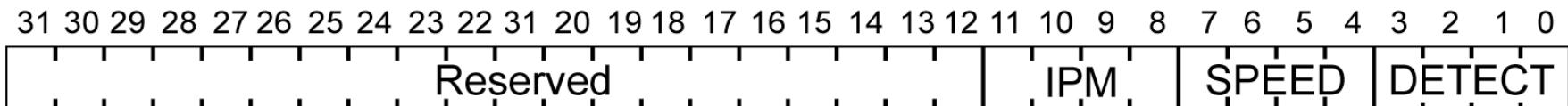
ATA Error Register

Figure 10-4, page 164



Phy Communication Status SStatus Register

Figure 10-4, page 164



IPM - Current state of Interface Power Management

0000 = No device present or communication not established

0001 = Active PM state

0010 = Partial PM state

0110 = Slumber PM state

Other values reserved

SPEED - Status of communication speed negotiated

0000 = No speed info; device not connected or no PHY communication

0001 = Base Communication rate established (Generation 1 speed)

Other values reserved

DETECT Reports Detect Status and the PHY State

0000 = Device not detected/no PHY communication

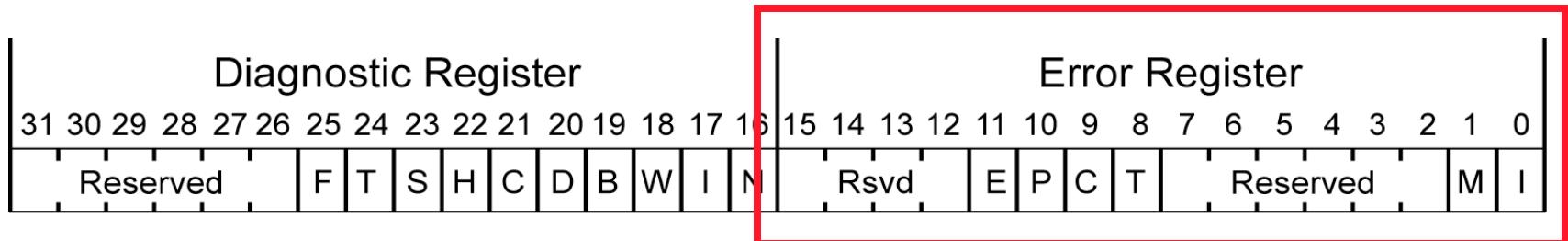
0001 = Device detected/no PHY communication

0011 = Device detected/PHY communicating

0100 = PHY offline (Interface disabled or running BIST Loopback)

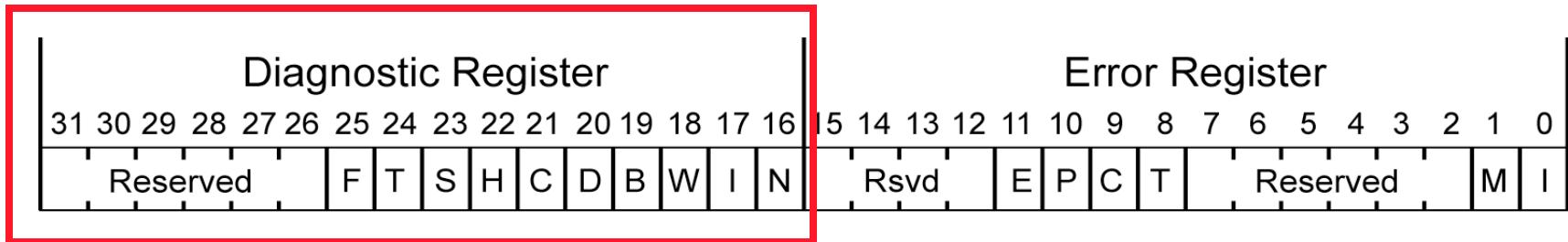
Other values reserved

SError Register



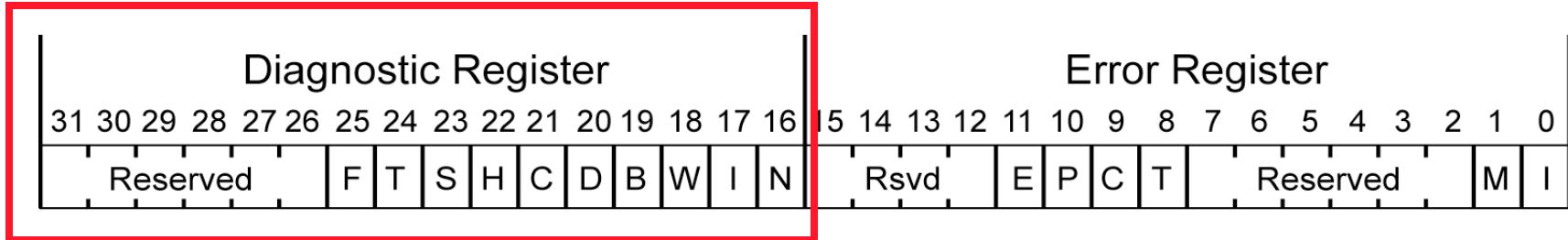
- C **Non-recovered persistent communication or data integrity error.** Persistent, so no retry attempted. Causes include bad interconnect with the device, device failure or removal, etc.
- E **Internal error.** Host bus adapter experienced an internal error that caused the operation to fail. Host software should reset the HBA before retrying the transaction.
- I **Integrity error, data recovered.** Retry solved the data integrity error. Causes include noise burst during transmission, voltage variation, etc. Host software not involved in correcting the problem, but may elect to track such failures.
- M **Recovered communications error.** Communications between device and host lost temporarily (PhyNRdy change), but was recovered. Causes include a device temporarily removed, temporary loss of synchronization, other sources that cause PhyNRdy to be signaled to the Link layer. Host software not involved in correcting the problem, but may elect to track such failures.
- P **Protocol error.** Serial ATA protocol violated. Causes include invalid or malformed FIS received, invalid state transitions, etc. Host software should reset the interface and retry the operation.
- R **Reserved** for future use and must be cleared (zero).
- T **Transient data integrity error, Non-recovered.** Because failure is suspected to be temporary host software should retry the operation.

Diagnostic Register



- B 10b to 8b Decode error: When set to a one, this bit indicates that one or more 10b to 8b decoding errors occurred since the bit was last cleared.
- C CRC Error: When set to one, this bit indicates that one or more CRC errors occurred with the Link Layer since the bit was last cleared.
- D Disparity Error: When set to one, this bit indicates that incorrect disparity was detected one or more times since the last time the bit was cleared.
- F Unrecognized FIS type: When set to one, this bit indicates that since the bit was last cleared one or more FIS's were received by the Transport layer with good CRC, but had a type field that was not recognized.
- I Phy Internal Error: When set to one, this bit indicates that the Phy detected some internal error since the last time this bit was cleared.
- N PhyRdy change: When set to one, this bit indicates that the PhyRdy signal changed state since the last time this bit was cleared.

Diagnostic Register, continued



- H Handshake error: When set to one, this bit indicates that one or more R_ERR handshake response was received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 10b/8b decoding error, or other error condition leading to a negative handshake on a transmitted frame.
- R Reserved bit for future use: Shall be cleared to zero.
- S Link Sequence Error: When set to one, this bit indicates that one or more Link state machine error conditions was encountered since the last time this bit was cleared. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition.
- T Transport state transition error: When set to one, this bit indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared.
- W Comm Wake: When set to one this bit indicates that a Comm Wake signal was detected by the Phy since the last time this bit was cleared.

Error Detection & Recovery

Goals:

- Maintain compatibility with legacy software.
Legacy software only has knowledge of parallel ATA, so some SATA error detection and reporting capabilities may be of little use without new software.
- Keep SATA's error detection and reporting capabilities modest to ensure low cost drives.

SATA Error Checks

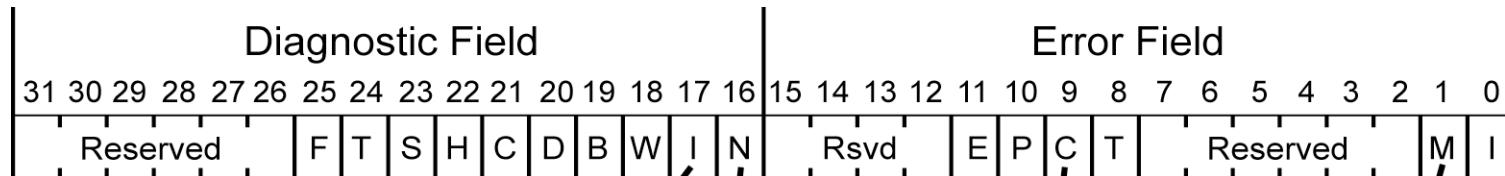
- Each FIS transferred across the link is CRC checked
- Disparity Error Checks
- Time Outs
 - Failure of host to deliver a valid command
 - Internal HBA errors
 - SATA transmission errors
 - Internal drive errors

Phy-Layer Errors/Actions

Physical Layer Errors

- Physical Layer errors typically result in communication failure
- No device present - reported via SStatus reg.
- OOB signaling sequence errors - reported via SStatus register
- Phy internal errors - reported via SStatus & SError register
 - Elasticity buffer overflow or underflow
 - PLL tracking errors
 - Clock variance errors
 - Electrical error conditions

Phy-Layer Errors/Actions (Reported via SError)



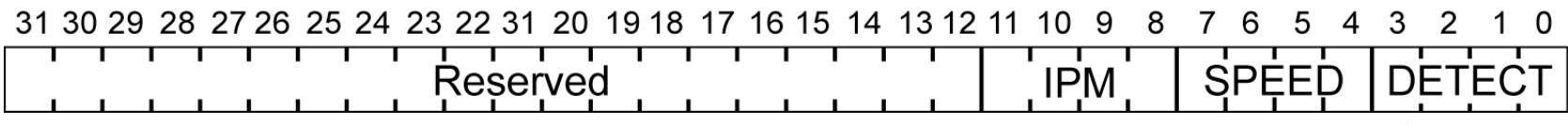
PHY Internal Error: All sources of PHY-specific errors are reflected in this bit.

PhyRdy Change: The PhyRdy indicator has changed state since last cleared.

Comm Errors, Unrecoverable: The error may have occurred due to an internal PHY error.

Recovered Comm Errors: The error has been recovered but may have been caused by an internal PHY error.

Phy-Layer Errors/Actions (Errors reported via SStatus)



- 0000 = Device not detected/no PHY communication
- 0001 = Device detected/no PHY communication
- 0011 = Device detected/PHY communicating
- 0100 = PHY offline

Link-Layer Errors/Actions

Link Layer Errors are of two categories:

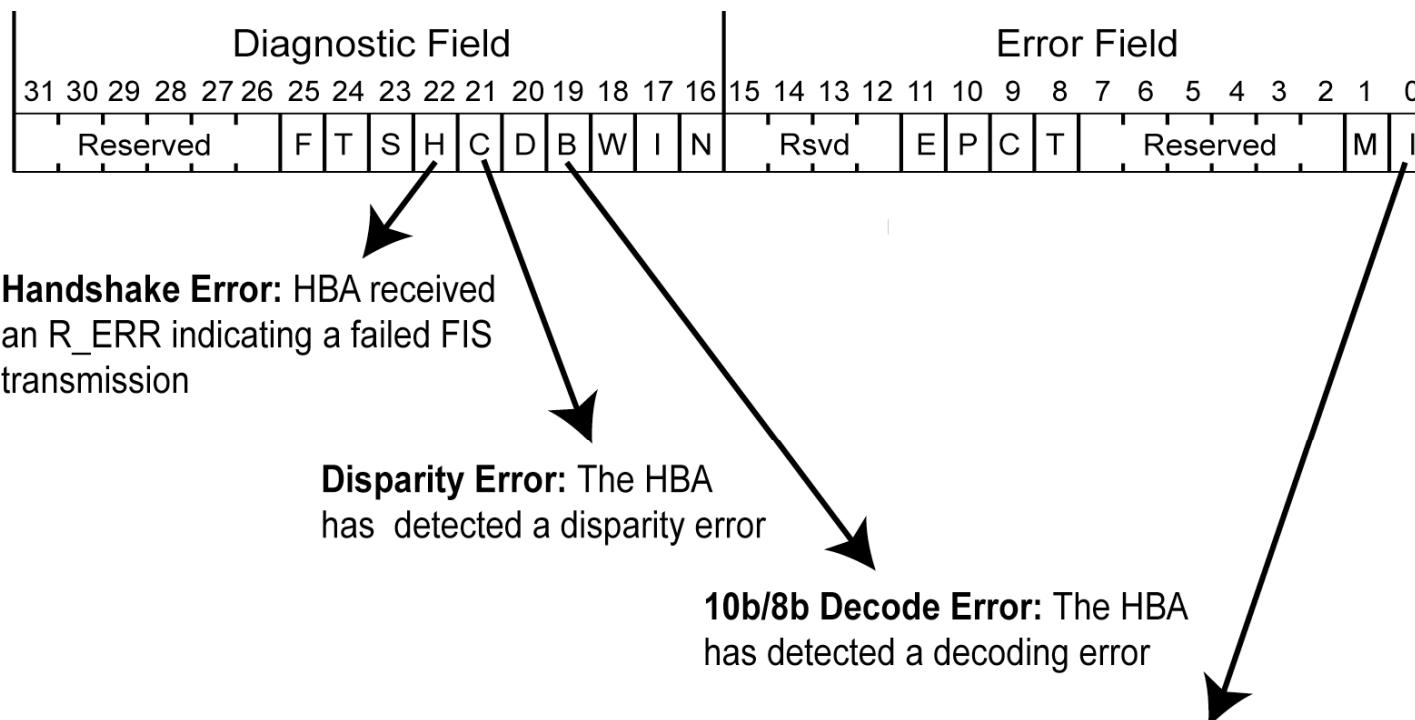
- Invalid State Transitions - when detected, subsequent behavior is designed to permit recovery back to the correct state
- Data Integrity Errors - these errors typically result from noise associated with the physical interconnect and include:
 - Disparity Errors
 - 10b/8b decoding errors
 - CRC errors

Link-Layer Errors/Actions

- Link Layer Errors are reported in several possible ways:
 - Errors may be accumulated and reported to the sending node during the FIS transaction handshake (via R_ERR). This action occurs when the error occurs during FIS transmission.
 - Bits within the Interface Error register may be updated
 - Errors may be reported to the Transport layer

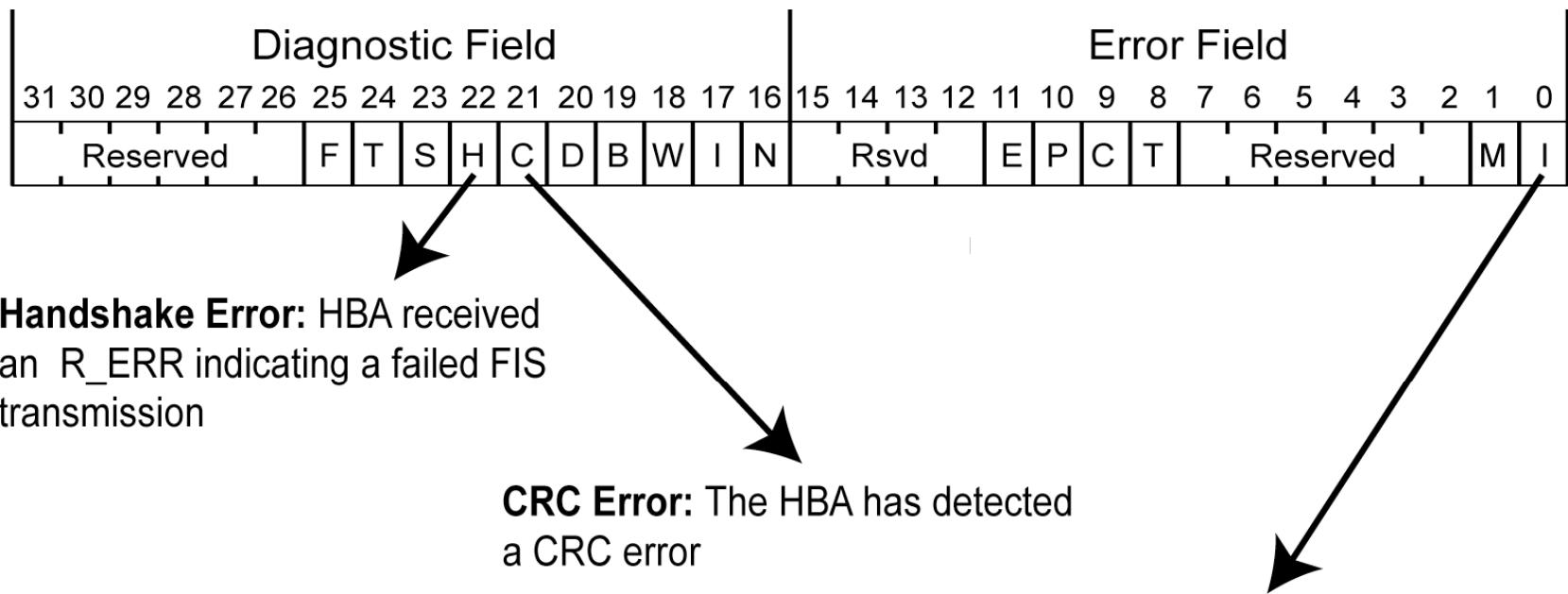
Link-Layer Errors/Actions (Disparity Error Reporting)

Disparity errors can result in several errors being detected.



Link-Layer Errors/Actions (CRC Error Reporting)

CRC errors can result in several errors being detected.



Handshake Error: HBA received an R_ERR indicating a failed FIS transmission

CRC Error: The HBA has detected a CRC error

Recovered Data Integrity Error: FIS Retry resulted in recovery from an error such as CRC or disparity.

Link-Layer Errors/Actions

(Invalid State Transitions)

Example cases:

- Following FIS delivery, if the transmitter receives a primitive other than SYNC, R_OK, or R_ERR the link layer must wait on a valid terminating primitive.
- Following reception of consecutive X_RDY primitives, if the next control character received is not SOF, the Link notifies the Transport Layer of the condition & transitions to the idle state.
- After transmission of X_RDY, if R_RDY is not received, no Link recovery action shall be attempted. Thus, the higher layers time out causing a reset.

Link-Layer Errors/Actions

(Invalid State Transitions)

Example cases:

- When SOF is sent before the receiver has signaled R_RDY, the receiving interface does not respond (stays in the idle state). The transmitting interface will eventually timeout and also return to the idle state.
- If the transmitter ends a frame with EOF and WTRM, but fails to receive R_OK or R_ERR within the predetermined timeout, no Link recovery is attempted and the higher layers will time out, causing interface reset.

Transport-Layer Errors/Actions

Errors reported by the link layer include:

- Errors detected in the lower layers of the SATA interface (link and PHY)
- Frame errors
- State Transition errors
- Internal Transport layer errors

Transport-Layer errors are typically handled via FIS retries.

Part Three

Command & Control

Protocols



Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

Command Protocols

(Device Command Protocols)

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

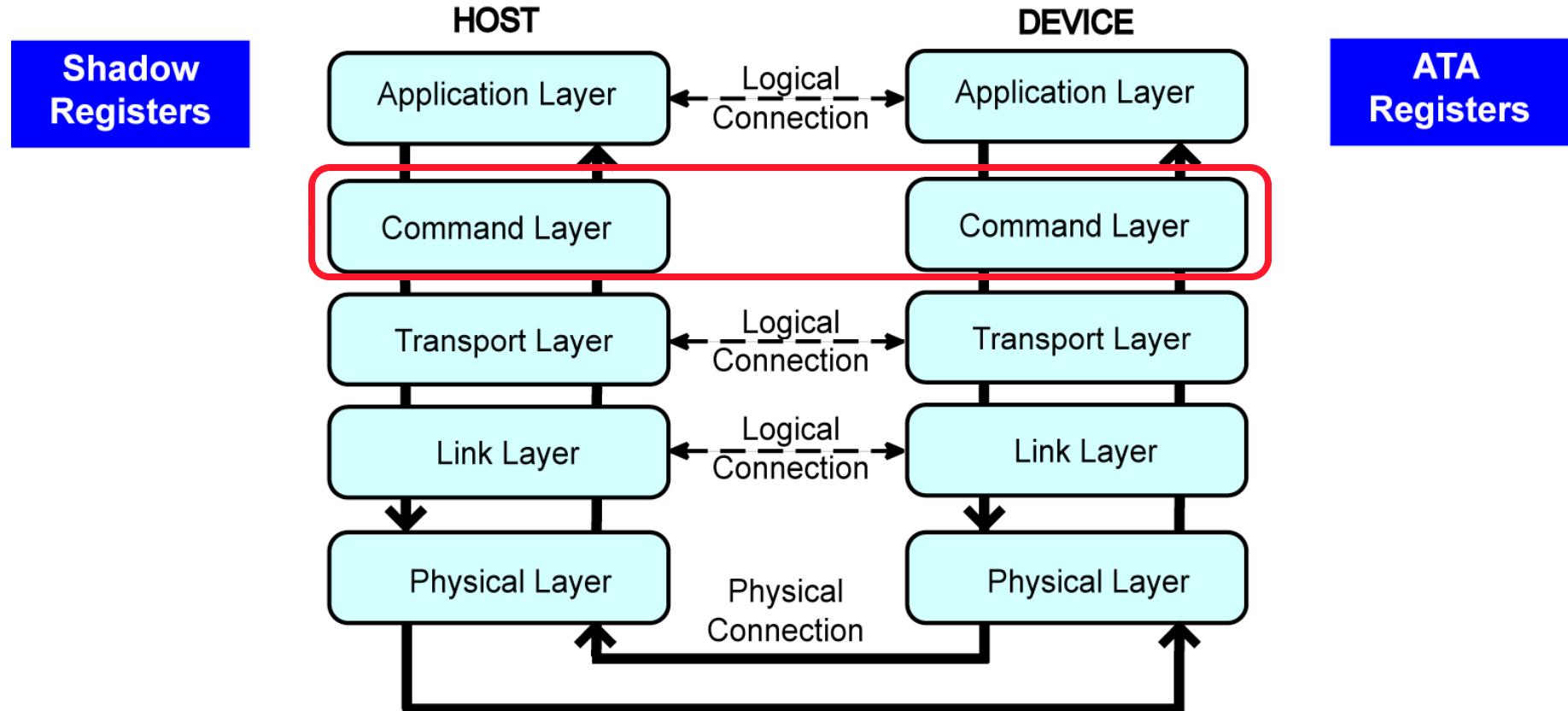


SATA Commands

Commands are issued by ATA host software by writing an 8-bit value to the Command Register.

Offset	Reads	Writes
Cmd+0		Data
Cmd+1	Error	Feature
Cmd+2		Sector Count
Cmd+3		LBA Low
Cmd+4		LBA Mid
Cmd+5		LBA High
Cmd+6		Device
Cmd+7	Status	Command
Ctrl+2	Alternate Status	Device Control

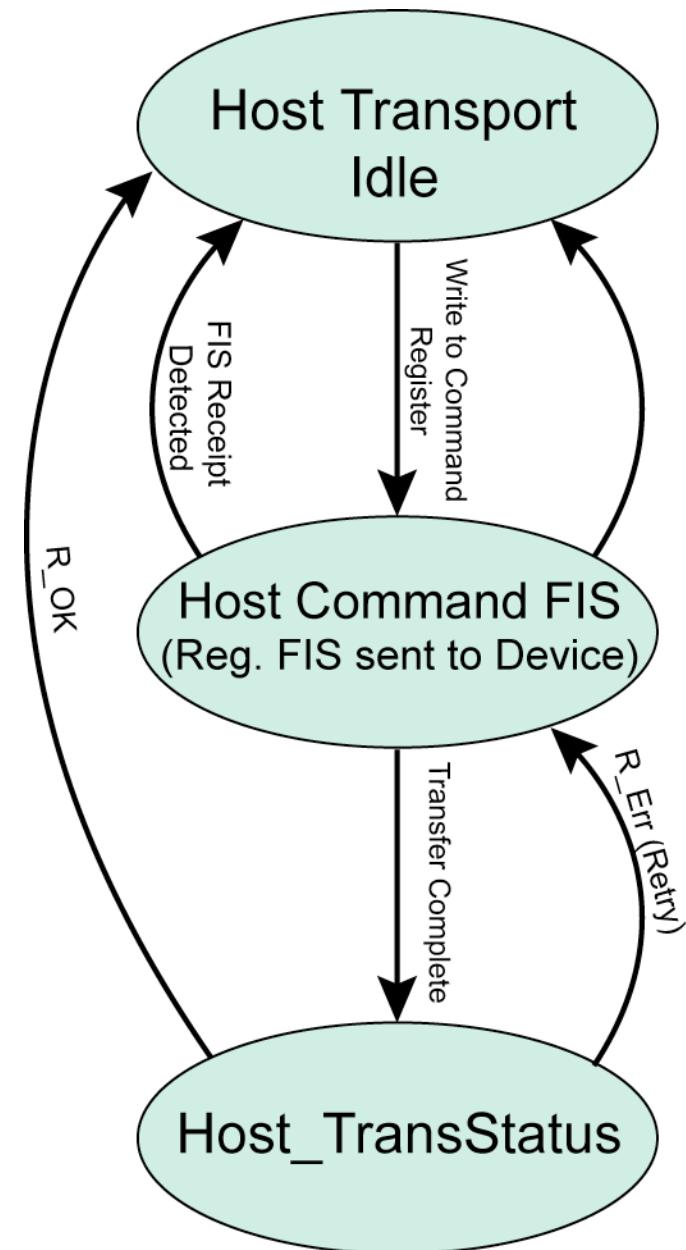
Command Layer



Command Delivery

HBA Sends Command via Register FIS to Drive

- Reg FIS transmission error results in Transport Layer retry
- Other results cause return to idle:
 - Successful transfer
 - illegal transitions
 - detection of FIS sent from device
 - Transmission error with Reset asserted



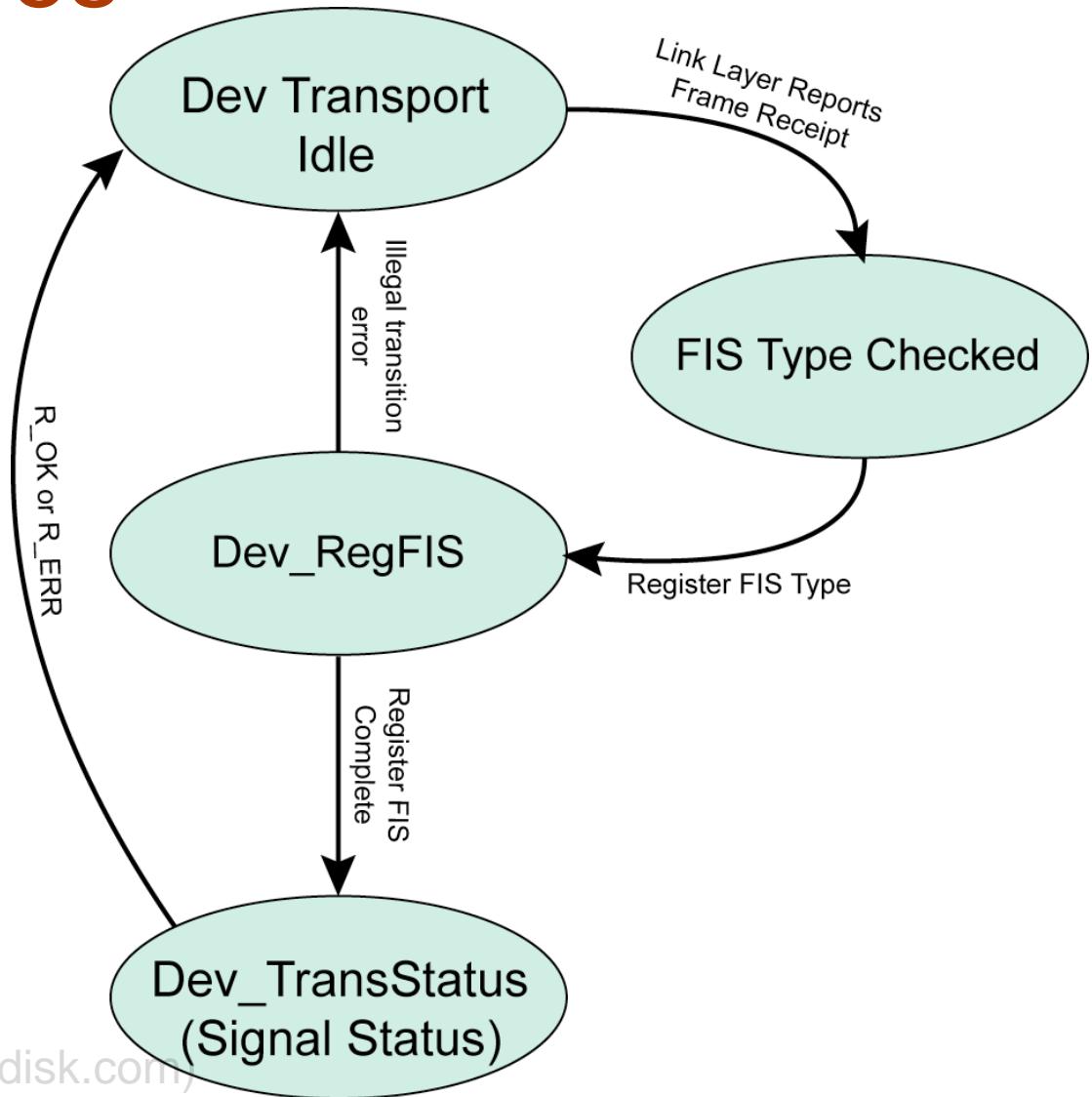
Register FIS Host-to-Device

“C” bit set to 1 indicates FIS was caused by a write to the command register (instead of the control register)

	+3	+2	+1	+0
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
DW 0	Features	Command	C R R Reserved	FIS Type (27h)
DW 1	Dev/Head	Cyl High	Cyl Low	Sector Number
DW 2	Features (exp)	Cyl High (exp)	Cyl Low (exp)	Sec Num (exp)
DW 3	Control	Reserved (0)	Sec Count (exp)	Sector Count
DW 4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

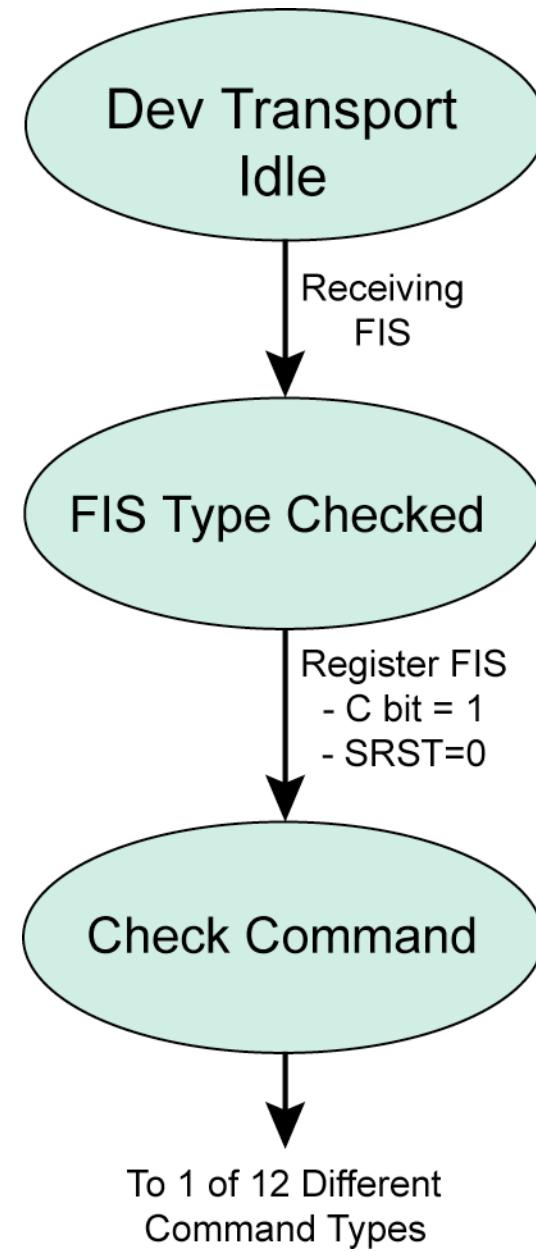
Command Delivery

Drive Receives RegFIS



Command Types

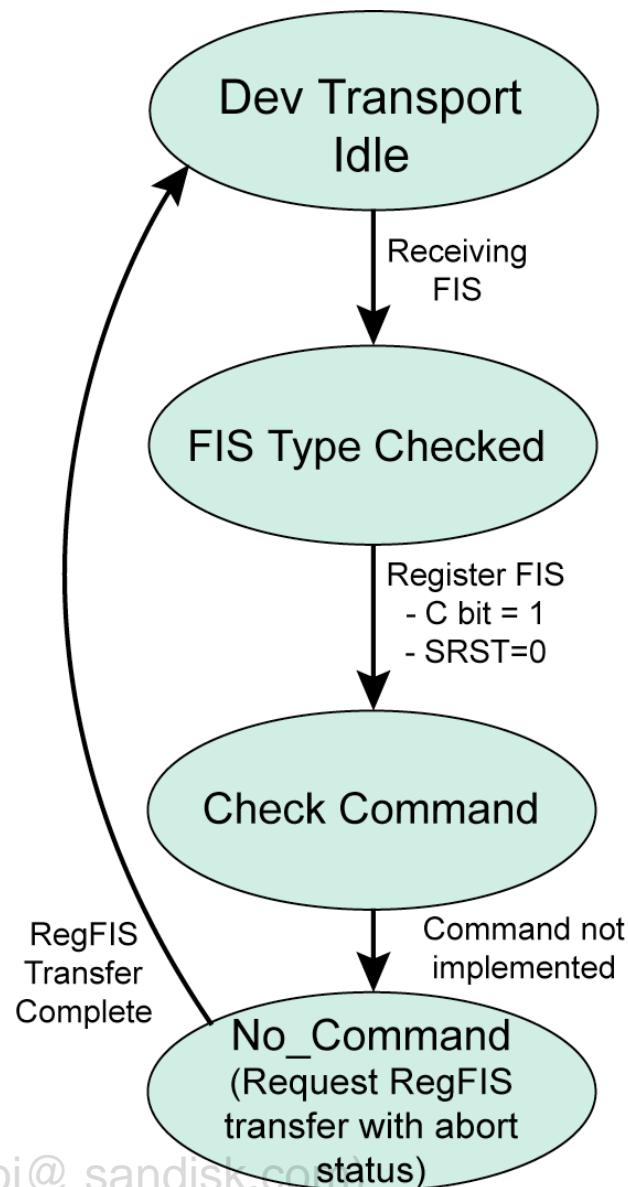
- Command protocols are based upon 12 different command protocol types.
- When a device receives a Register FIS it checks the command type and selects one of the 12 protocols that supports the specific command.



Command Protocol Types

Command Category	Number of Commands
Command Not Implemented	NA
Non -Data	34
PIO Data -In	13
PIO Data -Out	14
DMA -In	2
DMA -Out	2
DMA -In Queued	2
DMA -Out Queued	2
Packet (ATAPI)	1
Service	1
Device Reset	1
Execute Device Diagnostics	1

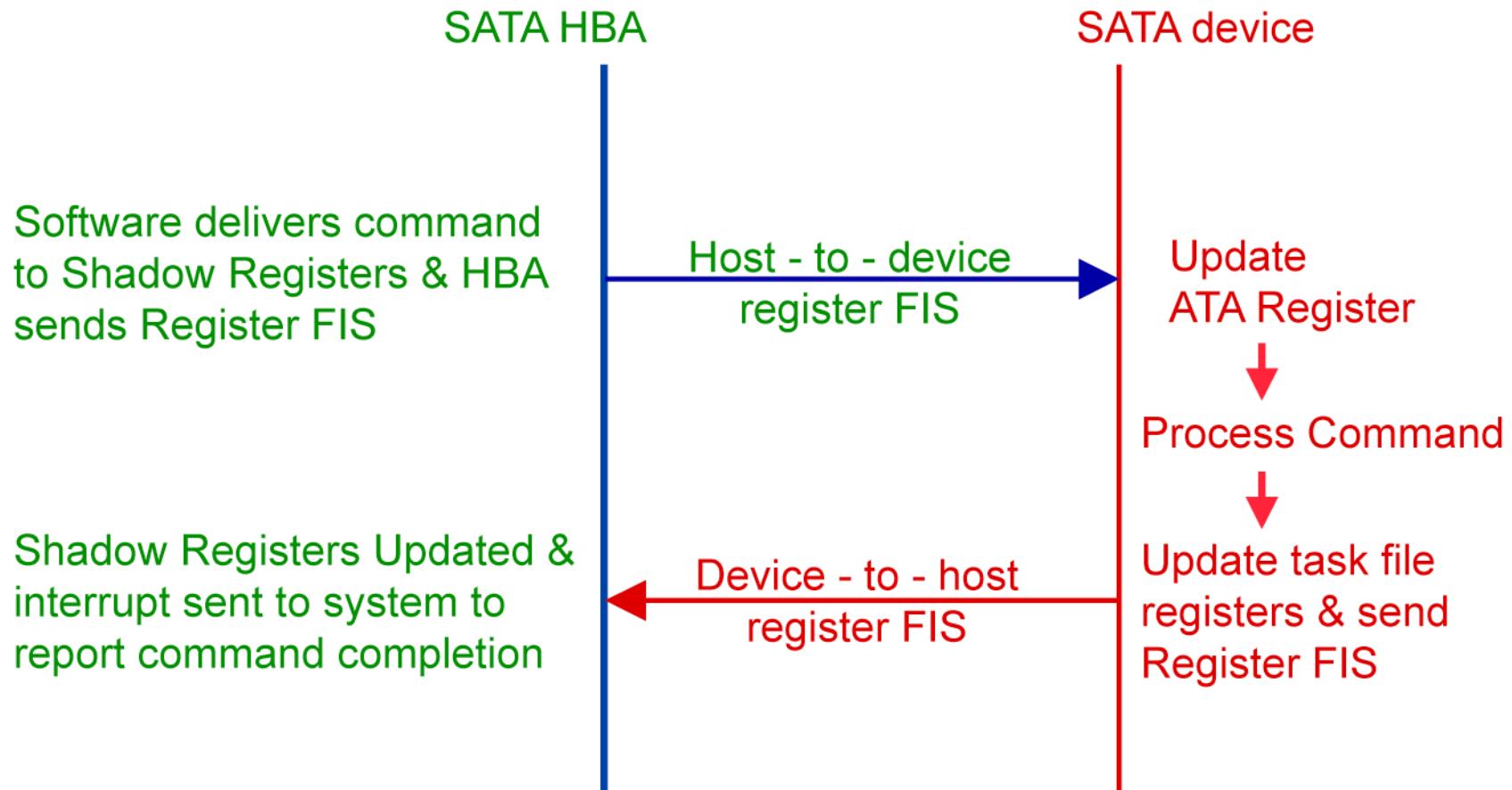
Command Not Implemented



Non-Data Commands

A large category of commands do not involve the movement of data to or from the drive, thus, no DATA FIS is used during in the command protocol.

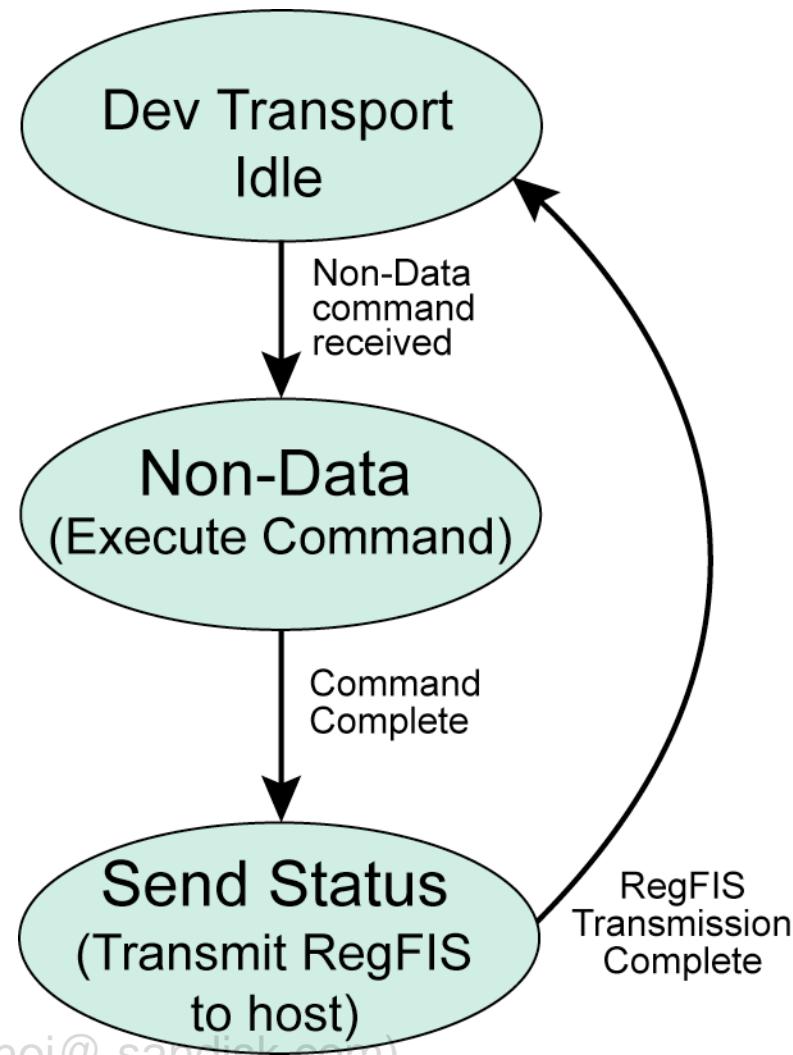
Example Non-Data Command



No Data - Commands

Command	Protocol	Code	No Packet Cmd	Packet Cmd
CFA Erase Sectors	ND	C0h	F	N
CFA Request Extended Error	ND	03h	O	N
Check Media Card Type	ND	D1h	O	N
Check Power Mode	ND	E5h	M	M
Configure Stream	ND	51h	O	O
Device Configuration Freeze Lock	ND	B1h	O	O
Device Configuration Restore	ND	B1h	O	O
Flush Cache	ND	E7h	M	O
Flush Cache Ext	ND	EAh	O	N
Get Media Status	ND	DAh	O	O
Idle	ND	E3h	M	O
Idle Immediate	ND	E1h	M	M
Media Eject	ND	EDh	O	N
Media Lock	ND	DEh	O	N
Media Unlock	ND	DFh	O	N
NOP	ND	00h	O	M
Read Native Max Address	ND	F8h	O	O
Read Native Max Address Ext	ND	27h	O	N
Read Verify Sector(s)	ND	40h	M	N
Read Verify Sector(s) Ext	ND	42h	O	N
Security Erase Prepare	ND	F3h	O	O
Security Freeze Lock	ND	F5h	O	O
Set Features	ND	EFh	M	M
Set Max	ND	F9h	O	O
Set Max Address Ext	ND	37h	O	N
Set Multiple Mode	ND	C6h	M	N
Sleep	ND	E6h	M	M
Smart Disable Operations	ND	B0h	O	N
Smart Enable/Disable Autosave	ND	B0h	O	N
Smart Enable Operations	ND	B0h	O	N
Smart Execute Off-Line Immediate	ND	B0h	O	N
Smart Return Status	ND	B0h	O	N
Standby	ND	E2h	M	O
Standby Immediate	ND	E0h	M	M

Non-Data Protocol



PIO Commands

The PIO command protocol has two primary variations based on direction of data movement:

- PIO Data-In — data movement from drive to memory
- PIO Data-Out — data movement from memory to the drive

PIO IN Commands

- Host initiates PIO IN by setting up the transfer and issuing command
- Host Transport Layer passes the command to the device via a Register FIS
- Device decodes command & begins access to disk to acquire requested data
- Device returns a PIO Setup FIS indicating that it's ready to return the DATA FIS
- Device sends Data FIS to host

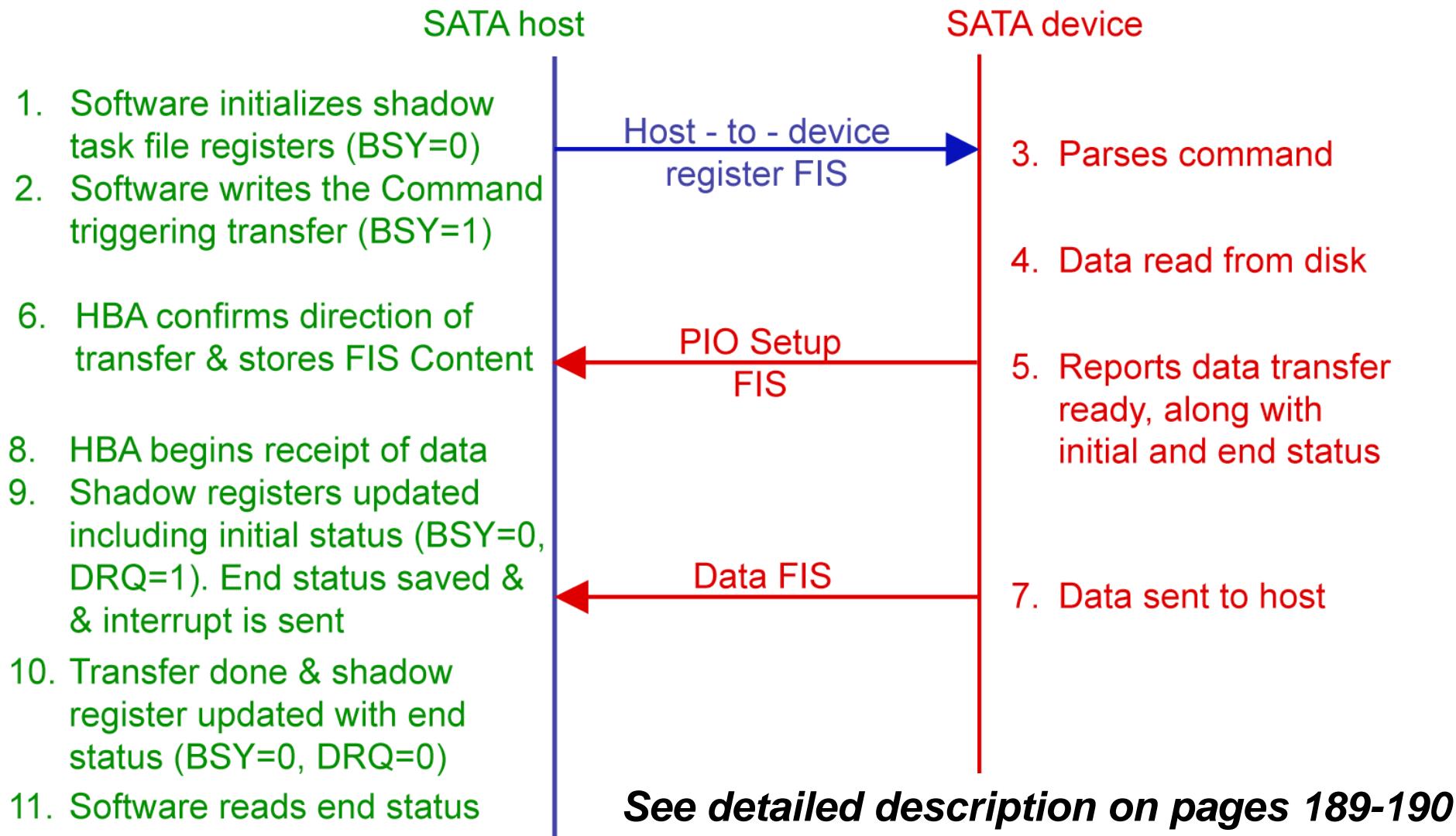
PIO Data-IN Commands

Command	Code	ATA Command	ATAPI Command
CFA Translate Sector	87h	O	N
Device Configuration Identify	B1h	O	O
Identify Device	ECh	M	M
Identify Packet Device	A1h	N	M
Read Buffer	E4h	O	N
Read Log Ext	2Fh	O	O
Read Multiple	C4h	M	N
Read Multiple Ext	29h	O	N
Read Sector(s)	EDh	M	M
Read Sector(s) Ext	DEh	O	N
Read Stream Ext	2Bh	O	N
Smart Read Data	B0h	O	N
Smart Read Log	B0h	O	N
M = Mandatory N = Not Allowed O = Optional			

PATA PIO Data-In Review

1. BSY and DRQ status bits are cleared (0) permitting host software to initialize and issue the command directly to the ATA registers (task file) within the disc drive. When the command is written the drive sets the BSY bit to notify software that the task file should not be updated.
2. The drive decodes the command and reads the entire first target sector from disc.
3. The first two bytes of data read from disc are transferred to the task file's Data register, making data available for host software to fetch.
4. The drive clears the BSY bit, sets the DRQ bit, and generates an interrupt to notify software that status has changed.
5. Host software reads status and detects BSY cleared and DRQ set, indicating that data is ready to be read from the Data register.
6. Host software continuously reads the contents of the Data register until an entire sector (512 bytes) has been read.
7. If the transfer count is satisfied, the drive will clear the DRQ and BSY bits to notify the host that the transfer is complete.
8. If the transfer count is not yet satisfied, the drive sets the BSY bit. Software reads status to determine if the command has completed. When the drive is ready to transfer the next sector, it repeats the process beginning at step 3.

PIO Data-IN Command Protocol



PIO Setup FIS - Dev to Host

	+3	+2	+1	+0
DW 0	Error	Status	R I D Reserved	FIS Type (5Fh)
DW 1	Dev/Head	Cyl High	Cyl Low	Sector Number
DW 2	Reserved (0)	Cyl High (exp)	Cyl Low (exp)	Sec Num (exp)
DW 3	E_Status	Reserved (0)	Sec Count (exp)	Sector Count
DW 4	Reserved (0)	Transfer Count		

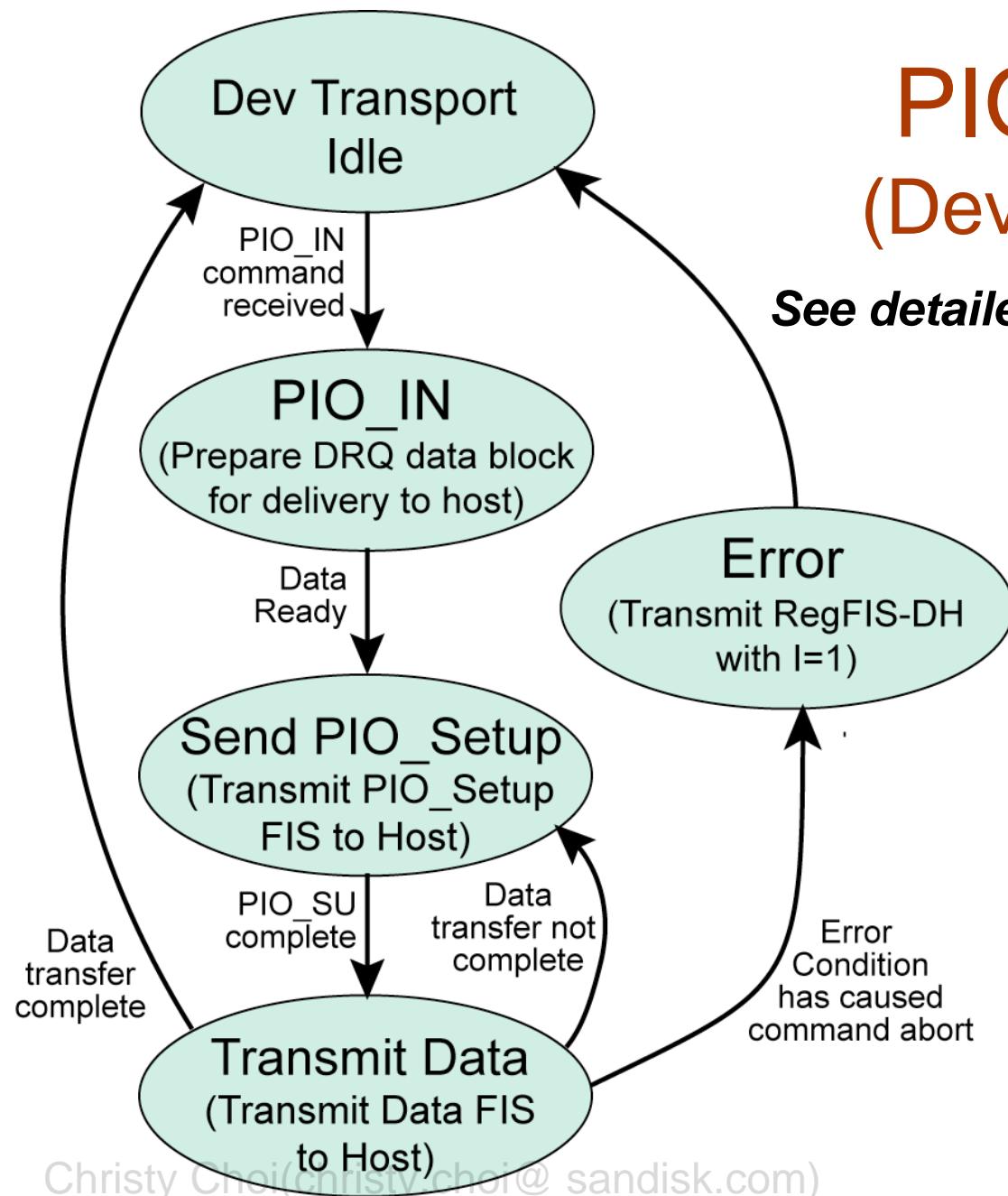
R = Reserved (0)

I = Interrupt Bit

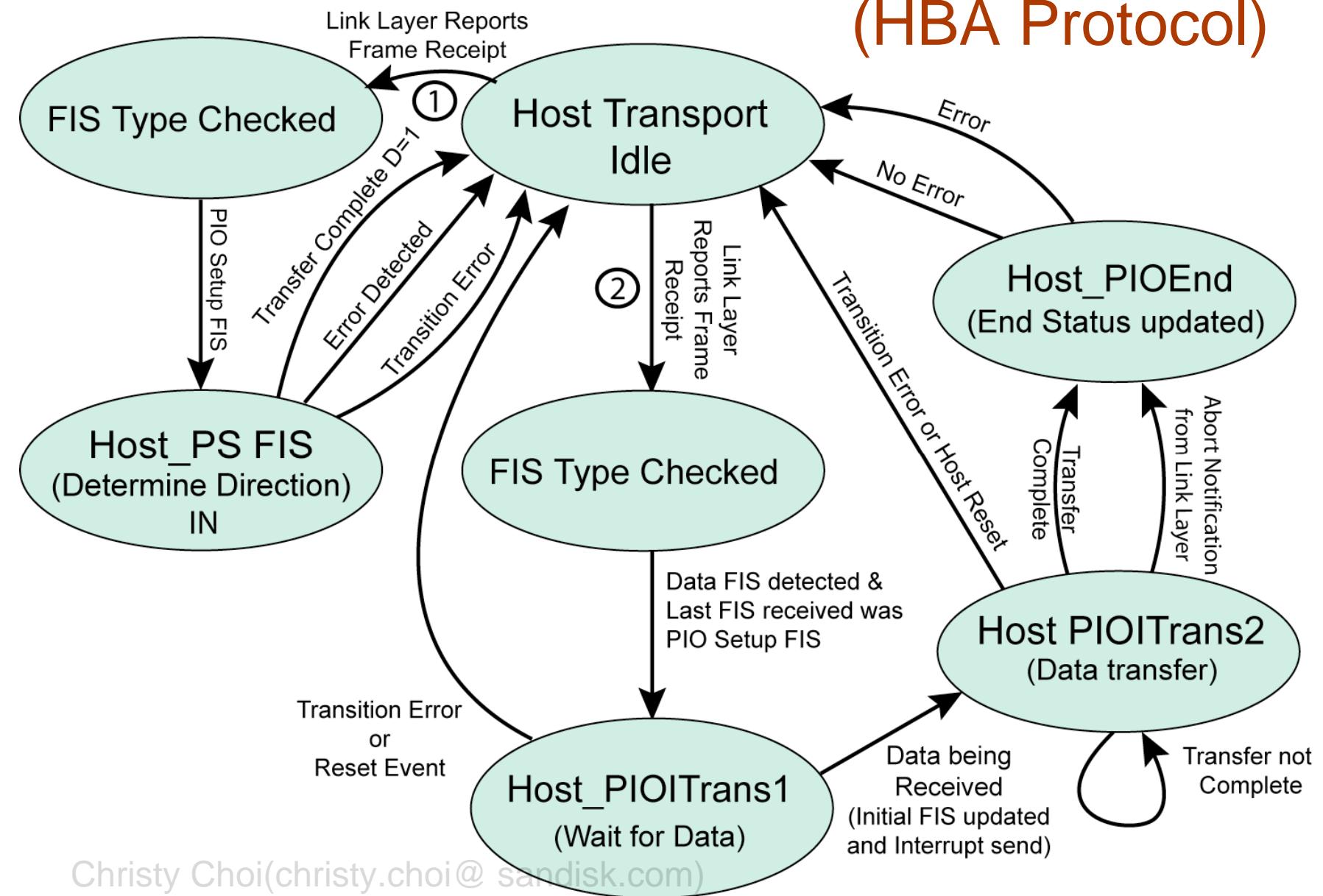
D = Direction of Transfer 0=Read from Host Memory

PIO Data-IN (Device Protocol)

See detailed description on pages 191-192



PIO In Command (HBA Protocol)



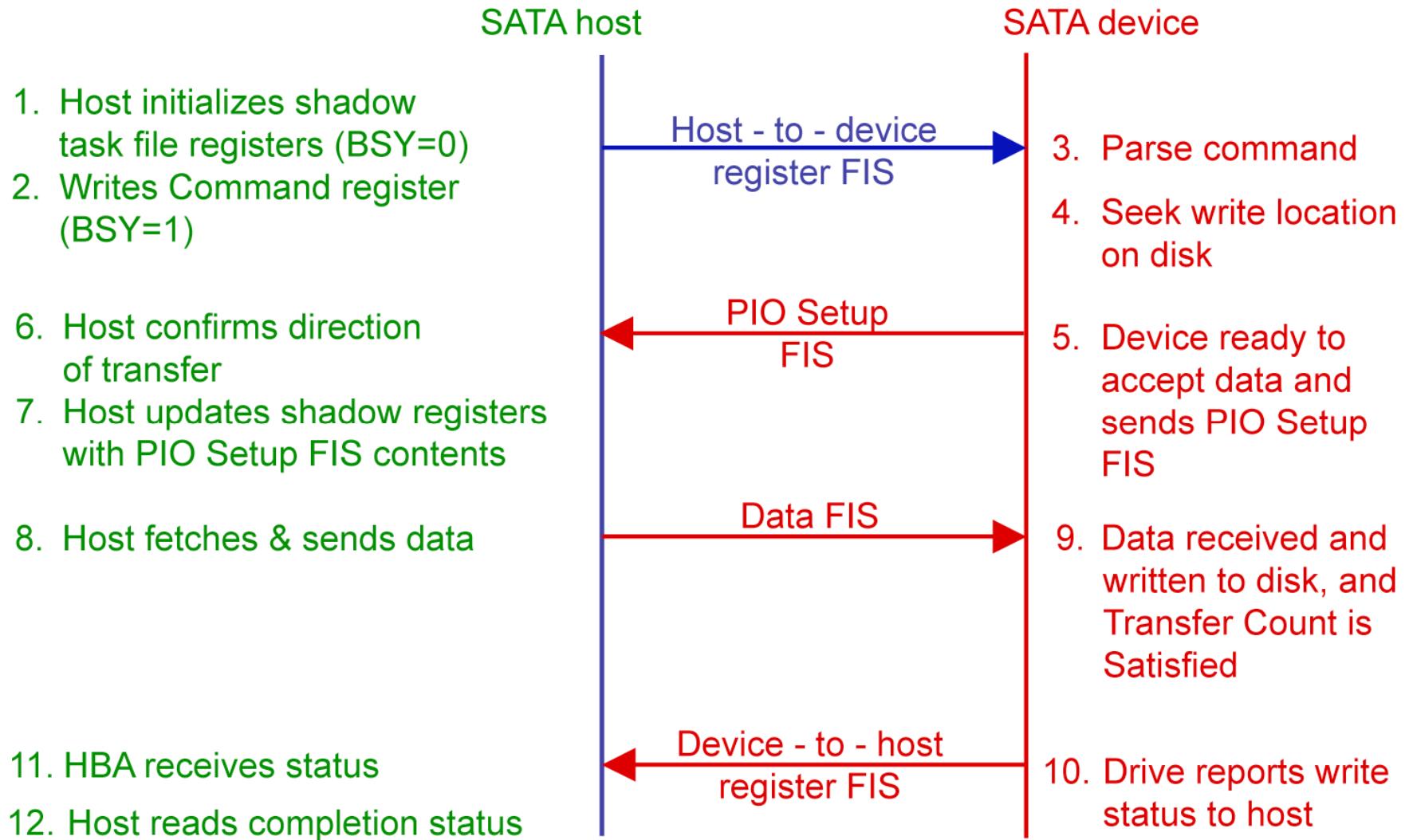
PIO Out Command

- Host initiates PIO Out by setting up the transfer and issuing command
- Host Transport Layer passes the command to the device via a Register FIS
- Device returns a PIO Setup FIS to Host thereby requesting the data transfer
- Host controller waits for host software to write Data register and accumulates the data until the transfer size specified in the PIO Setup FIS is satisfied
- Host sends Data FIS to device
- Device return Register FIS to host

PIO Data-Out Commands

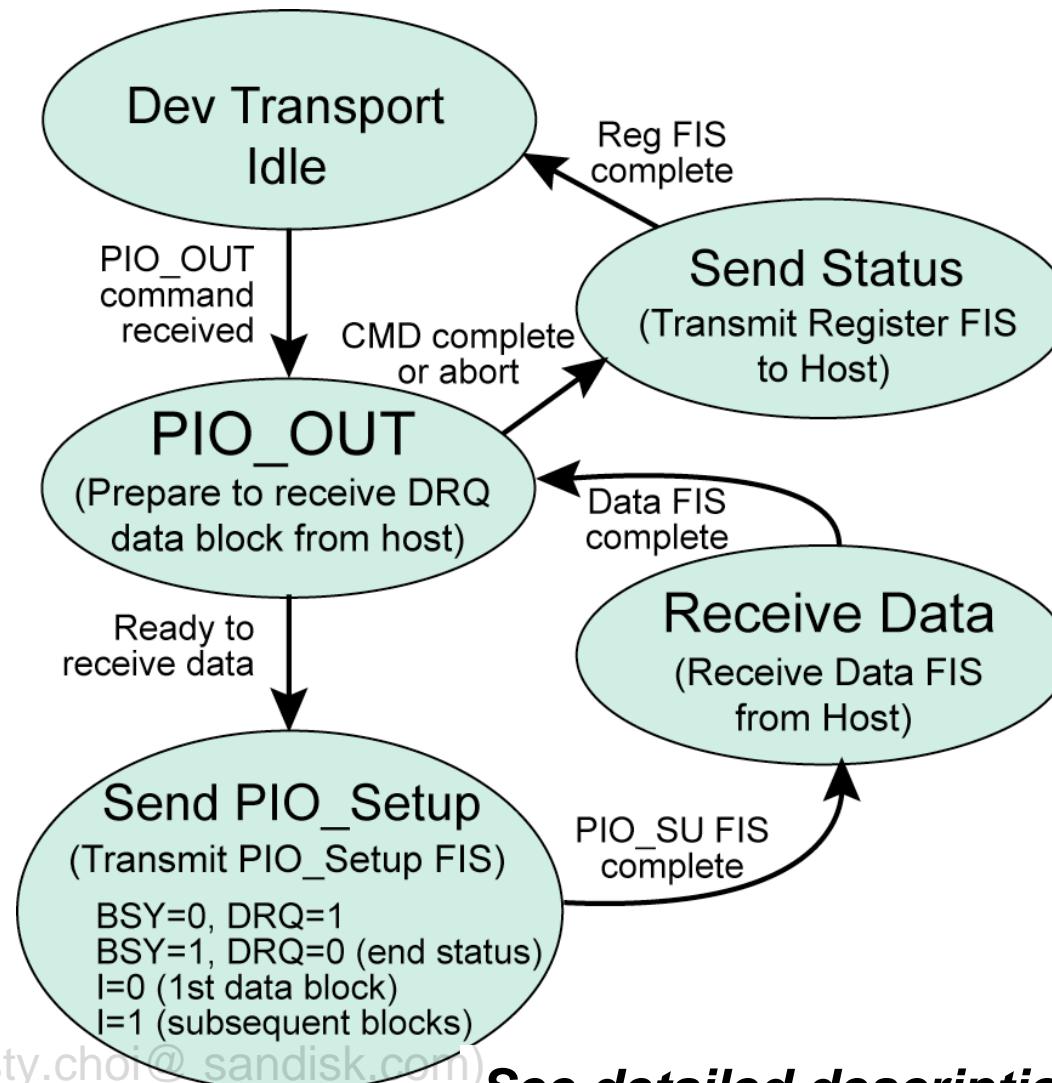
Command	Code	ATA Command	ATAPI Command
CFA Write Multiple Without Erase	CDh	O	N
CFA Write Sectors Without Erase	38h	O	N
Device Configuration Set	B1h	O	O
Download Microcode	92h	O	N
Security Disable Password	F6h	O	O
Security Erase Unit	F4h	O	O
Security Set Password	F1h	O	O
Security Unlock	F2h	O	O
Smart Write Log	B0h	O	N
Write Buffer	E8h	O	N
Write Log Ext	3Fh	O	O
Write Multiple	C5h	M	N
Write Multiple Ext	39h	O	N
Write Multiple FUA Ext	CEh	O	N
Write Sector(s)	30h	M	N
Write Sector(s) Ext	34h	O	N
Write Stream Ext	38h	O	N
M = Mandatory N = Not Allowed O = Optional			

PIO Data-OUT Command Protocol

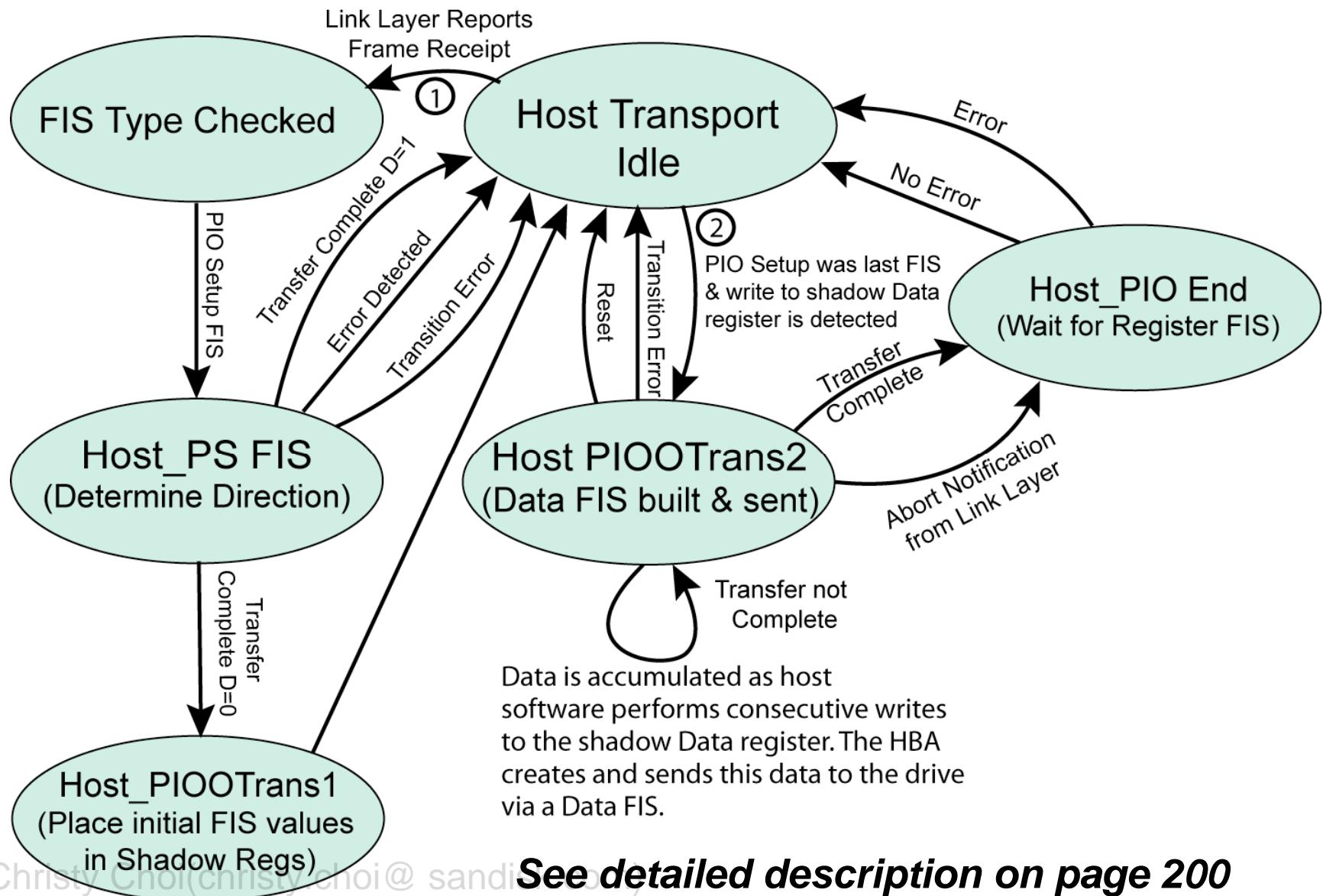


See detailed description on page 197

PIO Data-OUT Protocol (Device Protocol)



HBA PIO Out Protocol



DMA Commands

DMA commands come in several types that each have different command protocols:

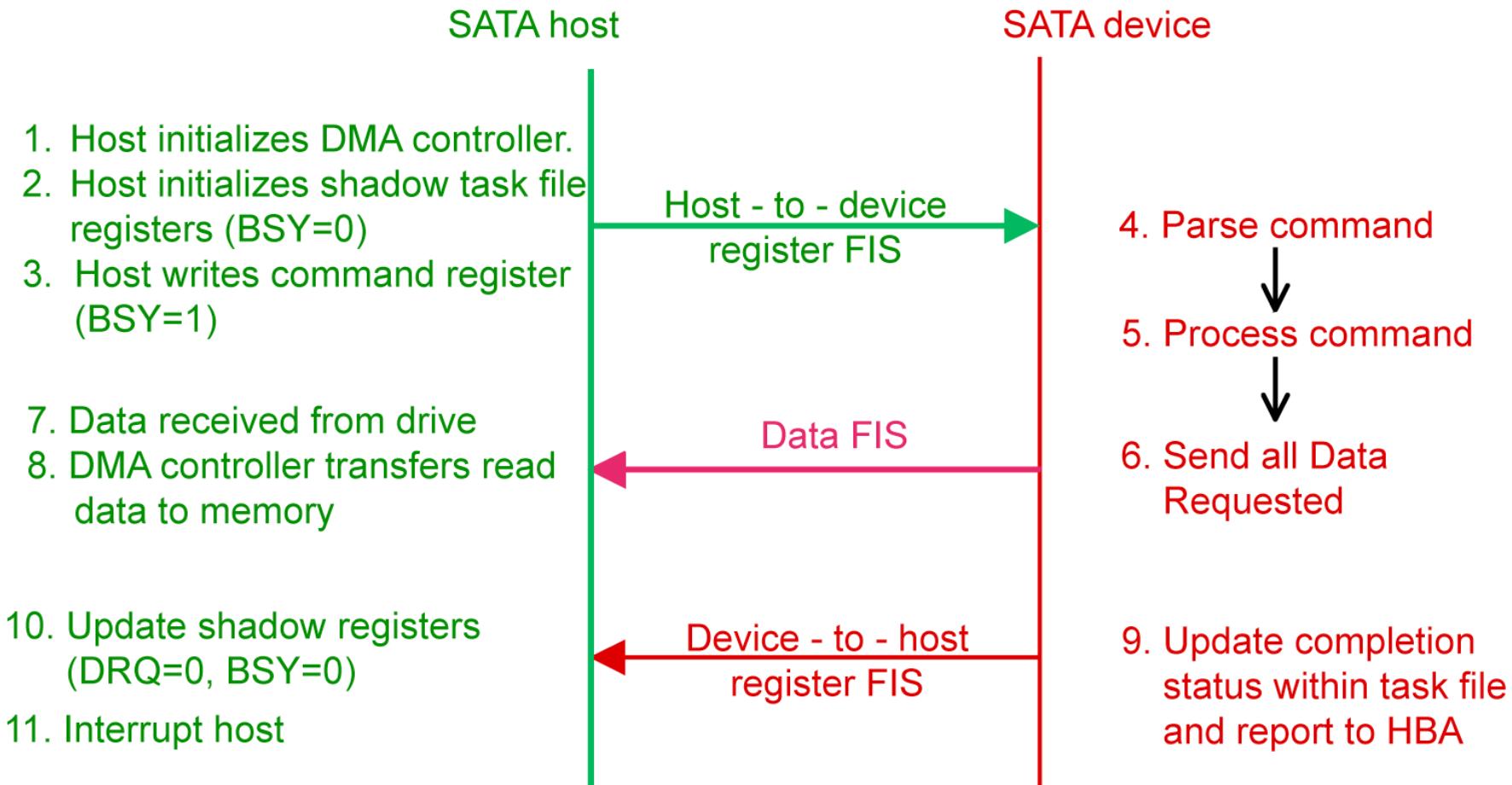
- DMA-In Commands
- DMA-Out Commands
- DMA-In Queued Commands
- DMA-Out Queued Commands
- First Party DMA Read Commands
- First Party DMA Write Commands

DMA-In (DMA Read) Protocol

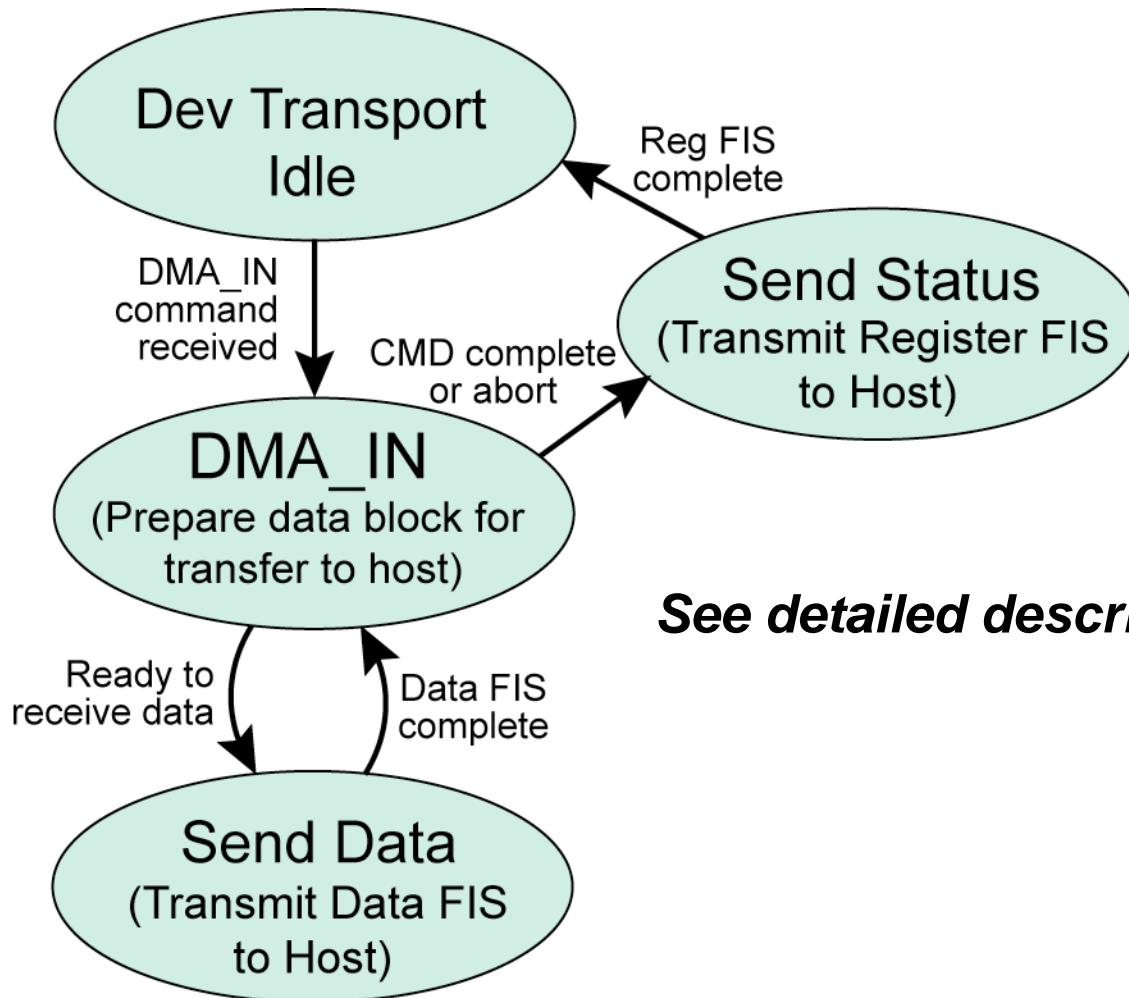
Commands that use the DMA-In Command Protocol

Command	Code	ATA Command	ATAPI Command
Read DMA	C8h	M	N
Read DMA Extended	25h	O	N
Read Stream DMA Ext	2Ah	O	N

DMA Read Command Protocol

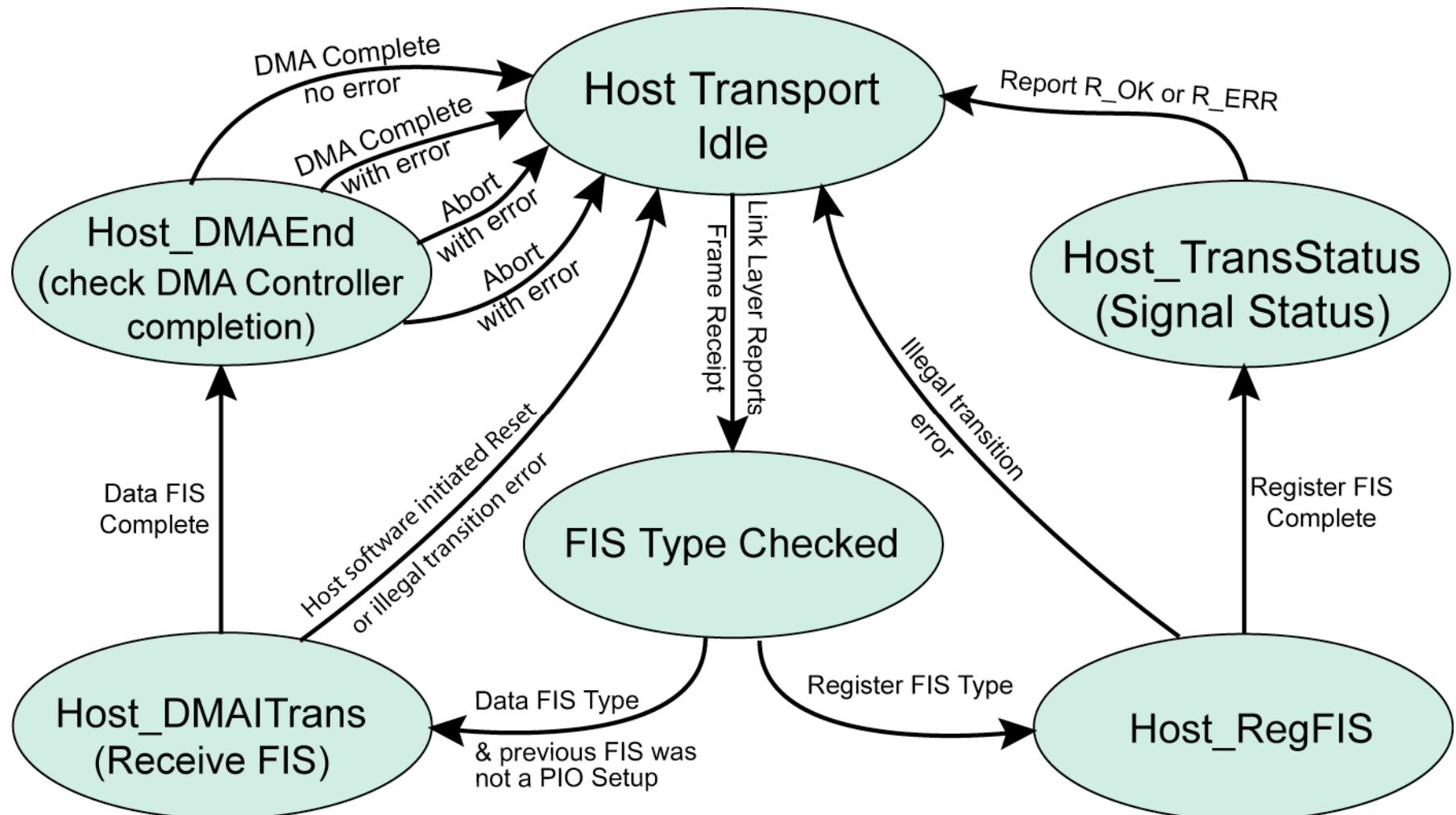


DMA Data-IN (Device Protocol)



See detailed description on page 20

DMA Data-IN (HBA Protocol)



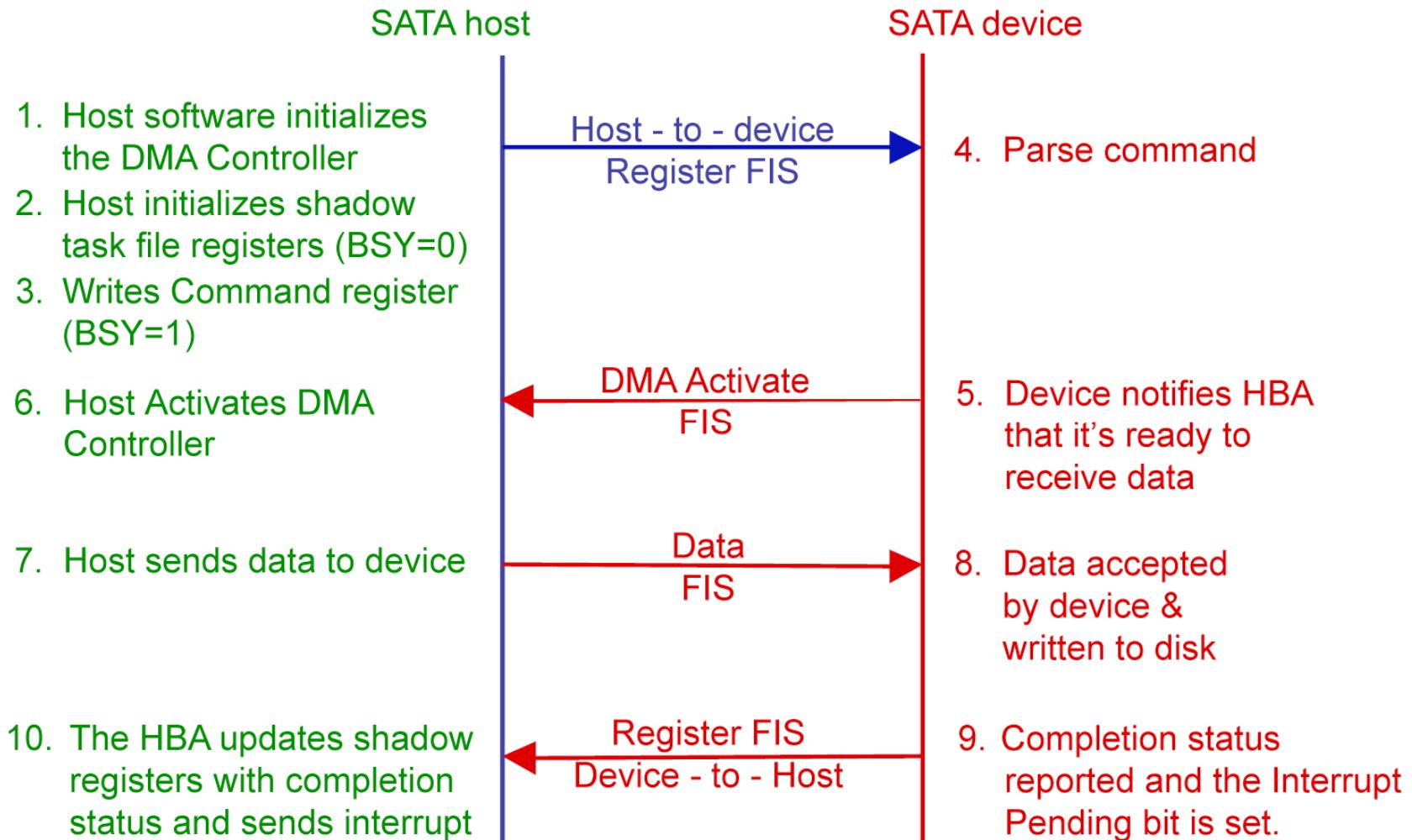
See detailed description on page 204-205

DMA Write Command

Four commands use the Write DMA protocol as listed in Table 11-6 on page 206.

Command	Code	ATA Command	ATAPI Command
Write DMA	CAh	M	N
Write DMA Extended	35h	O	N
Write DMA FUA	3Dh	O	N
Write Stream DMA Ext	3Ah	O	N

DMA Write Command Protocol



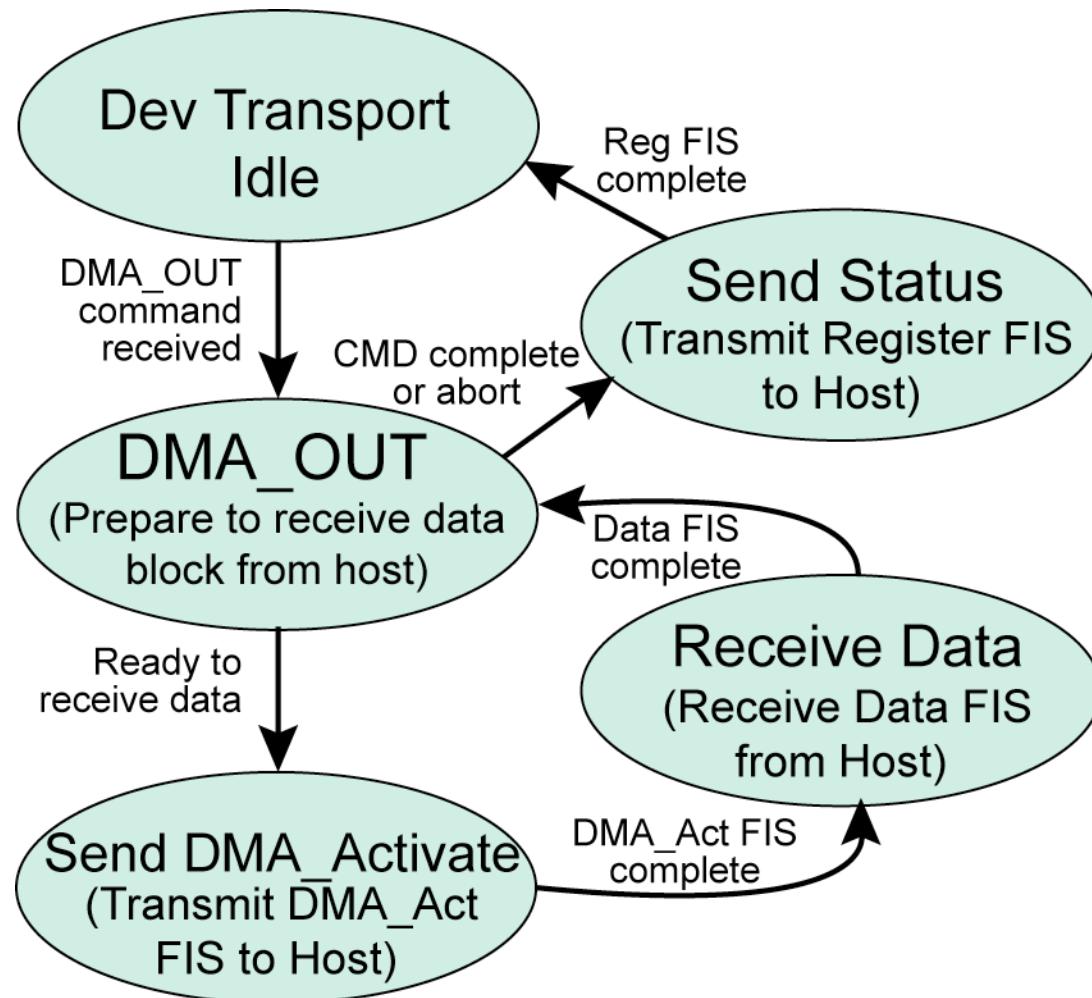
See detailed description on page 206-207

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

Copyright Mindshare Inc, 2009

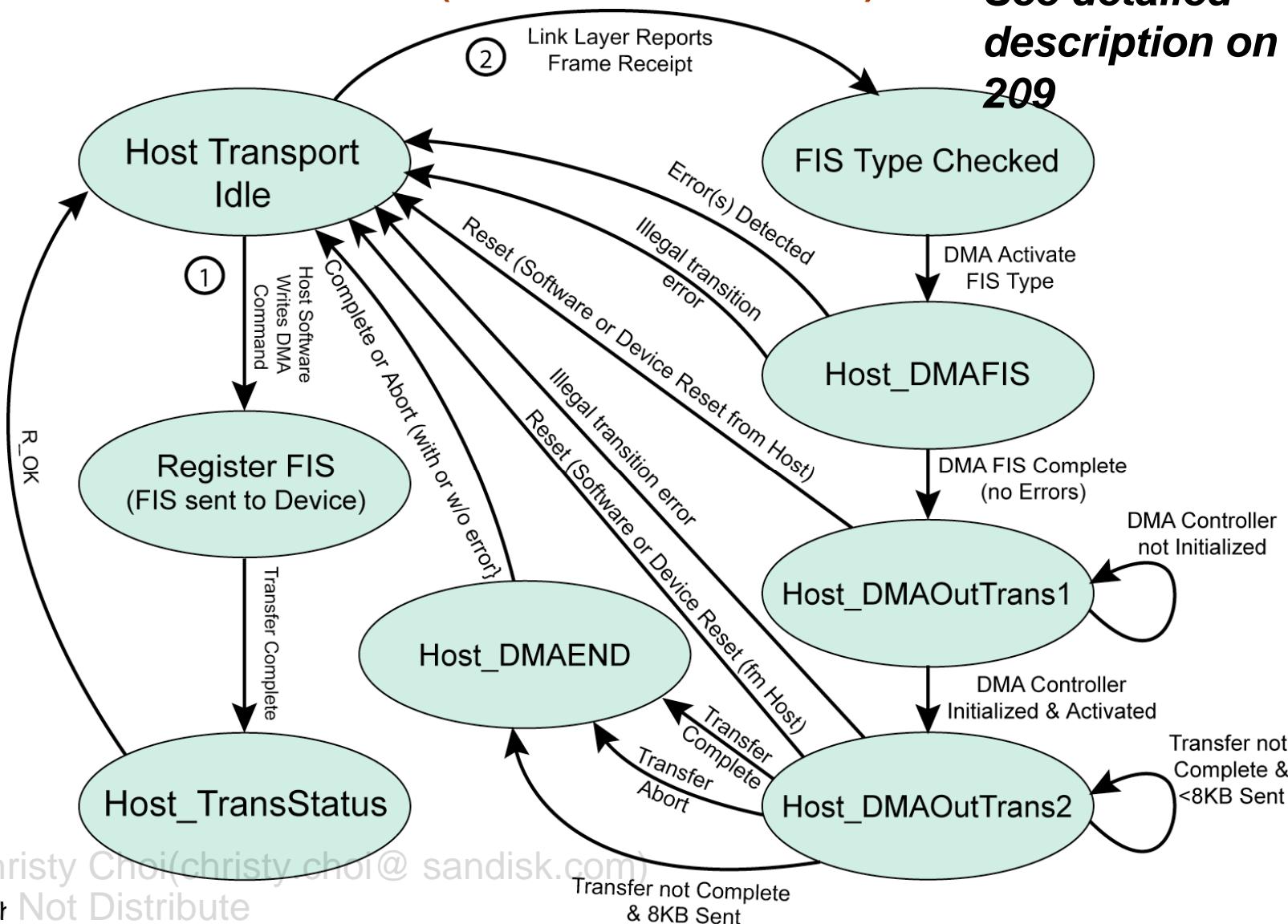
DMA Data-OUT (Device Protocol)



See detailed description on page 208

DMA Data-OUT (HBA Protocol)

See detailed description on page 209

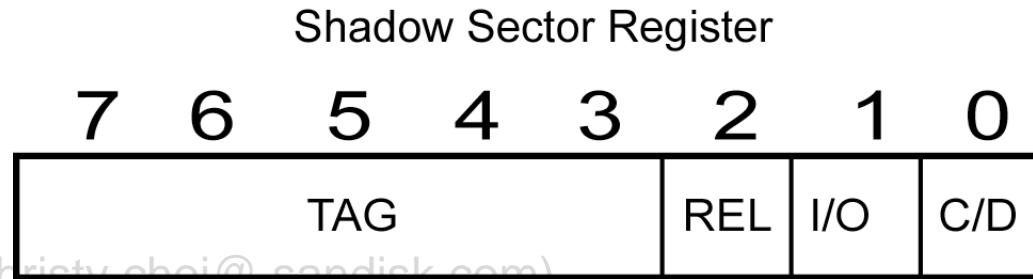
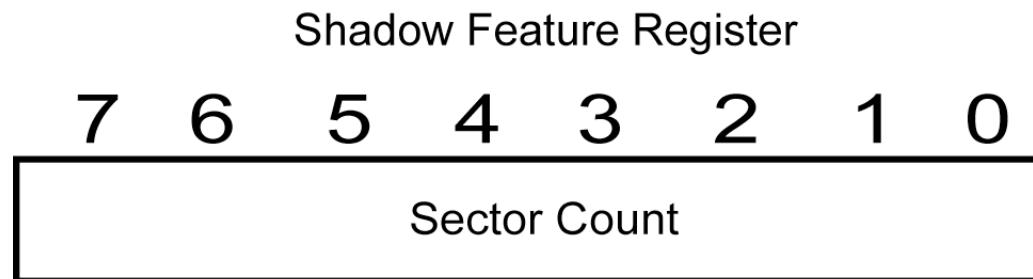


DMA Queued Commands

DMA Queued command never gained wide acceptance, therefore only an overview of the command protocols are provided.

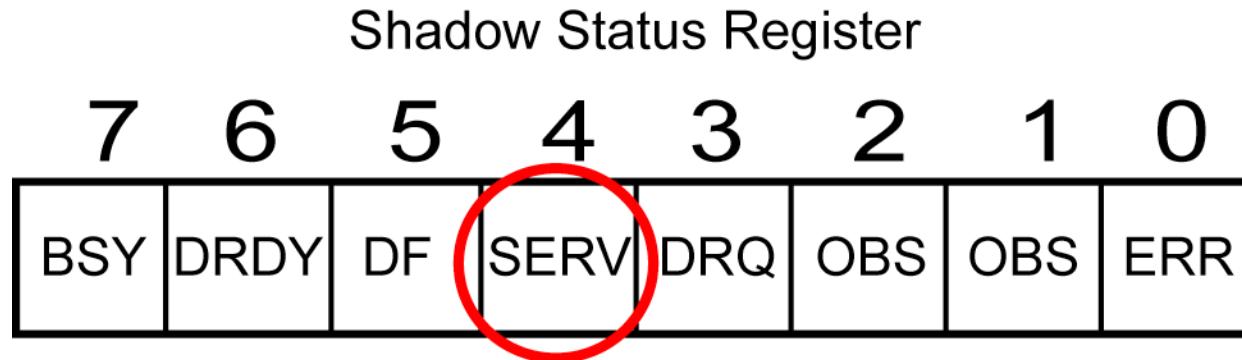
DMA Queued Command Initialization

The Shadow Feature and Sector Count registers change definition when a queued command is initialized. To support service requests, the shadow Status register includes an additional service bit.



DMA Queued Command Initialization

Status register includes an additional service bit.



DMA Queued Command

- Upon receipt of a Queued Command, a device must determine whether to complete the command immediately or to defer completion by releasing the bus.
- A bus release is signaled via a Device-to-Host Register FIS and notifies the host that the next command can be sent to the device.
- When device is ready to complete the released command it notifies the host via a Set Device Bits FIS.

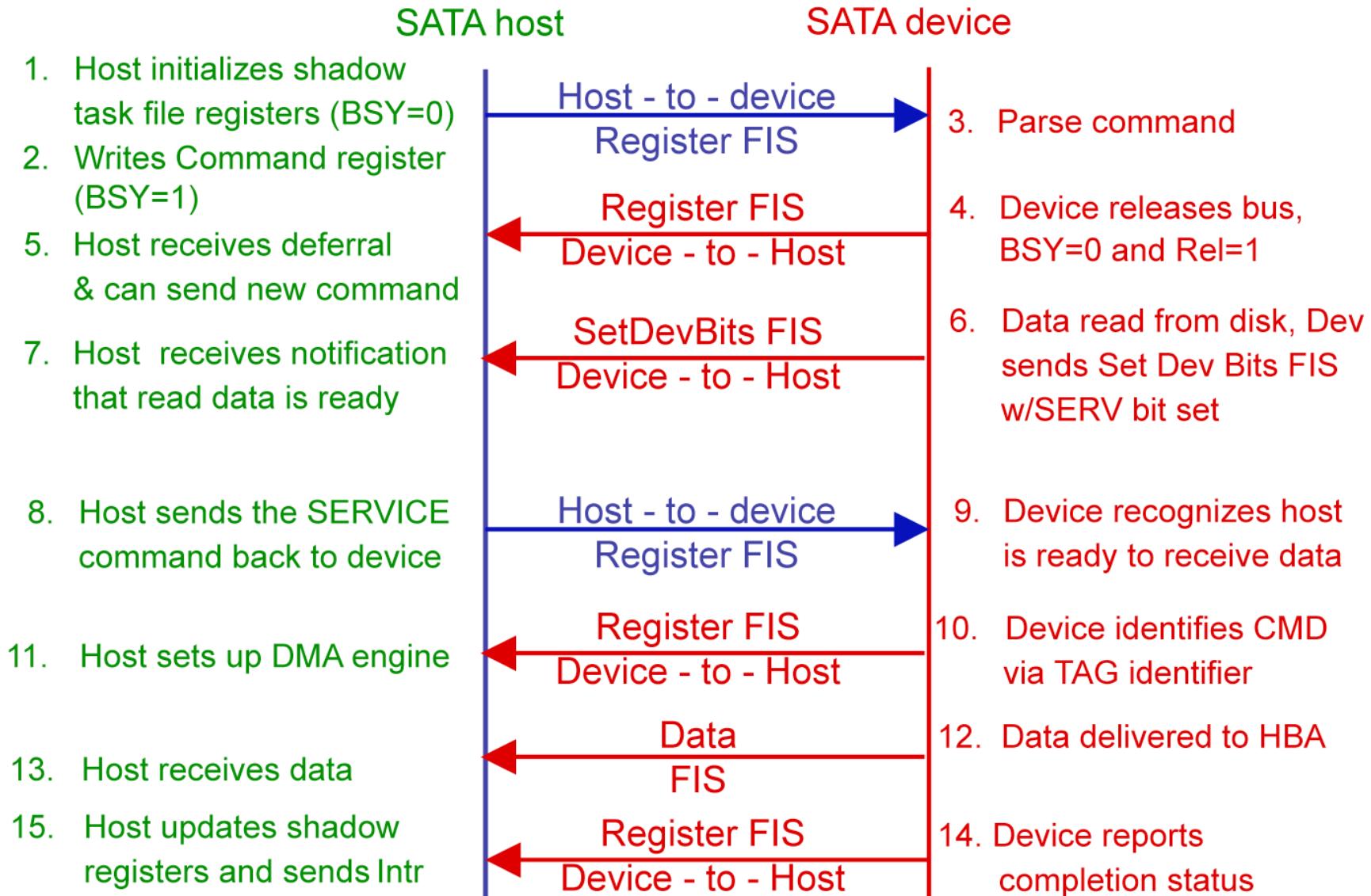
Completing the Released Command

- The host returns a Service command to request data transfer.
- Device either:
 - Proceeds directly to decoding the service request and completes the pending data transfer, or
 - First sends a Register FIS to return a tag and signal DRQ to the host before completing the data transfer.

Read DMA Queued Protocol Commands

Command	Code	ATA Command	ATAPI Command
Read DMA Queued	C7h	M	N
Read DMA Queue d Ext	26h	O	N

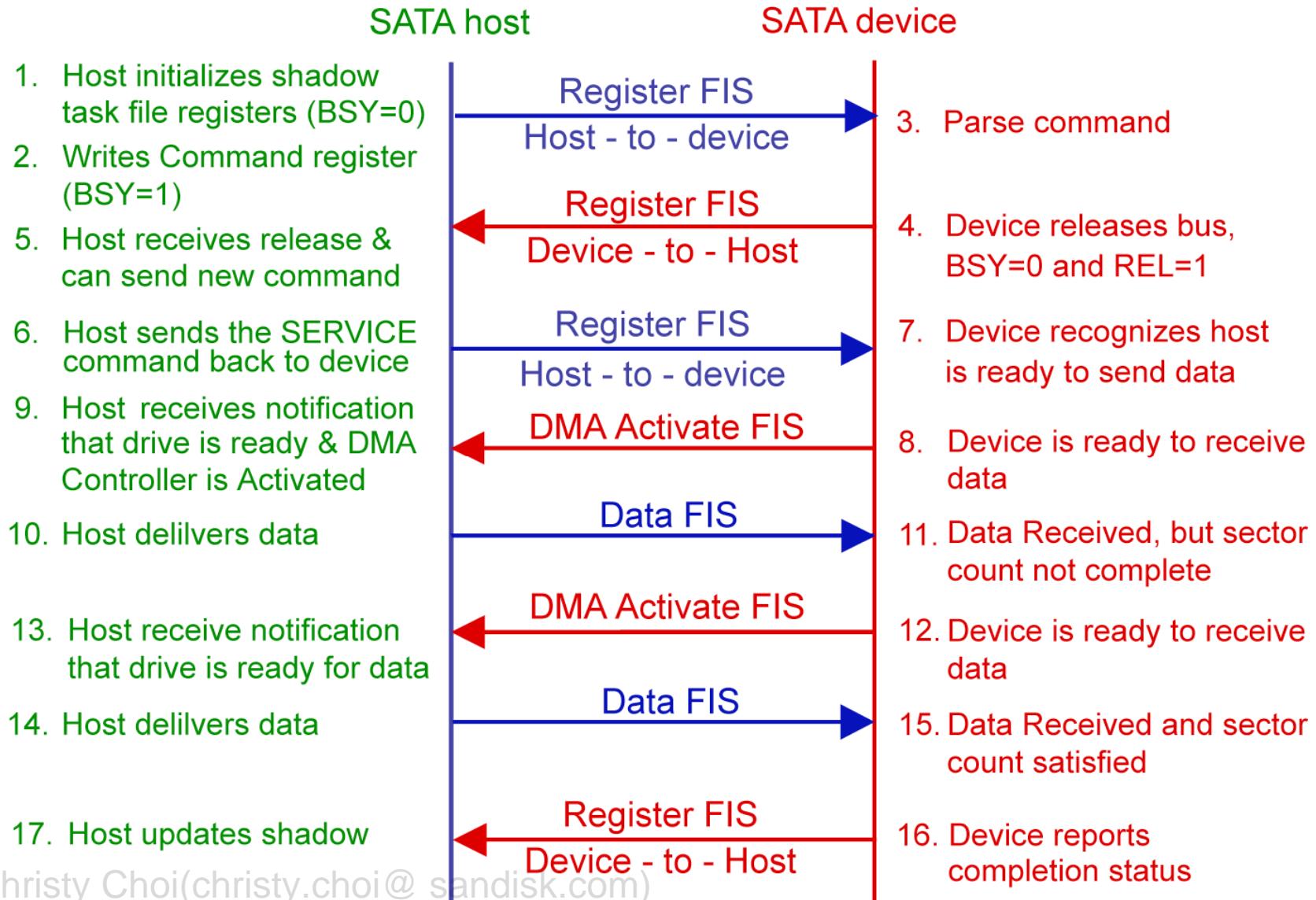
DMA Read Queued w/release



Write DMA Queued Protocol Commands

Command	Code	ATA Command	ATAPI Command
Write DM A Queued	CCh	O	N
Write DM A Queued Ext	36h	O	N
Write DM A Queued FUA Ext	3Eh	O	N

DMA-Out Queued Protocol



First Party DMA Commands

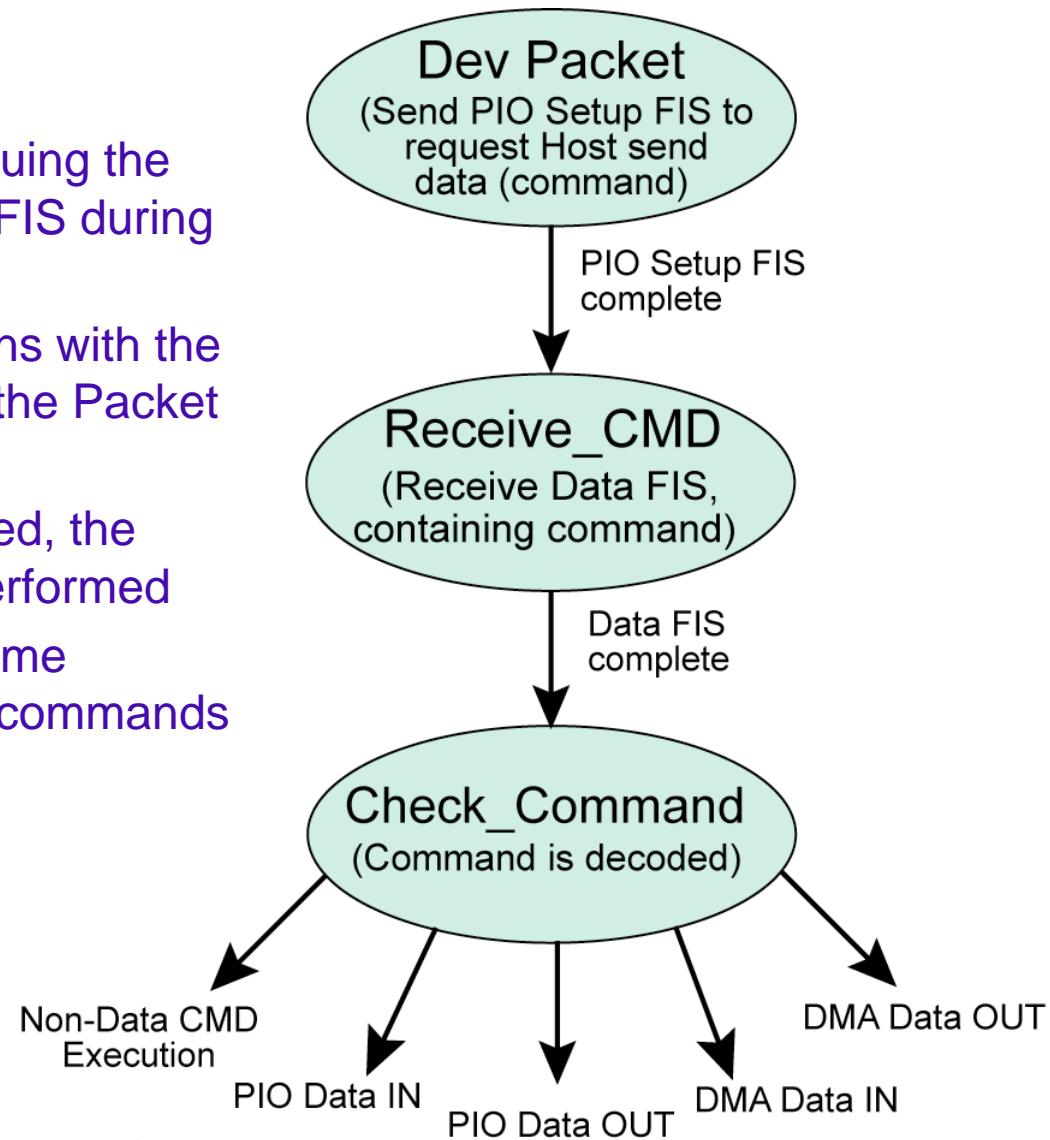
The First Party DMA Commands are used in conjunction with Native Command Queuing (NCQ). NCQ and the First Party DMA protocol is discussed in Chapter 14, entitled "Native Command Queuing," on page 235.

Packet Command Protocol

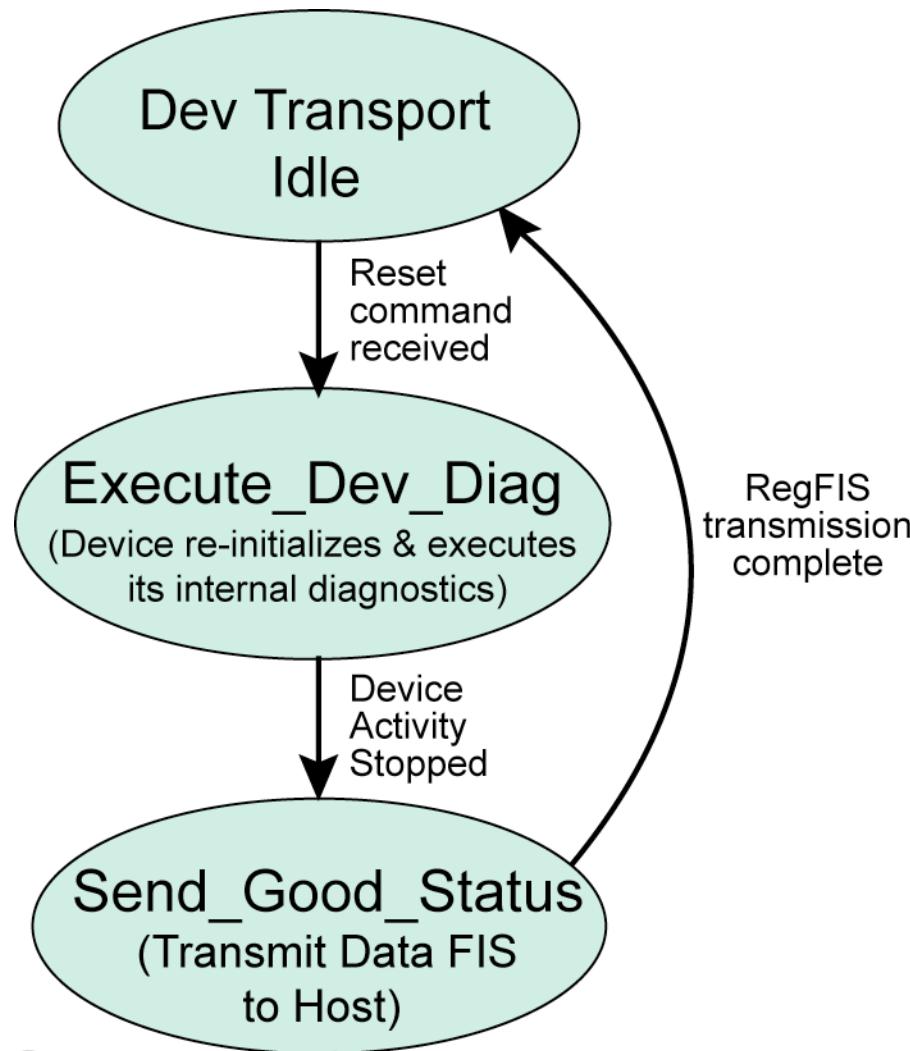
- PACKET command is used to send a SCSI CDB over ATA bus
- If command completes with an error, Error register contains a few fields with some of SCSI Sense Data
- Nearly all MMC (multimedia commands) devices use ATAPI
 - Although PACKET theoretically works with the ATA overlap and queuing features, they are never used
 - Because they have been focused on parallel ATA, to avoid hogging the bus shared with a disk drive, most commands complete immediately

Packet Protocol

- Packet protocol begins by issuing the Packet command via a Data FIS during a PIO Data Out sequence
- The diagram to the right begins with the PIO Setup FIS that requests the Packet command
- Once the command is decoded, the actual Packet command is performed
- ATAPI commands use the same sequence as the non-packet commands



ATAPI Device Reset



Device Reset (Register FIS - Good Status)

Good Status

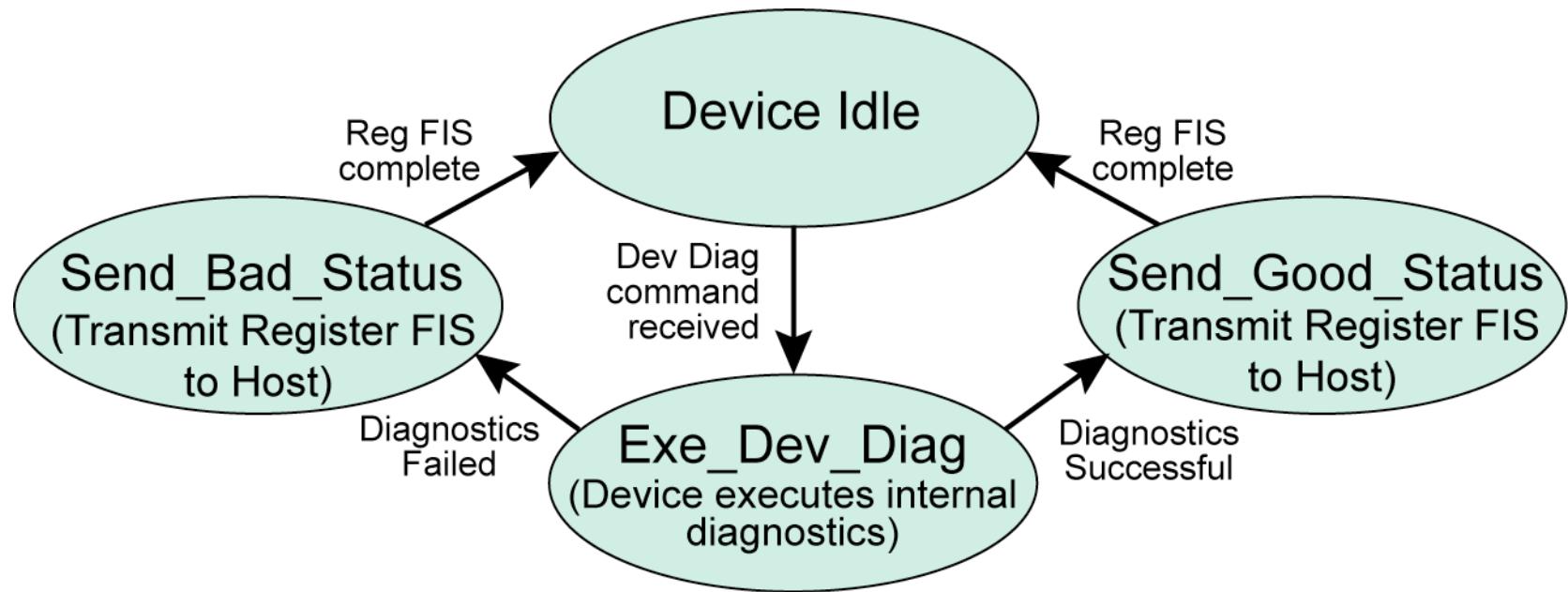
Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	01h
Status	00h

Execute Device Diagnostics

This command forces a device to perform internal diagnostics and to report the results back to the HBA. Figure 11-27 on page 219 illustrates the two possible responses following internal diagnostic having completed:

- Diagnostics have passed and Good Status is sent
- Diagnostics have failed and Bad Status is sent

Execute Device Diagnostics



Execute Device Diagnostics

(Register FIS Contents, no Packet Protocol)

Good Status

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	01h
Status	00h-70h*
* Bits 4-6 are device specific	

Bad Status

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	00h, 02h-27h
Status	00h-70h*
* Bits 4-6 are device specific	

Note that the RegFIS must also have the “I” bit set

Execute Device Diagnostics (Register FIS Contents w/Packet Protocol)

Good Status

Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	01h
Status	00h

Bad Status

Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device/Head	00h
Error	00h, 02h-7Fh
Status	00

Note that the RegFIS must also have the “I” bit set

Device Control Protocols

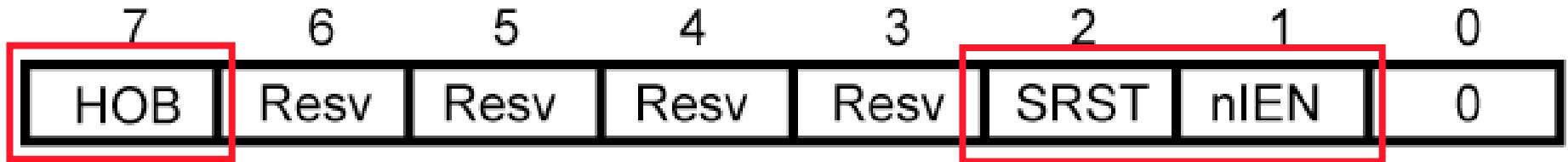
Christy Choi(christy.choi@sandisk.com)
Do Not Distribute



Control Register

Cmd Reg	Reads		Writes		Notes
Address	7	0	7	0	
01F0	Data		Data		16-bit accesses
01F1	Error		Feature		8-bit access only
01F2	Sector Count		Sector Count		8-bit access only
01F3	Sector #		Sector #		8-bit access only
01F4	Cylinder Low		Cylinder Low		8-bit access only
01F5	Cylinder High		Cylinder High		8-bit access only
01F6	Device	Head	Device	Head	8-bit access only
01F7	Status		Command		8-bit access only
Ctrl Reg					
03F6	Alternate Status		Device Control		8-bit access only

Control Register Bit Fields



- nIEN – Interrupt Enable, negative logic
- SRST – Soft Reset
- HOB – High Order Byte

Writes to the Control register do not cause a Host-to-Device Register FIS to be send unless a bit value has been changed

Register FIS - Host to Device

	+3	+2	+1	+0
DW 0	Features	Command	C R R Reserved	FIS Type (27h)
DW 1	Dev/Head	Cyl High	Cyl Low	Sector Number
DW 2	Features (exp)	Cyl High (exp)	Cyl Low (exp)	Sec Num (exp)
DW 3	Control	Reserved (0)	Sec Count (exp)	Sector Count
DW 4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

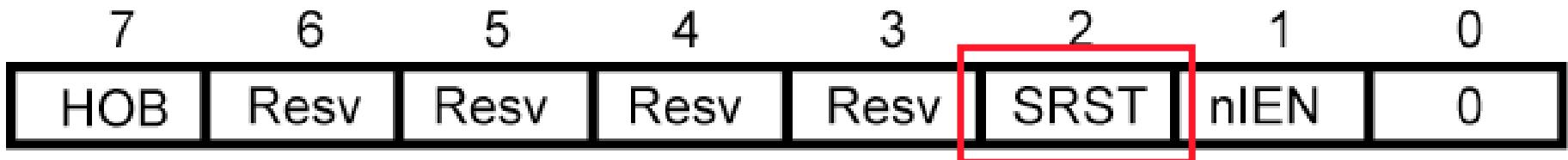
C = 0 write to Control Register

Interrupt Enable/Disable

- The nIEN bit provides legacy software support.
- when nIEN is cleared (0), interrupt request generation is enabled within the HBA and setting the bit disables interrupt generation. After an HBA reset the nIEN bit is cleared, and interrupts are enabled.
- Any write to the shadow control register clears an interrupt pending condition within the HBA.
- Unlike the PATA environment, the nIEN bit has no affect on a SATA drive's behavior.
- In the SATA environment the HBA actually signals interrupt requests and handles the nIEN function.

Software Reset (SRST) Behavior

Setting the SRST bit in the control register triggers the HBA to send a Register FIS to the device.



Software Reset

- Software resets a SATA drive in legacy fashion by writing a one to the SRST bit within the shadow Control register.
- This action causes the HBA to send a Register FIS (with “C” bit cleared) to the drive.
- Software must write a zero to the SRST bit to clear the reset.
- An SRST affects only the SATA device and has no effect on the link (unlike a COMRESET).

Soft Reset Status Values

Good Status

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	01h
Status	00h-70h*

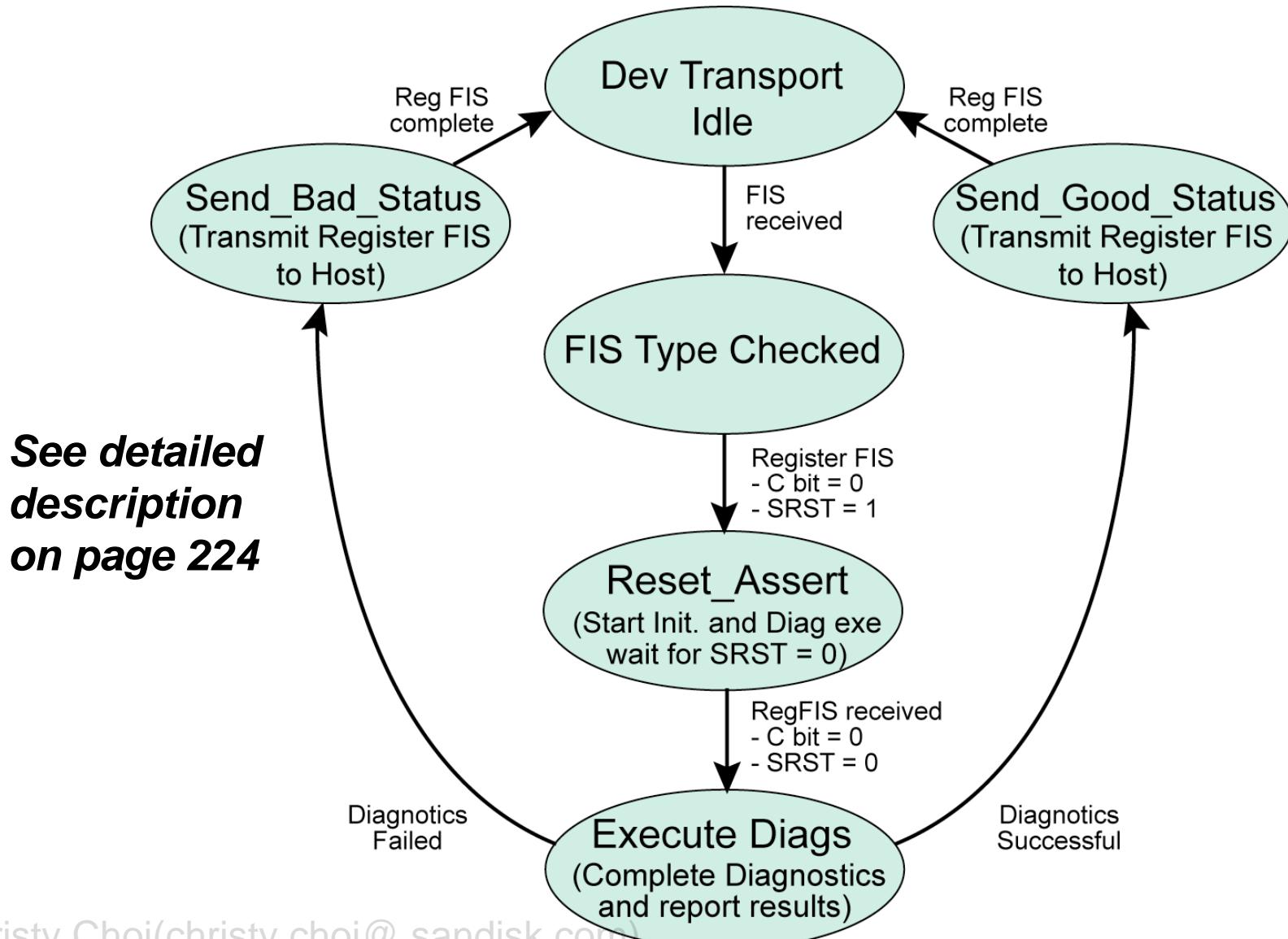
* Bits 4-6 are device specific

Bad Status

Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device/Head	00h
Error	00h, 02h-27h
Status	00h-70h*

* Bits 4-6 are device specific

Software Reset



High Order Byte

Setting the HOB bit allows reading the previously written byte from the selected 8-bit register.

See example, next slide

Start Sector Address (LBA)

LBA = Logical Block Address (48 bits)

Cmd Reg	Reads	Writes	Notes
Address	7	0	7
01F0		Data	16-bit accesses
01F1	Error	Feature	Two 8-bit accesses
01F2		Sector Count	Two 8-bit accesses
01F3		LBA Low (31:24 then 7:0)	Two 8-bit accesses
01F4		LBA Middle (39:32 then 15:8)	Two 8-bit accesses
01F5		LBA High (47:40 then 23:16)	Two 8-bit accesses
01F6	Device	Device	8-bit access only
01F7	Status	Command	8-bit access only
Ctrl Reg			
03F6	Alternate Status	Device Control	8-bit access only

Part 4

SATA II

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute



SATA II Features

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute



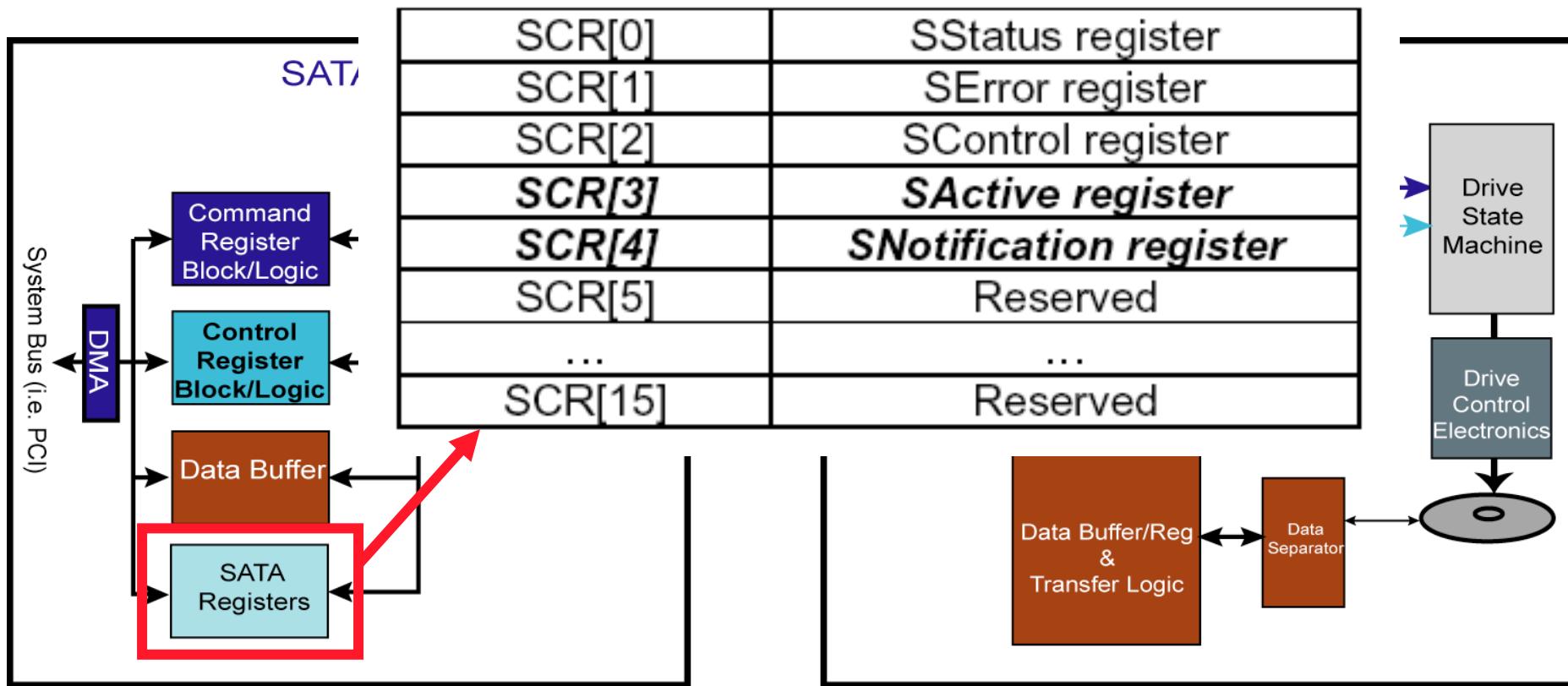
SATA II Enhancements

- Performance and Reliability Features — Higher transmission rates and support for Native Command Queuing
- Server-specific Support — SATA II added server-related enhancements that focus more on the SATA infrastructure and less on the drives themselves.
- Fixes and Enhancements to SATA — A variety of fixes and miscellaneous features were added to SATA II; including asynchronous event notification.

SATA II Enhancements

- New features require changes in Host and/or Devices:
 - Host changes to support Backplane Interconnect for Rack Mount Implementations
 - Transport and Command Layer changes to support FPDMA and NCQ
 - Enhanced Error Reporting via PHY Event Counters
 - New Identify Device/Set Feature definition to report new feature capability and to enable these features

New SATA-Specific Registers



Performance & Reliability Features

- Speed of 3.0 Gbps
- Native Command Queuing
- Asynchronous Signal Recovery

More Support for Server Apps

- Port Multipliers allow a single HBA port to support up to 15 drives
- Port Selectors provide “dual-port” functionality at the drives
- Hot Plug / Hot Docking support
- Multilane Cables
- Support for Enclosure Services and Management

Native Command Queuing

Christy Choi(christy.choi@sandisk.com)
Do Not Distribute

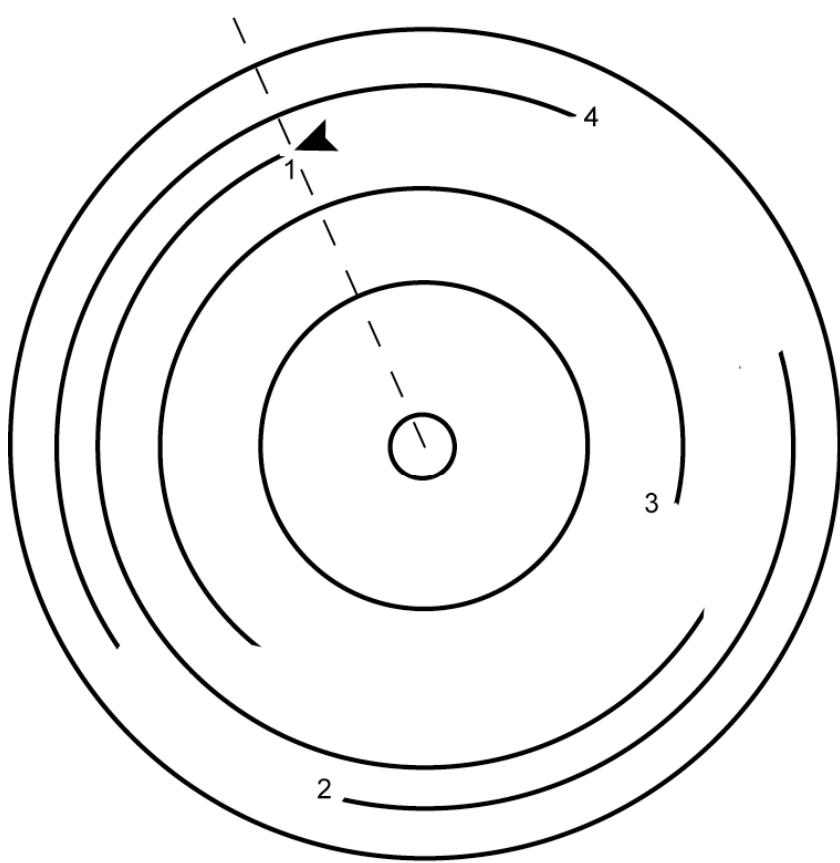


The Performance Problem

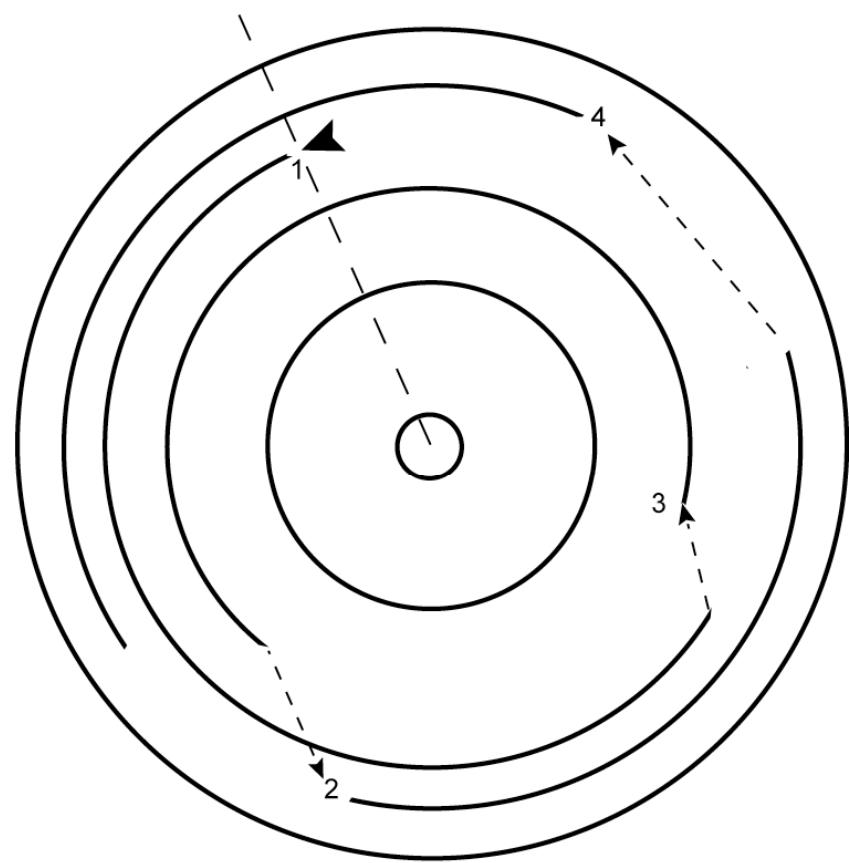
Problems Prior to command queuing:

- Earlier drives always process commands in the order received
- Seek time required to access data associated with subsequent commands can take several milliseconds
- Interrupts are generated upon completion of each command, increasing host software and CPU overhead
- Existing queued DMA operations also have high overhead

Native Command Queuing



A



B

Primary Benefits of Native Command Queuing

- Enhanced Drive Performance
 - Performance of 7200 rpm drive with NCQ is roughly equivalent to that of a standard 10k rpm drive
- Improved Drive Endurance
 - Less wear on drive's mechanical components
- Reduced Host Software/CPU Overhead

System Support Requirements

Native Command Queuing uses three features:

- First Party DMA Operation
- Race-free status return
- Interrupt Aggregation

“Race free”: no status notifications are lost or overwritten

“Aggregation”: if multiple commands complete within a short time, one interrupt can get service for all of them, reducing latency.

System Support Requirements

Other changes include:

- Extensions to state machines
- FPDMA Protocol
- New FPDMA Queued Commands
- Modification to some FISs
 - Set Device Bits
 - DMA Setup
- New SATA register - SActive

NCQ Drives

- Track an internal queue of pending commands
- Re-order command execution to improve performance to achieve shortest overall access time
- Dynamically include new commands in the Rotational Position Ordering schedule

Word Offset	R/O	Description
0 - 74		Defined in ATA/ATAPI-7
75	Optional	<p>Queue Depth 15-5 Reserved 4-0 Maximum queue depth-1</p>
76		<p>Serial ATA Capabilities 15-11 Reserved 10 Phy event counters supported 9 Host initiated link Power Management requests received 8 Native Command Queuing supported 7-4 Reserved 3 Reserved for SATA 2 Supports Serial ATA Gen-2 signaling 1 Supports Serial ATA Gen-1 signaling 0 Reserved (0)</p>
77		Reserved for SATA
78	Optional	<p>Serial ATA Features 15-7 Reserved 6 Software settings preservation supported 5 Reserved 4 In-order data delivery supported 3 Drive initiates link power management requests 2 DMA Setup FIS Auto-Activate optimization supported 1 DMA Setup FIS non-zero buffer offsets supported 0 Reserved (0)</p>
79	Optional	<p>Serial ATA Features Enabled 15-7 Reserved 6 Software settings preservation enabled 5 Reserved 4 In-order data delivery enabled 3 Drive initiates link power management requests 2 DMA Setup FIS Auto-Activate optimization enabled 1 DMA Setup FIS non-zero buffer offsets enabled 0 Reserved (0)</p>
80-255		Defined in ATA/ATAPI-7

New Commands

- First Party DMA Read command
- First Party DMA Write command

Note that non-NCQ commands can't be sent while NCQ commands are outstanding. If one is sent, the device will abort it.

First Party DMA

- SATA 1.0a does not fully describe the First Party DMA (FPDMA) protocol
- Because NCQ commands may complete out of order, FPDMA is a critical element in completing transactions
- FPDMA uses the Tag associated with each command when transferring data, permitting the HBA to locate the memory buffer used for the command.

FPDMA Read Queued Command

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			

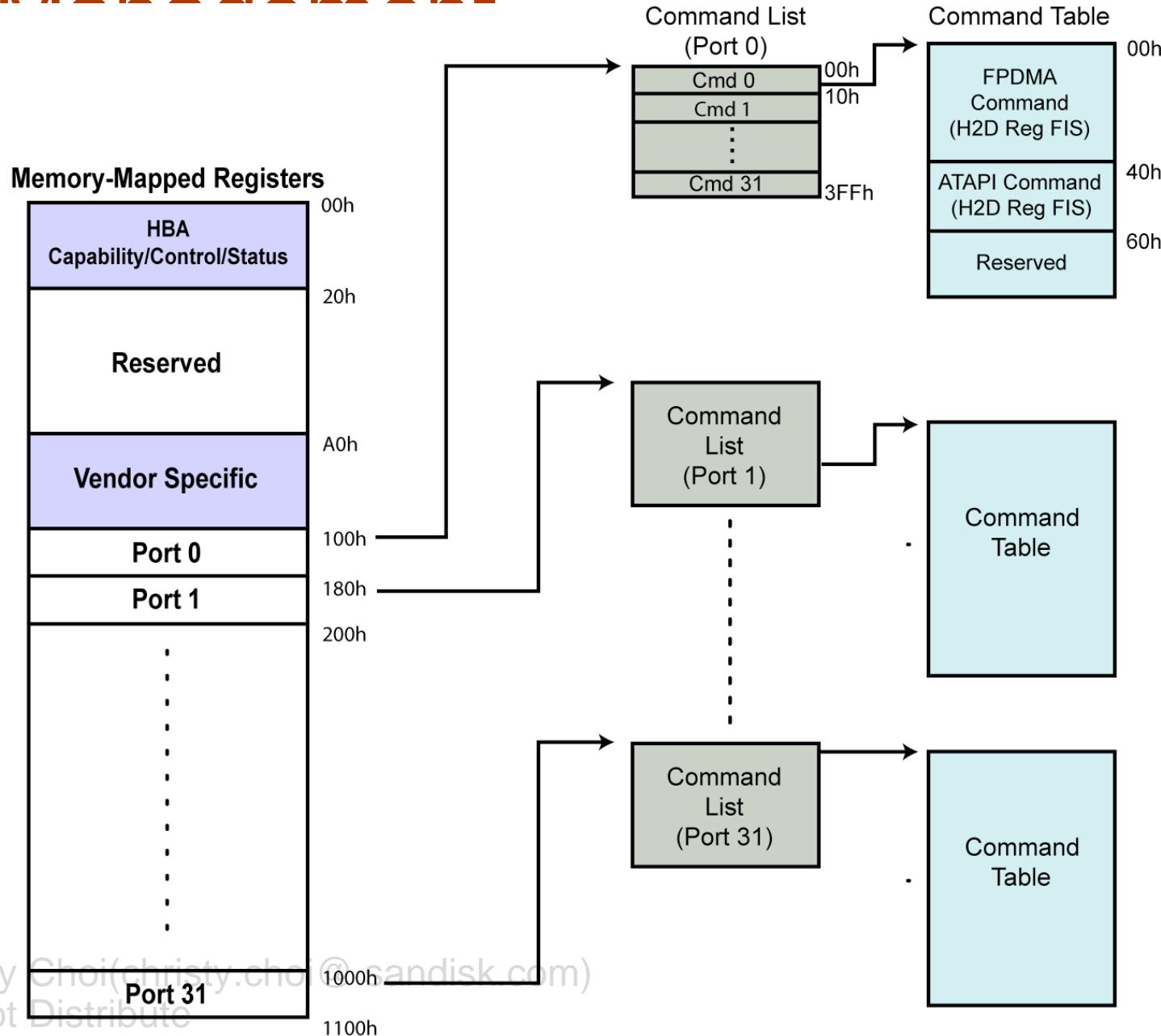
FUA (Force Unit Access): requires data be transferred to or from the media before completion can be indicated, even if caching is enabled

FPDMA Write Queued Command

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							

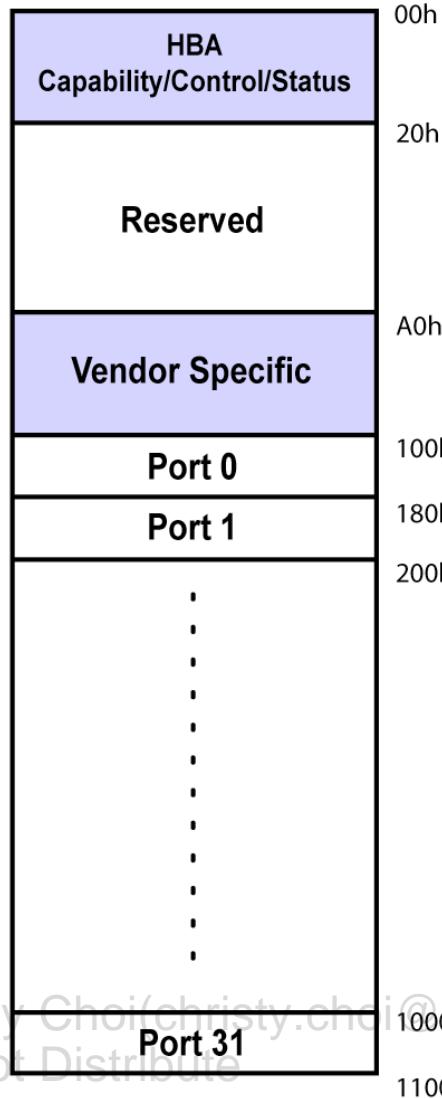
Queue

339



DMA Setup

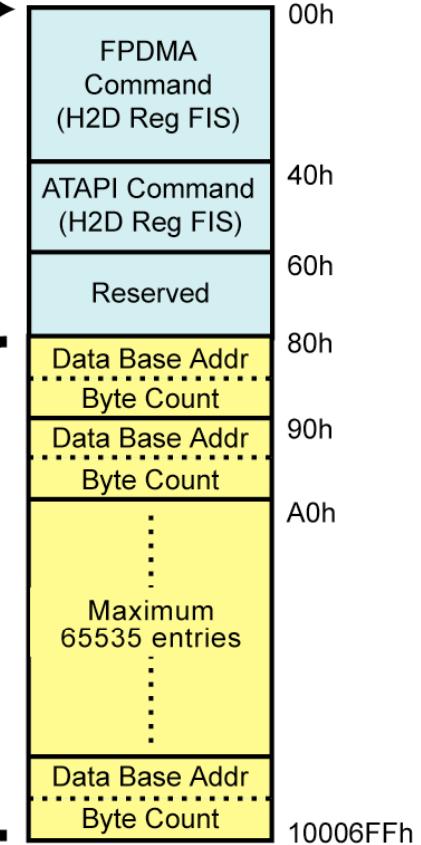
Memory-Mapped Registers



Command List
(Port 0)

Cmd 0	00h
Cmd 1	10h
⋮	⋮
Cmd 31	3FFh

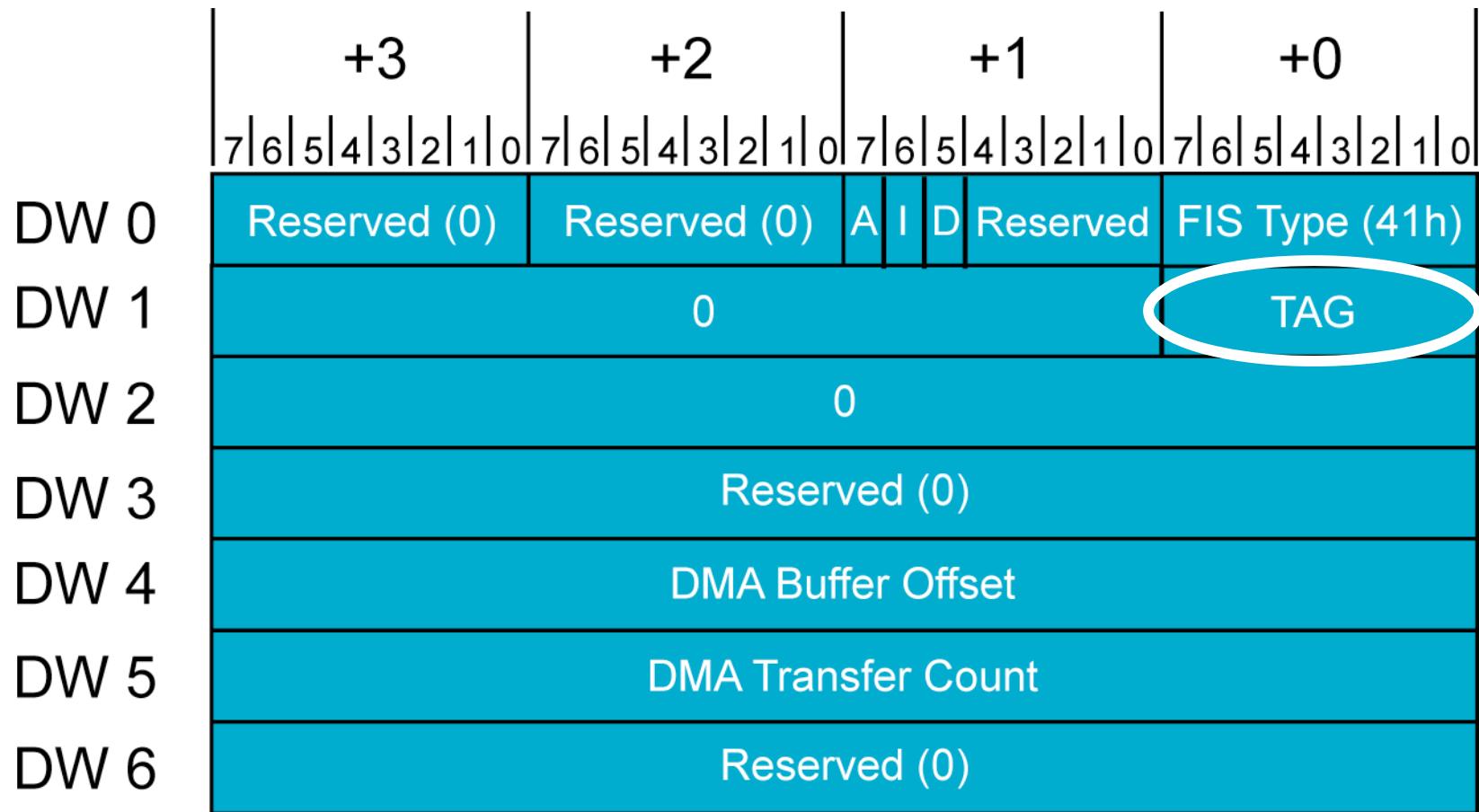
Command Table



Command List
(Port 31)

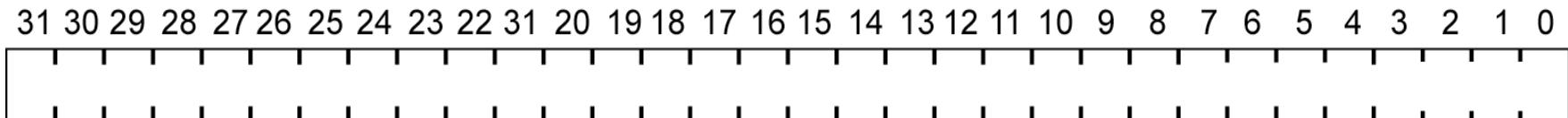
Command Table

Tag - FPDMA_Setup FIS



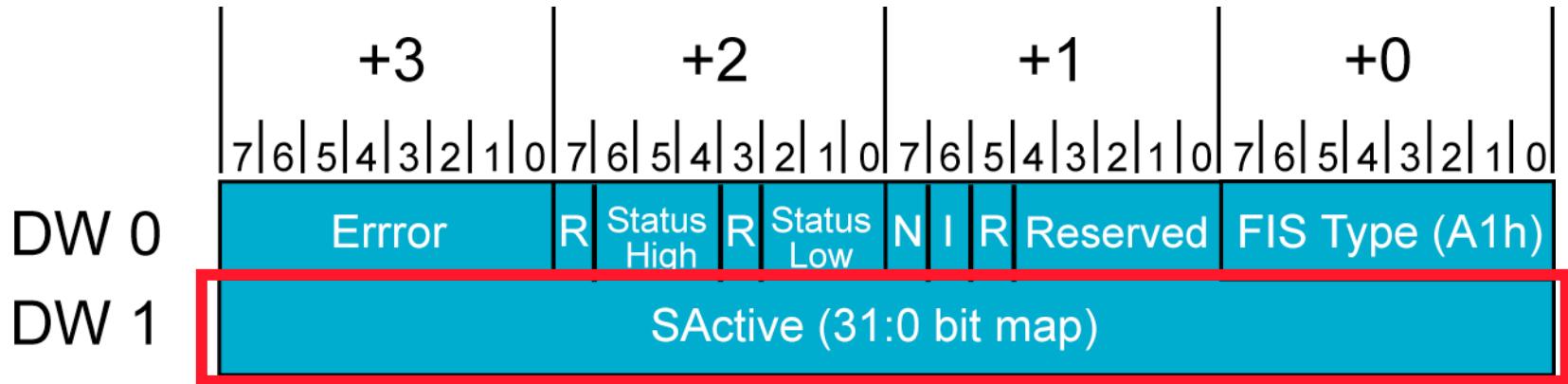
Active Register (Outstanding Queued Commands Active)

SATA II Enhancement



Each bit position represents a Queued command that has not yet completed.
The bit positions correlates to the TAG number of the outstanding Queued command.

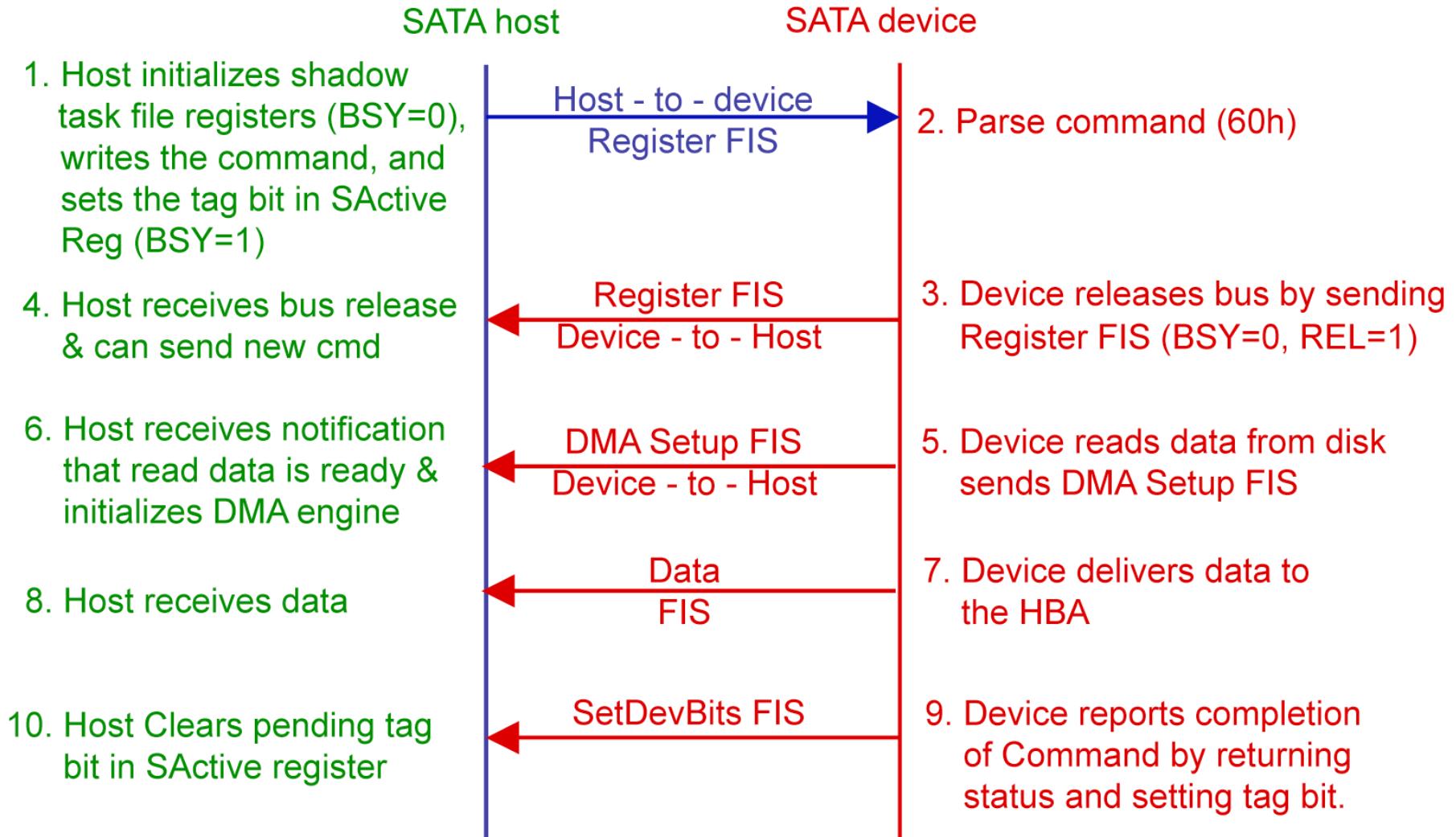
SActive - Set Device Bits FIS



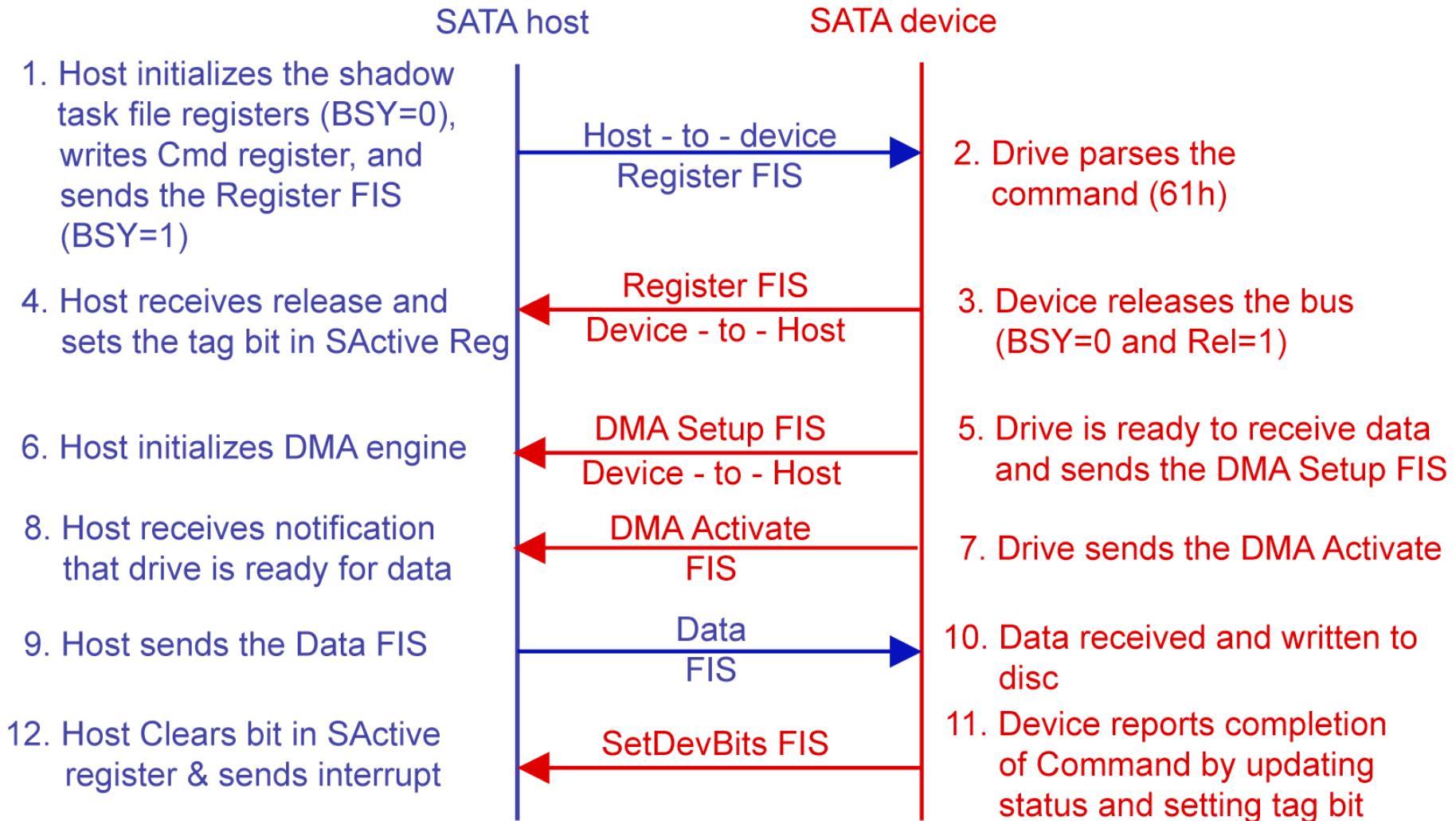
Device sets a bit in this register to indicate that the corresponding command has completed. BSY bit is not changed by this FIS and only conveys readiness to receive another command.

If a command fails, device will abort all outstanding commands and send a Register FIS so software can diagnose the problem. A READ LOG EXT command should be sent to fetch log page 10 and learn which tag failed.

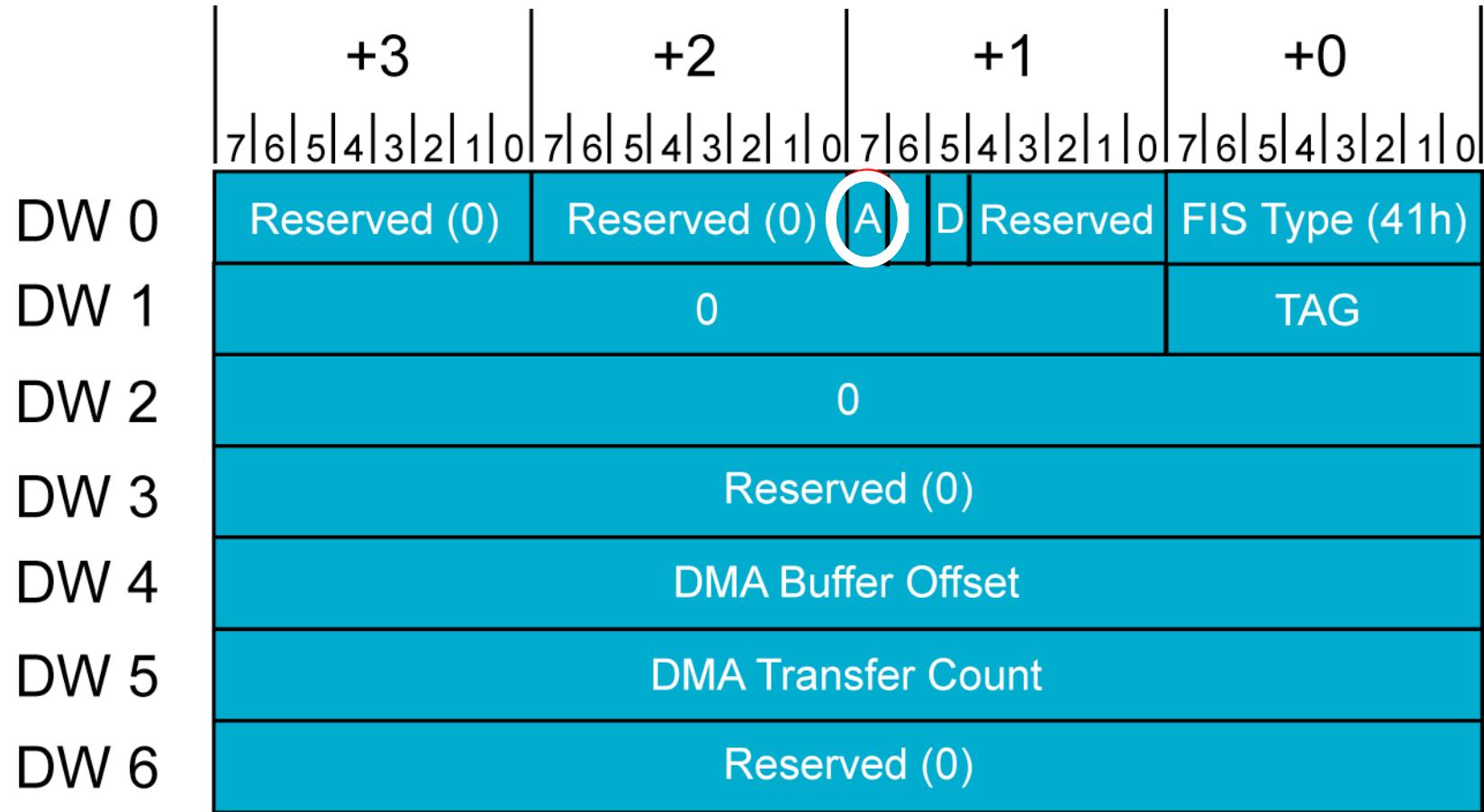
FPDMA Read



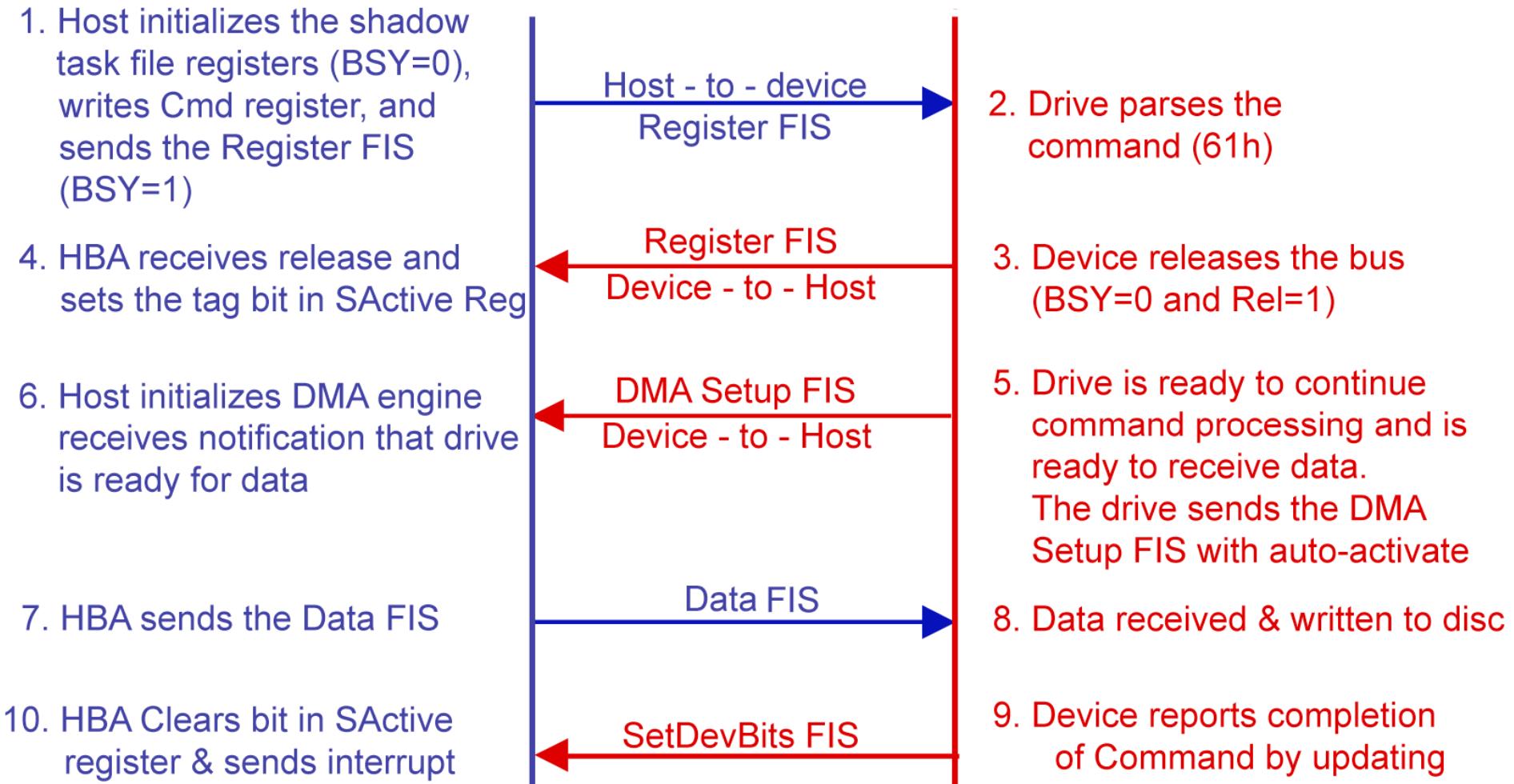
FPDMA Write



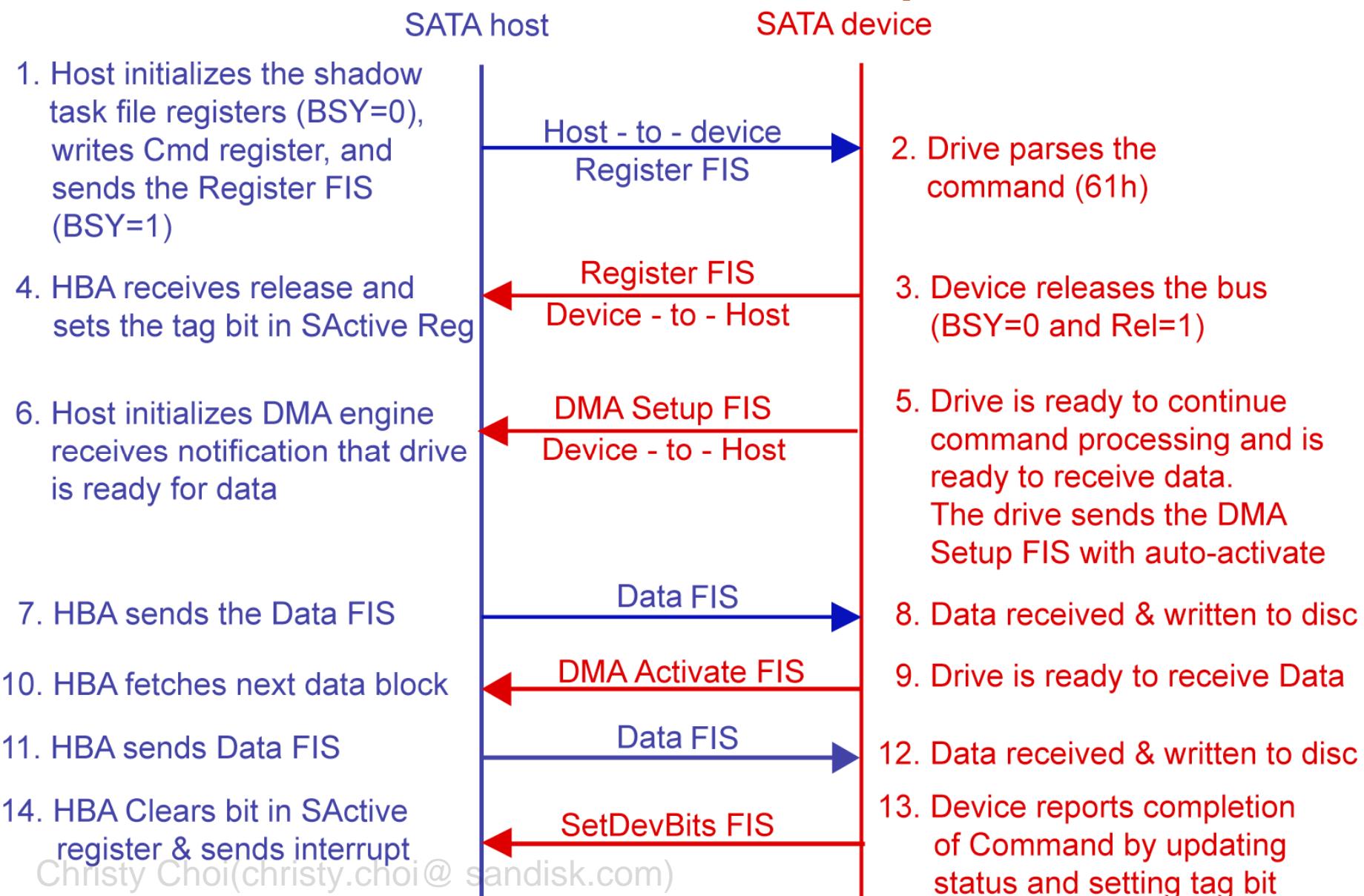
Auto Activate - DMA_Setup FIS



FPDMA with Auto Activate (1 Data FIS)



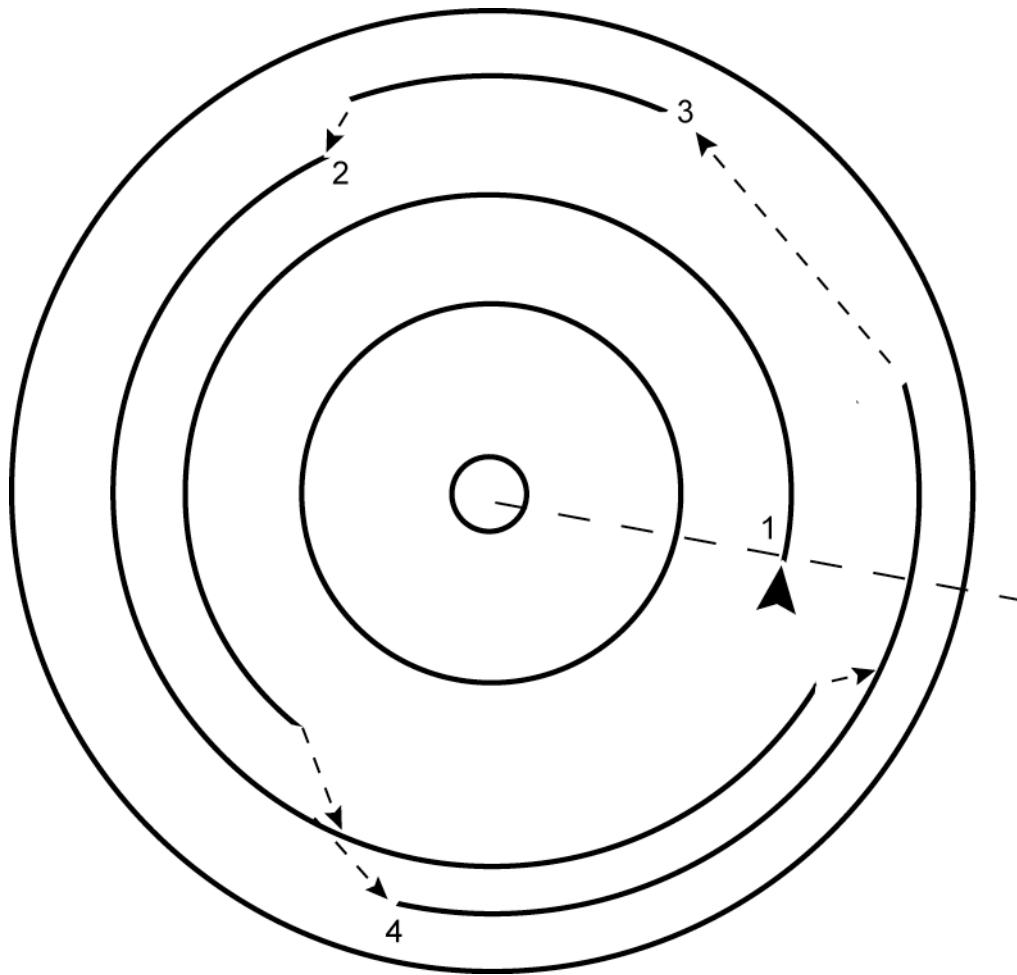
FPDMA Write – Multiple Data



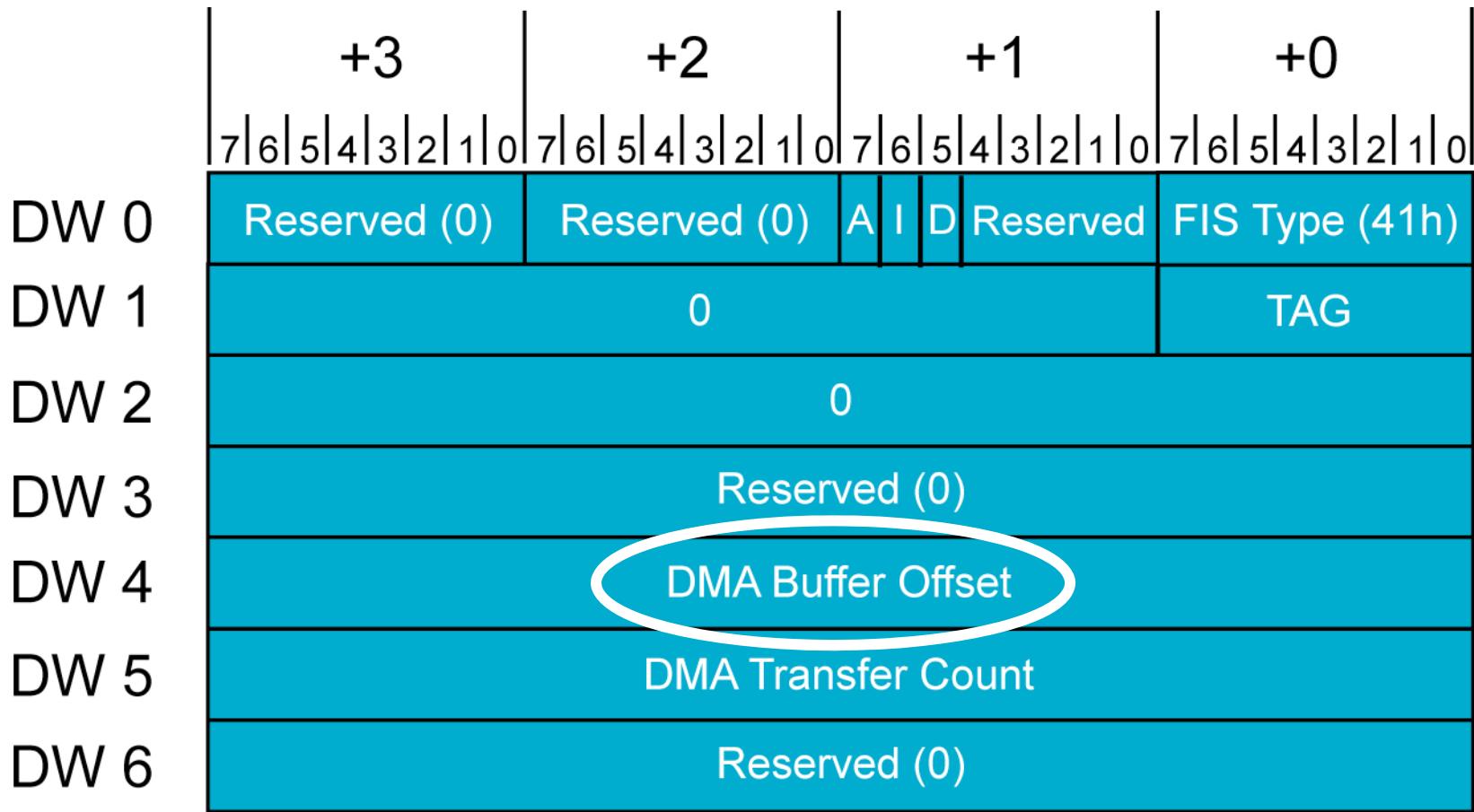
Device Identify Data - SATA II

Word Offset	R/O	Description
0 - 74		Defined in ATA/ATAPI-7
75	Optional	<p>Queue Depth</p> 15-5 Reserved 4-0 Maximum queue depth-1
76		<p>Serial ATA Capabilities</p> 15-11 Reserved 10 Phy event counters supported 9 Host-initiated link Power Management requests received 8 Native Command Queuing supported 7-4 Reserved 3 Reserved for SATA 2 Supports Serial ATA Gen-2 signaling 1 Supports Serial ATA Gen-1 signaling 0 Reserved (0)
77		Reserved for SATA
78	Optional	<p>Serial ATA Features</p> 15-7 Reserved 6 Software settings preservation supported 5 Reserved 4 In-order data delivery supported 3 Drive initiates link power management requests 2 DMA Setup FIS Auto-Activate optimization supported 1 DMA Setup FIS non-zero buffer offsets supported 0 Reserved (0)
79	Optional	<p>Serial ATA Features Enabled</p> 15-7 Reserved 6 Software settings preservation enabled 5 Reserved 4 In-order data delivery enabled 3 Drive initiates link power management requests 2 DMA Setup FIS Auto-Activate optimization enabled 1 DMA Setup FIS non-zero buffer offsets enabled 0 Reserved (0)
80-255		Defined in ATA/ATAPI-7

NCQ with Non-Zero Offsets



Non-Zero Offsets



Word Offset	R/O	Description
0 - 74		Defined in ATA/ATAPI-7
75	Optional	<p>Queue Depth</p> <p>15-5 Reserved 4-0 Maximum queue depth-1</p>
76		<p>Serial ATA Capabilities</p> <p>15-11 Reserved 10 Phy event counters supported 9 Host initiated link Power Management requests received 8 Native Command Queuing supported</p> <p>7-4 Reserved 3 Reserved for SATA 2 Supports Serial ATA Gen-2 signaling 1 Supports Serial ATA Gen-1 signaling 0 Reserved (0)</p>
77		Reserved for SATA
78	Optional	<p>Serial ATA Features</p> <p>15-7 Reserved 6 Software settings preservation supported 5 Reserved 4 In-order data delivery supported 3 Drive initiates link power management requests 2 DMA Setup FIS Auto Activate optimization supported 1 DMA Setup FIS non-zero buffer offsets supported</p> <p>0 Reserved (0)</p>
79	Optional	<p>Serial ATA Features Enabled</p> <p>15-7 Reserved 6 Software settings preservation enabled 5 Reserved 4 In-order data delivery enabled 3 Drive initiates link power management requests 2 DMA Setup FIS Auto Activate optimization enabled 1 DMA Setup FIS non-zero buffer offsets enabled</p> <p>0 Reserved (0)</p>
80-255		Defined in ATA/ATAPI-7

Port Multipliers

Christy Choi(christy.choi@sandisk.com)
Do Not Distribute



Port Multiplier Characteristics

Design Goal for the Port Multiplier:

- Serial ATA 1.0a devices may be attached without modification
- Link and Phy layer compatibility with Serial ATA 1.0a must be maintained for both hosts and devices
- No new primitives may be added as part of the definition
- No new FIS types may be added as part of the definition

PM limitations:

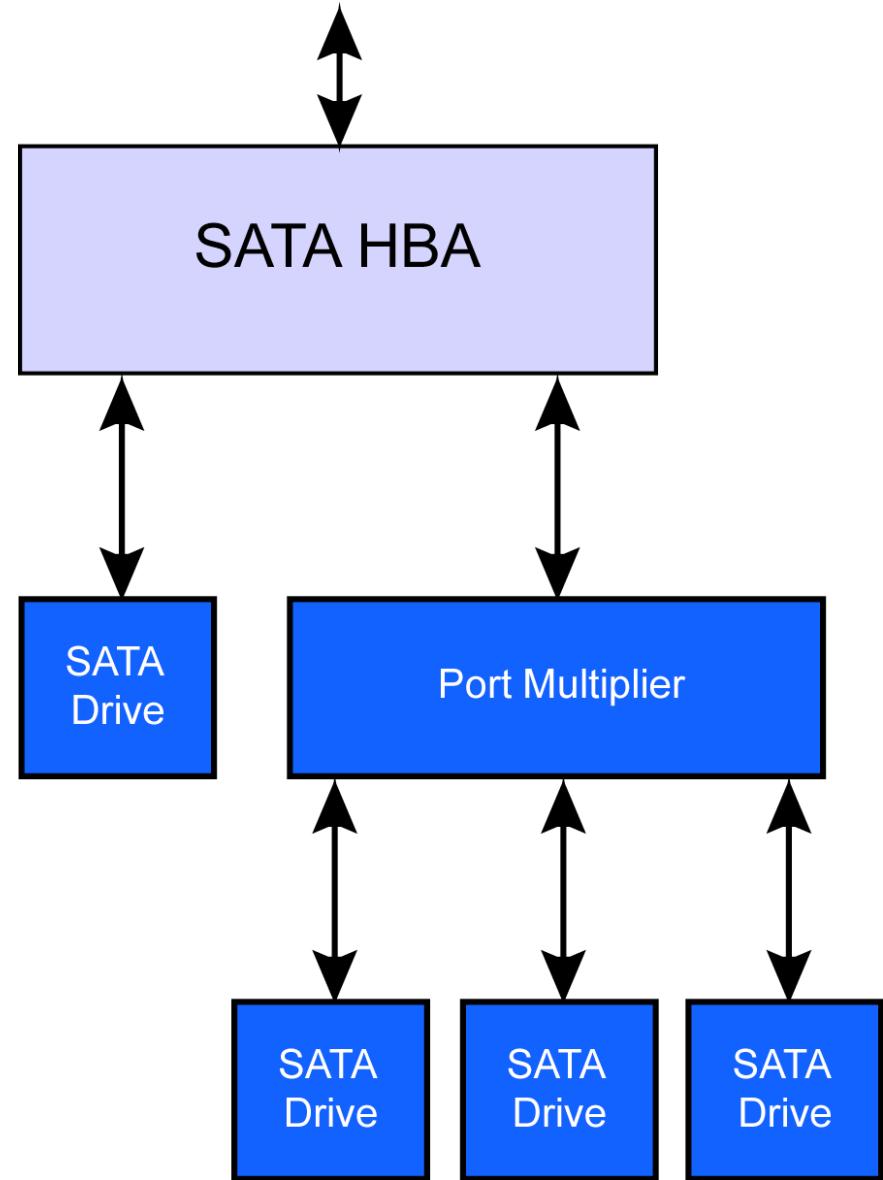
- Only one active host connection is supported
- Port Multipliers cannot be cascaded
- Maximum fan-out of 15 device connections

Port Multipliers support:

- booting with legacy software on device port 0h (required)
- staggered spin-up Avoids excessive power use when all drives spin up at once.
- hot plug Pin 11 of the power connector for the drive tells it whether to spin up immediately, after reset, or wait for initialization.

Port Multiplier

- PM uses store and forward mechanism to transfer each FIS between host and drive.
- Port numbers are used to route each FIS to the target drive



PM Port Number

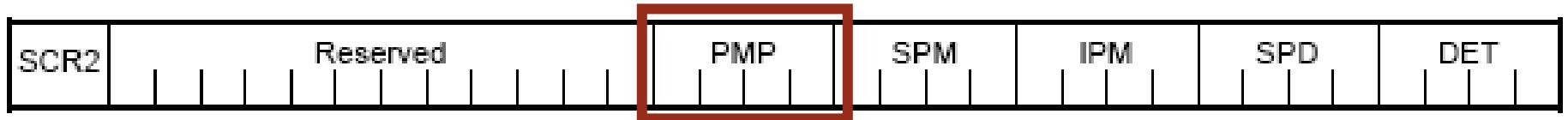
	+3	+2	+1	+0
DW 0	Features	Command	C R R R Port	FIS Type (27h)
DW 1	Device	LBA High	LBA Middle	LBA Low
DW 2	Features (exp)	LBA High (exp)	LBA Mid (exp)	LBA Low (exp)
DW 3	Control	Reserved (0)	Sec Count (exp)	Sector Count
DW 4	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

HBA Port Switching Types

- Command-Based Switching
 - Obtains Port Number from PMP field of the SCONTROL Register
 - Must complete current command before delivering a command to a different port
- Packet-Based Switching
 - Port Number is specified by the FIS being fetched from memory
 - Commands can be issued to multiple ports simultaneously

HBA Switching Types (Command-Based Switching)

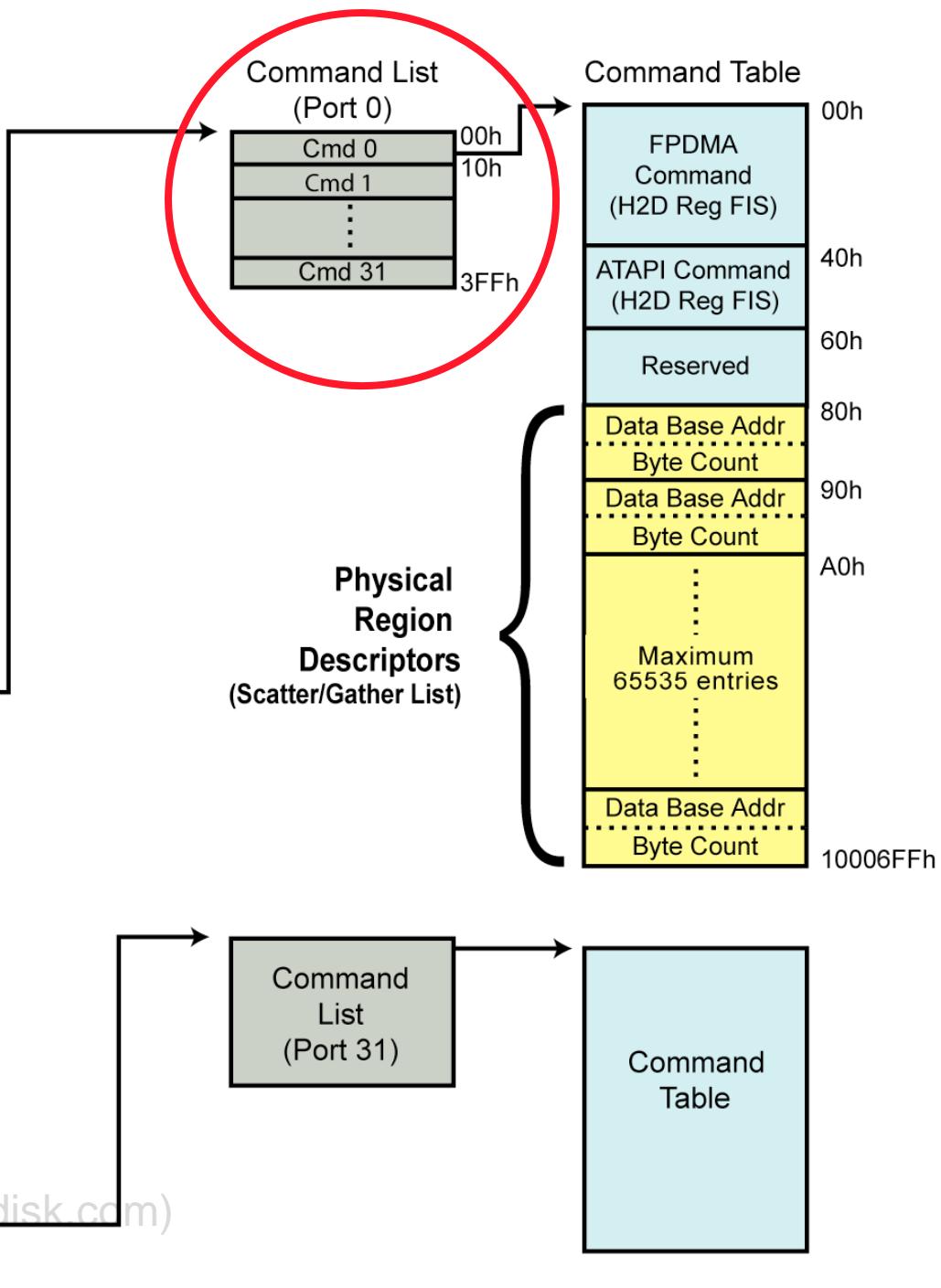
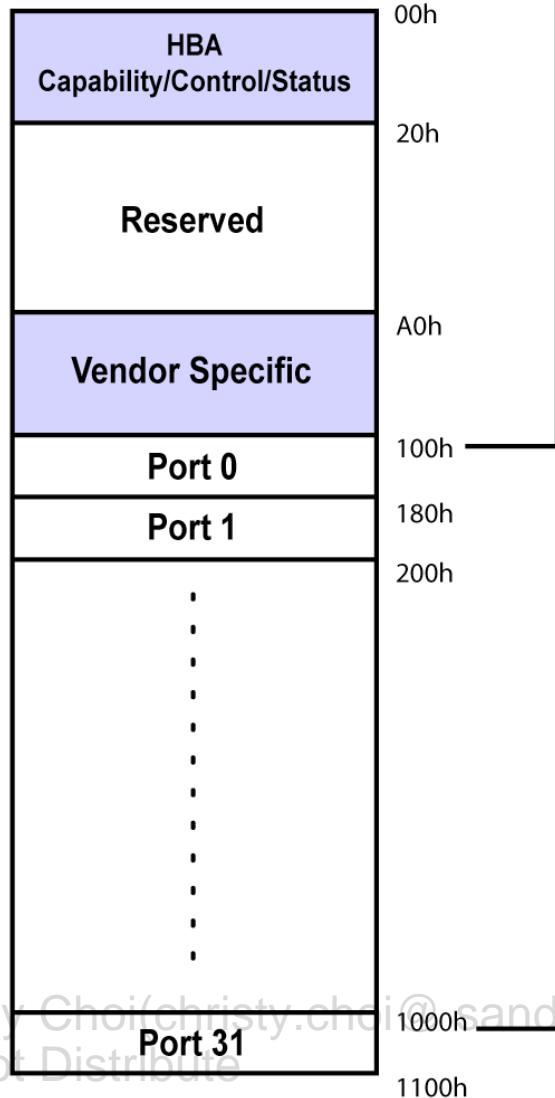
Source of Port number is SControl register



PMP -- The optional Port Multiplier Port (PMP) field specifies the 4-bit value to be placed in the PM Port field of all transmitted FISs. PM software loads this value during initialization. The value by default (following Reset) is 0h.

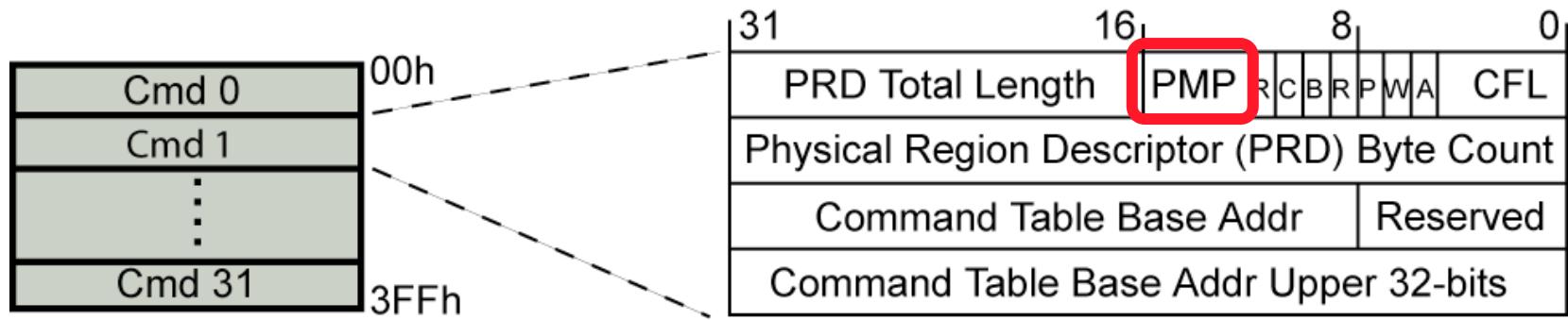
AHCI (Port Numbers)

Memory-Mapped Registers



HBA Switching Types

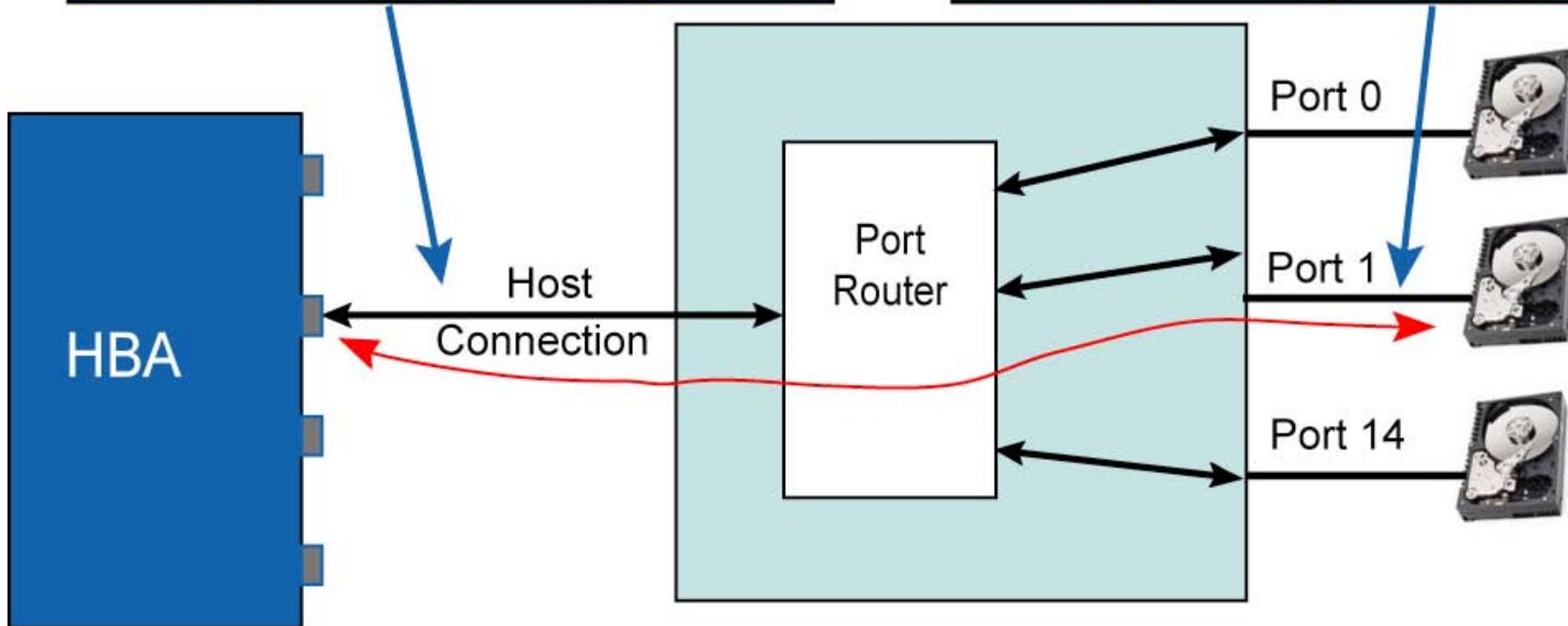
(Packet Switching - AHCI)



Frame Routing Across PM

+3	+2	+1	+0
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Features	Command	C R R R Port	FIS Type (27h)
Device	LBA High	LBA Middle	LBA Low
Features (exp)	LBA High (exp)	LBA Mid (exp)	LBA Low (exp)
Control	Reserved (0)	Sec Count (exp)	Sector Count
Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)

+3	+2	+1	+0
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Features	Command	C R R Reserved	FIS Type (27h)
Device	LBA High	LBA Middle	LBA Low
Features (exp)	LBA High (exp)	LBA Mid (exp)	LBA Low (exp)
Control	Reserved (0)	Sec Count (exp)	Sector Count
Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)



FIS Transmission - HBA to Drive (no errors)

1. The HBA arbitrates for link ownership and starts FIS transmission.
2. PM receives frame and returns primitives to the HBA, except the R_OK or R_ERR primitive.
3. PM detects the FIS type and port number and determines the port to which the FIS must be transferred.
4. The PM arbitrates for link ownership and begins transmitting the FIS to the drive. The PM must send the FIS to the drive unaltered, including the Port number field. Note: PM is not required to check and recalculate CRC before sending the FIS to the drive. This reduces latency and buffering.
5. Drive receives FIS and returns primitives to the PM in normal fashion. At this point the PM is involved in two separate and independent link transmissions of the same FIS.
6. When the PM receives the FIS in its entirety it does not return R_OK.
7. When FIS delivery completes to the drive, it returns R_OK to the PM, which in turn sends R_OK to the HBA thereby completing FIS delivery.

FIS Transmission - HBA to Drive

(Example Error Conditions)

Below are two error scenarios and how they must be handled:

1. When the PM checks the port number to determine routing, the port number is invalid (e.g., destination port not implemented). The PM must return a SYNC primitive to the HBA, terminating the FIS transmission.
2. If the X bit is set in the SError Register's Diagnostic field then there has been a change in device connection status. The PM is required to send a SYNC primitive in response, forcing FIS transmission to be terminated.

FIS Transmission - Drive to HBA (no errors)

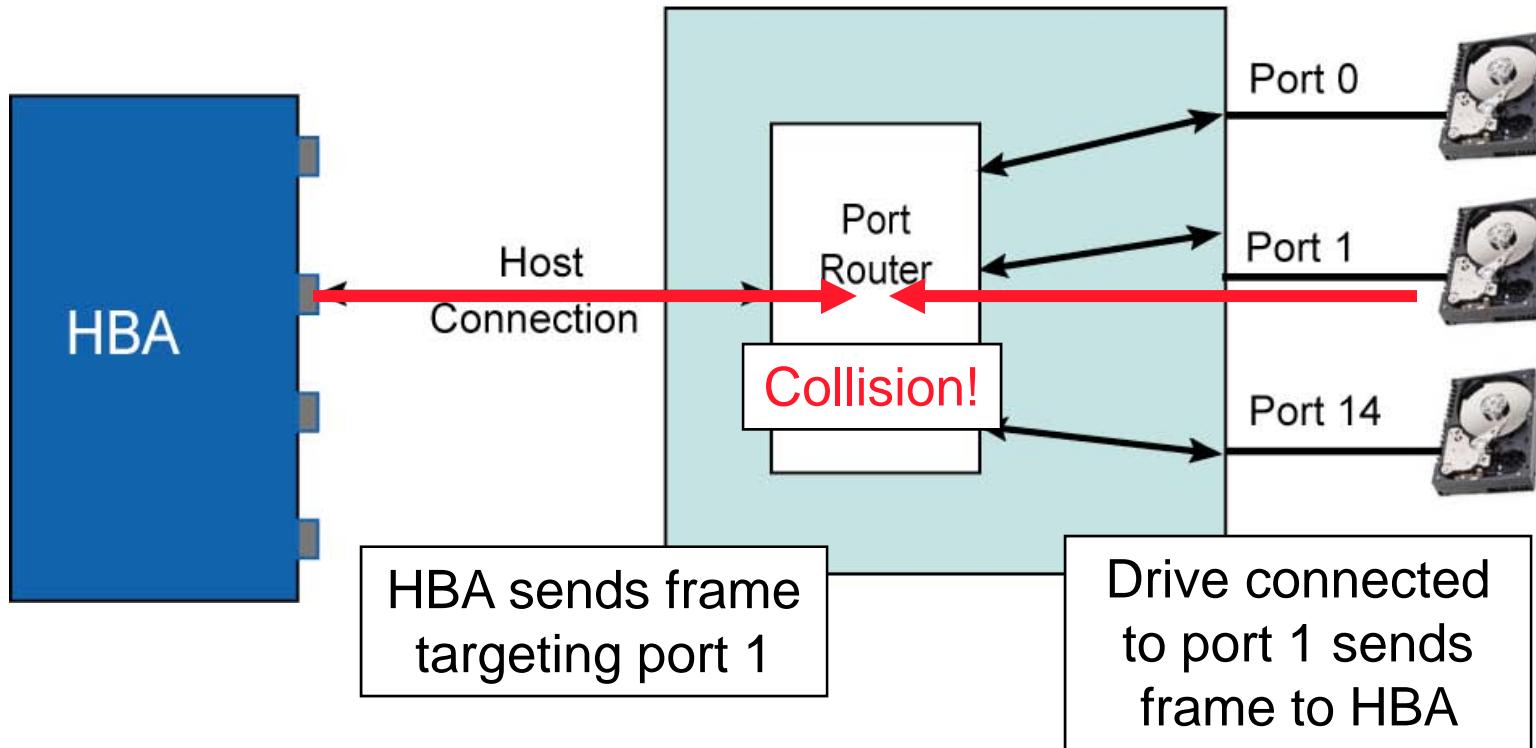
1. The drive arbitrates for link ownership and starts transmission of the FIS.
2. The PM receives the FIS from the drive and returns primitives in normal fashion, but it must not return R_OK until FIS transmission has completed to the HBA.
3. The PM begins to receive the FIS from the drive, but before sending the FIS on to the HBA, it loads the FIS's Port number field with the port number associated with the originating drive.
4. The PM arbitrates for HBA link ownership and begins transmitting the modified FIS to the HBA. (Note: To reduce latency and buffer space requirements, the PM is allowed to arbitrate for ownership of the HBA link prior to returning R_RDY to the drive.)
5. As the HBA receives the FIS it returns primitives to the PM in normal fashion. At this point the PM is involved in two separate and independent link transmissions of the same FIS.
6. During FIS reception the HBA must calculate and check CRC. The PM also delays delivery of the R_OK primitive to the drive until R_OK is received from the HBA.
7. During delivery of the modified FIS to the HBA, the PM must calculate and deliver a new CRC value.
8. When the HBA receives the FIS without error, it returns R_OK to the PM, which in turn sends R_OK to the drive, completing FIS delivery.

FIS Transmission - Drive to HBA

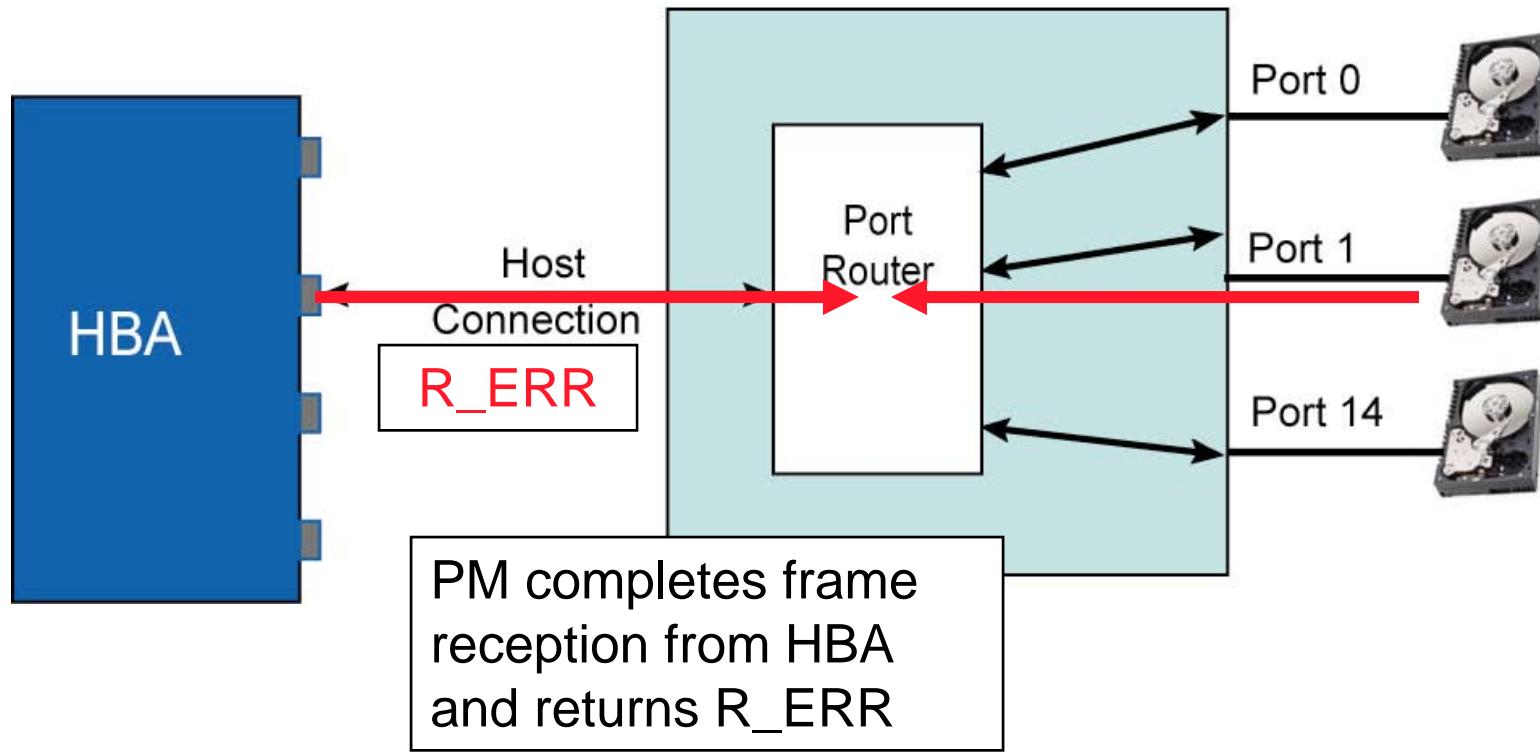
(Example Error Conditions)

- After receiving an X_RDY primitive from the device, the Port Multiplier determines if the X bit in the SError's Diag field is set. If set, the Port Multiplier must not return an R_RDY primitive to the drive until the X bit is cleared. The X bit indicates whether device connection status has changed. If the bit is set, a change has occurred but software may not have detected the change and has not cleared the bit.
- The PM must check the CRC of the FIS being received from the drive and if an error is detected, the PM must corrupt the CRC before forwarding it to the HBA.

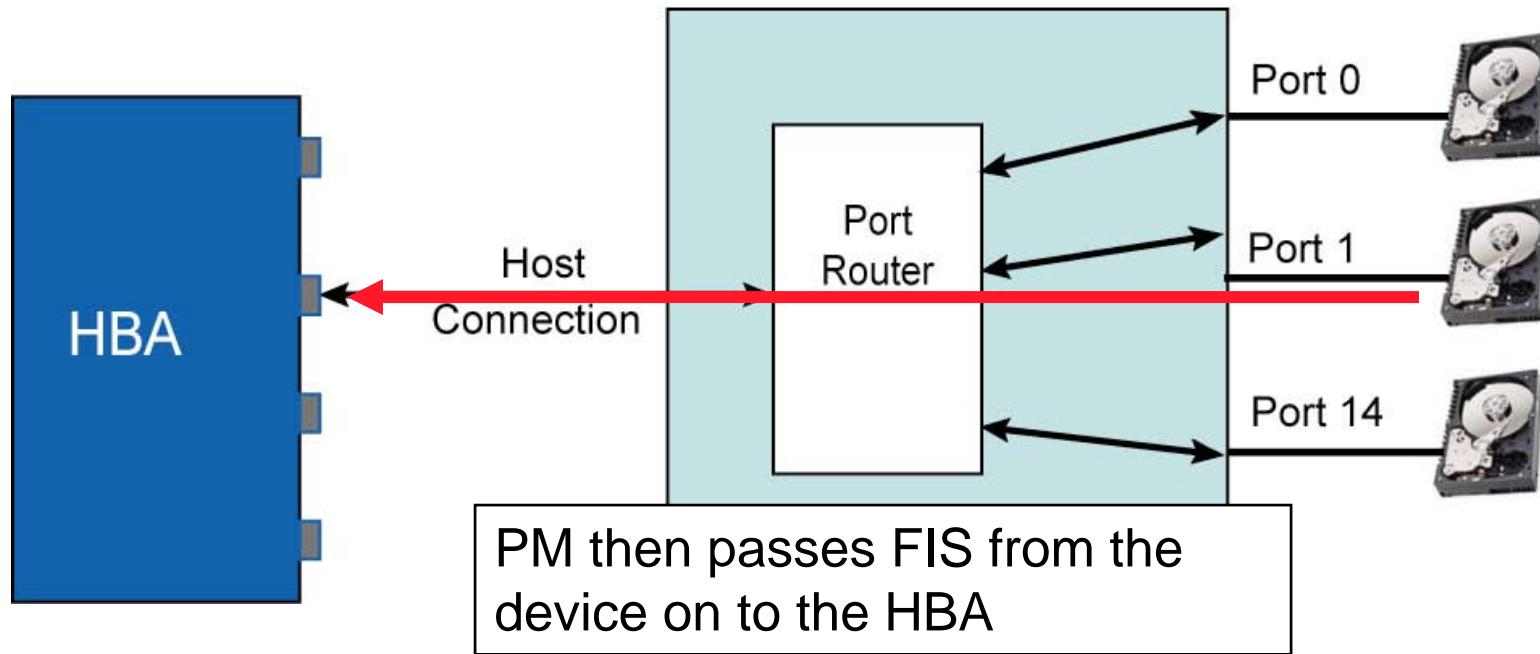
Collisions



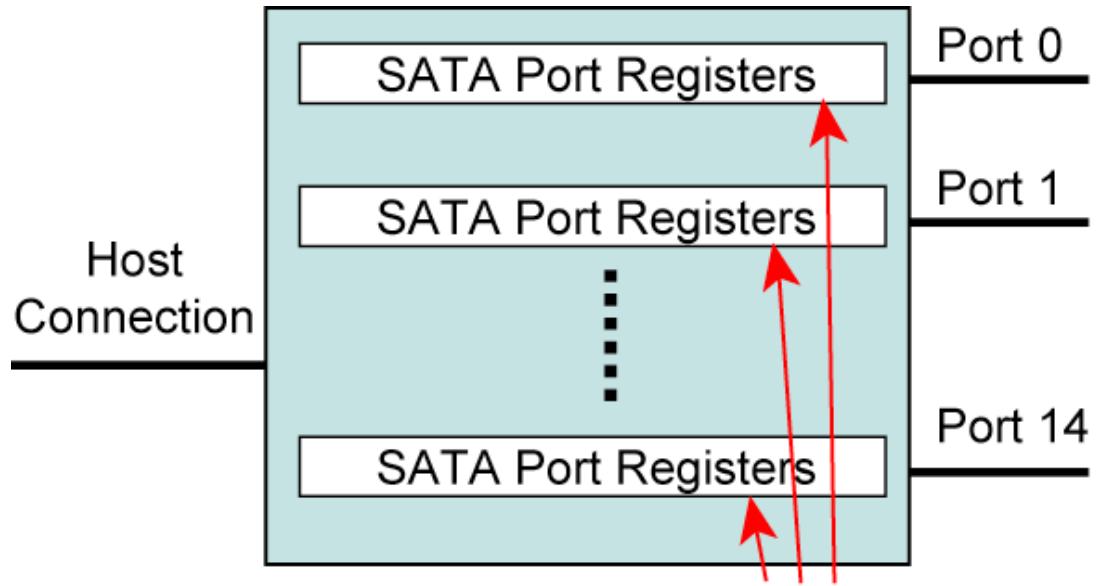
Collision Resolution



Collision Resolution



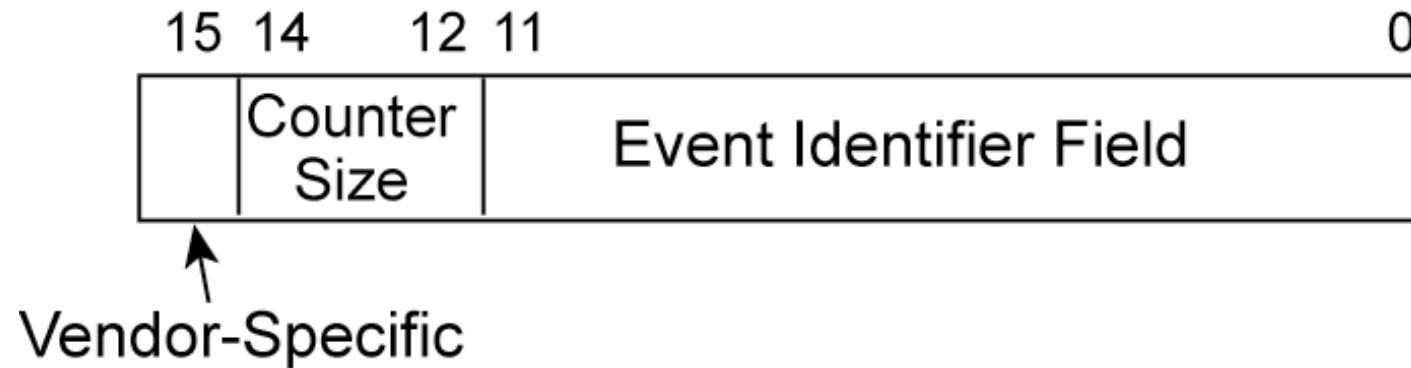
Port-Specific SCRs (PSCR[n])



PSCR[0]	SStatus Register
PSCR[1]	SError Register
PSCR[2]	SControl Register
PSCR[3]	SActive Register (not implemented)
PSCR[4]-PSCR[256]	Reserved
PSCR[257]-PSCR[2303]	PHY Event Counters
PSCR[2304]-PSCR[65535]	Reserved

Event Counters (Port Event Identifier Register)

Counter size can be 16, 32, 48, or 64 bits



Indicates counter is vendor specific and uses identifier range from 8000h to FFFFh.

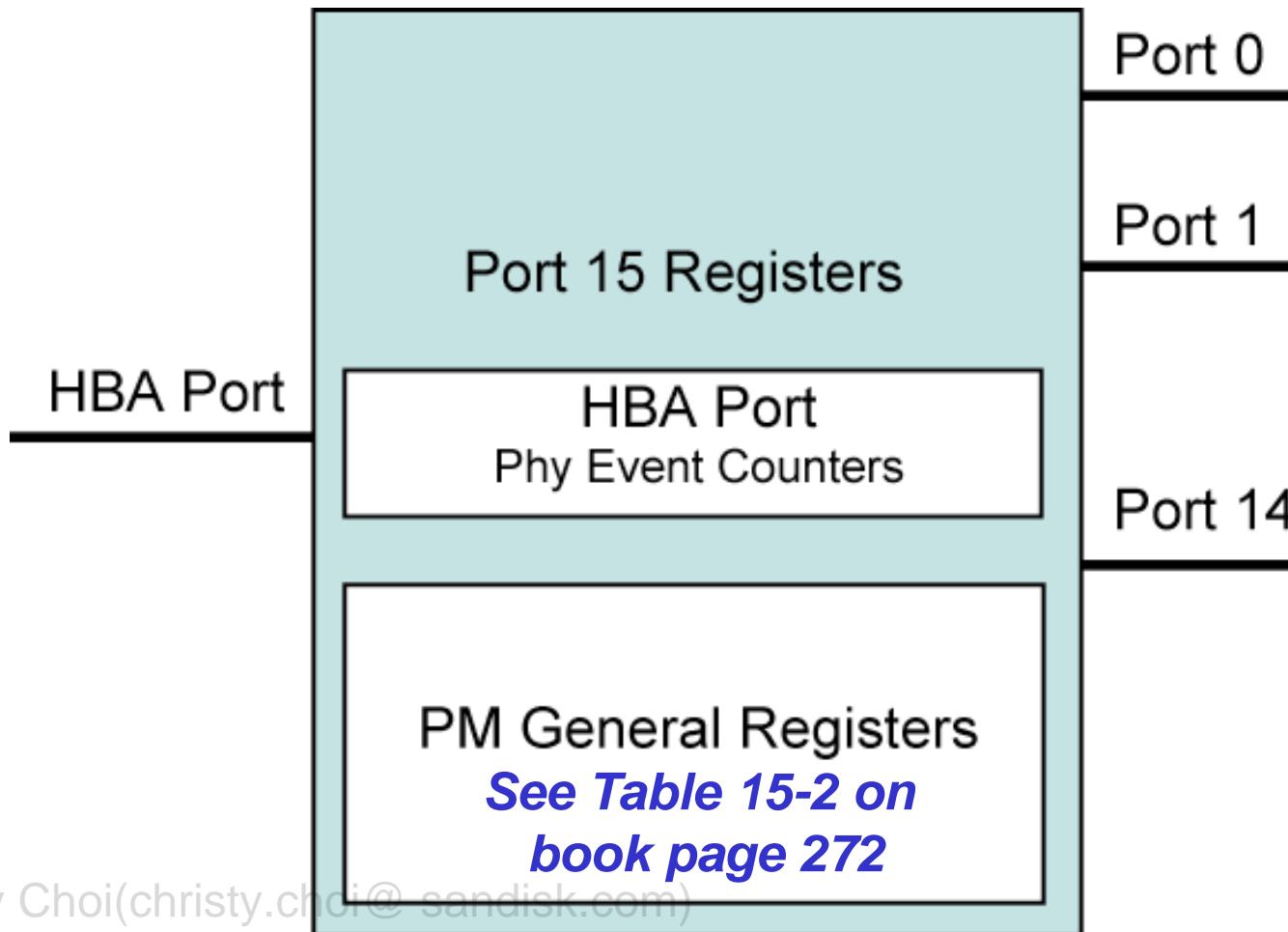
Refer to page 271 for additional details

Event Counters (Phy Event IDs)

Counter ID Bits (11:0)	Description
000h (Required)	Indicates end of counters in a given page
001h (Required)	Command failure and the ICRC error bit is set in the SError Register
002h	R_ERR response for Data FIS transmission
003h	R_ERR response from port for Device to Host Data FIS
004h	R_ERR response from port for Host to Device Data FIS
005h	R_ERR response from port for Non-Data FIS
006h	R_ERR response from port for Device to Host Non-Data FIS
007h	R_ERR response from port for Host to Device Non-Data FIS
008h	Device to Host Non-Data FIS retries
009h	Transition from Drive PHYRDY to Drive PHYRDYn
00Ah (Required)	Device to Host Register FIS sent due to COMRESET
00Bh	CRC Error detected in Host to Device FIS
00Dh	Non CRC Error detected in Host to Device FIS
00Fh	R_ERR response sent due to CRC error in Host to Device Data FIS
010h	R_ERR response sent due to Non-CRC error in Host to Device Data FIS
012h	R_ERR response sent due to CRC error in Host to Device Non-Data FIS
013h	R_ERR response sent due to Non-CRC error in Host to Device Non-Data FIS
C00h	PM only - R_ERR response due to Host to Device Non-Data FIS because of Collision
C01h	PM only - Device to Host Register FISes (Signature)
C02h	PM only - Corrupt CRC propagation, Device to Host FISes

Local PM Port [15]

Port 15 accesses the set of internal registers for the PM



PM Initialization

- Following reset, the Port Multiplier establishes connection with port zero only.
- During initialization, host system will detect the device attached to port zero and complete initialization in normal fashion when a Register FIS is returned from the drive.
- If no device is attached to port zero, a Register FIS will not be returned, but the HBA will still detect a device attached (the PM, though it's not yet recognized as such).

Detecting PM Presence

To determine Port Multiplier presence, host performs the following procedure:

- Determine whether communication was established on the host port by checking the host's SStatus register.
- If a device is present, issue a software reset with the PM Port field set to the local port (15), then check the signature value returned in the RegDev to Host FIS
- If the signature matches the Port Multiplier Signature, a Port Multiplier is attached (see next slide).
- If the signature is not that of a PM, the host may proceed with the normal initialization sequence for that device type

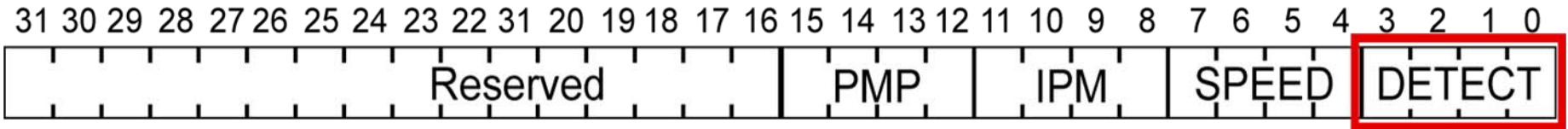
Note: The host must not rely on a device being attached to device port 0h to determine whether a Port Multiplier is present.

Port Multiplier Signature

Data	
Error	00h
Sector Count	01h
Sector Number	01h
Cylinder Low	69h
Cylinder High	96h
Device	00h
Status	

Configuring Device Ports (1 to n)

- All ports other than zero are disabled by default
- Software enables each port by writing a 1h to the DET field of the port's SControl register triggering Reset & Init for the link
- At the end of the link initialization sequence the attached drive will return a Register FIS



DETECT - Controls Host Adapter device detection and Initialization

0000 = No action requested

0001 = This value when written triggers the interface to be reset and initialized. This state is persistent until bit zero is cleared by software.

0100 = Disables interface and places PHY offline

Other values reserved

Accessing the Port Multiplier Registers

New commands used to access the PM Registers:

- Read PM Registers Command
- Write PM Registers Command

Read PM Command

(Contents of RegFis Host-to-Device)

Register Name	7	6	5	4	3	2	1	0
Features	Port Multiplier Register Number [7:0]							
Features (exp)	Port Multiplier Register Number [15:8]							
Sector Count	Reserved							
Sector Count (exp)	Reserved							
LBA Low	Reserved							
LBA Low (exp)	Reserved							
LBA Middle	Reserved							
LBA Middle (exp)	Reserved							
LBA High	Reserved							
LBA High (exp)	Reserved							
Device	na	RS1	RS2	na	Port Number			
Command	E4h							

See Table 15-3 for field definitions, book page 277

Read PM Command

(Contents of RegFis Device-to-Host)

Register Name	7	6	5	4	3	2	1	0
Error					0			
Sector Count								PM Register Value [7:0]
Sector Count (exp)								PM Register Value [39:32]
LBA Low								PM Register Value [15:8]
LBA Low (exp)								PM Register Value [47:40]
LBA Middle								PM Register Value [23:16]
LBA Middle (exp)								PM Register Value [55:48]
LBA High								PM Register Value [31:24]
LBA High (exp)								PM Register Value [63:56]
Device								Reserved
Status	BSY	DRDY=1	DF	na	DRQ	0	0	ERR=0

Read PM Command (RegFis Device-to-Host with Error Status)

Register Name	7	6	5	4	3	2	1	0
Error			Reserved			Abrt	Reg	Port
Sector Count				Reserved				
Sector Count (exp)				Reserved				
LBA Low				Reserved				
LBA Low (exp)				Reserved				
LBA Middle				Reserved				
LBA Middle (exp)				Reserved				
LBA High				Reserved				
LBA High (exp)				Reserved				
Device				Reserved				
Status	BSY	DRDY=1	DF	na	DRQ	0	0	ERR=1

Write PM Command (Contents of RegFis Host-to-Device)

Register Name	7	6	5	4	3	2	1	0
Features	Port Multiplier Register Number [7:0]							
Features (exp)	Port Multiplier Register Number [15:8]							
Sector Count	PM Register Value [7:0]							
Sector Count (exp)	PM Register Value [39:32]							
LBA Low	PM Register Value [15:8]							
LBA Low (exp)	PM Register Value [47:40]							
LBA Middle	PM Register Value [23:16]							
LBA Middle (exp)	PM Register Value [55:48]							
LBA High	PM Register Value [31:24]							
LBA High (exp)	PM Register Value [63:56]							
Device	na	RS1	na	na	Port Number			
Command	E8h							

See **Table 15-4 for field definitions, book page 280**

Write PM Command (Contents of RegFis Device-to-Host)

Register Name	7	6	5	4	3	2	1	0
Error					0			
Sector Count					Reserved			
Sector Count (exp)					Reserved			
LBA Low					Reserved			
LBA Low (exp)					Reserved			
LBA Middle					Reserved			
LBA Middle (exp)					Reserved			
LBA High					Reserved			
LBA High (exp)					Reserved			
Device					Reserved			
Status	BSY	DRDY=1	DF	na	DRQ	0	0	ERR=0

Write PM Command (RegFis Device-to-Host with Error Status)

Register Name	7	6	5	4	3	2	1	0
Error			Reserved			Abrt	Reg	Port
Sector Count				Reserved				
Sector Count (exp)				Reserved				
LBA Low				Reserved				
LBA Low (exp)				Reserved				
LBA Middle				Reserved				
LBA Middle (exp)				Reserved				
LBA High				Reserved				
LBA High (exp)				Reserved				
Device				Reserved				
Status	BSY	DRDY=1	DF	na	DRQ	0	0	ERR=1

Hot Plug Support

Port Multipliers must support Hot Plug capability.

See Hot Plug discussion on slide 466

Staggered Spinup/Drive Activity

Pin 11 of the SATA Power connector can be used for either:

- disabling staggered drive spin-up
- drive activity indicator signal

Port Selectors

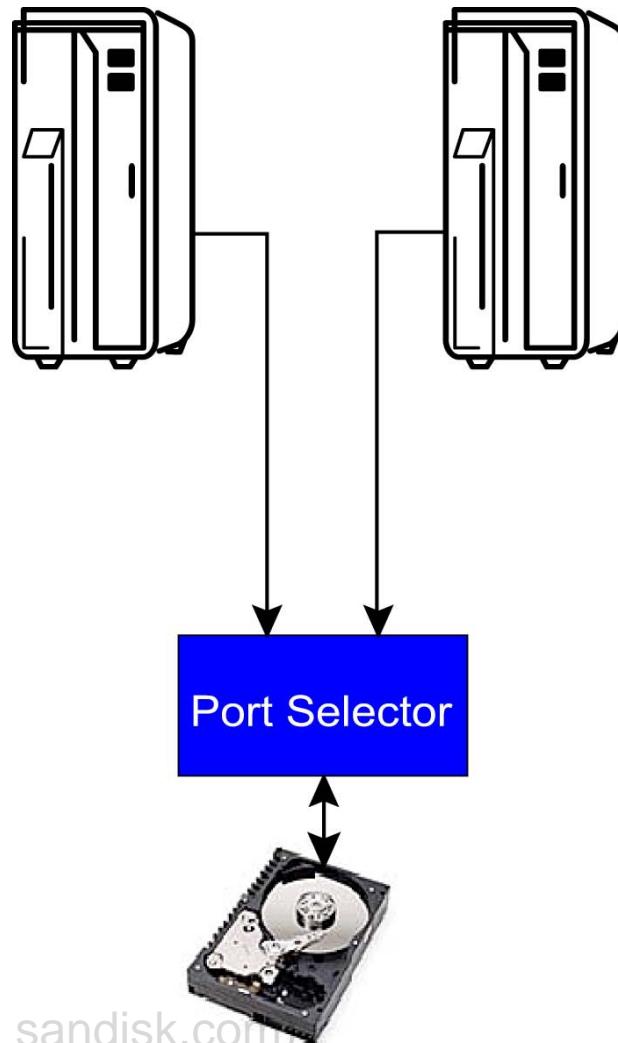
Christy Choi(christy.choi@sandisk.com)

Do Not Distribute



Port Selector

Allows a device to talk with two hosts



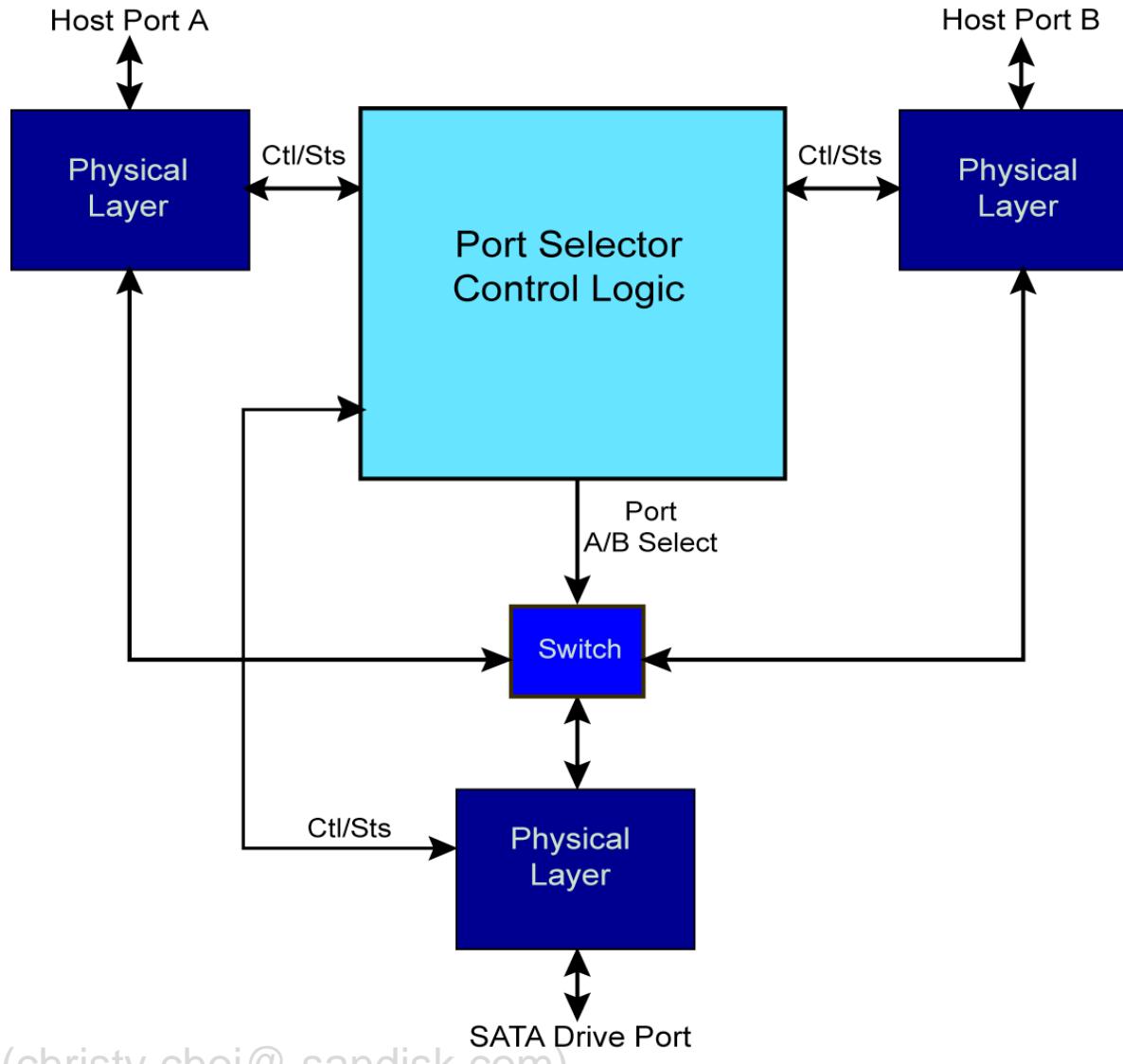
Port Selector Design Requirements

- No modifications to Serial ATA 1.0a devices required
- No hardware modification required for Host Adapters
- New software required
- No new primitives needed
- No new FIS types needed
- A port selector should not need full-function link and transport layers
- Host port connections limited to two
- Only one active port at a time
- Port selectors cannot be cascaded

Port Selector Functions

- Port Selectors participate in the OOB sequence
- The HBA and Port Selector may optionally support a mechanism to detect the presence of a Port Selector based on OOB signaling.
- Implement a mechanism to allow software to change the active port (e.g., protocol-based method of port selection or side-band signal).
- Port Selectors repeat frame and OOB traffic between the active host port and the drive.

Port Selector Functions

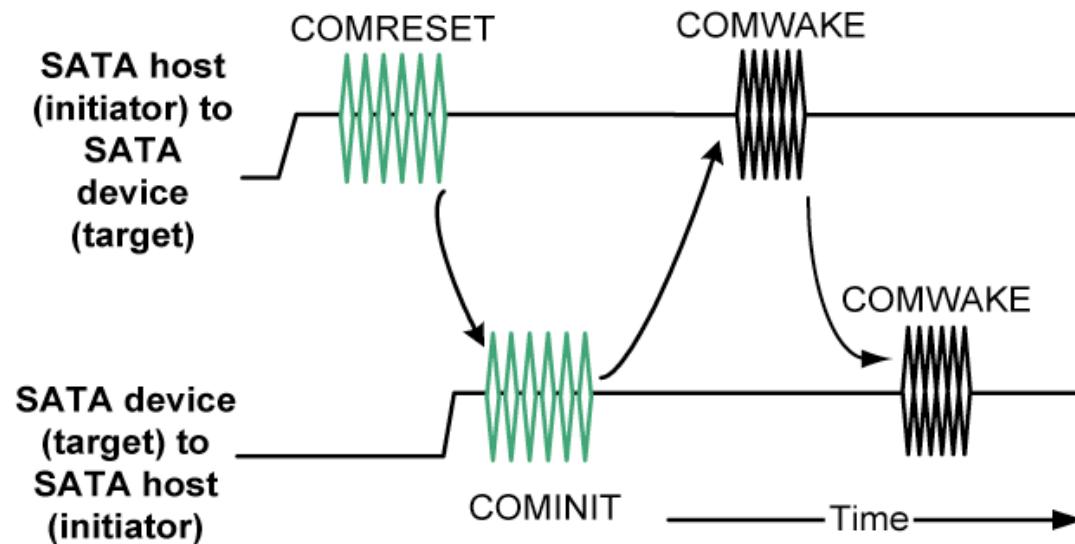


Detecting Port Selector Presence

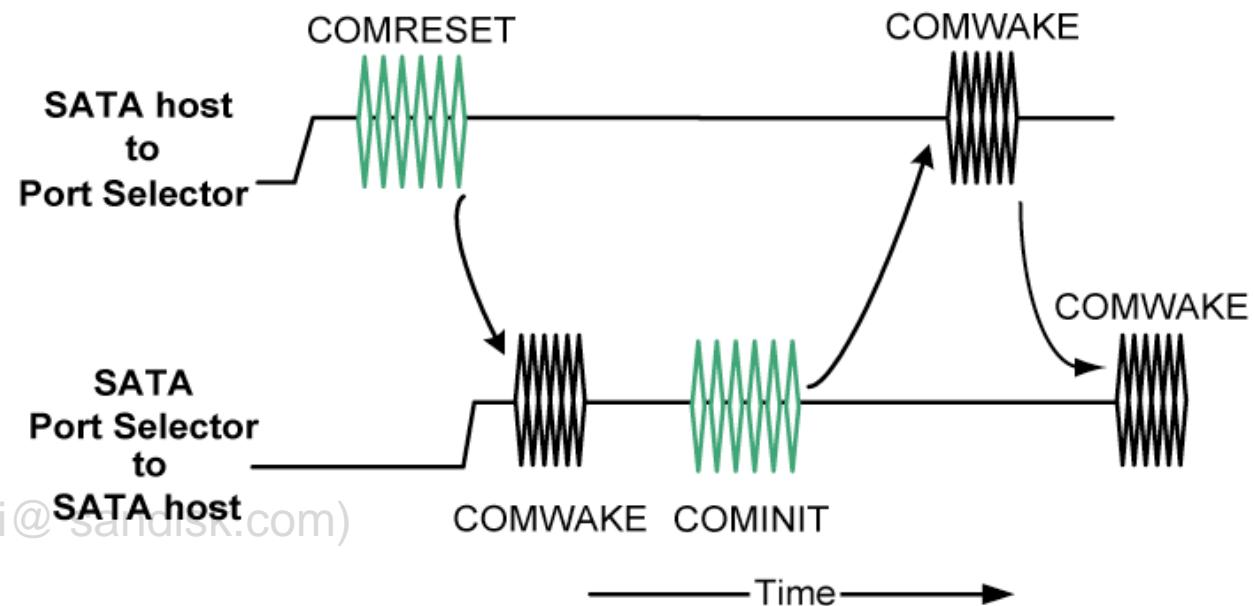
- Presence detection is the ability for a host to detect that a Port Selector is present on a port.
- Presence detection capabilities are defined for Port Selectors using protocol-based port selection only.
- The Port Selector signals its presence by returning COMWAKE after detecting COMRESET from Host before sending the expected COMINIT.

Port Selector Detection - optional

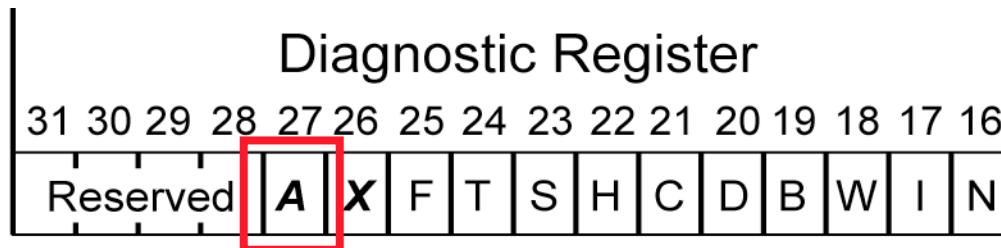
Normal OOB Sequence



Port Selector Presence OOB Sequence



Detection Port Selector Presence



X = Exchanged

- 0 = No Device presence change detected since bit last cleared
- 1 = Device presence has changed since this bit was cleared

A = Port Selector Presence Detected

- 0 = No Port Selector detected since bit last cleared
- 1 = Port Selector presence detected

All other bits are the same as defined by 1.0a

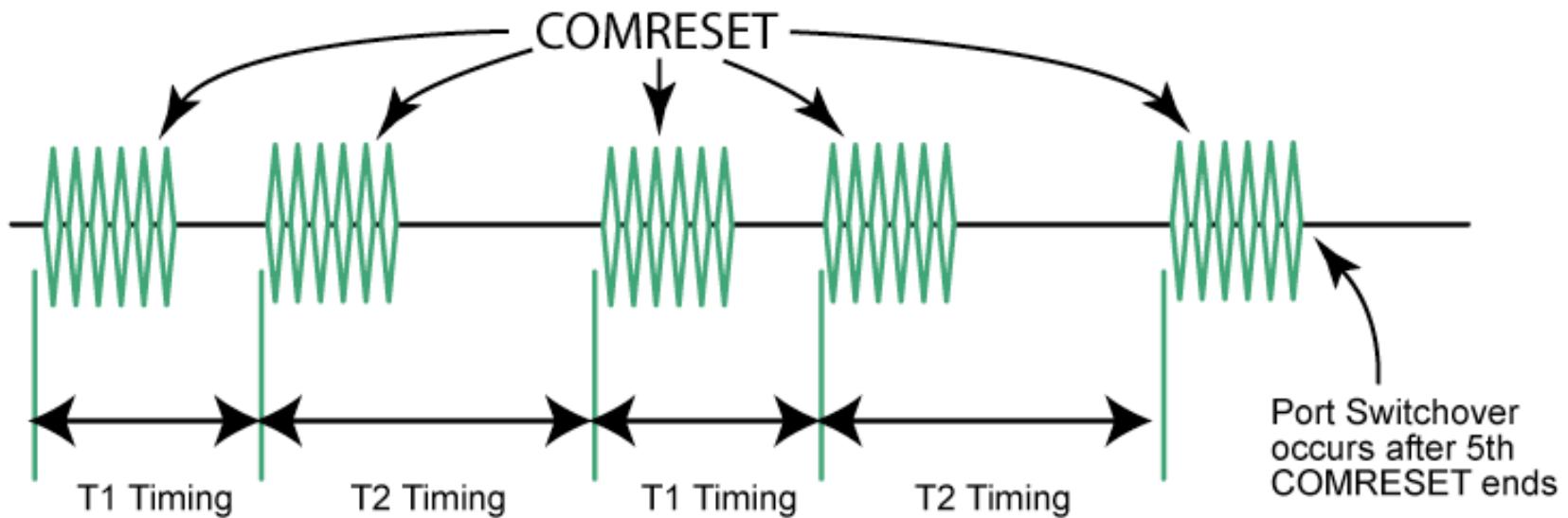
Selecting Active Host Port

Two methods are available to select the active host port:

- Protocol-based port selection using OOB signaling
- Side-Band selection - External signals can be used but not defined by the specification

Protocol Switching

The port selection signal is two back-to-back COMRESET timing sequences issued by a host adapter as illustrated below.



	Min	Nom	Max	Description
T1	1.6 ms	2.0 ms	2.4 ms	COMRESET delay for first selection event
T2	7.6 ms	8.0 ms	8.4 ms	COMRESET delay for second selection event

Enclosure Services



Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

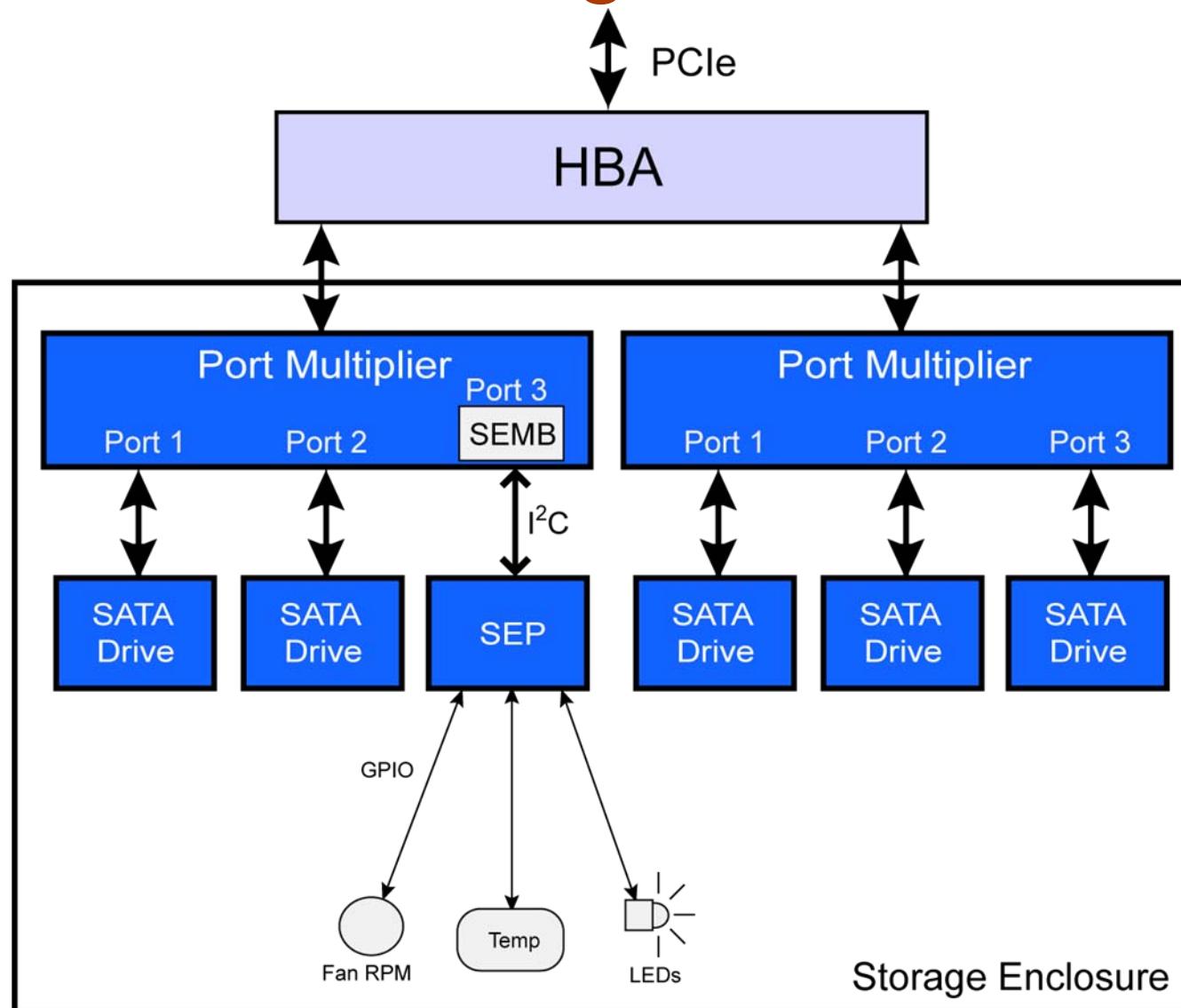
Enclosure Management Support

The SATA II specifications provide support for enclosure management features that are compatible with two industry standards:

- SAF-TE (SCSI Accessed Fault-Tolerant Enclosures)
- SES (SCSI Enclosure Services)

Note: This chapter does not detail the protocols and additional information defined by these standards.

Enclosure Management Elements



Enclosure Management Elements

- SEP (Storage Enclosure Processor) — interfaces with various sensors and indicators within a storage enclosure subsystem and responds to enclosure service commands. The required interface to the SEP is the I²C bus, over which the commands and data are passed.
- SEMB (Storage Enclosure Management Bridge) — the interface (bridge) between the SATA host and the Storage Enclosure Processor (SEP). The SEMB converts the SATA commands into I²C packets that are then delivered to the SEP.

SEP Commands

SEP commands are sent/received using the standard ATA Command Block Register set (or Serial ATA Register FIS) using the READ SEP and WRITE SEP commands associated with this interface.

Register Name	7	6	5	4	3	2	1	0
Features	SEP_CMD							
Features (exp)	Reserved							
Sector Count	LEN							
Sector Count (exp)	Reserved							
LBA Low	CMD_TYPE (80h or 82h)							
LBA Low (exp)	Reserved							
LBA Middle	Reserved							
LBA Middle (exp)	Reserved							
LBA High	Reserved							
LBA High (exp)	Reserved							
Device	Reserved	0	Reserved					
Command	SEP_ATTN (67h)							

Register Name	7	6	5	4	3	2	1	0
Features	SEP_CMD							
Features (exp)	Reserved							
Sector Count	LEN							
Sector Count (exp)	Reserved							
LBA Low	CMD_TYPE (00h or 02h)							
LBA Low (exp)	Reserved							
LBA Middle	Reserved							
LBA Middle (exp)	Reserved							
LBA High	Reserved							
LBA High (exp)	Reserved							
Device	Reserved	0	Reserved					
Command	SEP_ATTN (67h)							

Read SEP Command

Christy Choi(christy.choi@sandisk.com)
Do Not Distribute

Copyright Mindshare Inc, 2009

Write SEP Command

SEP Command Fields

Shadow Reg	Field Name	Description
Features	SEP_CMD	SEP Command — Defines parameters related to the SES and SAF_TE command protocol.
Sector Count	LEN	Length — defines the length of the data payload
LBA Low	CMD_TYPE	Command Type — encoding as follows: Upper nibble bits (7:4): 0 = SEP to Host Data Transfer 8 = Host to SEP Data Transfer Lower nibble bits (3:0): 0 = SAF-TE protocol used 2 = SES protocol used
Device	NA	Value specified is listed in Figure 17-2
Command	SEP_ATTN	SEP Attention Command (67h) — indicates that the SEMB that a SEP command is being issued

Write/Read Command

- Access enclosure information, such as:
 - Number of fans, power supplies, device slots, temperature sensors, etc.
 - Status of items listed above: fan status and speed, thermal alarms
 - Usage statistics: power-on minutes, power cycles,
 - Set power supply states
 - Number of device insertions
 - Slot status

Inquiry Command

- INQUIRY command only specifies LUN and number of bytes host wants to have returned
- Response has several fields, among them:
 - Vendor ID – 8 byte ASCII string
 - Product ID – 16 byte ASCII string
 - Enclosure ID – 58-bit ID number
- Combination of these 3 fields intended to uniquely ID any peripheral from any manufacturer

Part 5

Physical layer Details

Christy Choi(christy.choi@ sandisk.com)

Do Not Distribute



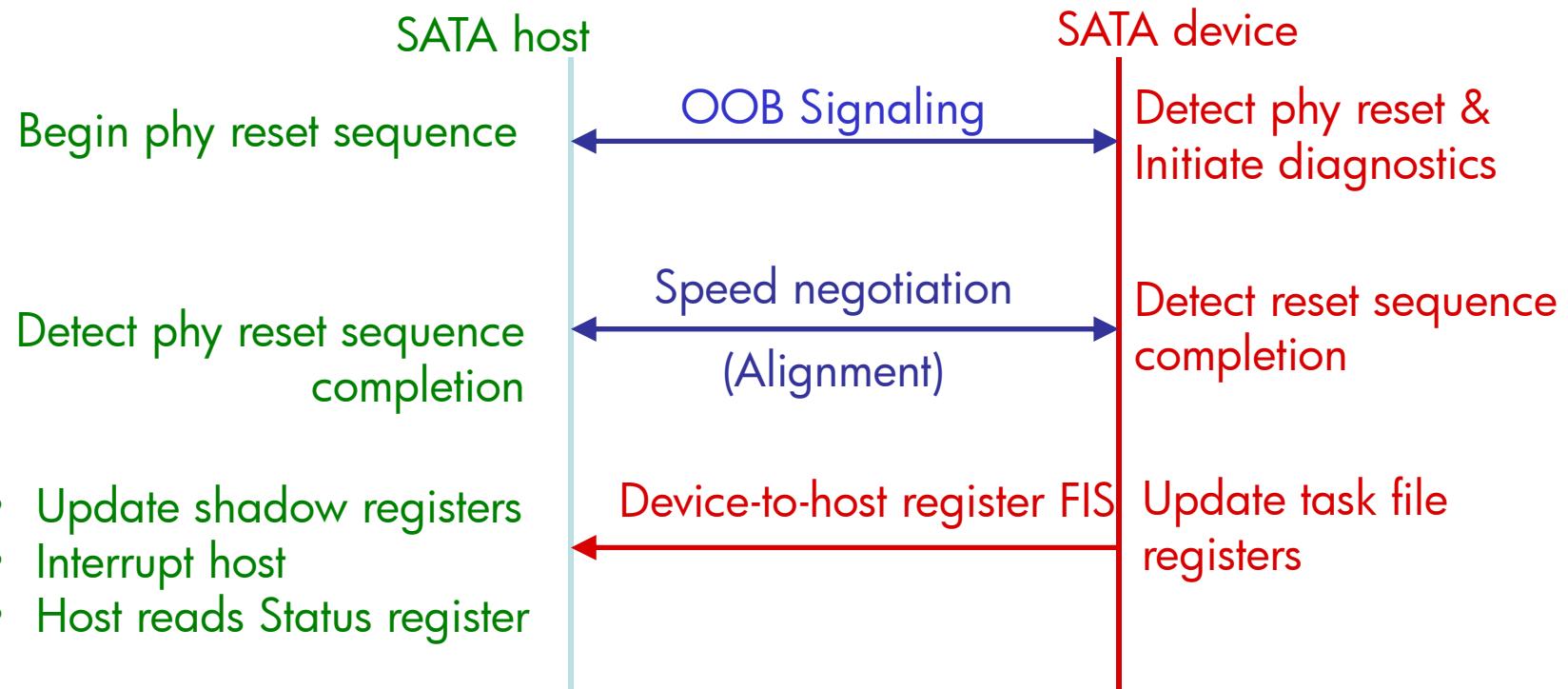
SATA Initialization

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

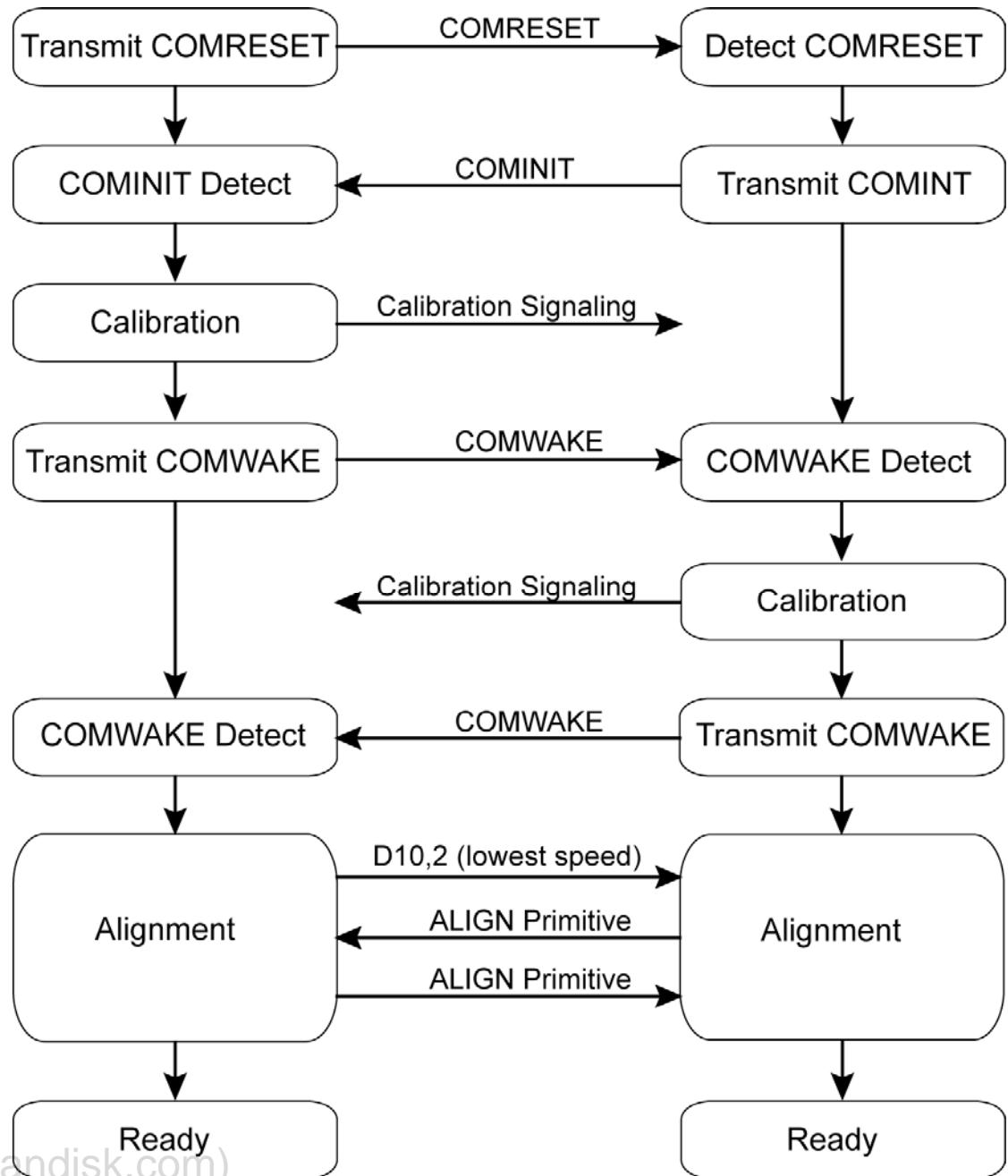


SATA Reset Sequence



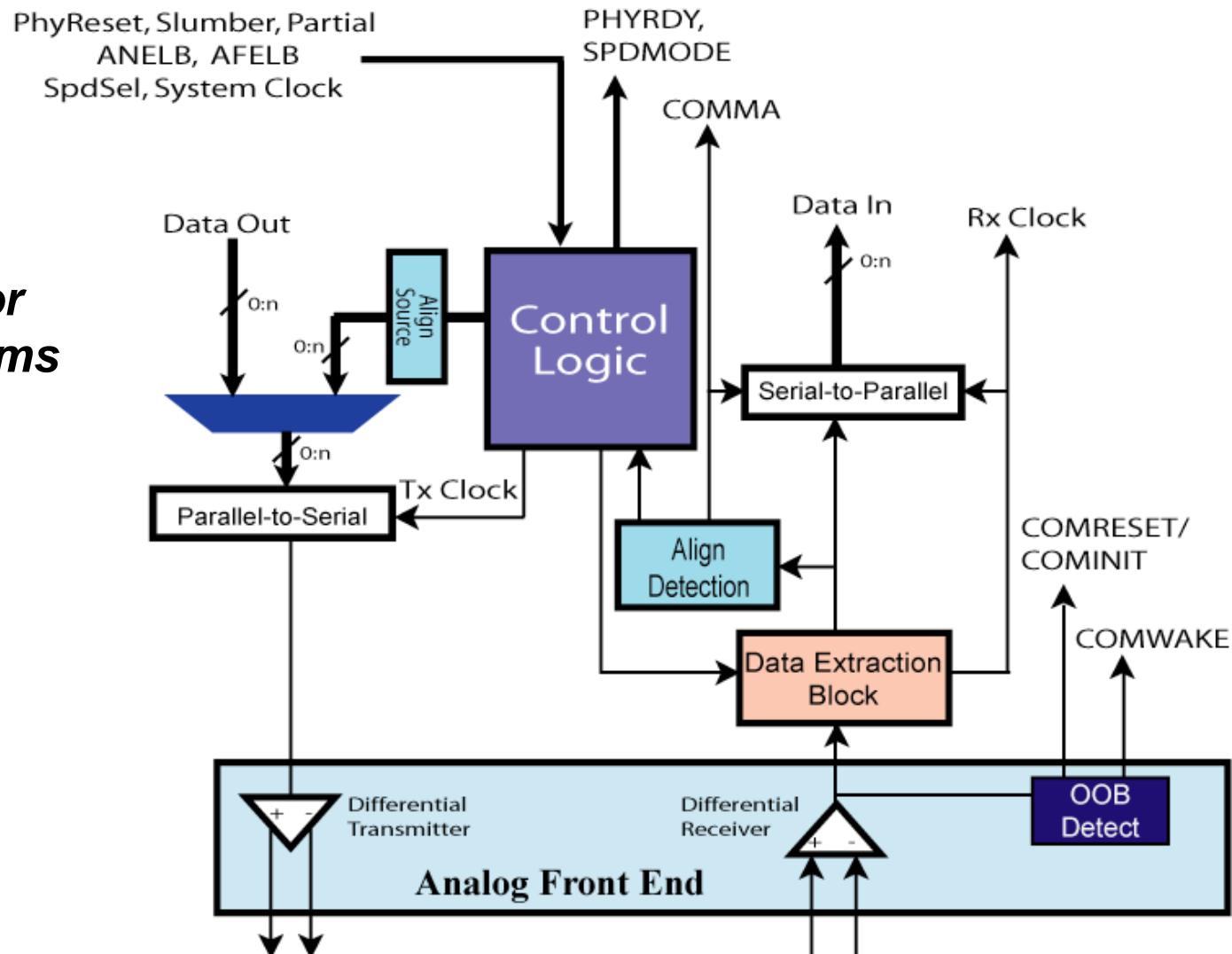
After completing a phy reset sequence, device runs diagnostics and transmits a FIS with results

OOB & Alignment



Physical Layer Block Diagram

See page 306 for definition of terms

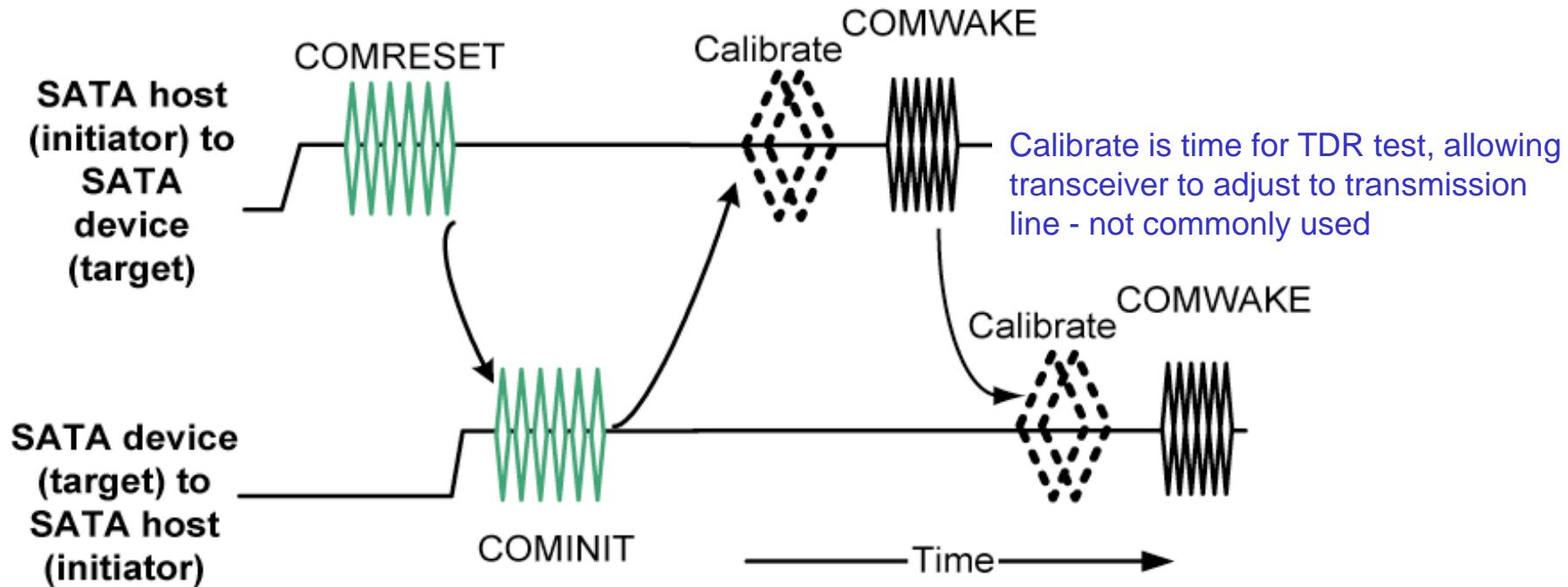


OOB (Out of Band) Signaling

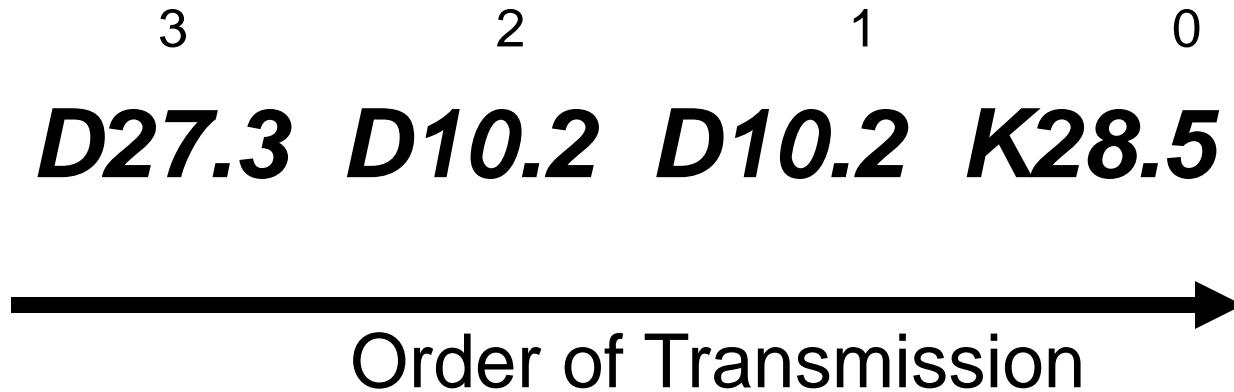
- Low-speed signal patterns detected by the phy that do not appear in normal data streams and are therefore out-of-band.
- Pattern of idle times and burst times, distinguished by length of time between idles
 - Idle time (and negation time)
 - Differential voltage = 0 V
 - No transitions (DC idle)
 - Burst time
 - Transmitted as a burst of ALIGN primitives
 - Received as activity (characters are irrelevant)
- Designed to be detectable by analog squelch detection logic

Initial SATA OOB Sequence

- Send and receive COMINIT
- Host then sends COMWAKE

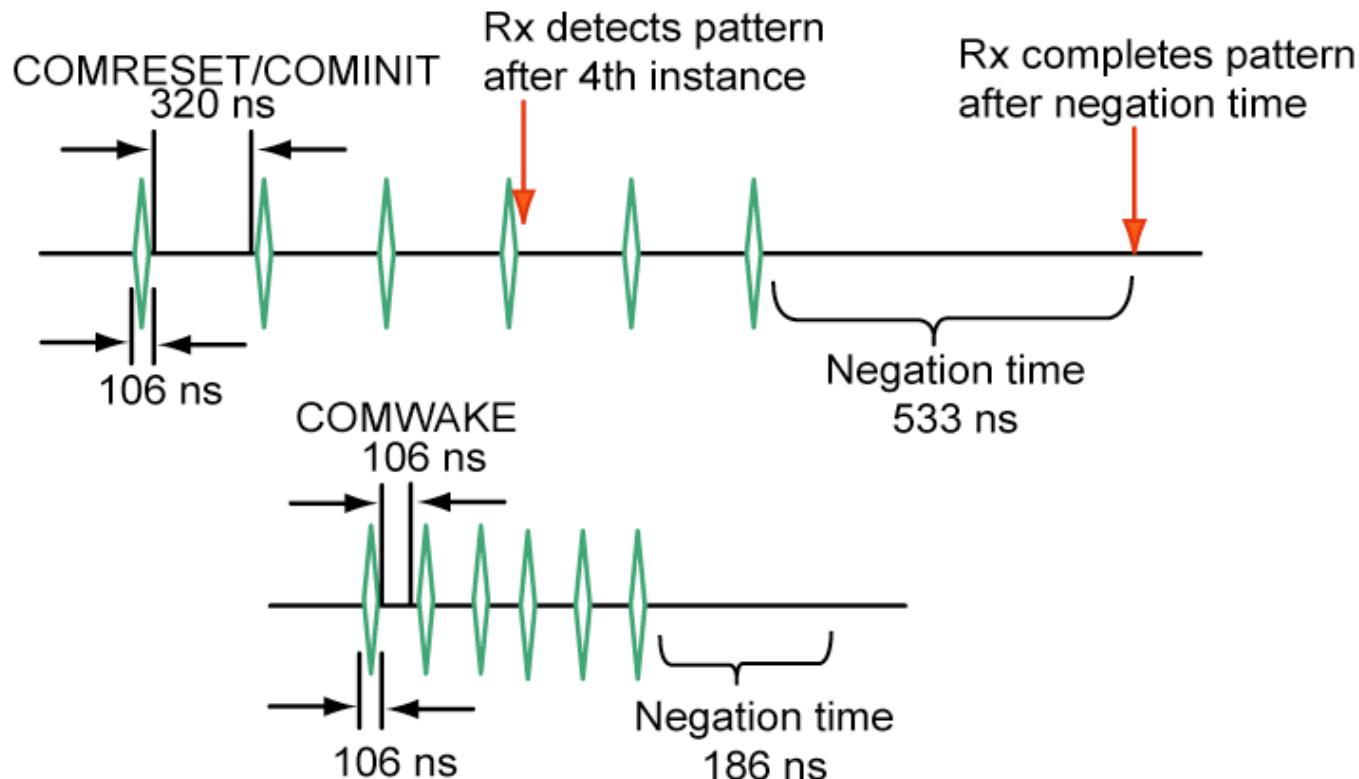


Align Primitive



(rd+)	(rd-)		
1100000101	0011111010	Align1	(K28.5)
0101010101	0101010101	Align2	(D10.2)
0101010101	0101010101	Align3	(D10.2)
1101100011	0010011100	Align4	(D27.3)

OOB Transmission



OOB Signal	Burst Time	Idle Time	Negation Time
COMINIT/COMRESET	160 OOB (103.47 - 109.87ns)	480 OOB (310.40 - 329.60 ns)	800 OOB (517.33 - 549.34 ns)
COMWAKE	160 OOB (103.47 - 109.87ns)	160 OOB (103.47 - 109.87ns)	280 OOB (181.07- 192.60 ns)

OOB Reception

Receiver OOB Idle Detection Timing

OOB Signal	Burst Time	Idle Time	Negation Time
COMWAKE	160 OOB UI (103.47 - 109.87ns)	160 OOBI (103.47 - 109.87ns)	280 OOBI (181.07- 192.60 ns)
COMINIT/ COMRESET	160 OOBI (103.47 - 109.87ns)	480 OOBI (310.40 - 329.60 ns)	800 OOBI (517.33 - 549.34 ns)

Receiver OOB Negation Detection Timing

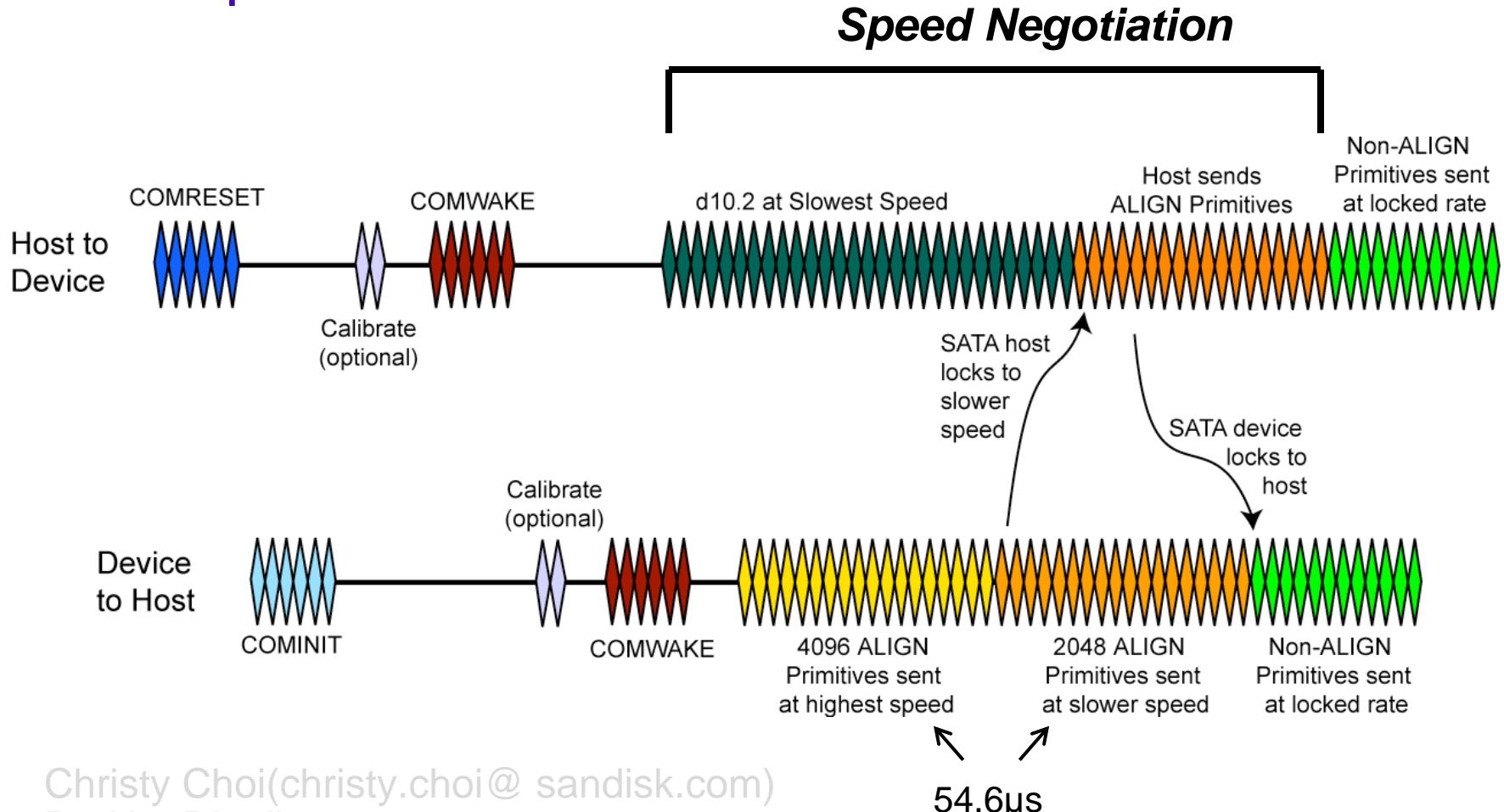
OOB Signal	Shall Detect
COMWAKE	> 175 ns
COMINIT/ COMRESET	> 525 ns

SATA Speed Negotiation

- Fast to slow progression
 - SATA target device sends ALIGN primitives at the fastest supported rate
 - Waits for host to reply with ALIGNs
 - If no reply after sending 2048 (i.e., the host doesn't support this speed), step down to next slower speed and try again

SATA Speed Negotiation

When host replies with ALIGNs, it has locked at the current frequency and negotiation is complete



Asynchronous Signal Recovery

- When communication is lost between the Host and SATA device (i.e., no signals are received), a phy that supports asynchronous signal recovery will attempt to re-establish communication.
- Phys use a variable named the RetryInterval to determine the rate at which signal recovery polling is attempted (10 ms minimum).
- This condition can occur when the cable is unplugged during normal operation and then replaced while power is still present.

Asynchronous Signal Recovery (HBA Not Receiving)

- When HBA in the Ready state but no longer detects a signal, it attempts to recover communications by sending COMRESET.
- The HBA then awaits receipt of COMINIT from the device and completes initialization
- If COMINIT has not been received when the RetryInterval elapses, the phy transitions back to the Reset state causing COMRESET to be sent again.
- This process continues until communications are re-established

Asynchronous Signal Recovery (HBA Receives Unsolicited COMINIT)

- When the HBA detects an unsolicited COMINIT (i.e., COMINIT received in any state, except for AwaitCOMINIT) the HBA Phy transitions to the Reset state.
- This causes the host phy to deliver a COMRESET, thereby initiating the Initialization sequence.

Asynchronous Signal Recovery (Device Phy Sends COMINIT)

- SATA devices also attempt signal recovery when the received signal disappears.
- Devices first transition to the Error state and wait for the RetryInterval to elapse.
- This causes the Phy to enter its Reset state at which time the drive sends a COMINIT and awaits a response from the Host.

Software Initialization

When Reset & Link Initialization complete the drive sends a Register FIS to the HBA, reporting:

- results of the drive's internal diagnostics
- the device signature

ATA Devices

Data	
Error	01h
Sector Count	01h
Sector Number	01h
Cylinder Low	00h
Cylinder High	00h
Device	00h
Status	00h-70h

ATAPI Devices

Data	
Error	01h
Sector Count	01h
Sector Number	01h
Cylinder Low	14h
Cylinder High	EBh
Device	00h
Status	00h

Software Initialization

- Software can read the contents of the shadow registers in the host to check the device's power-on signature.
- Next, system software issues the Identify Device command and parses the information to determine device capabilities
- Finally, software may configure the device using the Set Features command.

SATA II Initialization Issue

The book chapter includes the initialization information for Port Multipliers and Port Selectors here for ease of reference, but we don't need to repeat it here.

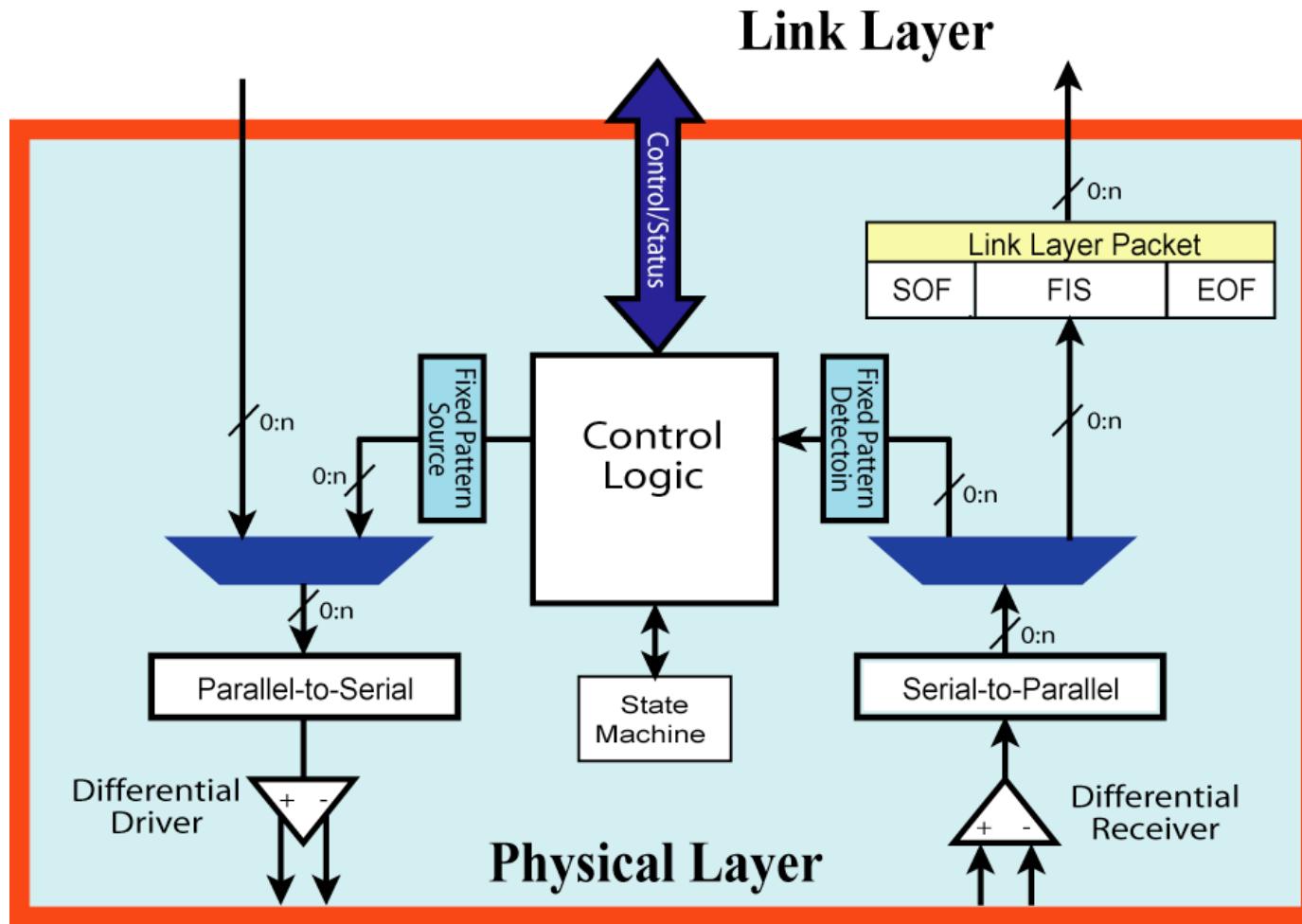
Physical layer - Electrical characteristics

Christy Choi(christy.choi@sandisk.com)

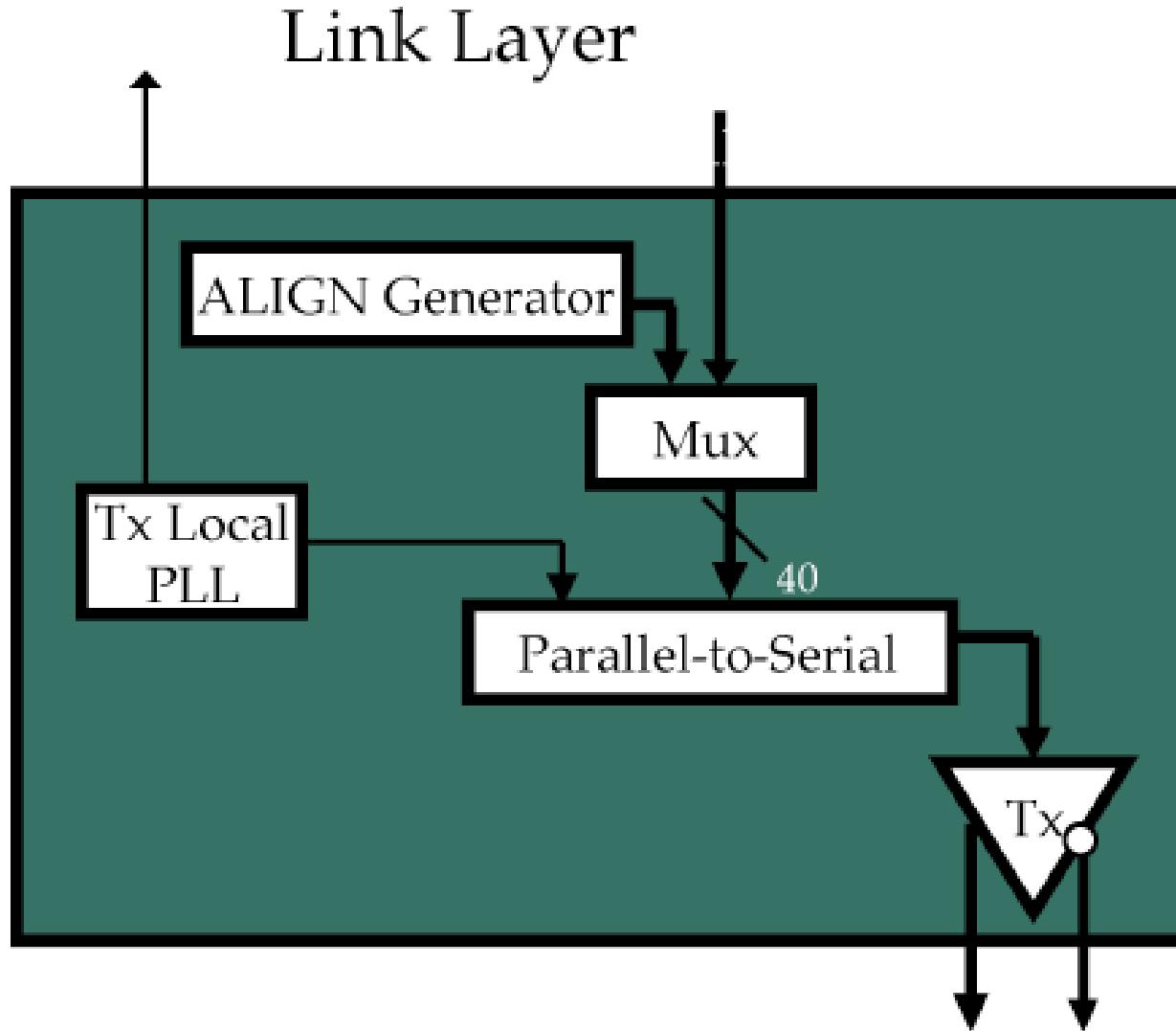
Do Not Distribute



Physical Layer Functions



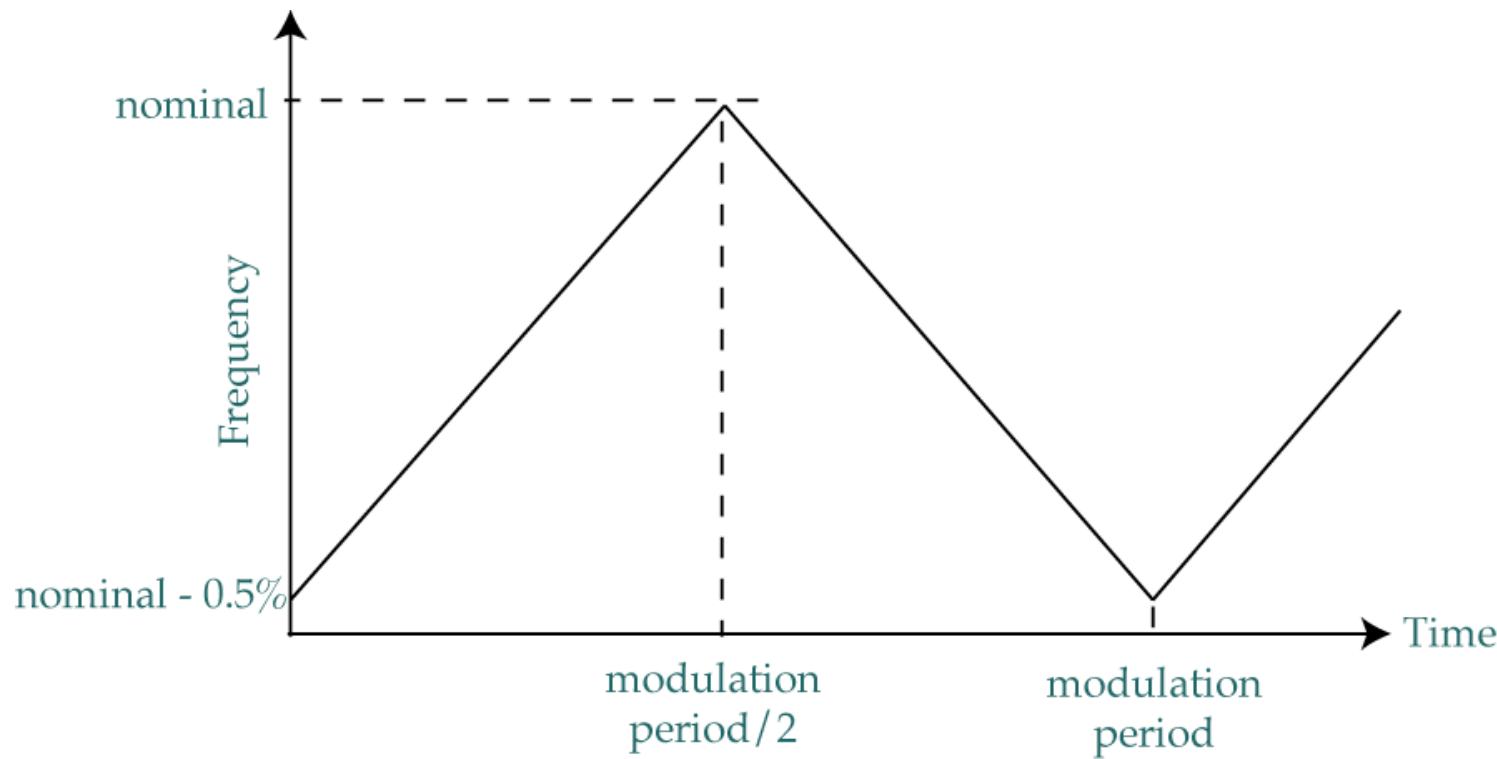
Physical Layer - Transmit



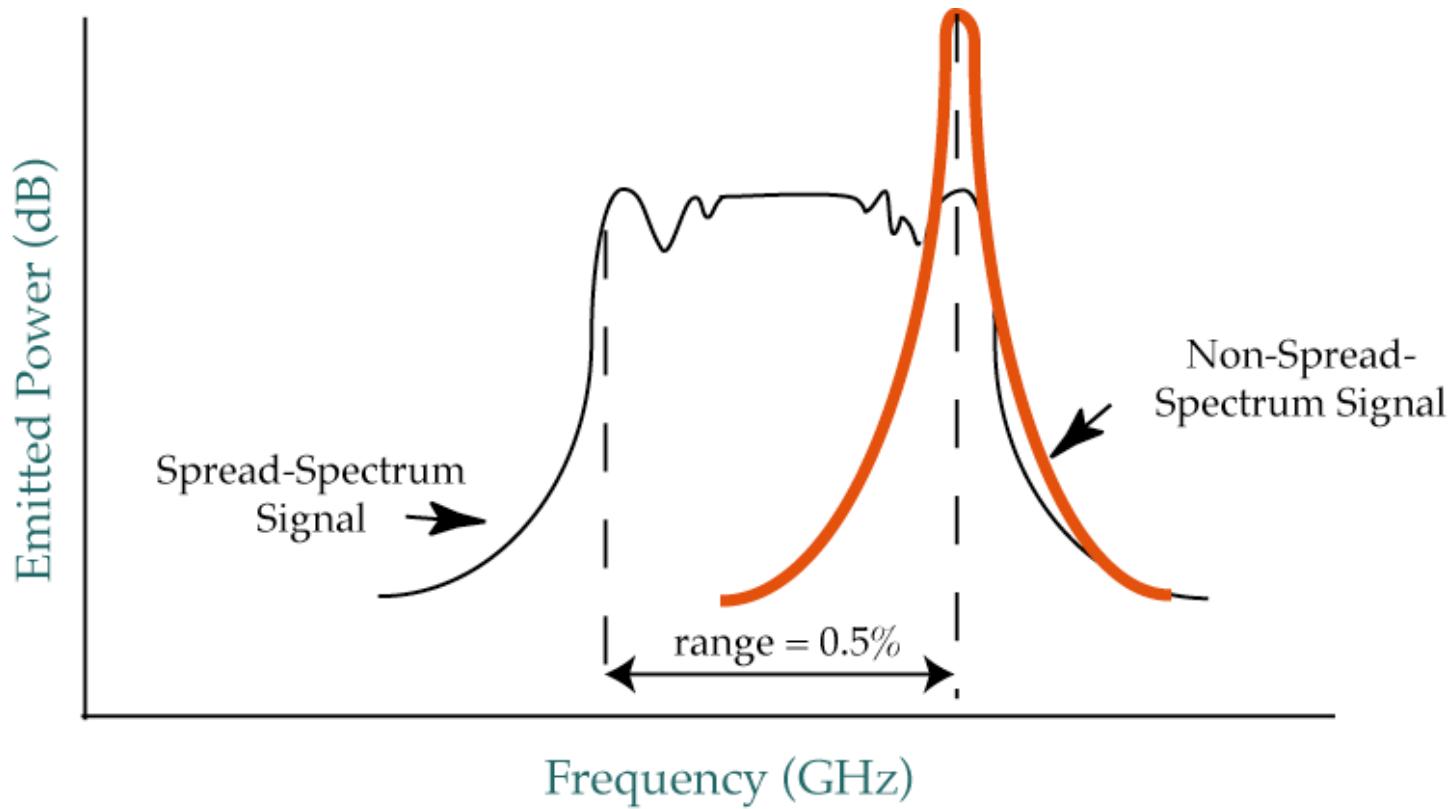
Reducing EMI with SSC

- The clocks must be accurate to within +/- 350 ppm, for a total maximum separation of 700 ppm.
- SSC (Spread Spectrum Clocking) adds another 5000 ppm, resulting in a total difference between the clocks that could be as high as 5,700ppm.
- SSC varies the clock over a range of frequencies rather than keeping it fixed as a pure reference clock, for the purpose of reducing EMI (electro-magnetic interference).

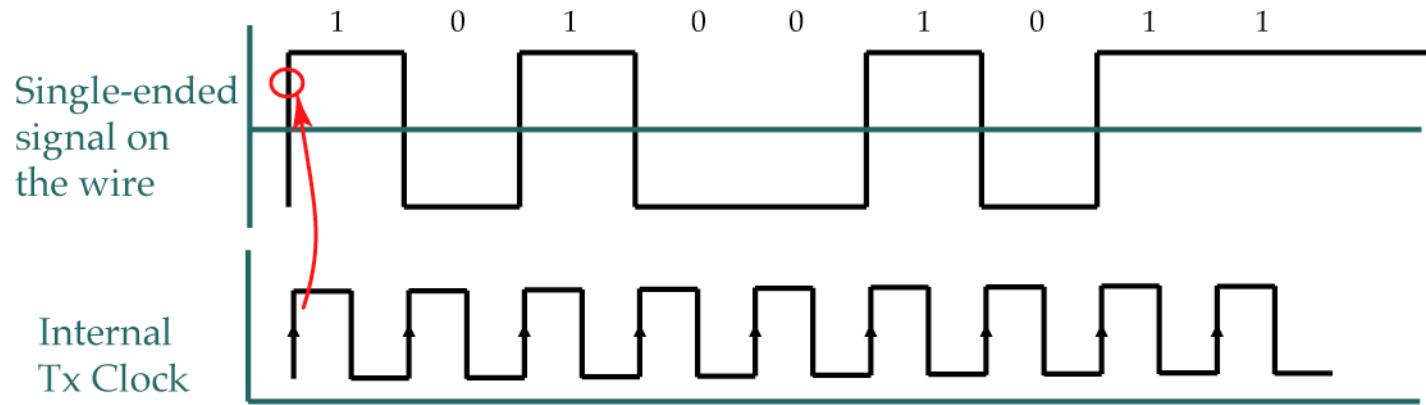
SSC Frequency Modulation



SSC Frequency Analysis

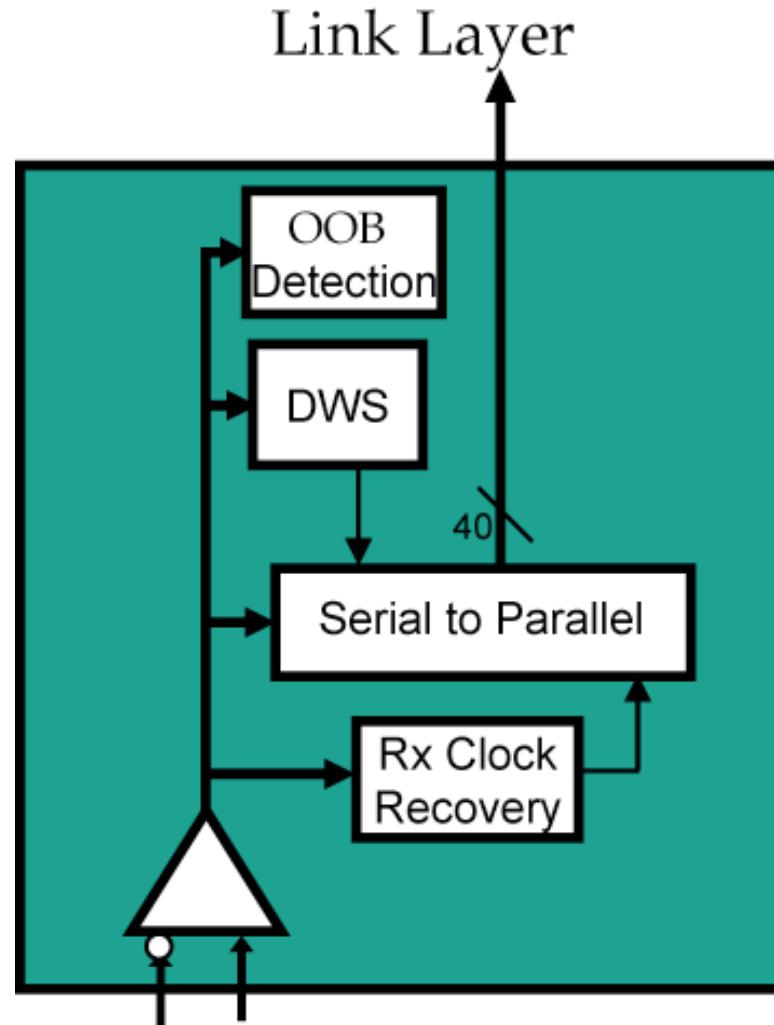


Bit Frequency

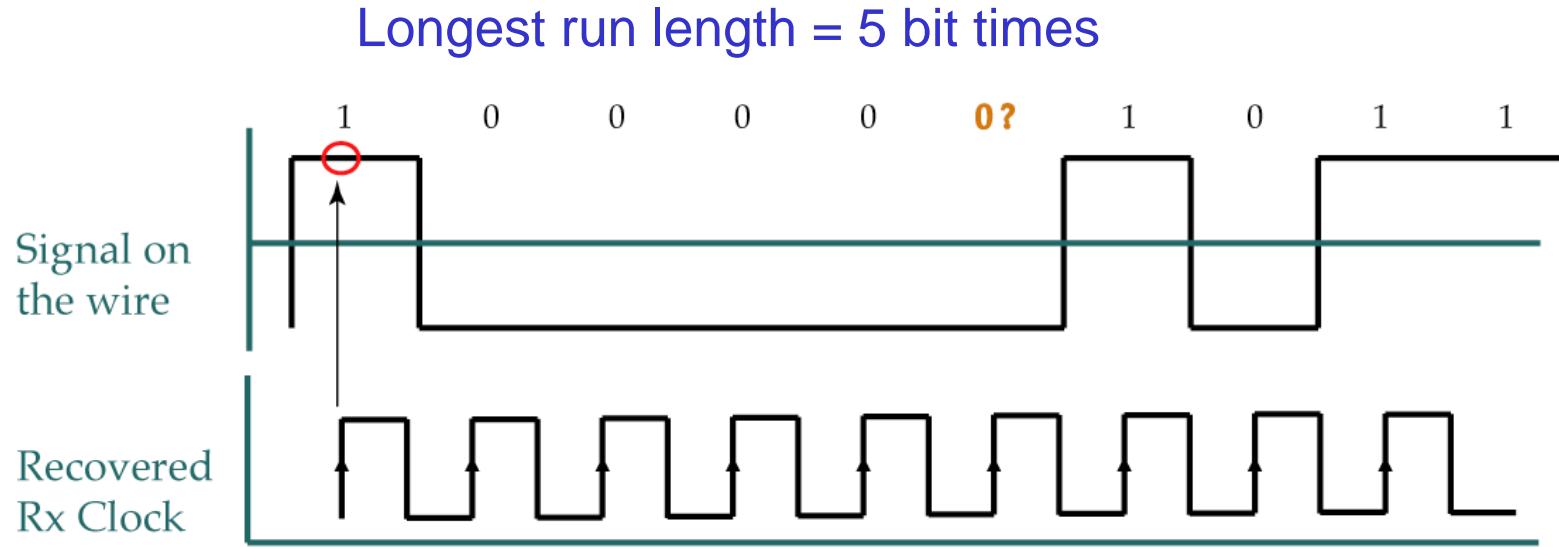


- The frequency of the signals seen on the wire using NRZ encoding must necessarily be less than or equal to half of the transmit frequency.
- Frequencies radiated by the system are most often those that appear on the transmission medium, meaning the frequency of concern for EMI in a 3.0 Gbps SATA system would actually be 1.5 GHz.

Physical Layer - Receive Functions



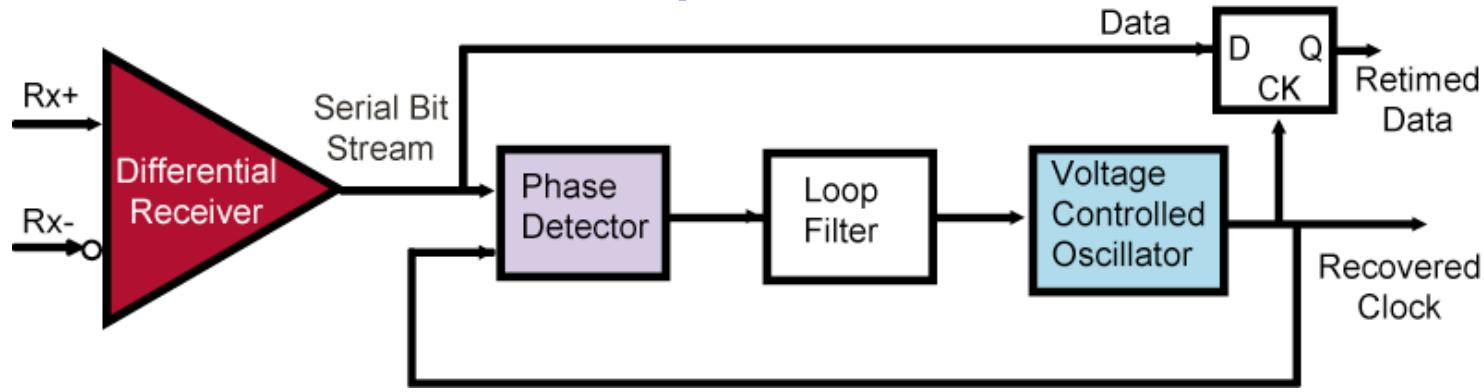
Recovering the Receive Clock



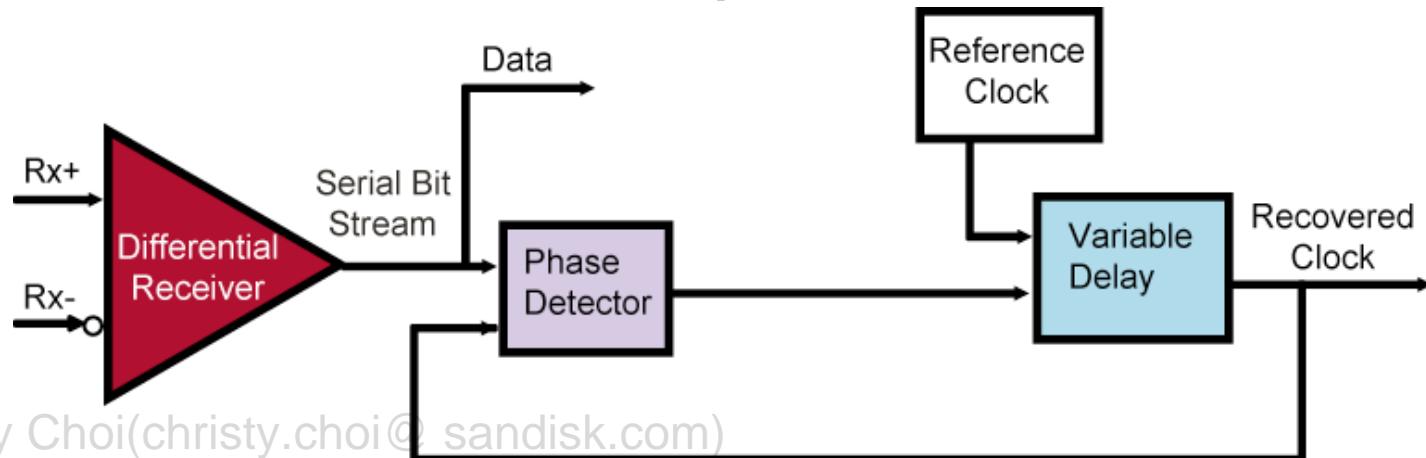
The receiver uses a PLL (tracking architecture) or over sampling circuit to recover the clock

Tracking Architectures

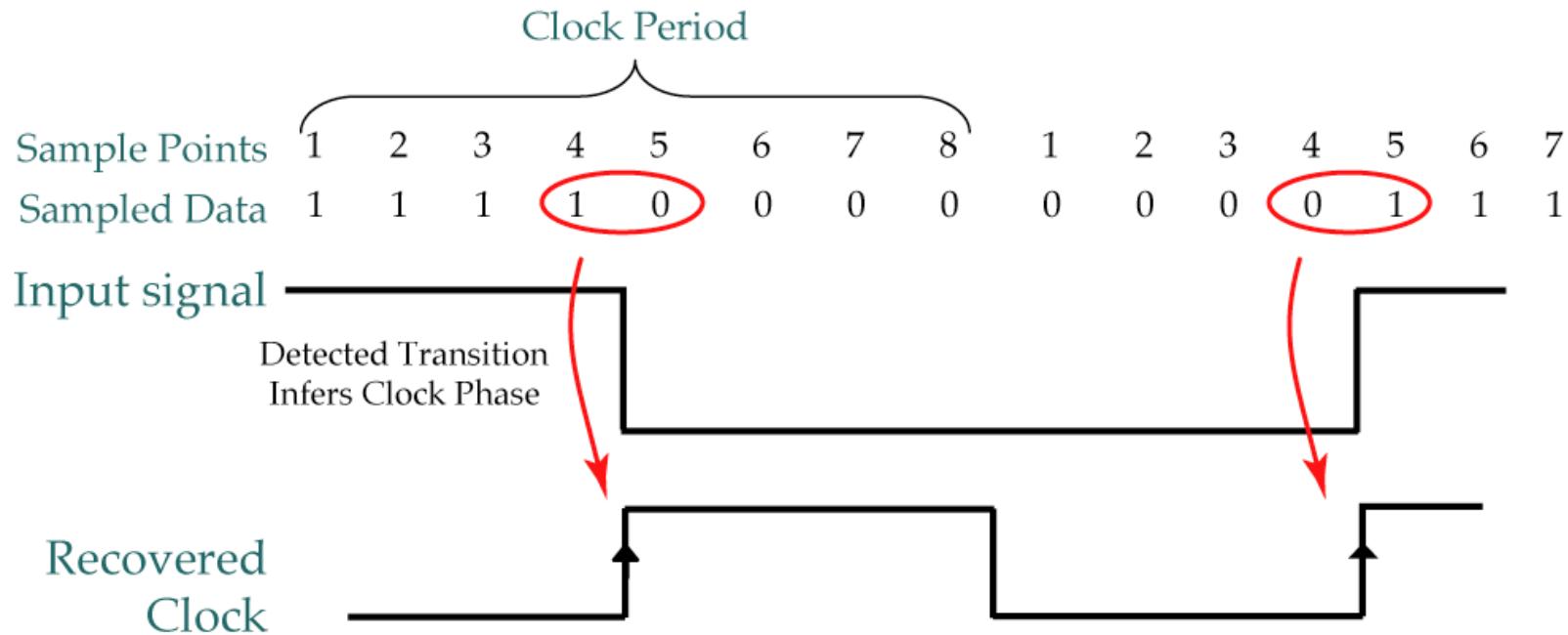
Simplified PLL



Simplified DLL



Oversampling Example



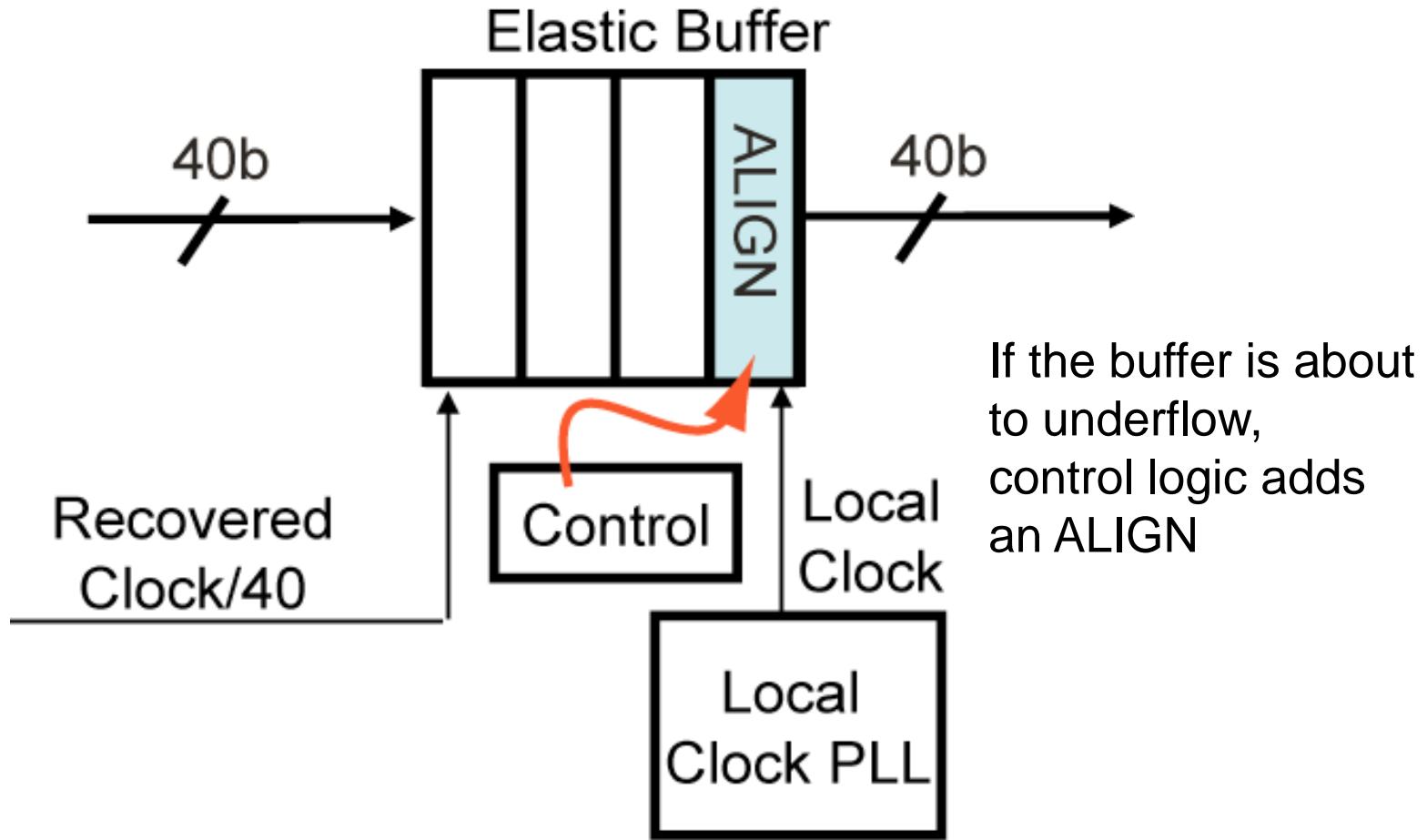
Clock Compensation

- Because the local clock domains of the HBA and Device are slightly different (+/- 350ppm) some form of compensation must be used.
- The elasticity buffer enables the receiver to accept a slightly different incoming frequency and still avoid buffer overruns or under-runs.

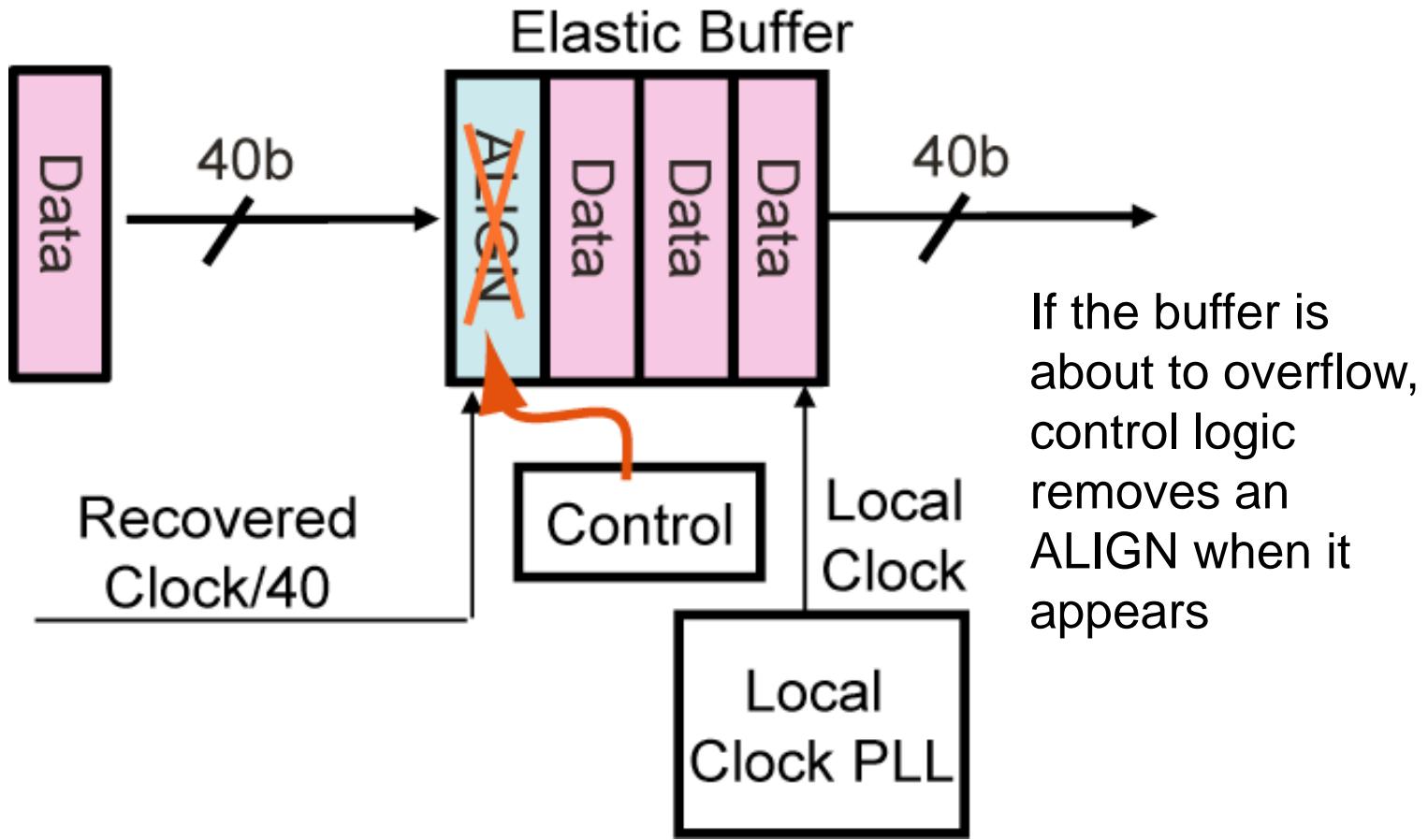
Elasticity Buffer & ALIGN Primitives

- The buffer needs to be able to add or remove some parts from the incoming bit stream.
- To facilitate this, the transmitter periodically injects ALIGN primitives at a specified rate of two ALIGNs (always sent in pairs) within at least every 256 dwords.
- Control logic monitors the buffer and adds or removes ALIGNs as required. (See following slides.)

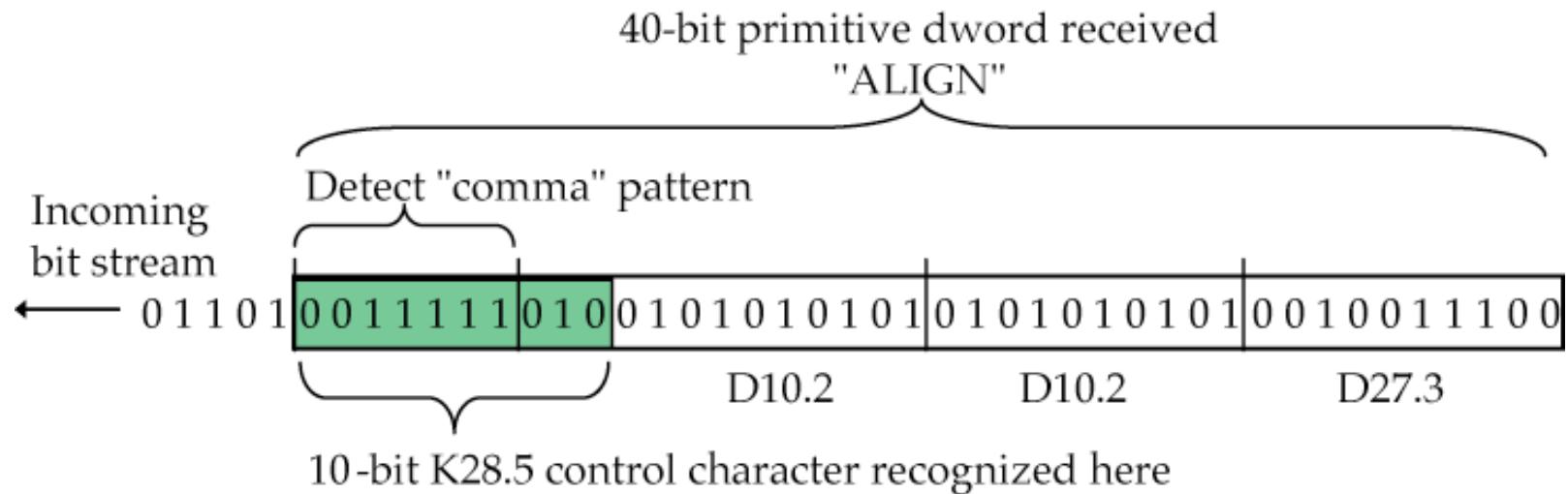
Elasticity Buffer - Dry Condition



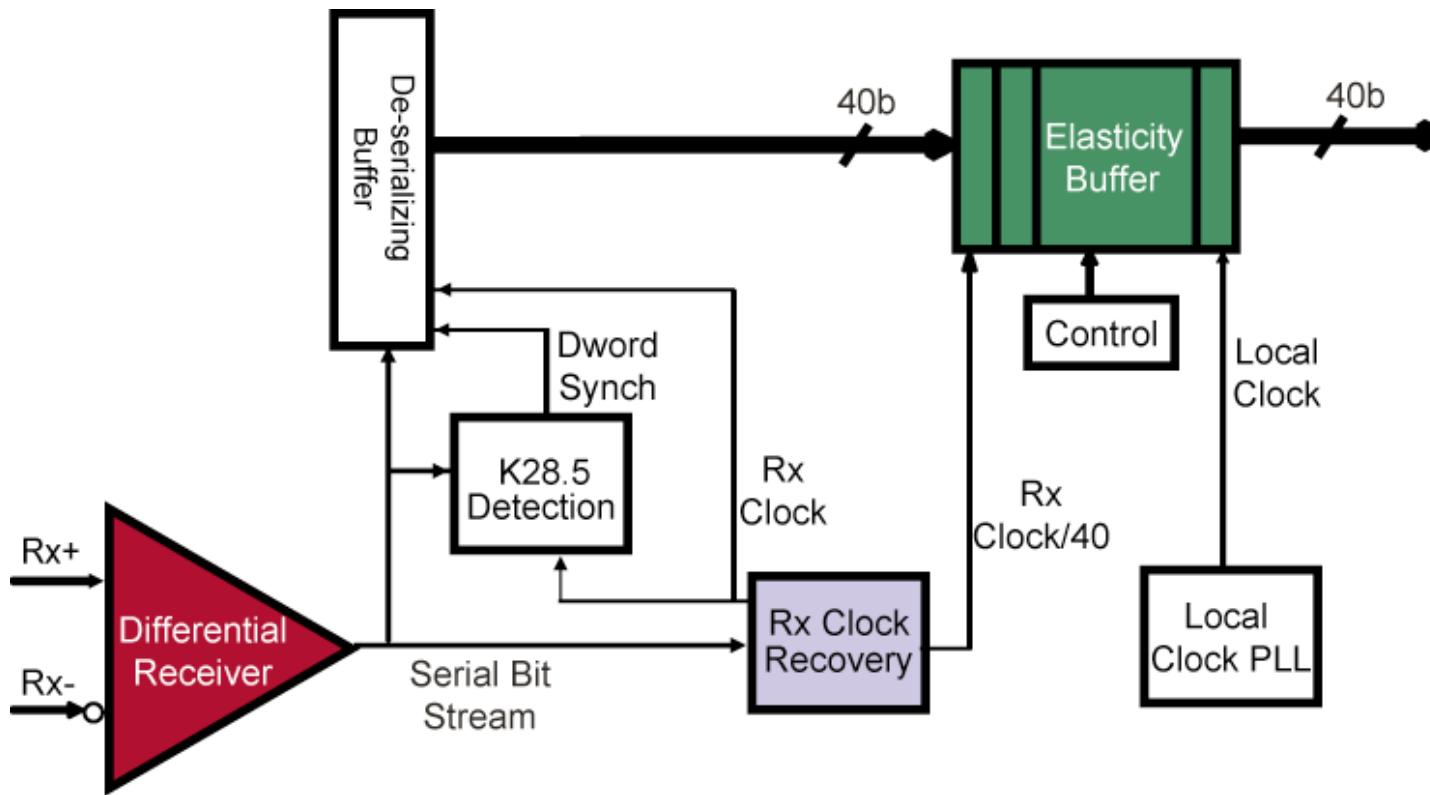
Elasticity Buffer - Overflow Condition



Dword Synchronization



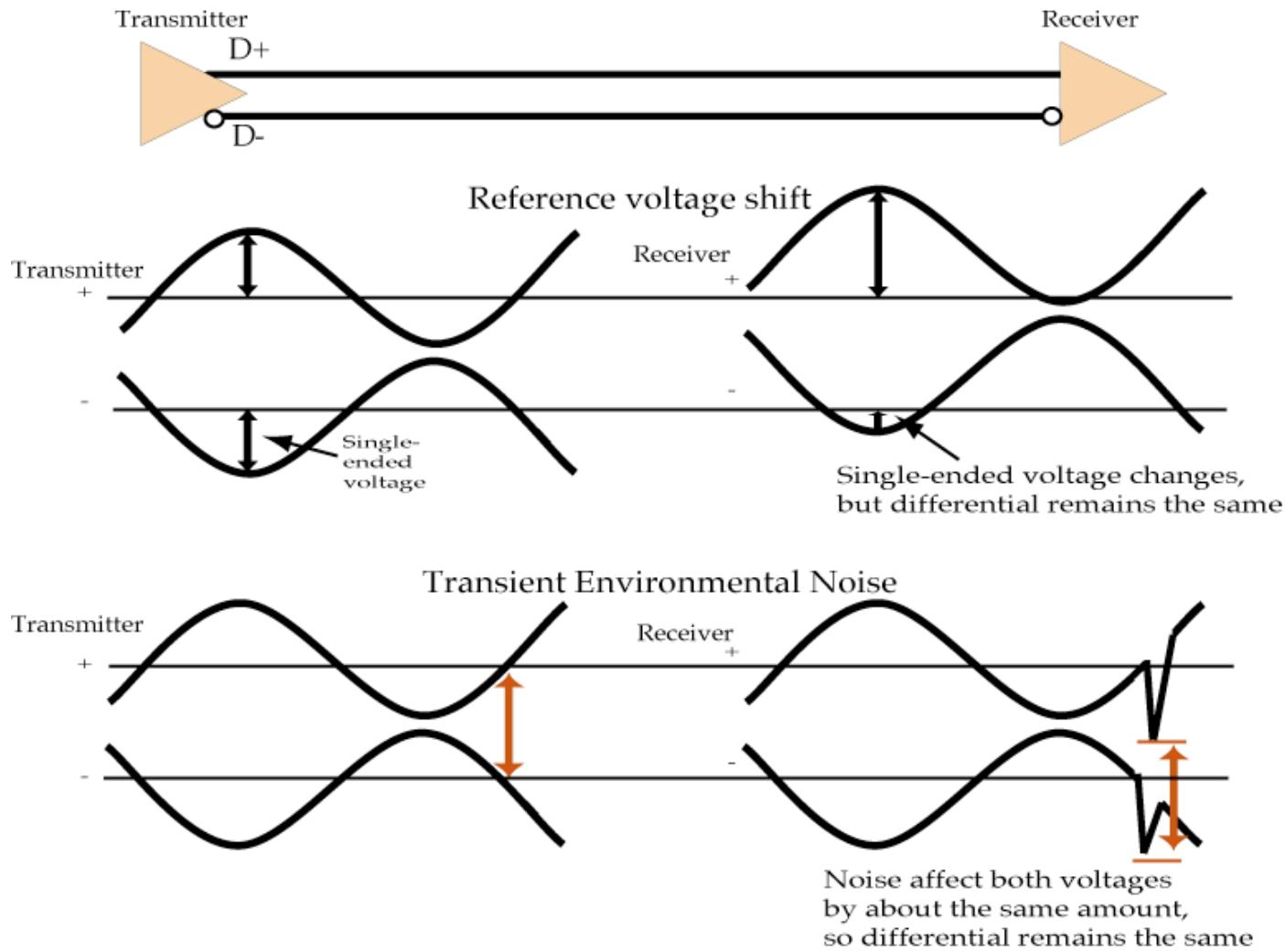
Example Clock Recovery Block



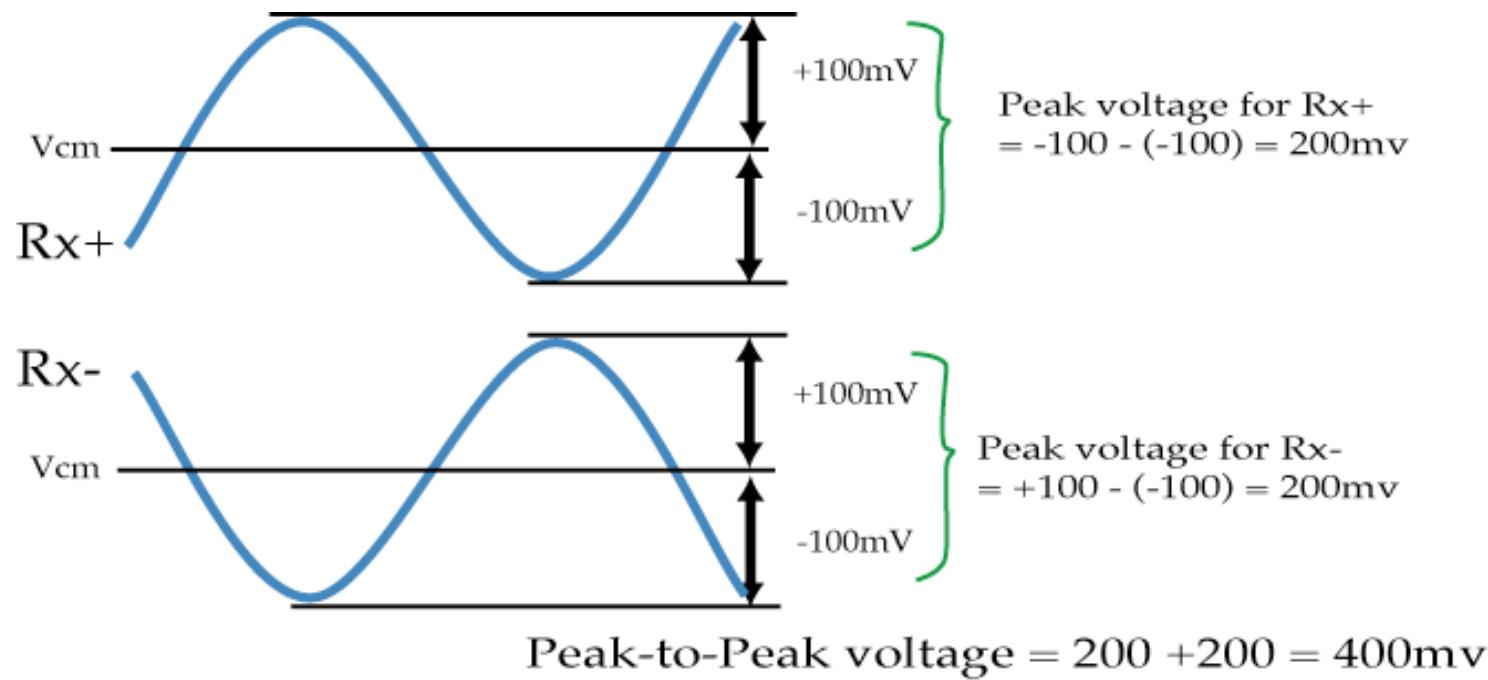
Transmission Characteristics

Characteristic	1.5 Gb/s (Gen-1)	3.0 Gb/s (Gen-2)
Physical Link Rate	150 GB/s	300 GB/s
Unit Interval (UI) (Nominal)	666.66 ps	333.33 ps
Differential Impedance (Nominal)	100 ohms	
A.C. Coupling capacitor (max)	12 nF	

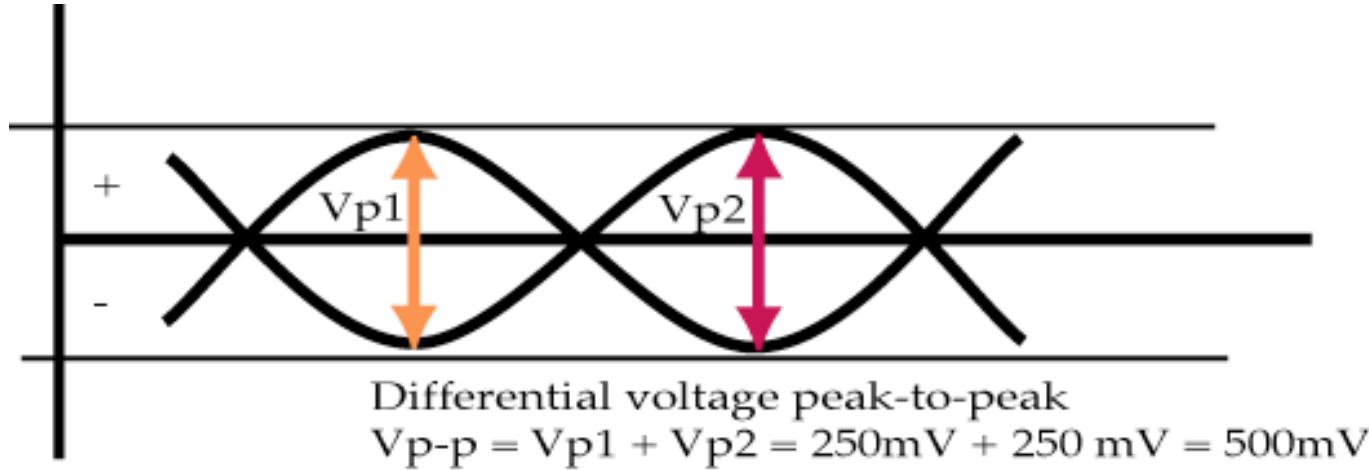
Noise Rejection



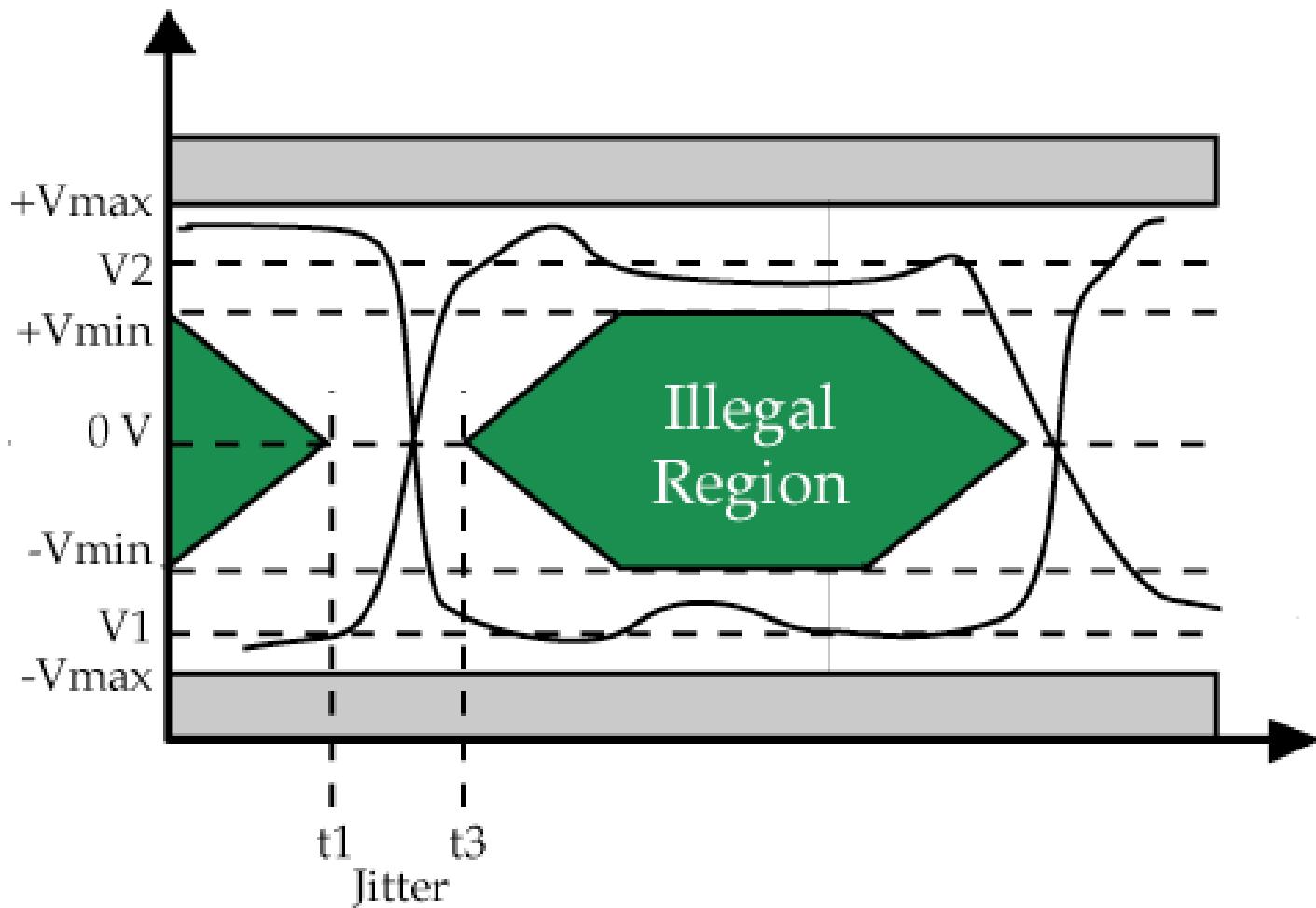
Single-Ended & Differential Voltages



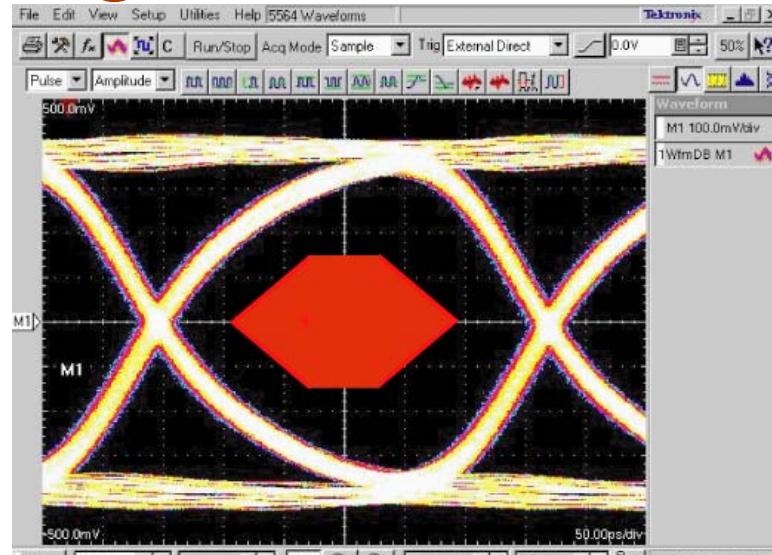
Differential Voltages Overlay View



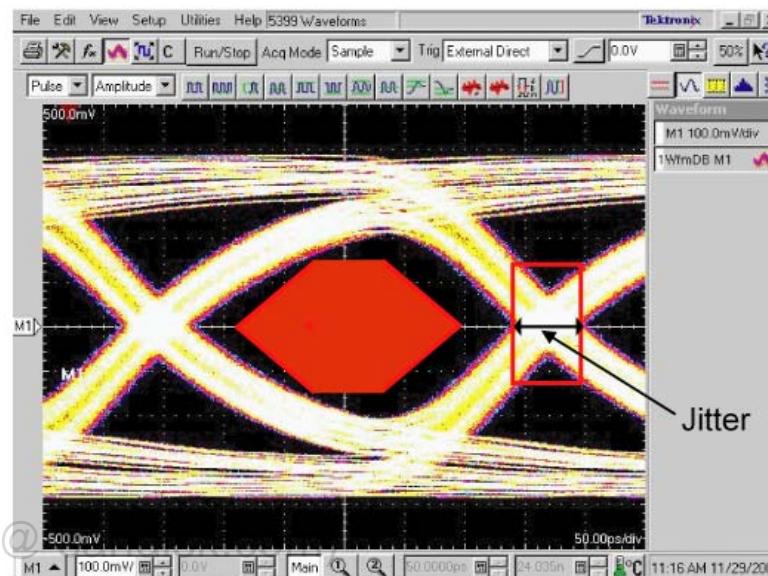
Receiver Eye Diagram



Eye Diagram - Short Cable

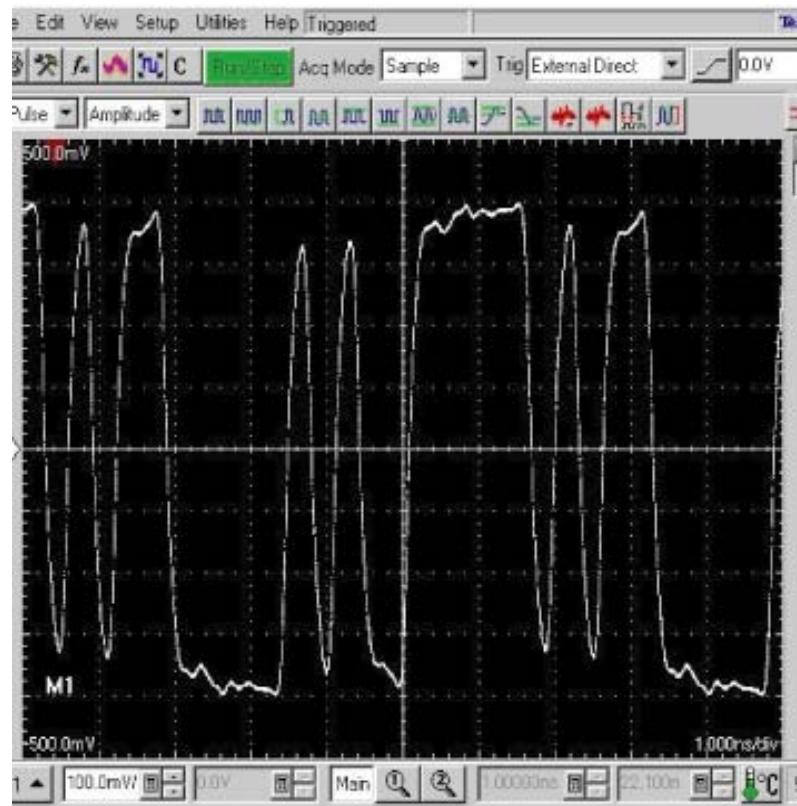


Transmitter

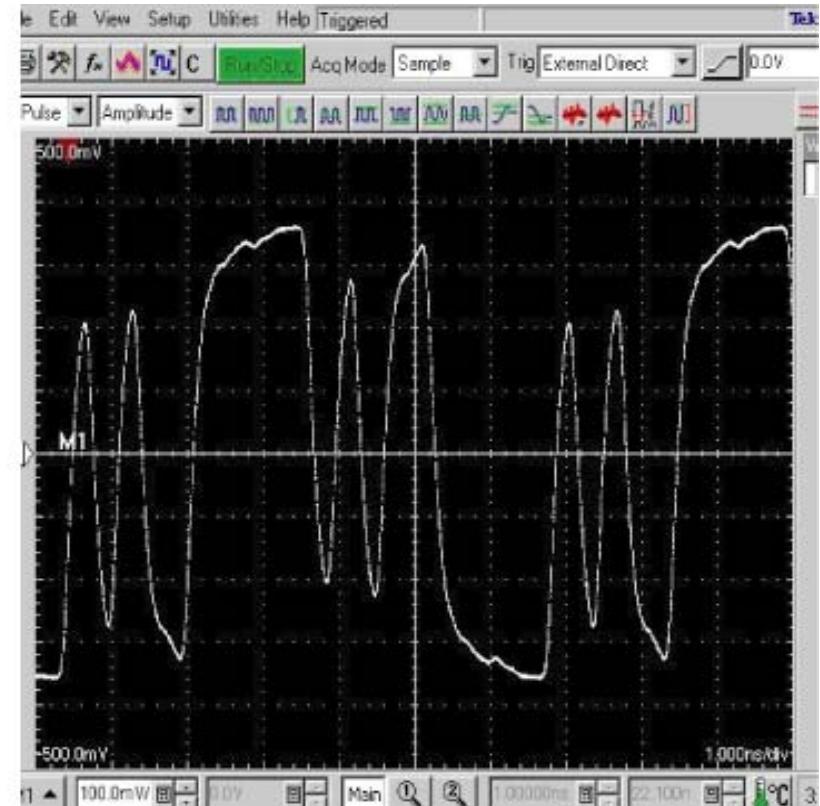


Receiver

Inter-Symbol Interference (ISI) Example

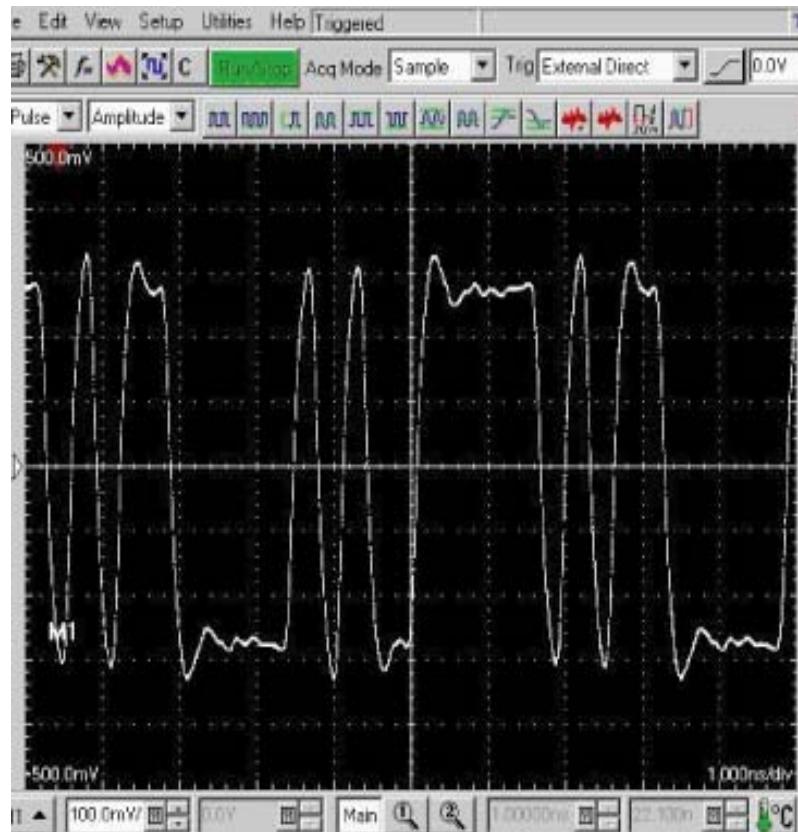


Transmitter



Receiver

ISI Reduced with De-emphasis

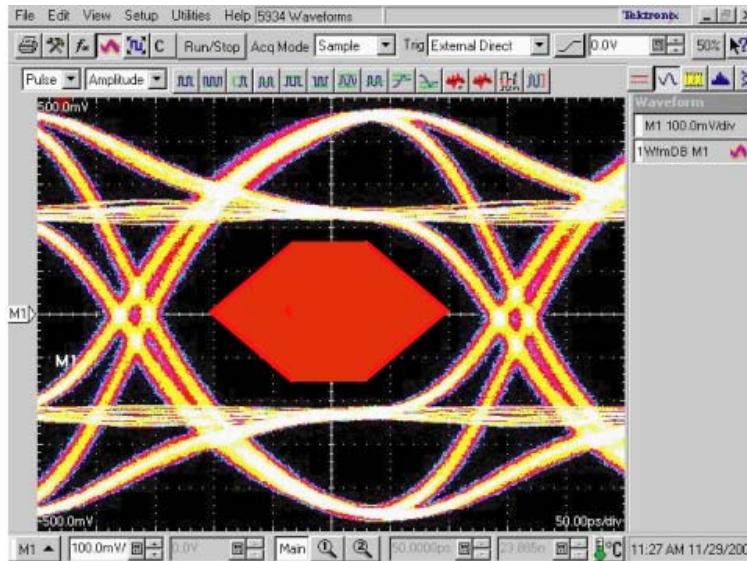


Transmitter

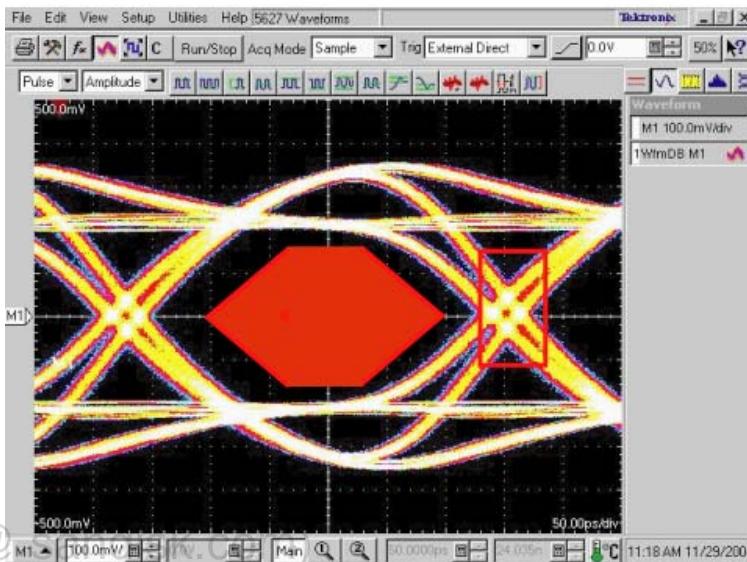


Receiver

Short Cable with Pre-emphasis



Transmitter



Receiver

Extreme Signaling Environments

In addition to internal signaling, SATA also supports extreme environments. The environments supported include:

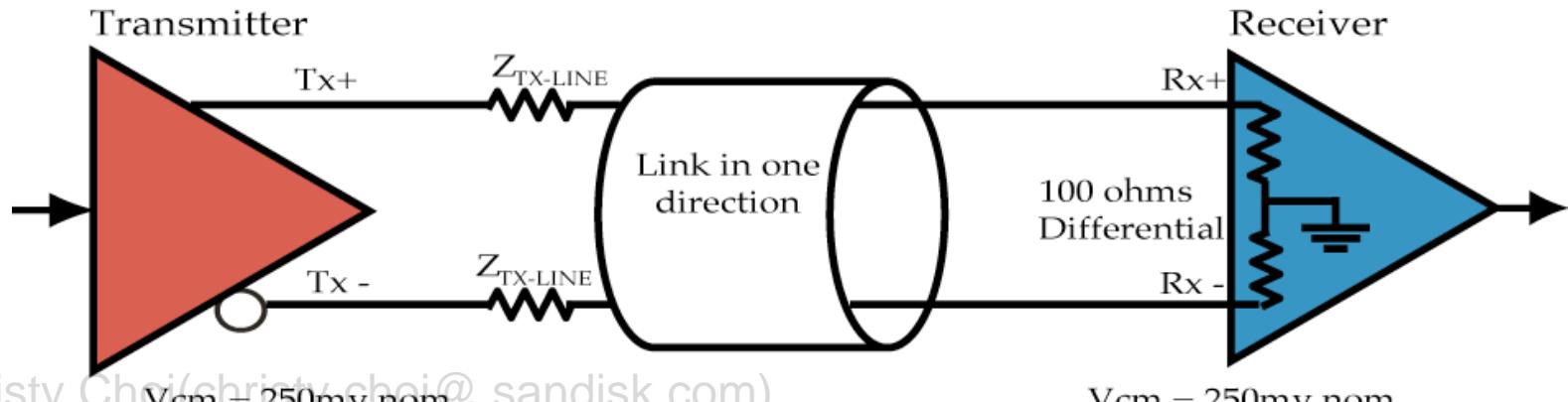
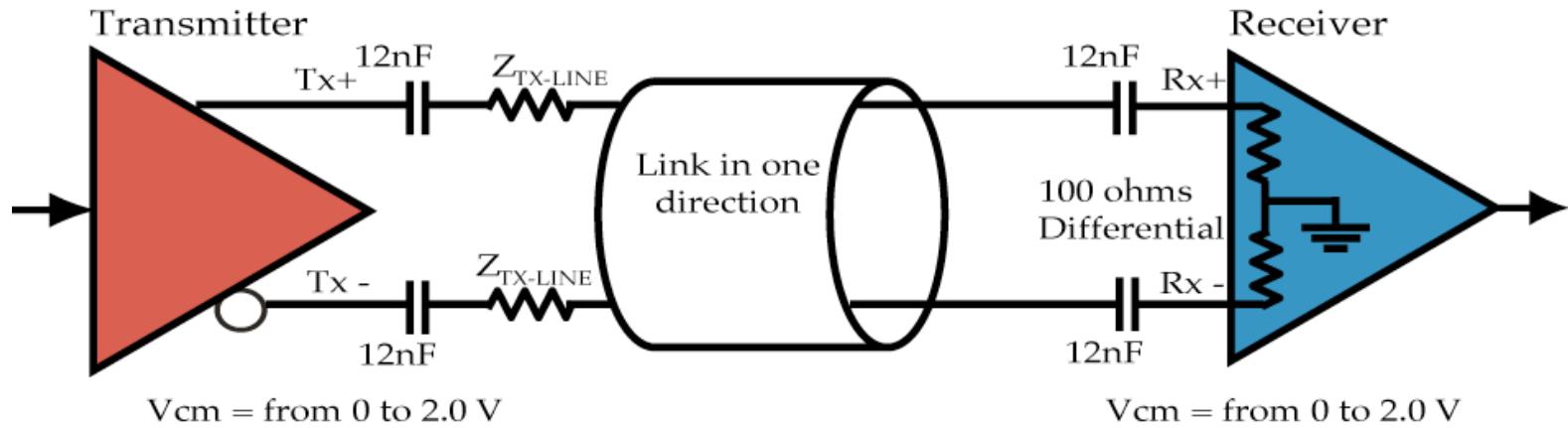
- Internal (i) Environment (Gen1i and Gen2i) - maximum cable length of 1 meter.
- Intermediate (M) Environment (Gen1m and Gen2m) - Short Backplane and External Desktop applications with cable lengths up to 2 meters.
-
- Extreme (X) Environment (Gen1x and Gen2x) - used in long backplane and longer cabled environments.

Extreme Signaling Environments

The specification makes the following statement regarding coupling:

- Gen1i implementations may be either DC- or AC-coupled.
- Gen1m implementations are allowed to be DC-coupled, but AC-coupling is recommended.
- Gen1x, Gen2i, Gen2m, and Gen2x implementations must be AC-coupled.

AC- Versus DC-Coupled Links



Christy Choi (christy.choi@sandisk.com)

Do Not Distribute

Copyright Mindshare Inc, 2009

HBA Must Support Extreme Signaling

- Signals transmitted by the host must compensate for the increased attenuation of the environment and drive signals with greater amplitude.
- Similarly, the host side receiver must have greater sensitivity to compensate for the attenuation of the signals transmitted by the drives.

HBA Compensation Example

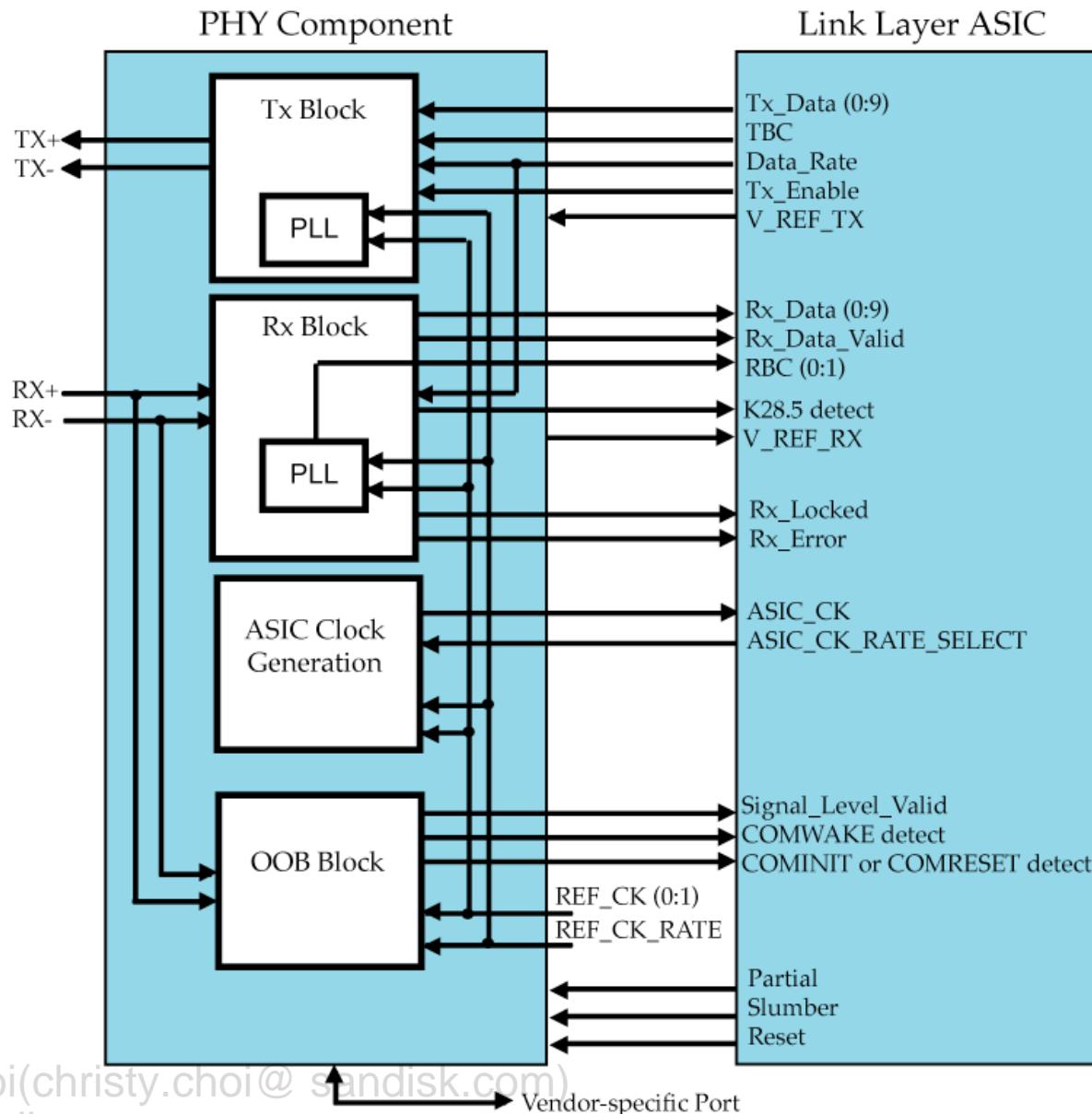
HBA Transmitter Output Voltages

Parameter	Limit	Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Units
Tx Diff. Output Voltage	Min	400	500	800	400	500	800	mVppd
	Nom	500		--	--			
	Max	600		1600	700		1600	

HBA Receiver Input Voltages

Parameter	Limit	Gen1i	Gen1m	Gen1x	Gen2i	Gen2m	Gen2x	Units
Differential Input Voltage	Min	325	240	275	275	240	275	mVppd
	Nom	400		--	--			
	Max	600		1600	750	750	1600	

SAPIS Block Diagram



Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

Copyright Mindshare Inc, 2009

Cables & Connectors

Christy Choi(christy.choi@sandisk.com)

Do Not Distribute



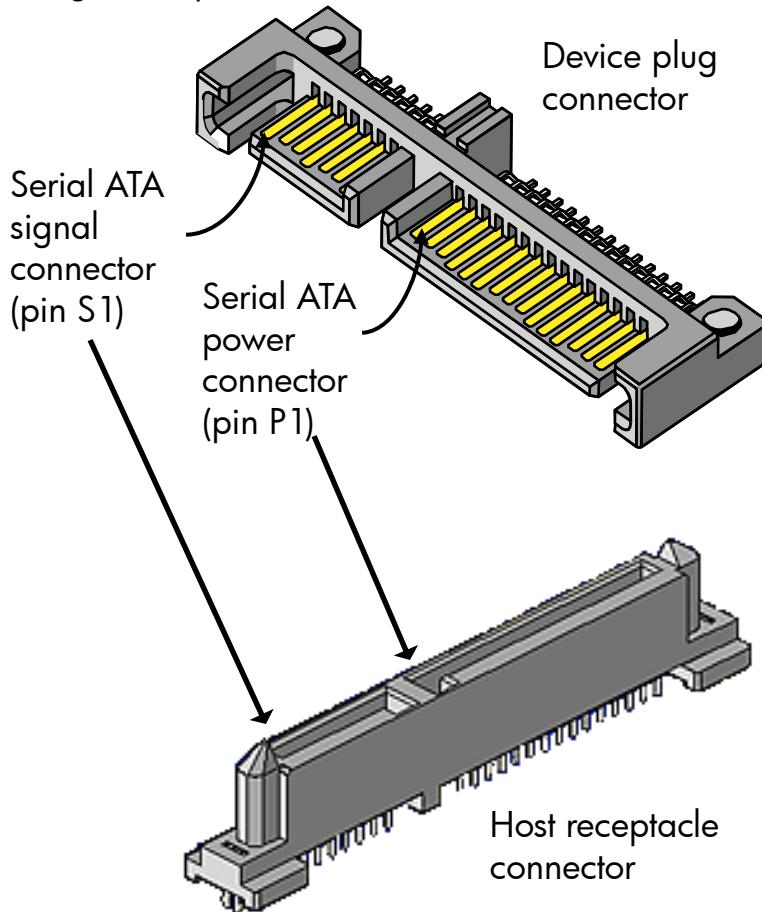
Usage Models / Form Factors

- Internal 1-meter cable
- Internal 4-lane cabled disk array
- System to device external interconnect (long)
- Short backplane to device
- Long backplane to device
- System to System Interconnect (very long)

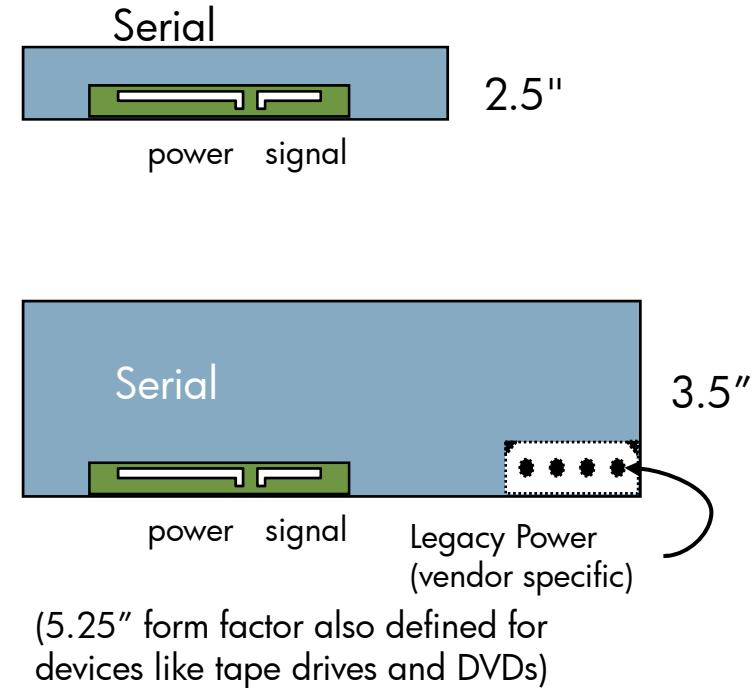
SATA device connector

Appearance of Serial ATA Connectors

(Drawing courtesy of Molex)



Device connector sizes and locations



Connector Finger Lengths

- The connector fingers have two different lengths, long and short, allowing for three possibilities.
 - First mate (long to long) for grounds and “precharge” power pins
 - Second mate (long/short or short/long) for power pins
 - Third mate (short/short) for signals
- Intended to guarantee some microseconds settling time before signals make connection.

Internal connector signals

Host connector	
Pin	Signal
S1	GROUND
S2	TP+
S3	TP-
S4	GROUND
S5	RP-
S6	RP+
S7	GROUND

Device connector	
Pin	Signal
S1	GROUND
S2	RP+
S3	RP-
S4	GROUND
S5	TP-
S6	TP+
S7	GROUND

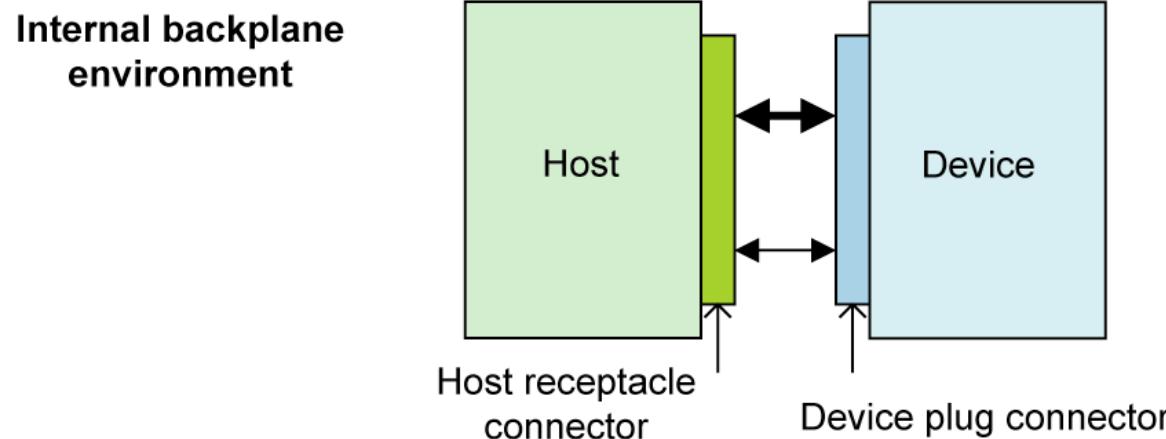
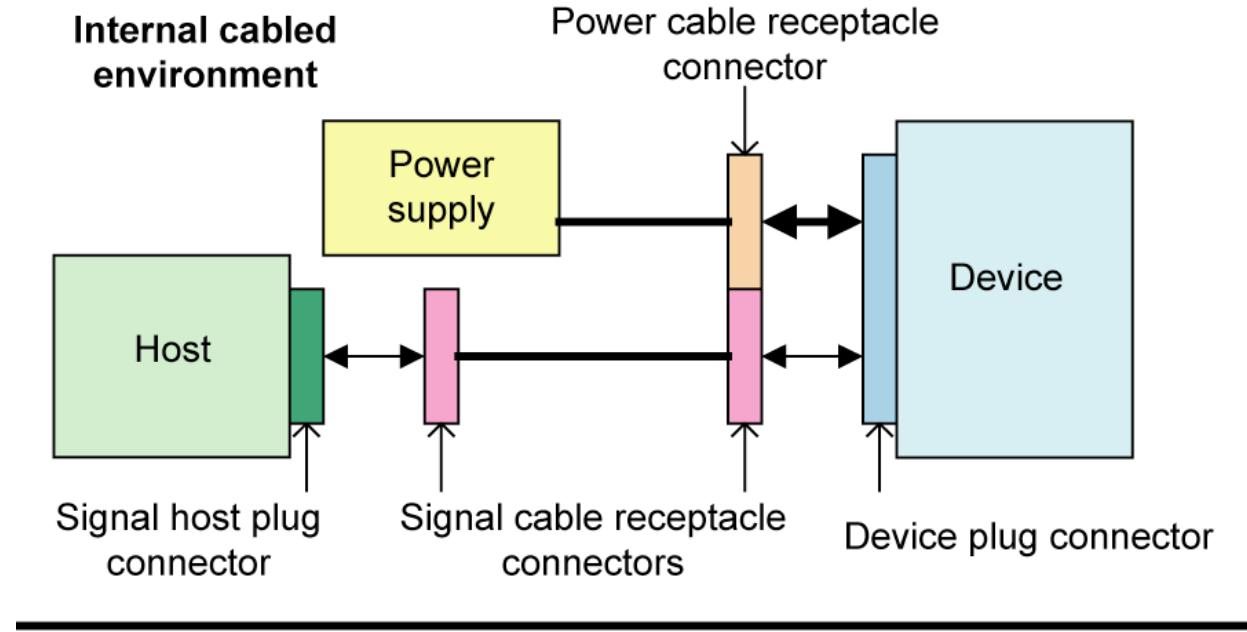
The cable or backplane connects Tx on one side to Rx on the other side

Internal device connector power assignments

- Three voltages are provided
 - 3.3 V (new for serial interfaces)
 - 5 V
 - 12 V (for drive motors)
- All the pins of one voltage are tied together by the target device
- Precharge pins are longer
- Signal cables do not carry power
 - Separate power cable
 - Converts cables from old large power connector to the SATA hot plug capable power connector

Pin	Signal
P1	V3.3
P2	V3.3
P3	V3.3, precharge
P4	GROUND
P5	GROUND
P6	GROUND
P7	V5, precharge
P8	V5
P9	V5
P10	GROUND
P11	READY LED
P12	GROUND
P13	V12, precharge
P14	V12
P15	V12

Internal Cable & Connectors



Single Lane Cables



COMAX



COMAX

External Cables & Connectors (eSATA)

- Connectors designed to handle many unplug/plug cycles
- Improved connector retention
- Support for faster transmission rates
- Improve transmission characteristics
- Longer cables (2 meters and beyond)
- Support for hot plugging
- Improved ability to control Electromagnetic Interference (EMI)
- Requirements for handling electrostatic discharge (ESD)

eSATA Single-Lane Cable



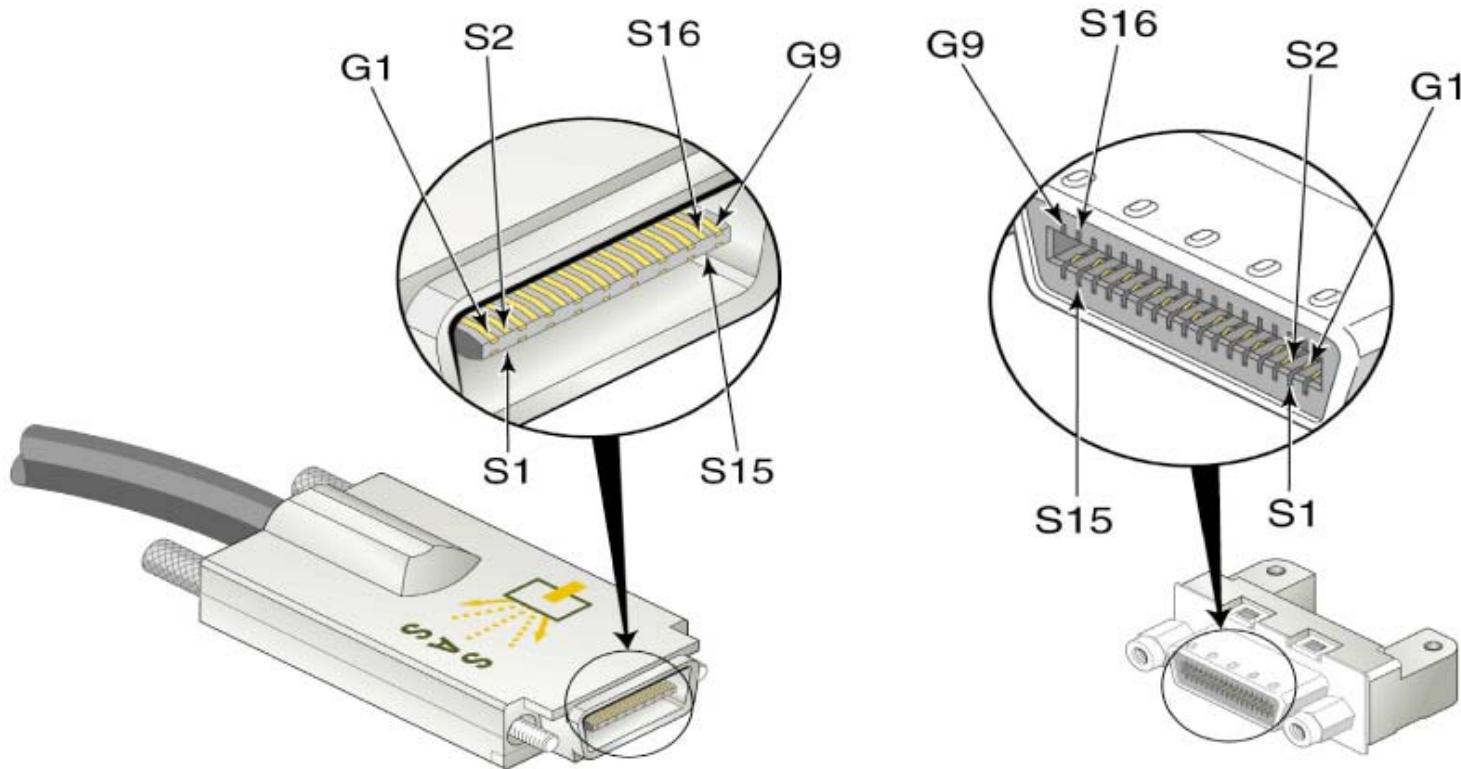
COMAX

Christy Choi(christy.choi@ sandisk.com)

Do Not Distribute

Copyright Mindshare Inc, 2009

eSATA Multiple-Lane Cables



Hot Plug

Christy Choi(christy.choi@sandisk.com)
Do Not Distribute



SATA Hot Plug Requirements

- Gen1i/Gen2i Implementations do not require Hot Plug support; however, if implemented in a short backplane implementation, Hot Plug capability is required.
- Gen1m/Gen2m implementations require support for Hot Plug.
- Gen1x/Gen2x Implementations are required to support Hot Plug.

SATA Hot Plug Requirements

The SATA specifications define the following key features that may be implemented in Hot Plug solutions:

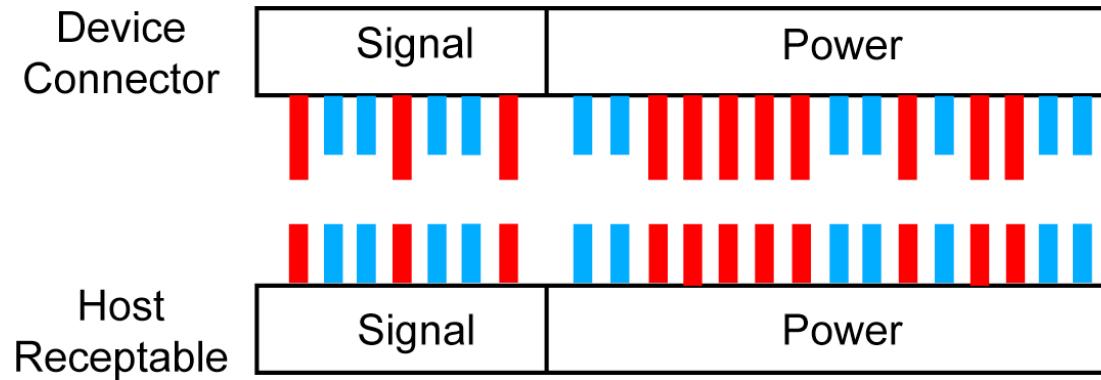
- Asynchronous Signal Recovery (SATA II)
- Two Connector Contact Lengths to support Hot Plug Connectors
- In-rush Current Limiting Definition (SATA II)
- Drive Presence Detect (SATA II)
- Asynchronous Event Notification (SATA II)

SATA Hot Plug Operations

When a Hot Plug operation is performed two possibilities exist:

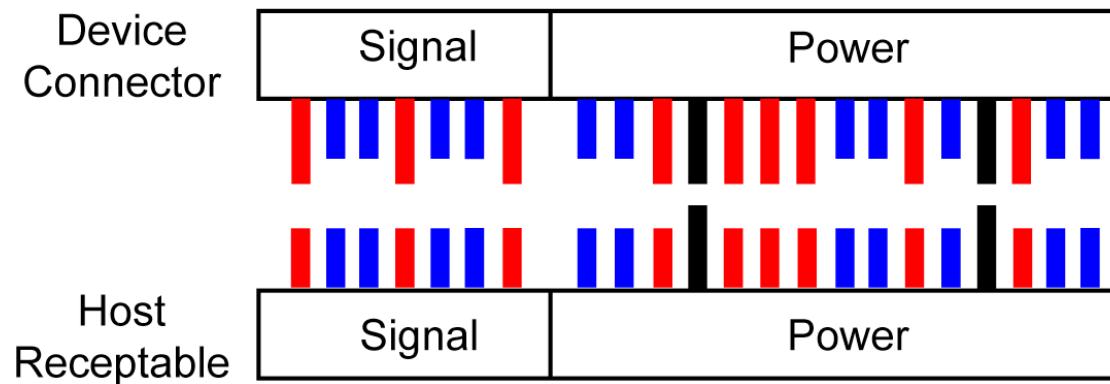
- Power is present at the connector — insertion and removal is considered a surprise event because software has no prior notice of the Hot Plug event.
- Software has previously been notified by the user that a Hot Plug operation is requested. Under software control power is removed prior to the drive being removed and replaced.

Hot Plug Connection - Cable



Contact Combinations	Mating Points	Contact Assignments
Long to Short	First	3.3vdc Precharge Contact 5vdc Precharge Contact 12vdc Precharge Contact All ground Contacts
Short to Short	Second	Other 3.3vdc Contacts (2) Other 5vdc Contacts (2) Other 12vdc Contacts (2) Device Activity/Staggered Spinup disable Differential Signal Pairs (4 contacts)

Hot Plug Connection (Backplane)



Contact Combinations	Mating Points	Contact Assignments
Long to Long	First	Two Ground Contacts
Long to Short	Second	3.3vdc Precharge Contact 5vdc Precharge Contact 12vdc Precharge Contact All Other Ground Contacts (6)
Short to Short	Third	Other 3.3vdc Contacts (2) Other 5vdc Contacts (2) Other 12vdc Contacts (2) Device Activity/Staggered Spinup disable Differential Signal Pairs (4 contacts)

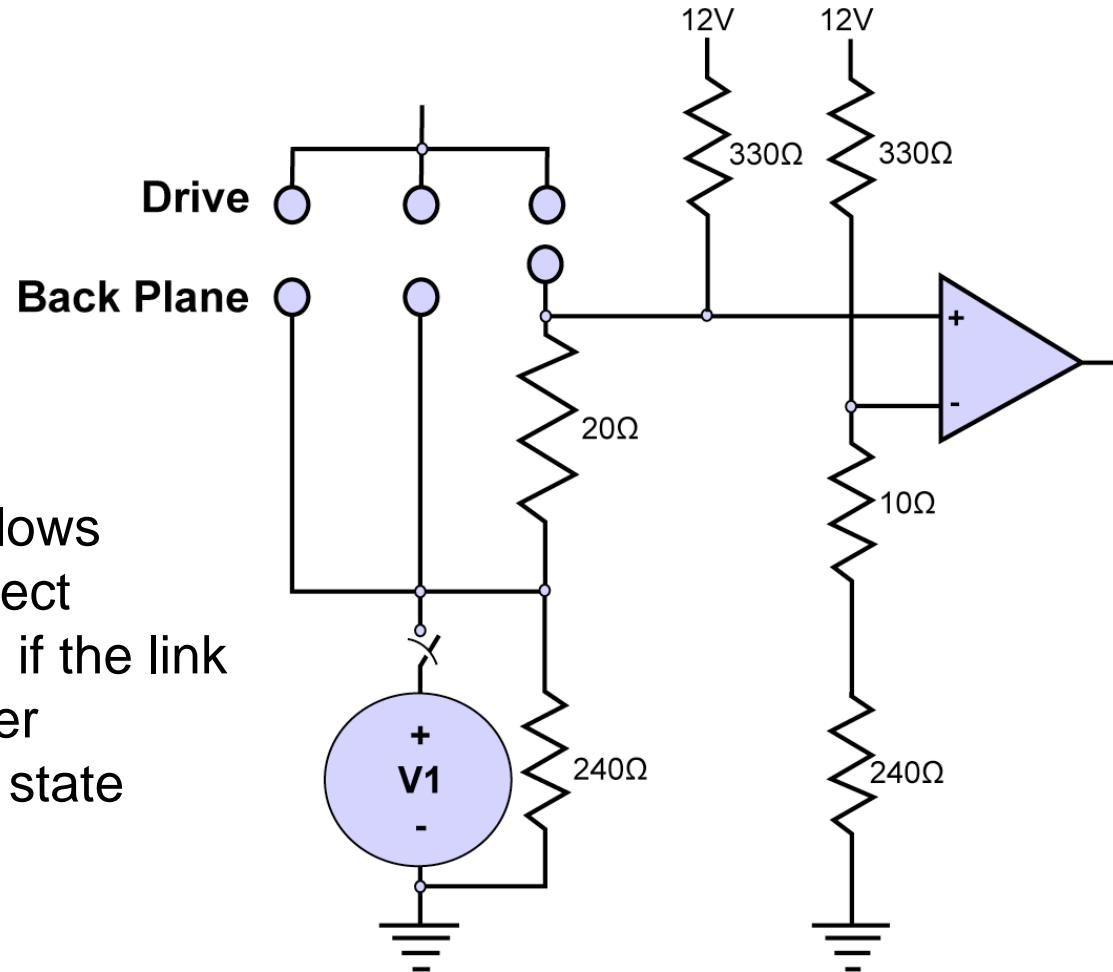
Drive Plug/Unplug detection

The specification requires that SATA devices bus the power contacts together for each supply voltage. The contact assignments are:

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

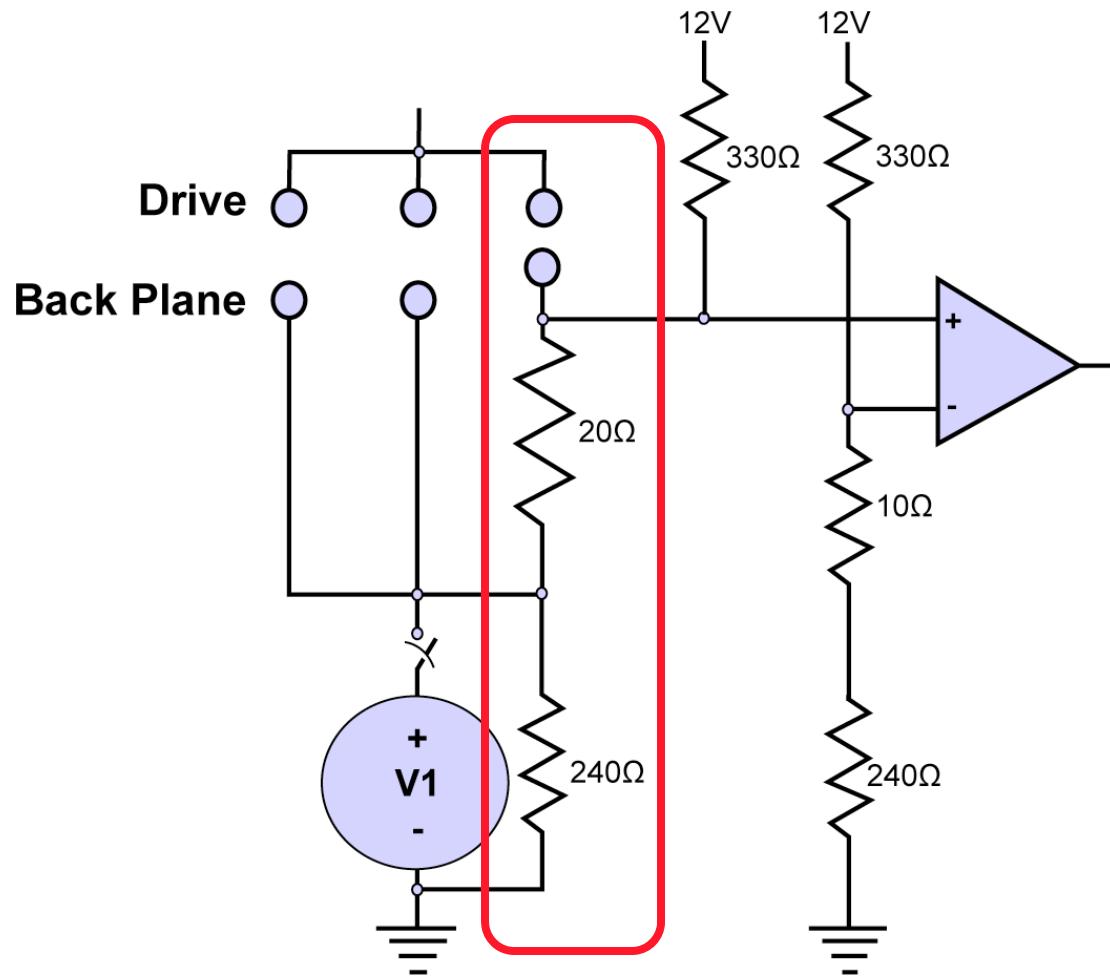
When device attachment or removal is detected, the host interface sets the “X” bit in the SError register’s DIAG field.

Example Removal Detect Circuit

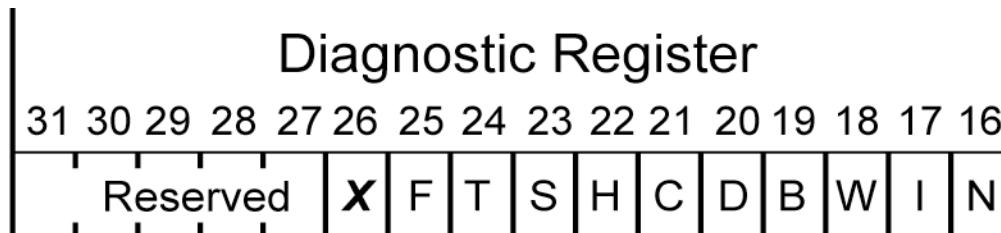


This circuit allows system to detect removal even if the link was in a power management state

Current Limiting



Modification to SError Register (Device Changed Detection)

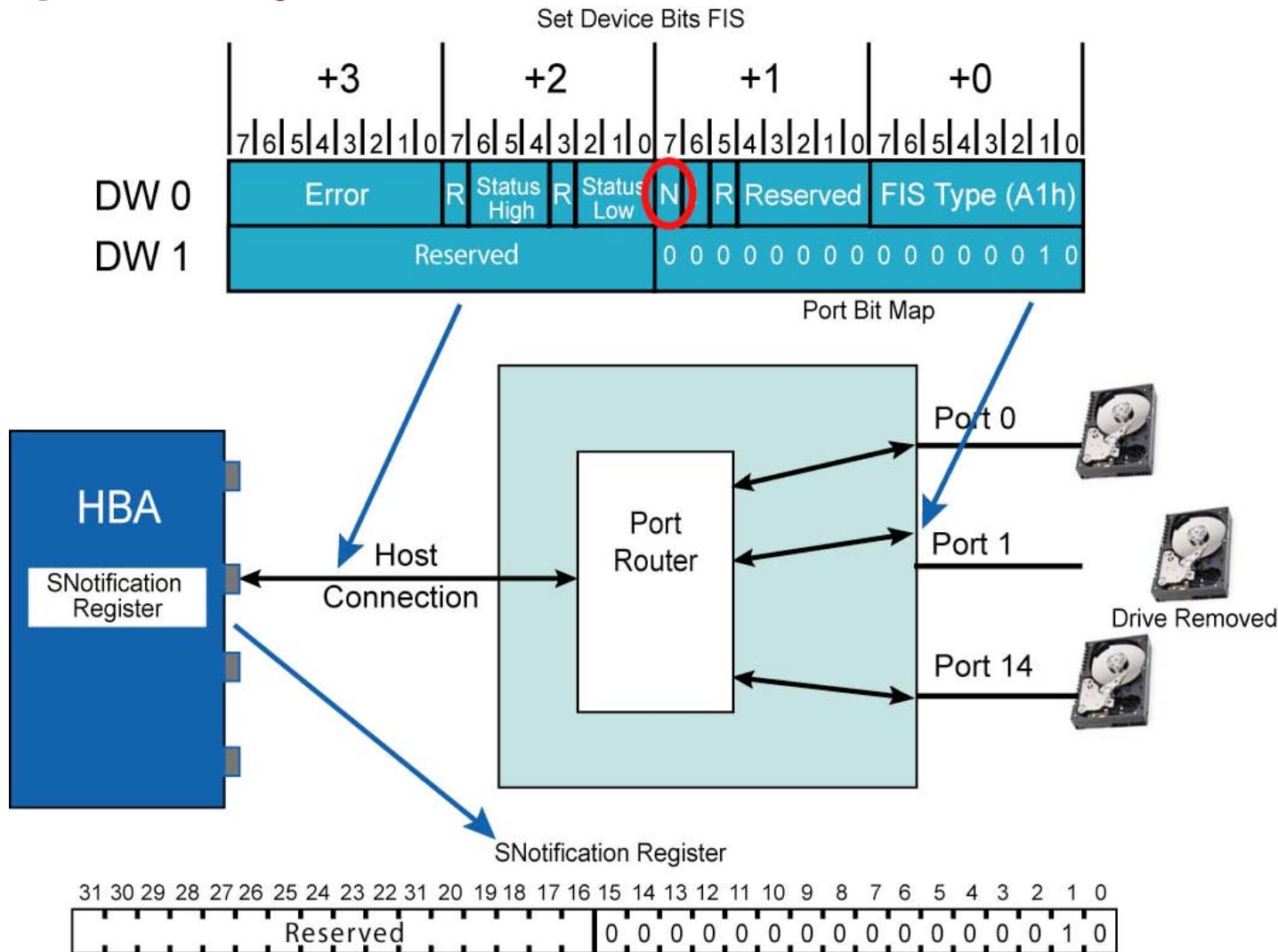


X = Exchanged

- 0 = No Device presence change detected since bit last cleared
- 1 = Device presence has changed since this bit was cleared

All other bits are the same as defined by 1.0a

Example Asynchronous Event Notification



Notify - Each bit position represents a Port Multiplier Port number (ports 15:0) indicating whether a device with the corresponding port number has sent a Set Device Bits FIS with the notification bit set.

Link Power Management

Christy Choi(christy.choi@ sandisk.com)

Do Not Distribute



Link Power Management

- Power Management support is optional
- Intended for Mobile use
- Two low-power modes supported:
 - Partial
 - Slumber
- The power limits in these states are not specified and are vendor specific

Drive Capabilities

Identify Data defines a drive's support for link power management, this includes whether a drive supports:

- receipt of Host-initiated commands to enter Link Power Management.
- initiation of commands that place the link into the partial or slumber states.

See Table 22-1 on book page 385

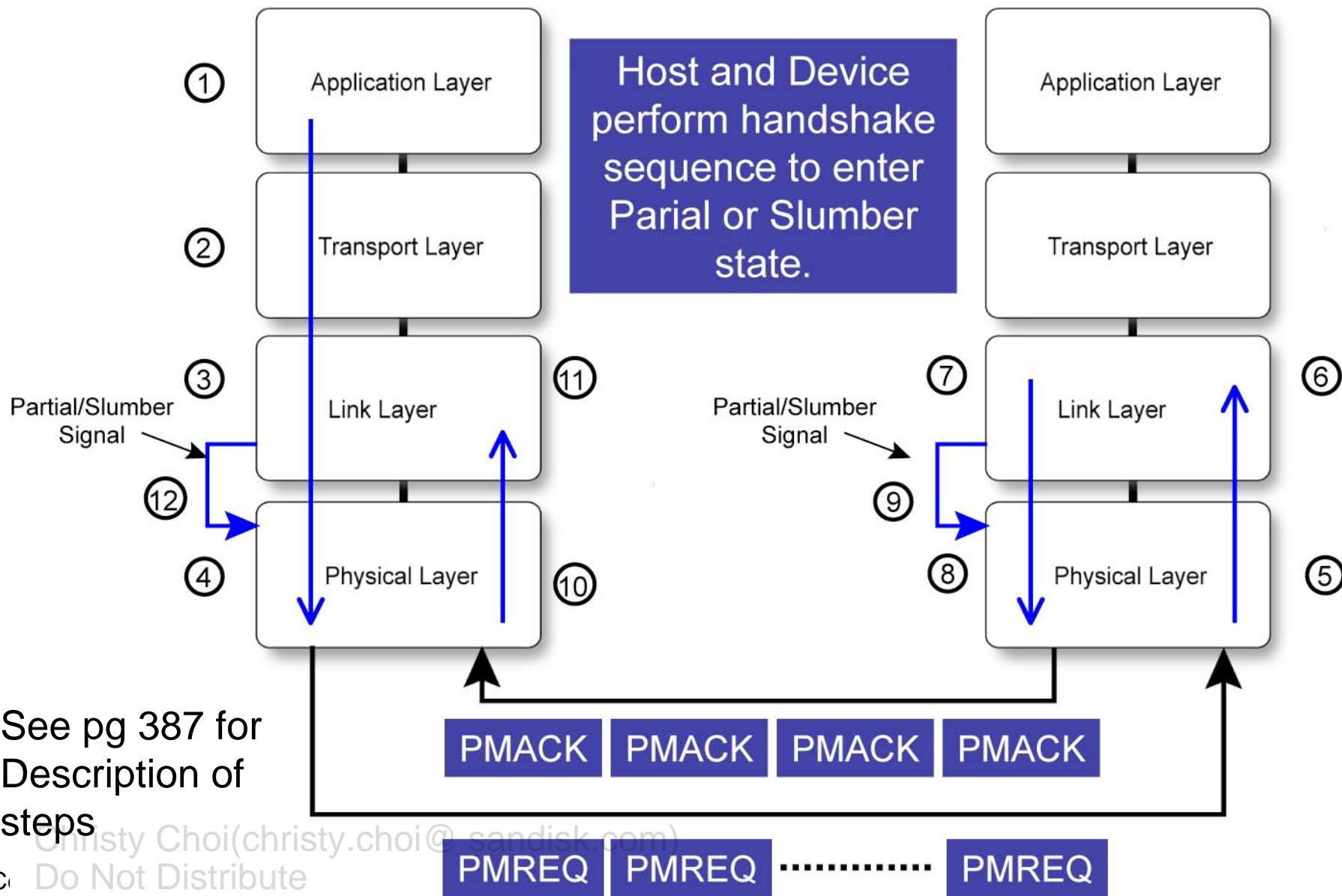
Primitives – Power Management Command Execution

Primitive	Name	Description
PMREQ_S	Power Management Request to enter Slumber state	This primitive is sent continuously until PMACK or PMNAK is received. When PMACK is received, current node (host or device) will stop PMREQ_S and enters the Slumber power management state.
PMREQ_P	Power Management Request to enter Partial PM state	This primitive is sent continuously until PMACK or PMNAK is received. When PMACK is received, current node (host or device) will stop PMREQ_P and enters the Partial power management state.
PMACK	Power Management Acknowledgement	Sent in response to a PMREQ_S or PMREQ_P when a receiving node is prepared to enter a power mode state.
PMNAK	Power Management No Acknowledgement	Sent in response to a PMREQ_S or PMREQ_P when a receiving node is not prepared to enter a power mode state or when power management is not supported.

Device Identify Data - SATA II

Word Offset	R/O	Description
0 - 74		Defined in ATA/ATAPI-7
75	Optional	<p>Queue Depth</p> <p>15-5 Reserved 4-0 Maximum queue depth-1</p>
76		<p>Serial ATA Capabilities</p> <p>15-11 Reserved 10 Phy event counters supported 9 Host-initiated link Power Management requests received 8 Native Command Queuing supported 7-4 Reserved 3 Reserved for SATA 2 Supports Serial ATA Gen-2 signaling 1 Supports Serial ATA Gen-1 signaling 0 Reserved (0)</p>
77		Reserved for SATA
78	Optional	<p>Serial ATA Features</p> <p>15-7 Reserved 6 Software settings preservation supported 5 Reserved 4 In-order data delivery supported 3 Drive initiates link power management requests 2 DMA Setup FIS Auto-Activate optimization supported 1 DMA Setup FIS non-zero buffer offsets supported 0 Reserved (0)</p>
79	Optional	<p>Serial ATA Features Enabled</p> <p>15-7 Reserved 6 Software settings preservation enabled 5 Reserved 4 In-order data delivery enabled 3 Drive initiated link power management requests enabled 2 DMA Setup FIS Auto-Activate optimization enabled 1 DMA Setup FIS non-zero buffer offsets enabled 0 Reserved (0)</p>
80-255		Defined in ATA/ATAPI-7

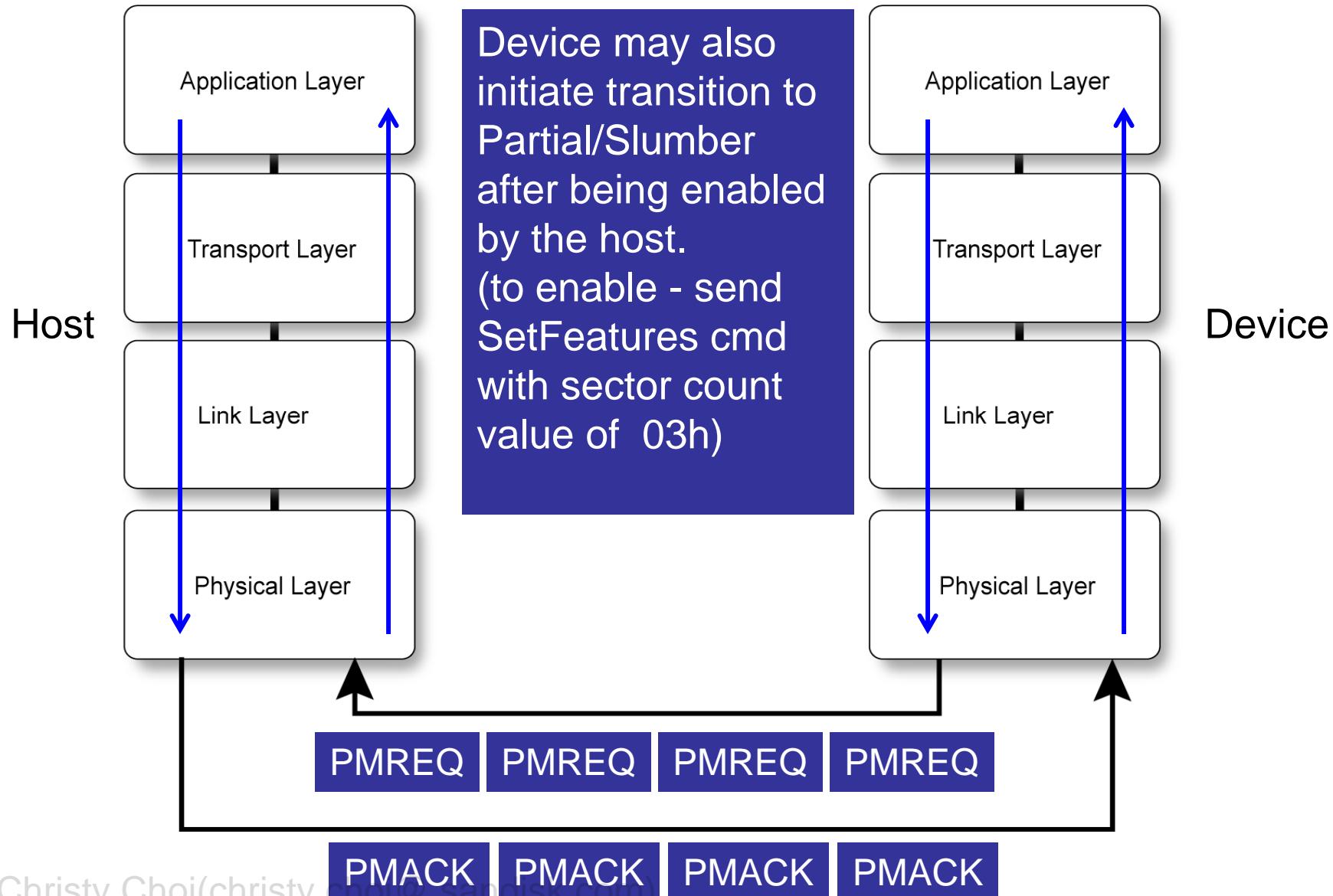
Host Initiated Handshake Sequence



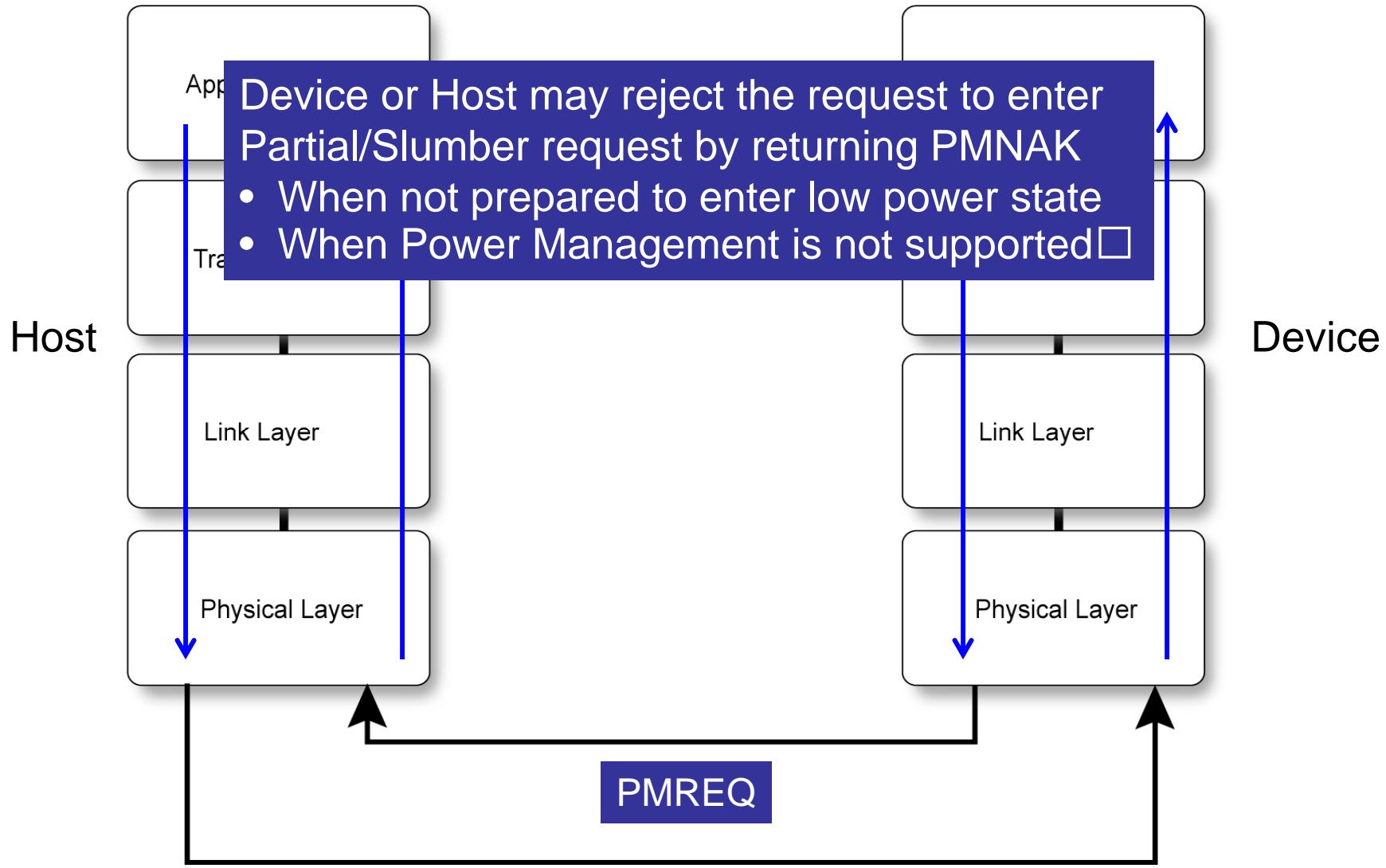
Device Identify Data - SATA II

Word Offset	R/O	Description
0 - 74		Defined in ATA/ATAPI-7
75	Optional	Queue Depth 15-5 Reserved 4-0 Maximum queue depth-1
76		Serial ATA Capabilities 15-11 Reserved 10 Phy event counters supported 9 Host-initiated link Power Management requests received 8 Native Command Queuing supported 7-4 Reserved 3 Reserved for SATA 2 Supports Serial ATA Gen-2 signaling 1 Supports Serial ATA Gen-1 signaling 0 Reserved (0)
77		Reserved for SATA
78	Optional	Serial ATA Features 15-7 Reserved 6 Software settings preservation supported 5 Reserved 4 In-order data delivery supported 3 Drive initiates link power management requests 2 DMA Setup FIS Auto-Activate optimization supported 1 DMA Setup FIS non-zero buffer offsets supported 0 Reserved (0)
79	Optional	Serial ATA Features Enabled 15-7 Reserved 6 Software settings preservation enabled 5 Reserved 4 In-order data delivery enabled 3 Drive initiated link power management requests enabled 2 DMA Setup FIS Auto-Activate optimization enabled 1 DMA Setup FIS non-zero buffer offsets enabled 0 Reserved (0)
80-255		Defined in ATA/ATAPI-7

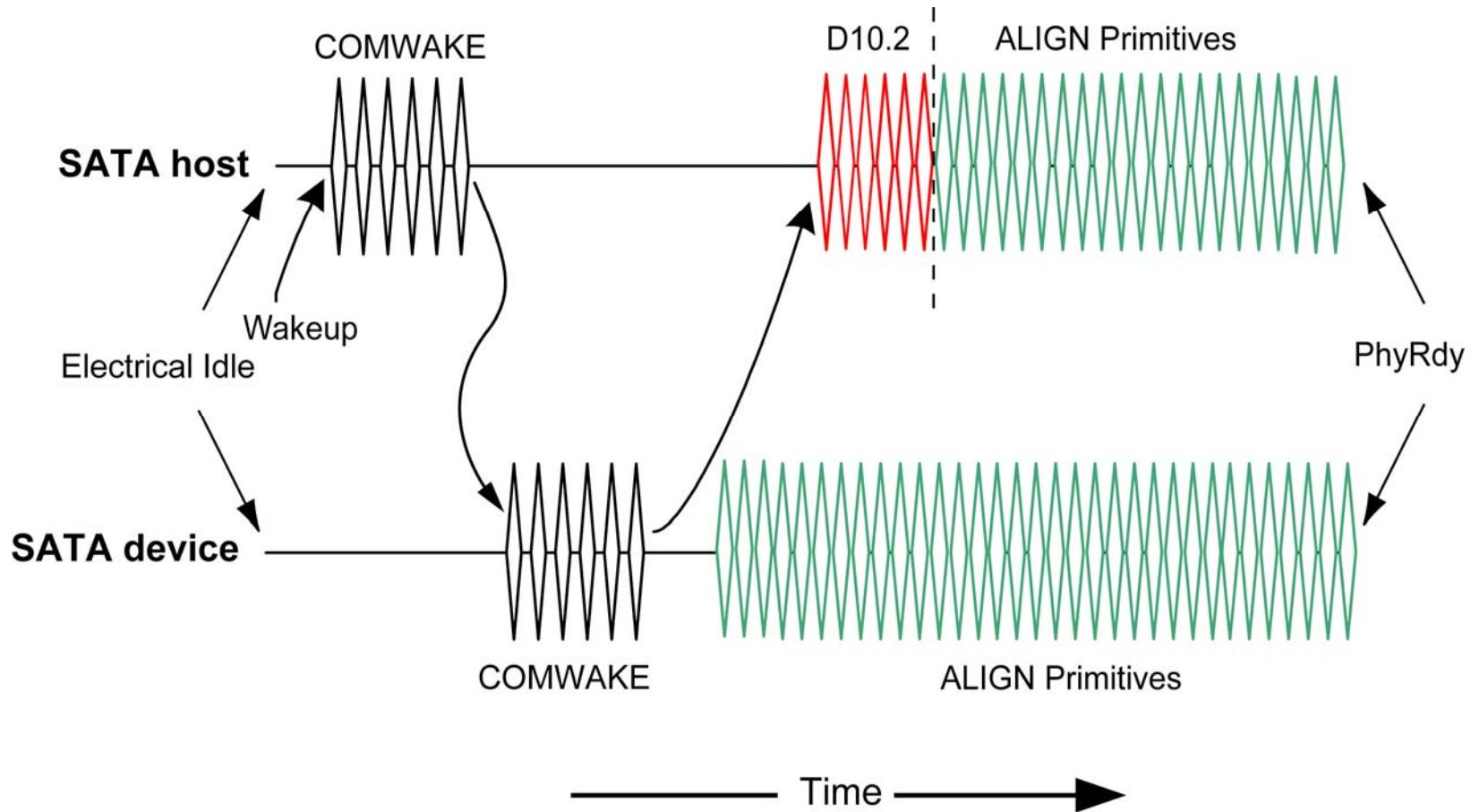
Entering Partial Slumber State



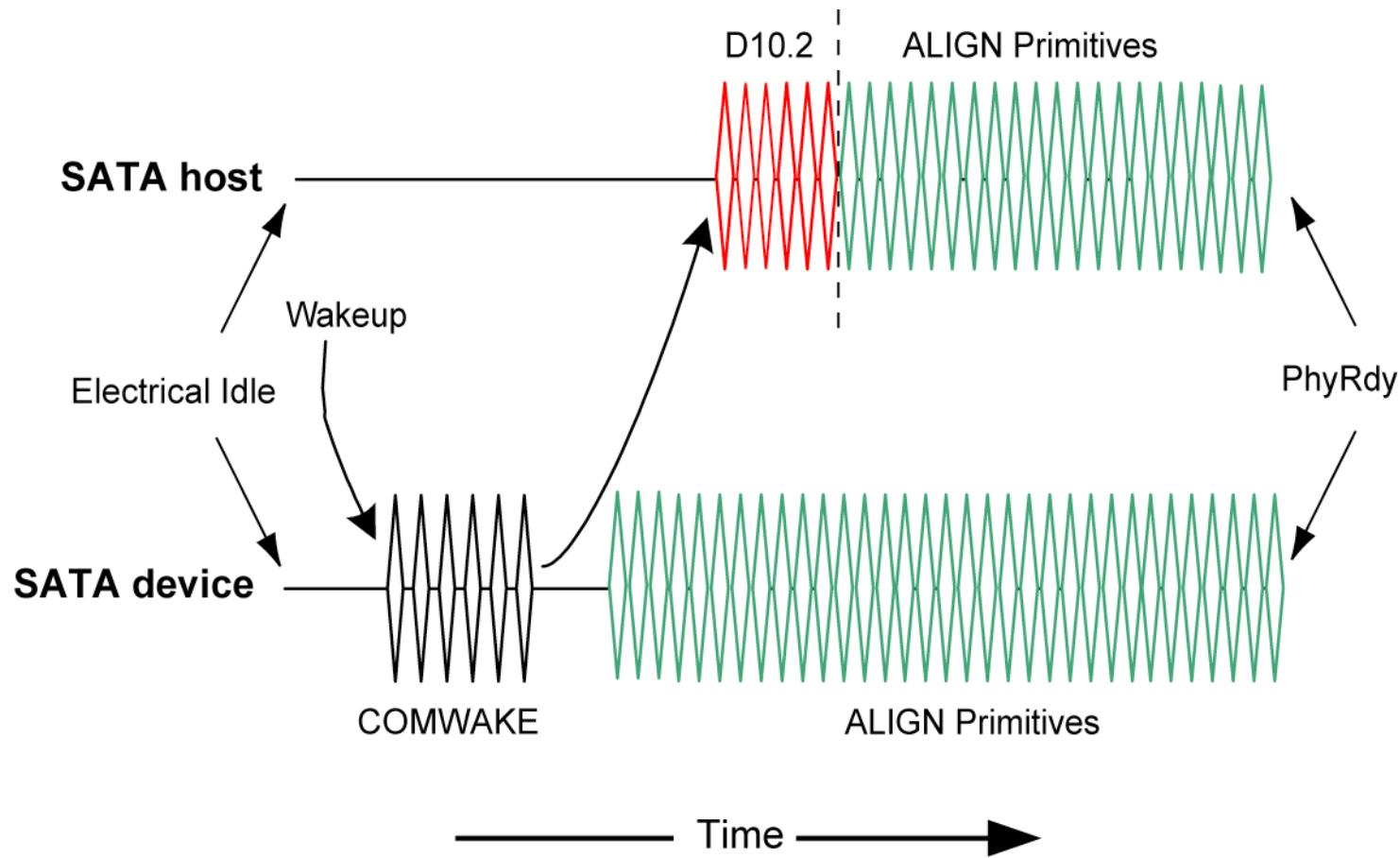
PM No Acknowledge



COMWAKE Protocol (Host Initiated Wakeup)



COMWAKE Protocol (Device Initiated Wakeup)



BIST



Christy Choi(christy.choi@sandisk.com)

Do Not Distribute

The Tests

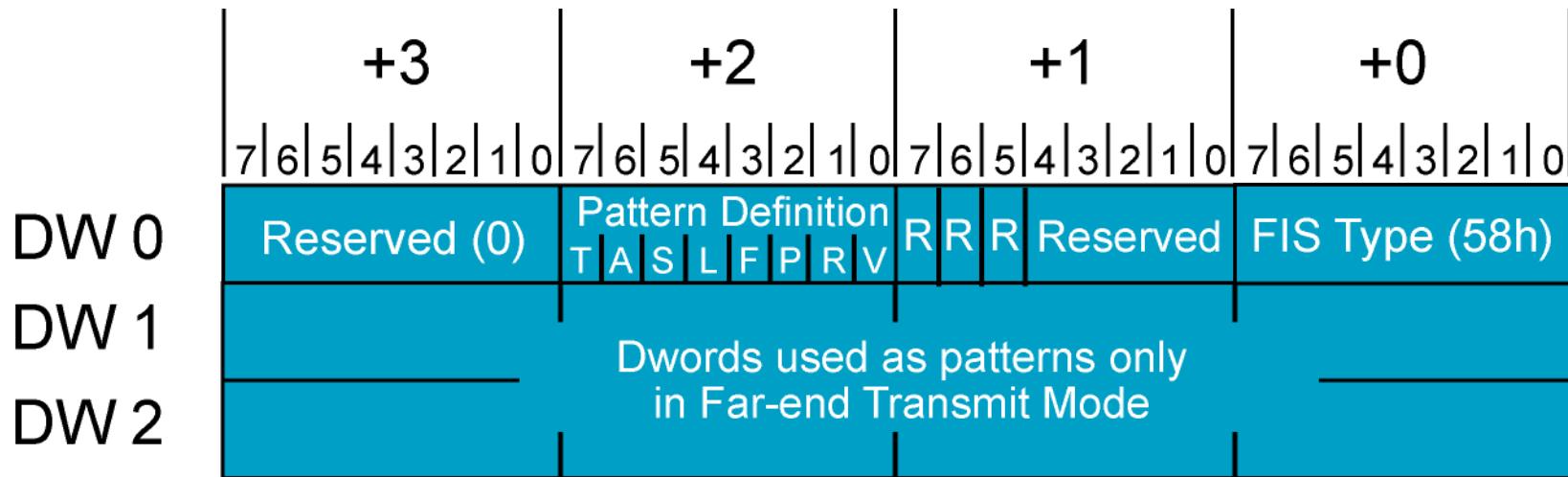
The SATA specification defines several Built-In Self Tests that can be entered under either HBA or Drive control. These tests include:

- Far End Retimed Loopback
- Far End Analog Loopback (optional)
- Far End Transmit Only (optional)

BIST Capability

- Commands are used to initiate delivery of BIST Activate FISs
- The following slides illustrate BIST Activate Transmission and Reception and the possible Transport actions

BIST_Activate (bidirectional) FIS



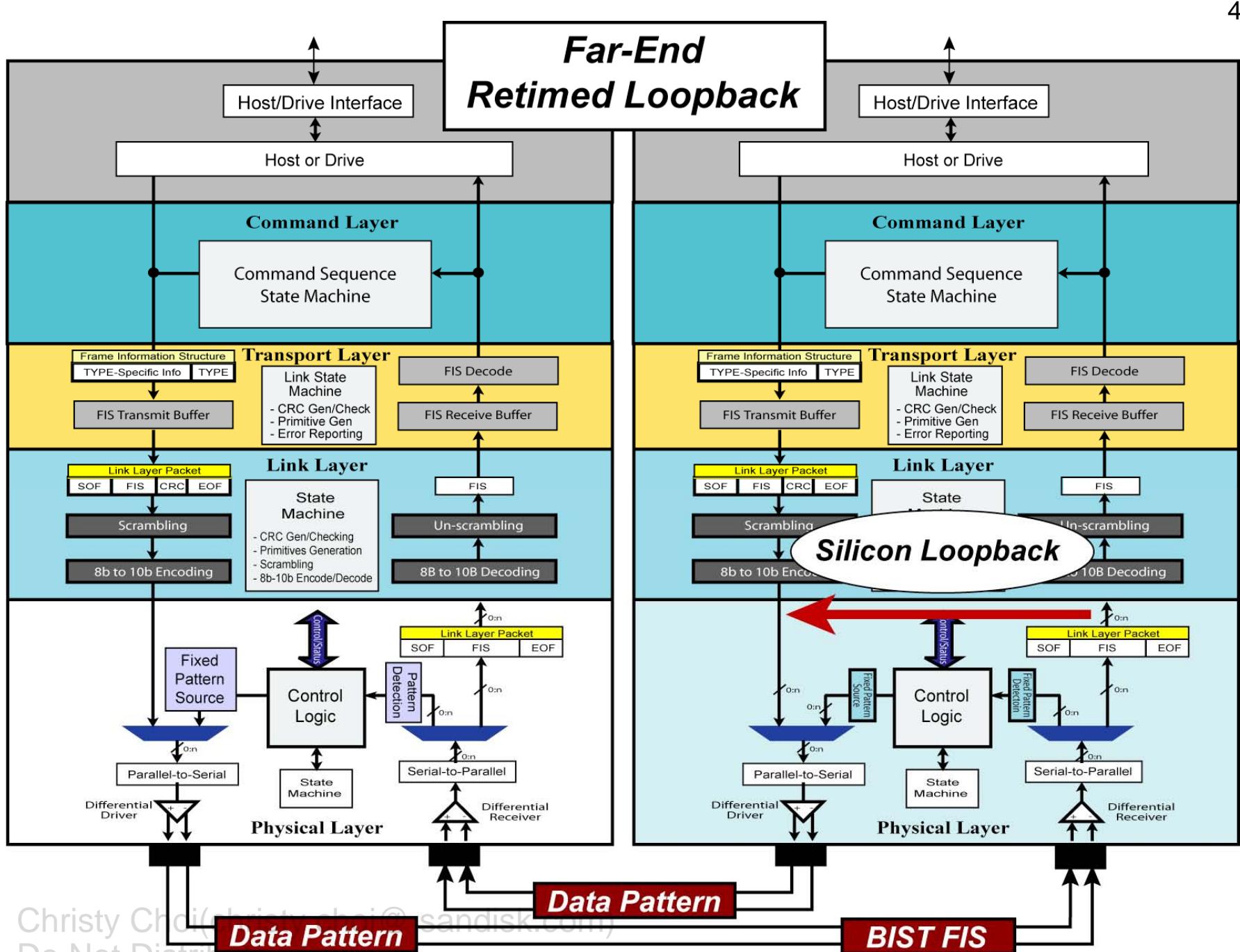
See Table 23-1
For FIS field
descriptions

T = Far end transmit-only mode
 A = ALIGN Bypass (only when T=1)
 S = Bypass Scrambling (only when T=1)
 L = Far End Retimed
 F = Far End Analog Loopback
 P = Primitive bit (only when T=1)
 R = Reserved (0)
 V = Vendor Unique Test Mode

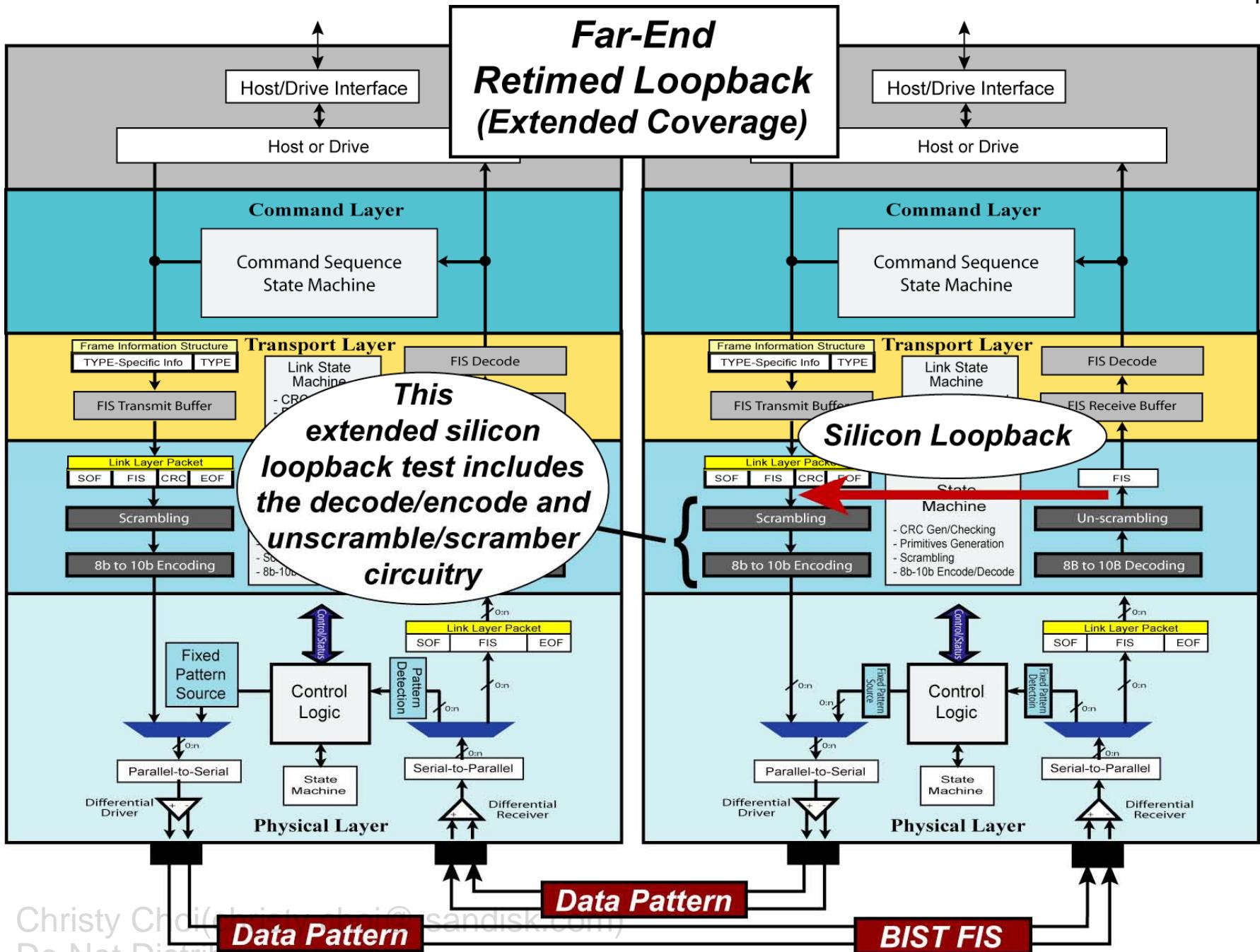
Test Variations

BIST Mode	F	L	T	P	A	S	V
Far End Retimed Loopback	1	0	0	0	0	0	0
Far End Analog Loopback	0	1	0	0	0	0	0
Far End Transmit Only with ALIGN primitives and scrambled data	0	0	1	0	0	0	0
Far End Transmit Only with ALIGN primitives but without scrambled data	0	0	1	0	0	1	0
Far End Transmit Only without ALIGN primitives but with scrambled data	0	0	1	0	1	0	0
Far End Transmit Only without ALIGN primitives and without scrambled data	0	0	1	0	1	1	0
Far End Transmit primitives with ALIGNs	0	0	1	1	0	na	0
Far End Transmit primitives without ALIGNs	0	0	1	1	1	na	0
Vend or Specific BIST mode	na	na	na	na	na	na	1

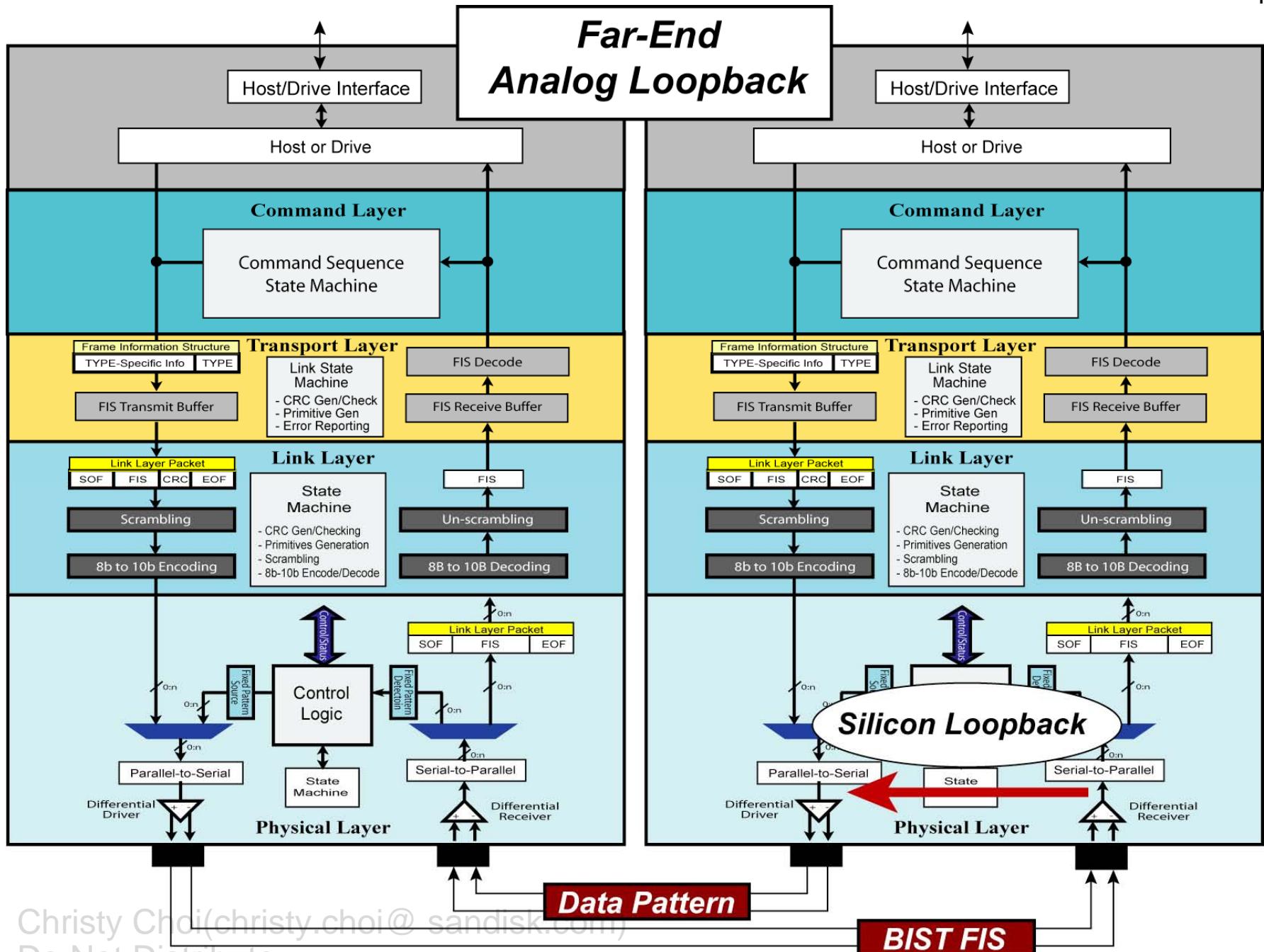
Far-End Retimed Loopback



Far-End Retimed Loopback (Extended Coverage)



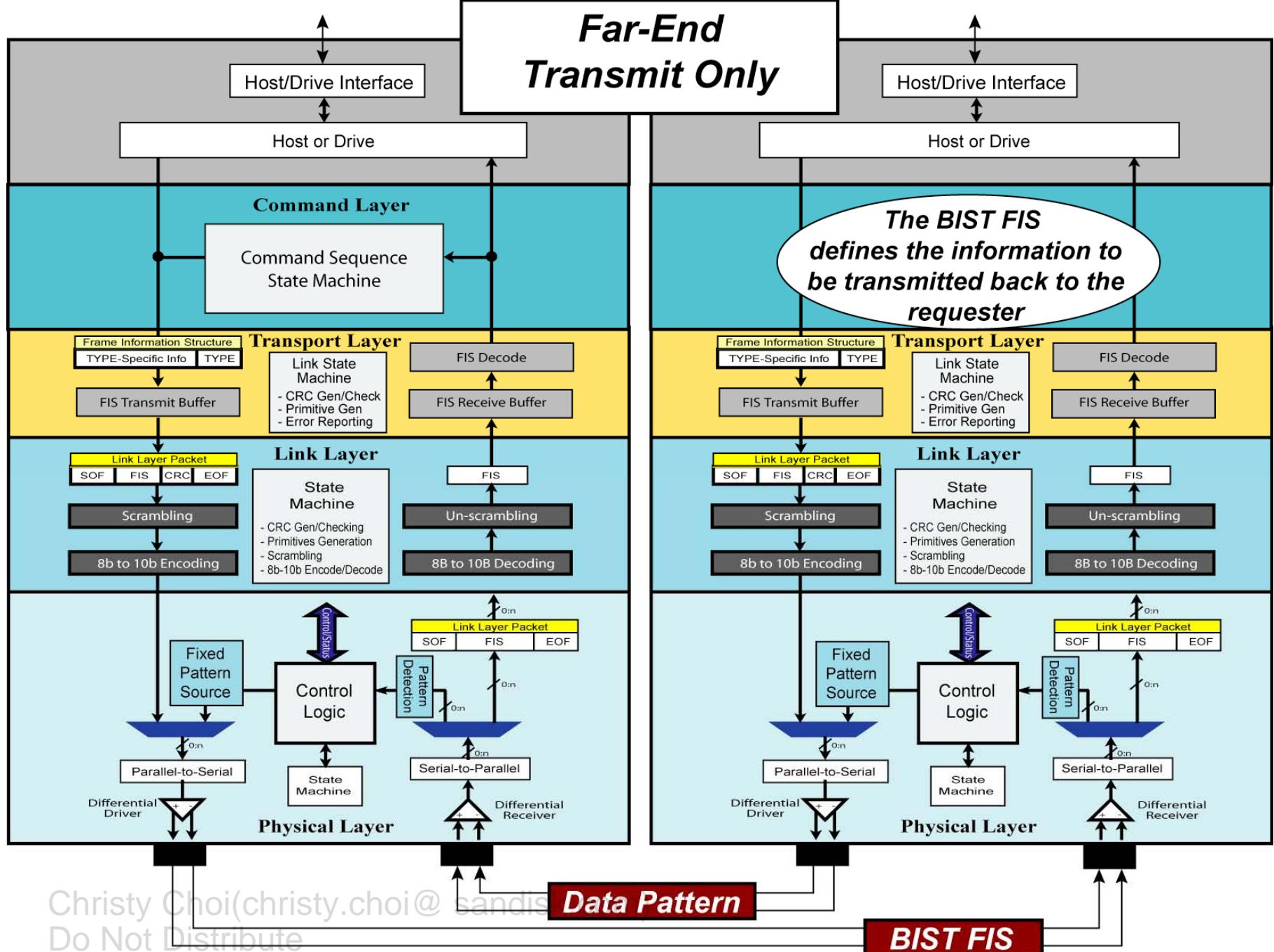
Far-End Analog Loopback



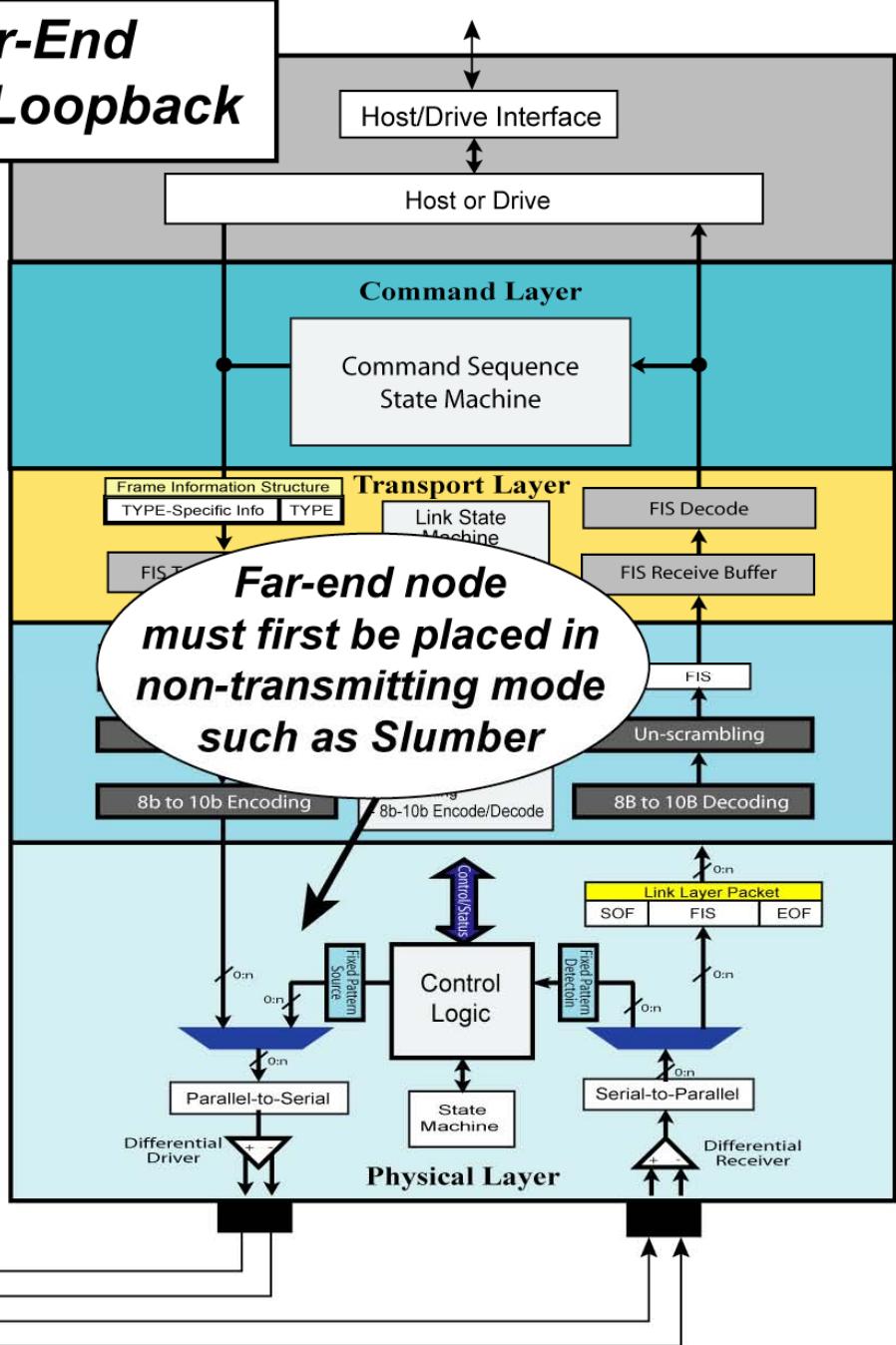
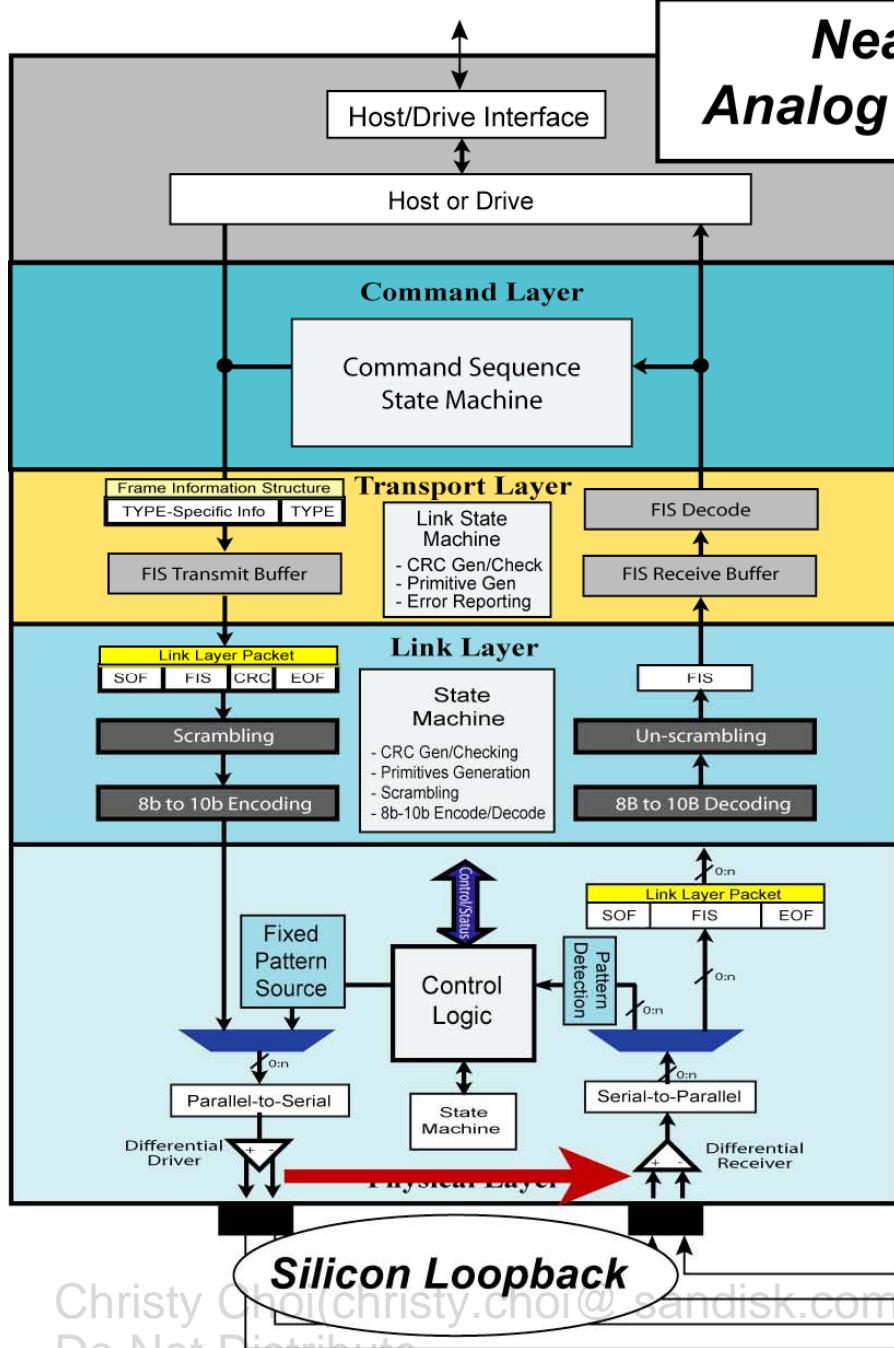
Far End Transmit Only Mode

- In this mode, the device at the far end of the link begins to transmit test patterns. The patterns are designed to check interface compliance and signal integrity.
 - Low transition density bit patterns
 - High transition density bit patterns
 - Low frequency spectral component bit patterns
 - Simultaneous switching outputs bit patterns

Far-End Transmit Only



Near-End Analog Loopback



Test Patterns

The specification defines both compliant and non-compliant test patterns. Their intended use is as follows:

- Non-compliant patterns are simply a set of encoded values that are repeated continuously and need not be presented in a standard Data FIS format. The intended use of the non-compliant patterns is for jitter measurements, electrical parameters tests, and signal quality assessment across the physical environment.
- Compliant patterns are presented in a standard frame format as illustrated in Figure 23-7 (below). Also, ALIGN primitives are injected into the data stream as required. These patterns are intended for frame error rate evaluation and in-system testing.

