

Serial Attached SCSI

SCSI upper layers



by Rob Elliott

HP Industry Standard Servers

Server Storage Advanced Technology

elliott@hp.com <http://www.hp.com>

30 September 2003

- These slides are freely distributed by HP through the SCSI Trade Association (<http://www.scsita.org>)
- STA members are welcome to borrow any number of the slides (in whole or in part) for other presentations, provided credit is given to the SCSI Trade Association and HP
- This compilation is © 2003 Hewlett-Packard Corporation



SAS standard layering

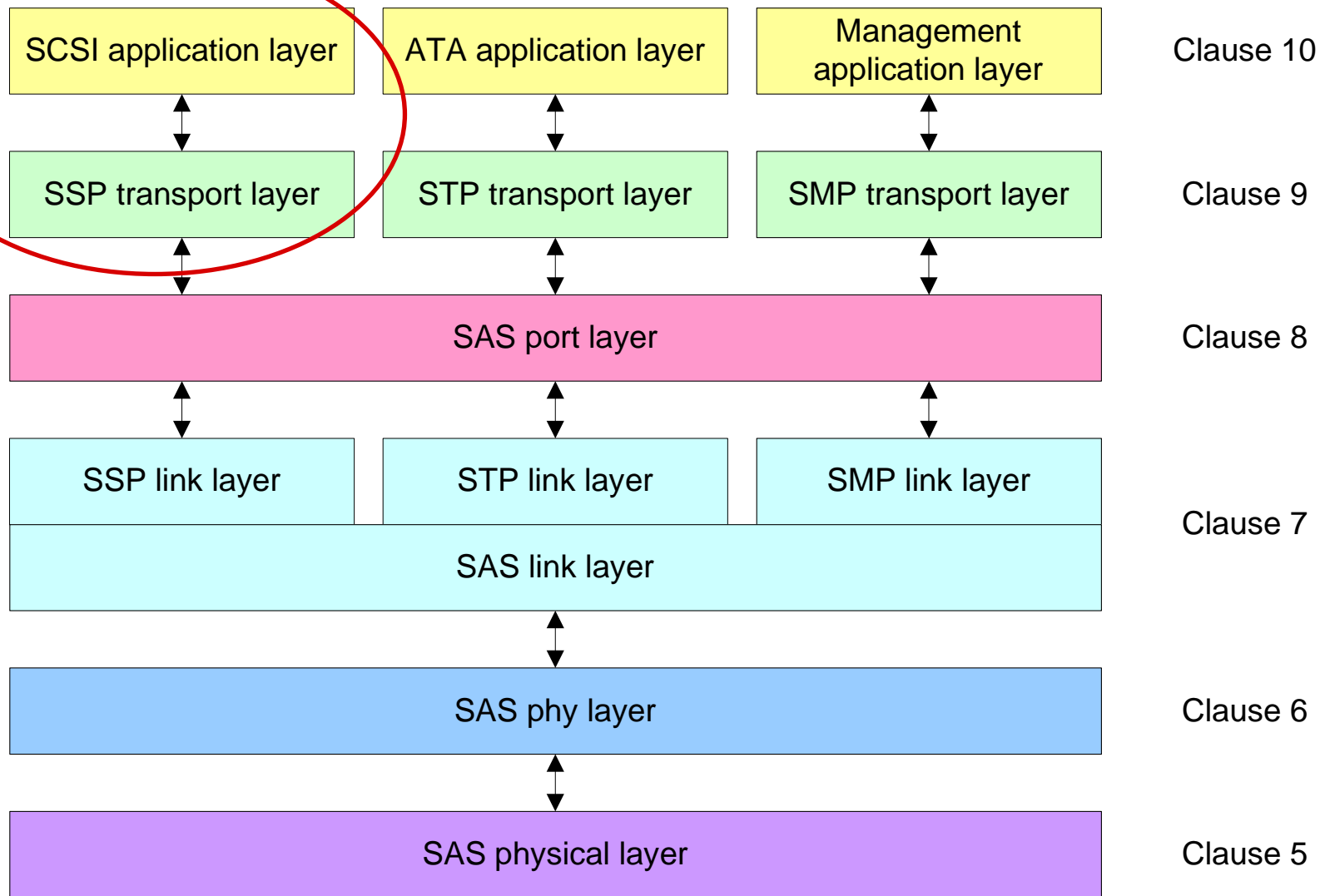


Table of contents

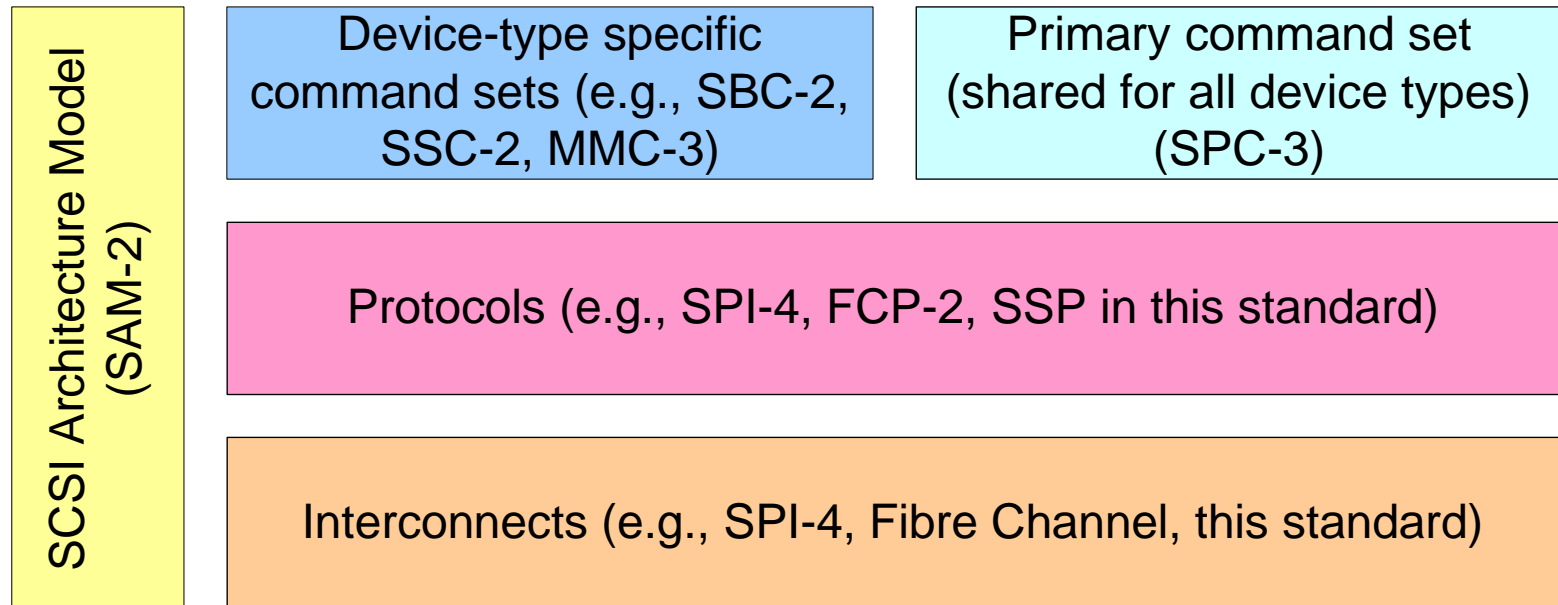
- SCSI standards
- SCSI architecture
 - SCSI commands
 - SCSI data and status
 - SCSI task management functions
 - SCSI application layer – protocol services
- SSP transport layer
- SCSI application layer – mode pages
- SCSI application layer – log pages
- SCSI application layer – power conditions and spinup
- Wrap up

SCSI standards

SCSI standards structure



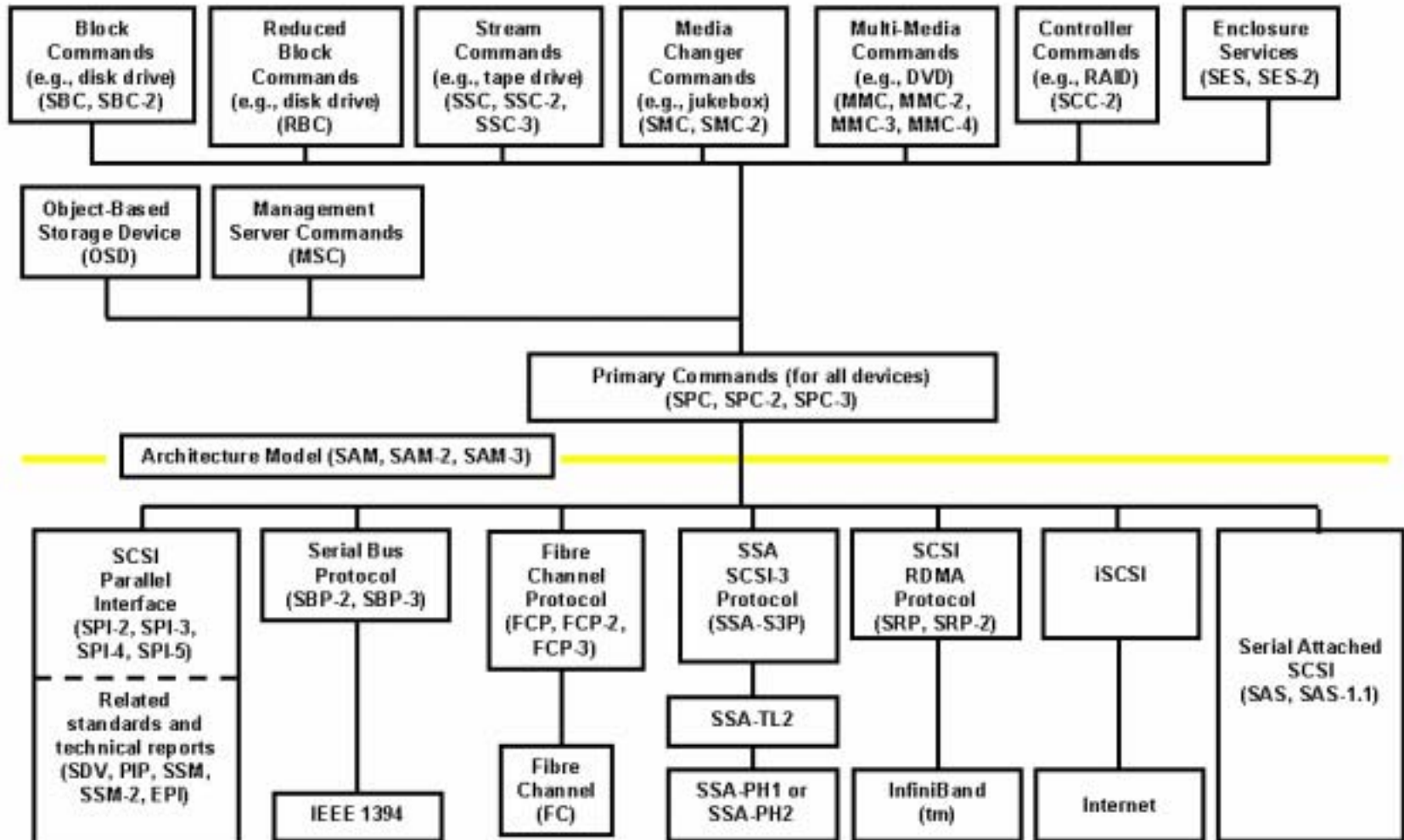
- SCSI-3 is split into different standards
- SAS is both a protocol and interconnect standard



SCSI standards roadmap (comprehensive)



SCSI Architecture Roadmap

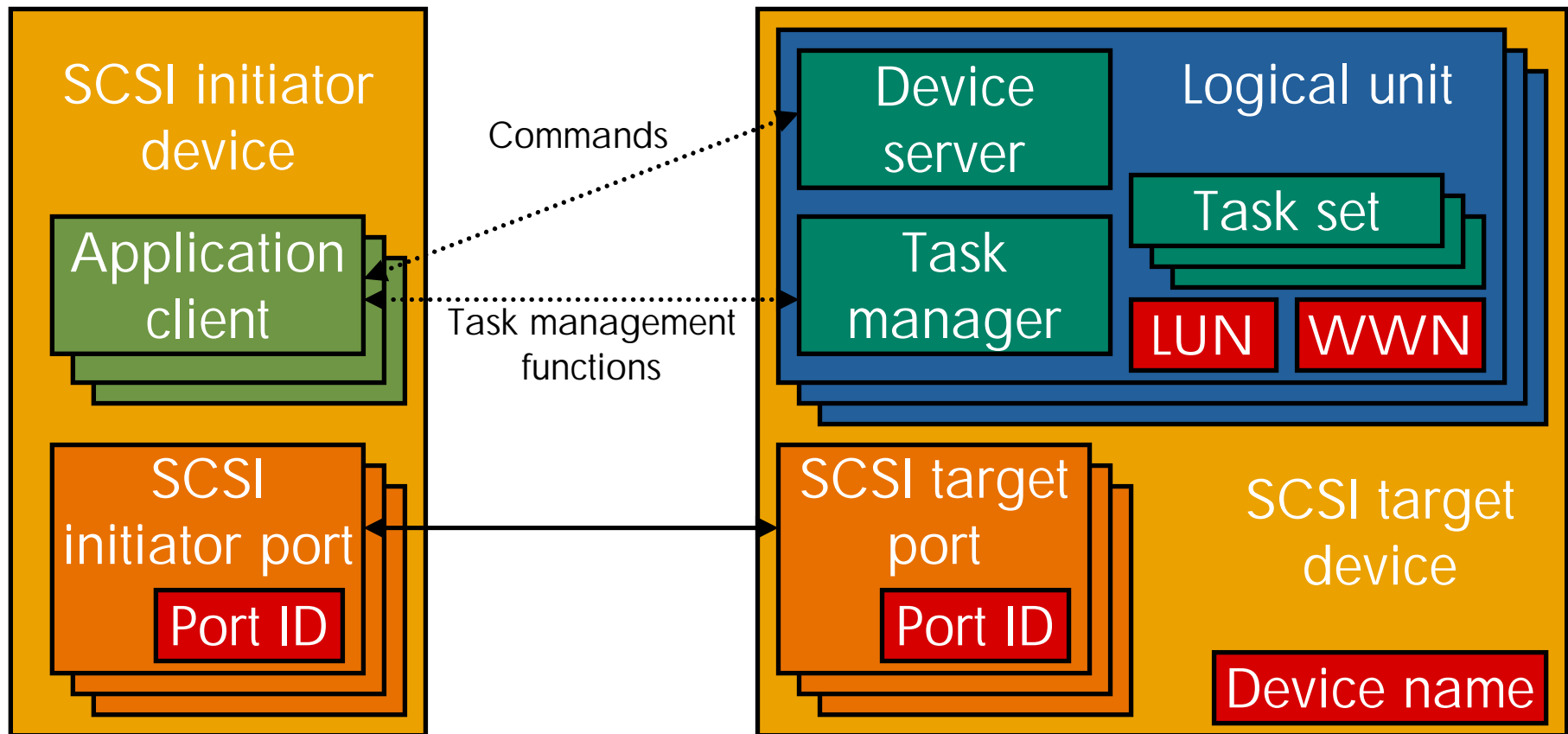


SCSI architecture

SCSI objects



- SCSI domain contains SCSI initiator devices and SCSI target devices
- Application clients communicate with device servers and task managers using SCSI initiator ports and SCSI target ports



SCSI addressing

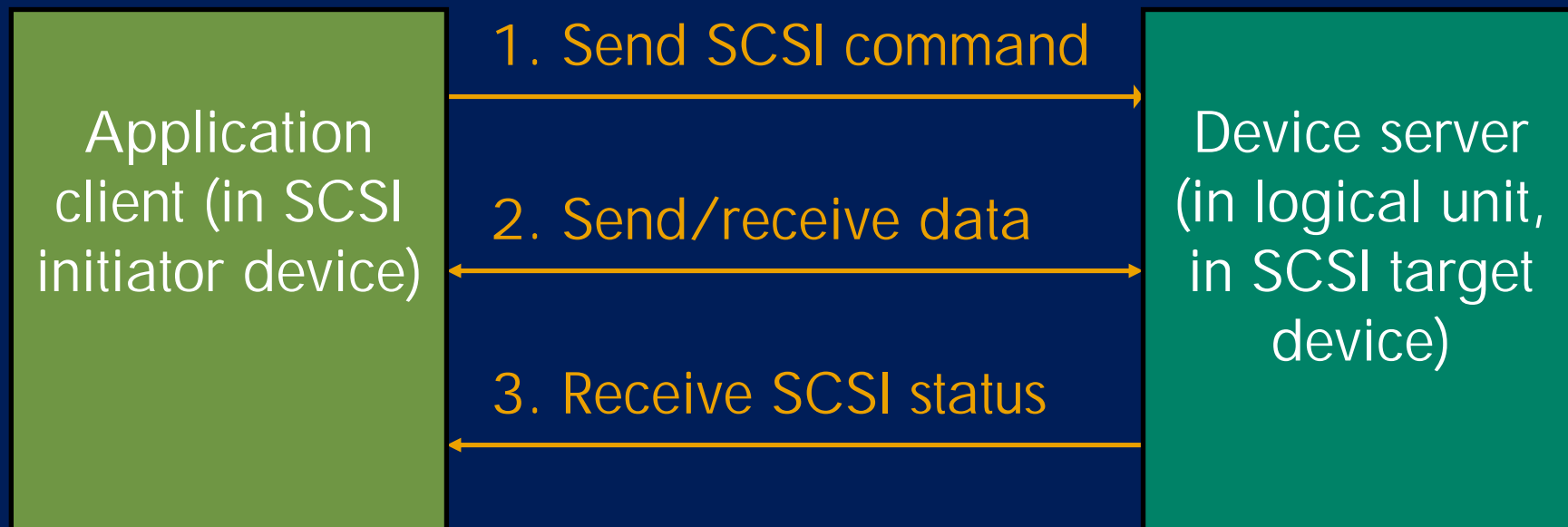


- Each SCSI initiator port and SCSI target port has a **SCSI port identifier**
 - unique in the SCSI domain
 - In SAS, the 64-bit SAS address serves as the SCSI port identifier
- Each logical unit is assigned a **logical unit number (LUN)** within the SCSI target device
 - 64 bits
 - Subdivided into 4 “levels” 16 bits each
 - LUN 0 is required
- Each logical unit also has a **logical unit name**
- Each SCSI target device with SAS target ports has a **device name**

SCSI command processing



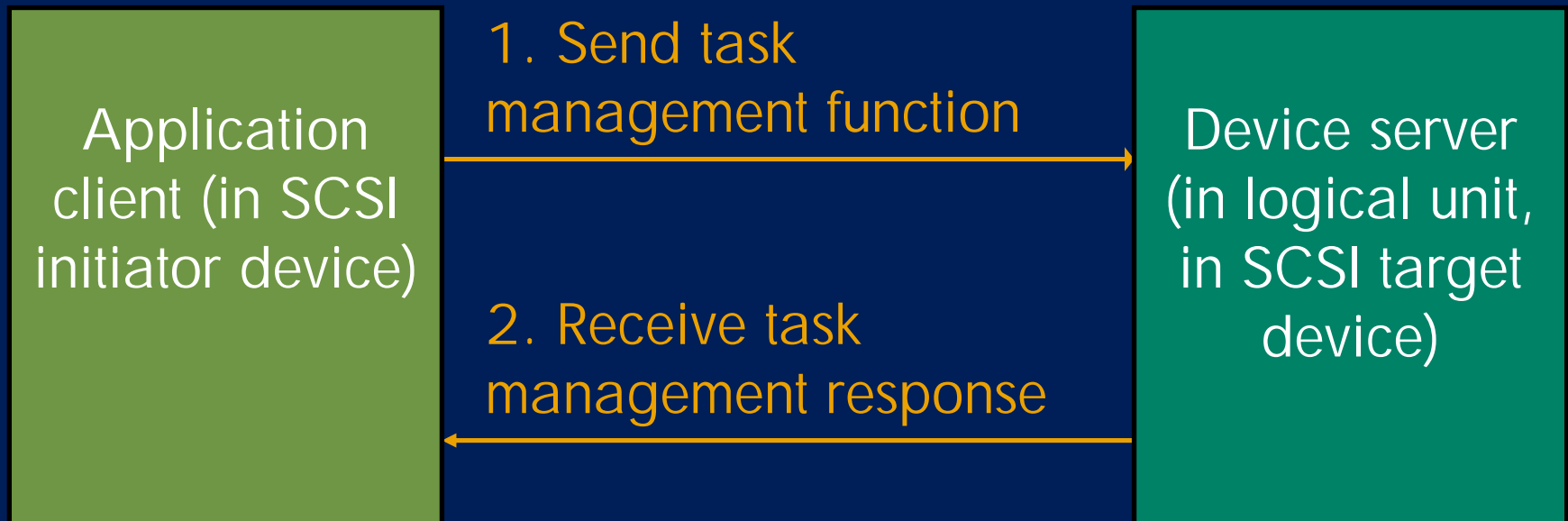
- Application client sends a SCSI command (task) to device server
 - using a SCSI initiator port and SCSI target port
- Device server transfers data
- Device server returns SCSI status



SCSI task management function processing



- Application client sends task management functions to device server
 - using a SCSI initiator port and SCSI target port
- Device server returns task management response
- No data transfer involved



Miscellaneous SCSI terminology



- **Nexus** means relationship
 - **I_T nexus** – initiator and target
 - **I_T_L nexus** – initiator, target, and logical unit
 - **I_T_L_Q nexus** – initiator, target, logical unit, and tag
- A **task** is usually synonymous with a command
- Each command has a unique **tag**
 - In SAS, 16 bits
 - In SAS, the tag is unique within the I_T nexus
 - shares its namespace with the task management function associations
- Each task management function has a unique **association**
 - In SAS, 16 bits
 - In SAS, the association is unique within the I_T nexus
 - shares its namespace with the command tags

Miscellaneous SCSI terminology 2



- A command is sent in a **Command Descriptor Block (CDB)**
 - **Operation Code** field distinguishes commands
 - CDBs can be 6, 10, 12, or 16 bytes long
 - Variable-length longer CDBs also allowed
 - CDB formats defined in SPC-3
- Task management function is sent in a protocol-specific manner
 - Only one byte generally needed to specify the function

SCSI commands

SCSI command categories



- Four categories of SCSI commands
 - **Non-data commands (N)**
 - No data is transferred
 - **Write commands (W)**
 - data transferred from initiator to target
 - Write data is called Data-Out (out of initiator)
 - e.g. write to disk
 - **Read commands (R)**
 - Data transferred from target to initiator
 - Read data is called Data-In (in to initiator)
 - e.g. read from disk
 - **Bidirectional commands (B)**
 - Data transferred in both directions
 - Data-Out and Data-In
 - So far, just for specialized uses (disk drives and object storage devices)

SCSI command sets



- INQUIRY command returns which command sets a logical unit supports (**Peripheral Device Type** and some other fields)
- Commands are defined in command set standards
 - SPC-3 annex lists all the commands and their **Operation Codes**

Standard	Name	Types of logical units that implement
SPC-3	Primary	All
SBC-3	Block	Disk drives
SSC-2	Streaming	Tape drives
SES-2	Enclosure Services	Enclosures (JBODs, external RAID)
MMC-5	Multimedia	CDs and DVDs
SMC-2	Media Changer	Tape libraries
SCC-2	Controller	RAID controllers (no known implementations follow this, but many do use its Peripheral Device Type)
OSD	Object	Object-based storage devices

SCSI primary commands



- Primary commands can be implemented by all SCSI logical units
- Four commands are **required** for all logical units:

Command	Type	Description
INQUIRY	R	Returns miscellaneous information about the logical unit. Peripheral Device Type (disk, tape, etc.), Vendor ID, Product ID, Serial Number, etc. Also returns vital product data (VPD) pages which contain Device Identifiers (worldwide names).
REPORT LUNS	R ⁰	Returns a list of the logical unit numbers present in the SCSI target device (mandatory for LUN 0)
REQUEST SENSE	R	Returns logical unit's current sense data
TEST UNIT READY	N	Returns GOOD if the logical unit is ready to accept media-access commands

SCSI primary commands - optional



Command	Type	Description
LOG SELECT/ LOG SENSE	W/R	Read (sense) or write (select) statistical information from the logical unit. Information is defined by log pages .
MODE SELECT/ MODE SENSE	W/R	Read (sense) or write (select) control information from the logical unit. Information is defined by mode pages .
PERSISTENT RESERVATION IN/OUT	R/W	Prevent other initiators from accessing the logical unit.
READ BUFFER/ WRITE BUFFER	R/W	Read or write a memory buffer (for testing and firmware downloads)
REPORT SUPPORTED OPERATION CODES	R	Return a list of commands supported
REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS	R	Returns a list of task management functions supported
SEND DIAGNOSTIC/ RECEIVE DIAGNOSTIC RESULTS	W/R	Perform diagnostics. Information is defined by diagnostic pages .

SCSI block commands



- **Block devices** are **disk drives** or RAID controller volumes
- Random access
- Always read or write blocks at a time
 - 512 byte block common
 - 520 bytes for some applications (e.g. Tandem NonStop servers)
 - 4096 bytes may eventually be used
- SBC-1
 - 32-bit LBA (logical block address) limits disk sizes to 2 TB with a 512 byte block size
- SBC-2
 - 64-bit LBA breaks the 2 TB limit
- Only 3 commands are mandatory:

Command	Type	Description
FORMAT UNIT	W	Format the medium
READ	R	Read from the medium
READ CAPACITY	R	Return the capacity of the medium

SCSI block commands - optional



- Some of the more important optional SBC commands:

Command	Type	Description
LOCK UNLOCK CACHE	N	Force specified blocks to remain in cache
PREFETCH	N	Transfer specified blocks into cache
READ DEFECT DATA	R	Return list of defective blocks
REASSIGN BLOCKS	W	Reassign selected blocks to another area on the media
START STOP UNIT	N	Start or stop the rotating media
SYNCHRONIZE CACHE	N	Force cache flush for specified blocks
VERIFY	W	Test reading from the medium, optionally comparing with specified data
WRITE	W	Write to the medium
WRITE AND VERIFY	W	Write to the medium, then read and verify it
WRITE SAME	W	Write one block of data to the medium several times in a row

SCSI stream commands



- Stream devices are tape drives
- Not random access
- Position dependent on previous commands
- SSC-1
 - Tape position is implied
 - Hard to recover from errors
 - Lose a write command and data is written to the wrong position
- SSC-2
 - Adds “explicit address mode” commands where each command specifies the offset of the data
 - Tape drive checks the position before transferring data
 - Supports larger tape sizes
- Tape volumes are divided into partitions
 - Partitions contain data, filemarks, and setmarks

SCSI stream commands - mandatory



- These SSC commands are mandatory:

Command	Type	Description
ERASE	N	Erase all or part of the medium
LOCATE	N	Position to specified location on the medium
READ	R	Read from medium
READ BLOCK LIMITS	R	Report block sizes supported
READ POSITION	R	Report the current position of the medium
REPORT DENSITY SUPPORT	R	Report supposed densities (types of tapes)
REWIND	N	Rewind to beginning of partition
SPACE	N	Move medium forward or backwards
WRITE	W	Write to medium
WRITE FILEMARKS	N	Write filemarks to the medium

SCSI enclosure services



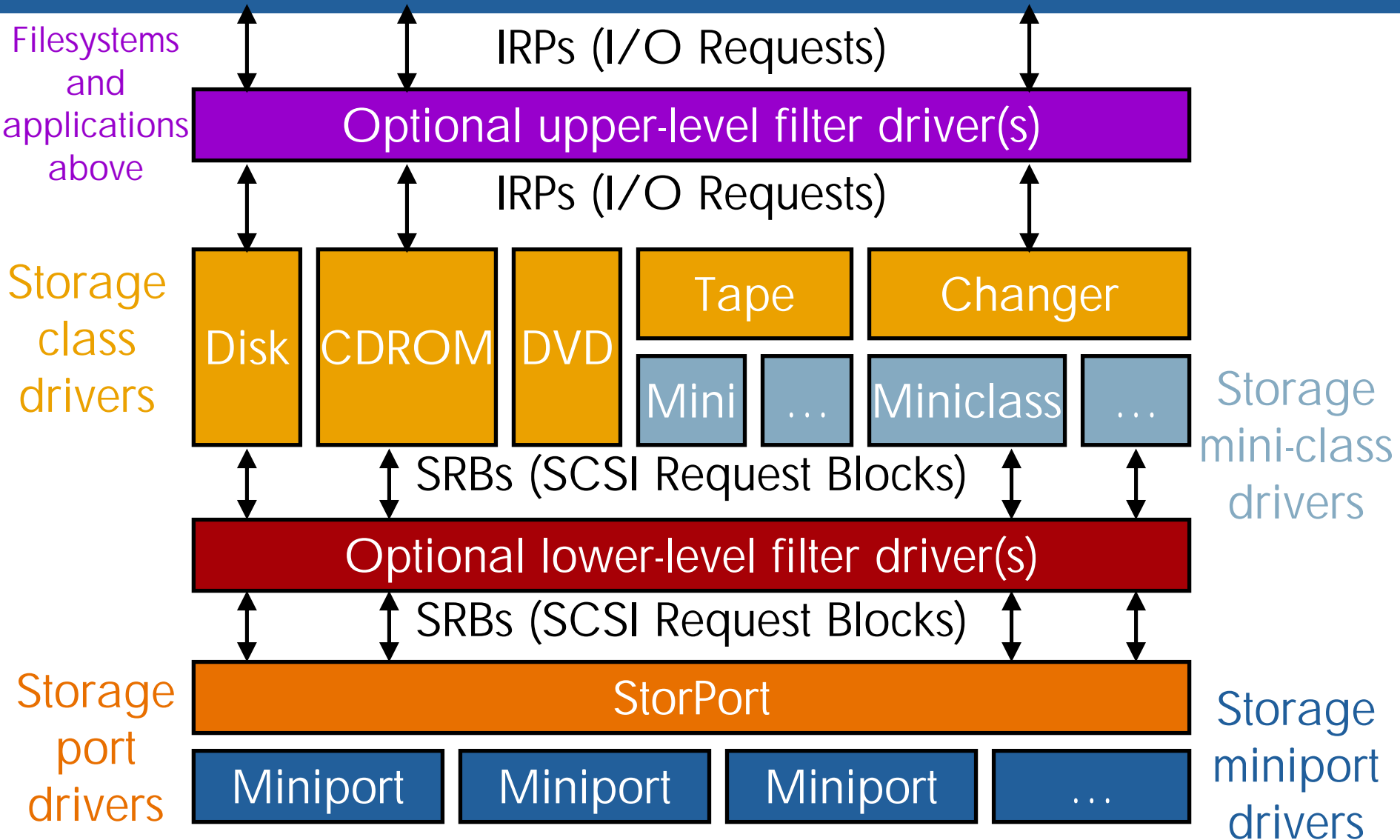
- Enclosures may include enclosure service processors to manage fans, temperature sensors, LEDs, etc.
- SES defines a series of **diagnostic pages** to communicate with the enclosure services processor over SCSI
 - SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands used to transmit the pages
 - Diagnostic page codes 00h-1Fh are defined by SES, regardless of the **Peripheral Device Type** reported in INQUIRY

Standalone or attached enclosure services



- Enclosure services can be a standalone logical unit or a part of another logical unit (attached)
 - Fibre Channel disk drives implement the SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands
 - Forwards accesses to **diagnostic pages** 00h – 01Fh through the “ESI interface” on the drive connector
 - Backplane connects ESI to an enclosure services processor which services the requests
 - Disk drive handles other **diagnostic pages** itself
 - In SAS, some expanders host a standalone SES logical unit
 - Has its own SAS address and logical unit number (LUN 0)
 - responds to SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands

Microsoft Windows storage driver stack





SCSI data and status

SCSI data transfers

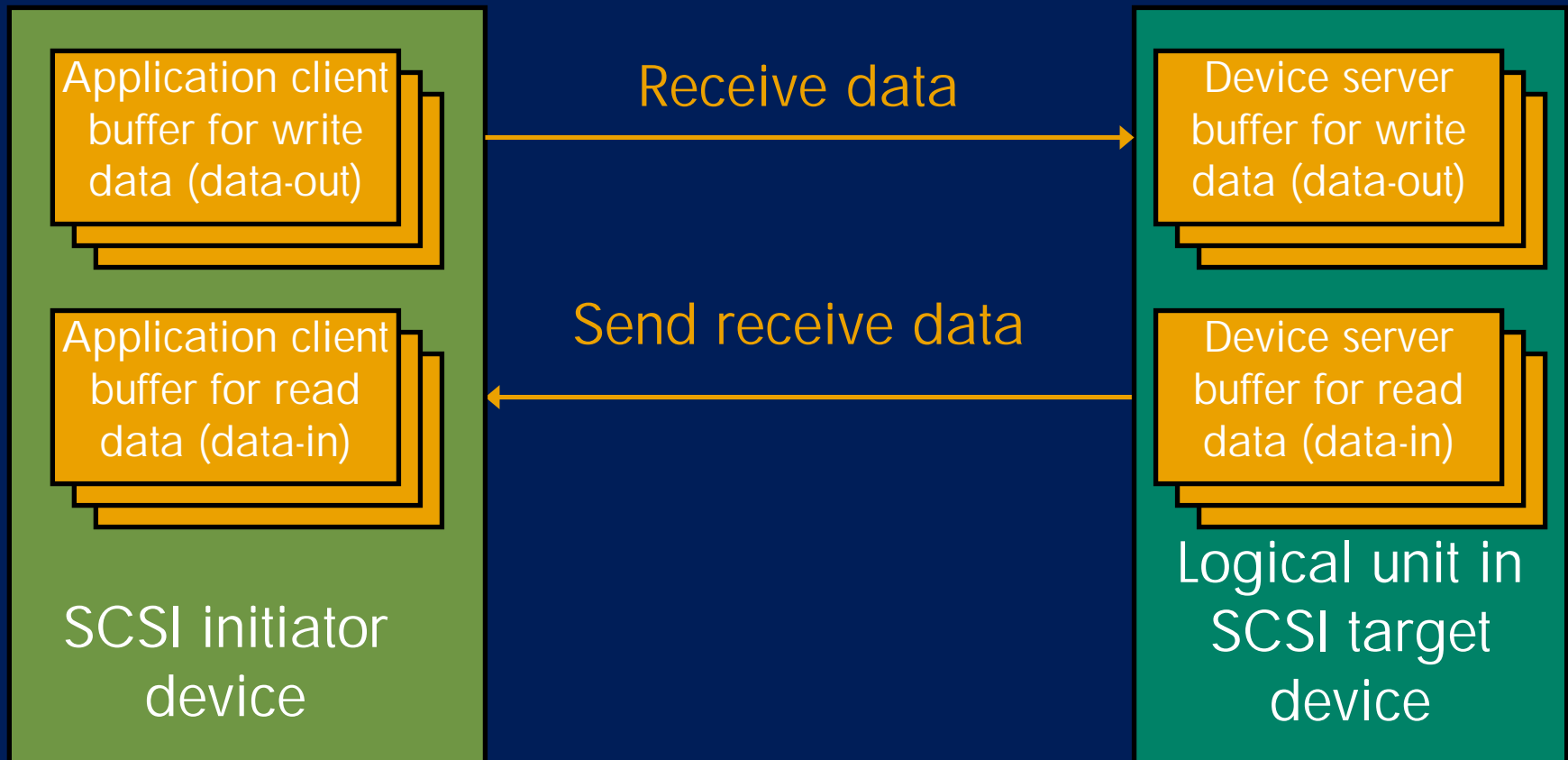


- Logical unit controls when data transfers occur
- When an application client sends a command, it must be prepared to provide all the data at any time
- After a logical unit receives a command, it can wait as long as it wants to fetch the write data or deliver the read data
- Initiator does not “push” data to the target
 - Reduces target buffer requirements
 - No need to reserve space for unsolicited data
 - Exception: “first burst” data sometimes supported
 - more important for long-latency protocols like iSCSI than low-latency environments like SAS

SCSI data transfer buffers



- Application client's buffers for write data and read data are separate from each other and from other commands



SCSI status codes



- When the command is done, the logical unit returns a **Status** byte
- Most common **Status** codes:

Value	Status	Description
00h	GOOD	The command completed successfully
02h	CHECK CONDITION	The command failed. Sense Data (a > 8 byte data structure) is returned along with the Status indicating why.
08h	BUSY	Command refused - the logical unit is temporarily busy (unknown reason). Try again.
28h	TASK SET FULL	Command refused - the logical unit is busy handling other commands. Try again when a command completes (if that time can be determined).
18h	RESERVATION CONFLICT	Command refused - the logical unit is reserved by some other initiator port.

- Complete list of **Status** codes defined in SAM-3 (not SPC-3)

SCSI sense keys



- When CHECK CONDITION status is returned, **sense data** is sent explaining the reason
 - Includes a **Sense Key** field (4 bits) explaining the basic reason
 - Complete list of **Sense Keys** in SPC-3
 - Some common **Sense Keys**:

Value	Sense key	Description
0h	NO SENSE	No additional information
1h	RECOVERED ERROR	Command succeeded, barely
2h	NOT READY	Logical unit not ready for the command (e.g. not spinning when a READ was requested)
3h	MEDIUM ERROR	Non-recovered error (e.g. bad tape)
4h	HARDWARE ERROR	Non-recovered error (e.g. bad controller)
5h	ILLEGAL REQUEST	Problem with the requested command (e.g. bad field in CDB)
6h	UNIT ATTENTION	Command ignored; some other event happened that needs to be reported (e.g. a reset occurred)
Bh	ABORTED COMMAND	Command aborted for some reason

SCSI additional sense codes



- Sense data includes **Additional Sense Code** which provides more detail than the Sense Key
 - Comprised of two bytes
 - ASC (additional sense code)
 - ASCQ (additional sense code qualifier)
 - **Additional Sense Codes** are defined in SPC-3 (see Annex)
- Examples:

Status	Sense Key	Additional Sense Code
CHECK CONDITION (02h)	Illegal Request (5h)	Invalid Field in CDB (ASC=24h, ASCQ=00h)
CHECK CONDITION (02h)	Unit Attention (6h)	Power On Occurred (ASC=29h, ASCQ=01h)
GOOD	N/A	N/A
BUSY	N/A	N/A

SCSI sense data



- Additional fields
 - **Information** – defined by the device type or command
 - **Command-Specific Information** – defined by the device type or command
 - **Sense Key-Specific** – meaning differs based on the Sense Key
 - **Field Replaceable Unit Code** - optional
 - **Filemark** – for tape commands
 - **EOM** (end of medium) – for tape commands
 - **ILI** (incorrect length indicator) – for tape and block commands
- Sense data format is defined in SPC-3
 - Fixed length (18 bytes) and variable-length formats
 - Variable length format required for > 2 TB disk drives (see SBC-2)

SCSI task management functions

SCSI task management functions



- Task management functions are defined in SAM-3
 - Behavior is defined in SAM-3
 - How to request them is defined by each transport protocol
- Each function has either I_T, I_T_L, or I_T_L_Q nexus scope

Command	Scope	Description
QUERY TASK	I_T_L_Q	Check if a specified command is still being processed
ABORT TASK	I_T_L_Q	Abort a specific command
ABORT TASK SET	I_T_L	Abort all commands from an initiator
CLEAR TASK SET	I_T_L	Abort all commands from all initiators
CLEAR ACA	I_T_L	Clear an auto contingent allegiance condition
LOGICAL UNIT RESET	I_T_L	Reset the logical unit

- TARGET RESET was previously defined but is now obsolete

SCSI application layer – protocol services

Transport protocol services



- SAM-3 defines command and task management processing using transport protocol service procedure call model
 - Interface between application client and SCSI initiator port
 - Interface between device server and SCSI target port
- In SAS, these serve as the interface between the application layer and the transport layer state machine
 - Transport layer state machines are part of the SCSI port object
 - SAS does not define application layer state machines
 - The application layer is assumed to comply with SAM-3's definition of the transport protocol services

Initiator transport protocol services



- Interface between application client (AC) and SCSI initiator port (IP)

Protocol service	Caller	Description
1. Send SCSI Command	AC	Send a command
2. Command Complete Received	IP	SCSI status has arrived
or		
1. Send Task Management Request	AC	Send a task management function
2. Received Task Management Function-Executed	IP	Task management function response has arrived

Initiator transport protocol service example



- Conceptual model
 - Application client calls Send SCSI Command (...);
 - This function is in a library provided by the SCSI initiator port
 - SCSI initiator port sends the command
 - Application client registers a function pointer to Command Complete Received (...) with the SCSI initiator port library
 - When a SCSI status arrives
 - SCSI initiator port invokes Command Complete Received (...)
 - This notifies the application client

Target transport protocol services



- Interface between device server (DS) and SCSI target port (TP)

Protocol service	Caller	Description
1. SCSI Command Received	TP	A command arrived
2. Send Data-In	DS	Send some read data
3. Data-In Delivered	TP	Read data sent
2. Receive Data-Out	DS	Fetch some write data
3. Data-Out Received	TP	Write data fetched
4. Send Command Complete	DS	Send SCSI status
or		
1. Task Management Request Received	TP	A task management function arrived
2. Task Management Function Executed	DS	Send task management function response



SSP transport layer

SSP frames



- One common SSP frame format
 - **Frame header:** 24 bytes
 - **Information Unit:** 0 to 1024 bytes
 - **Fill bytes:** 0 to 2 bytes
 - **CRC:** 4 bytes

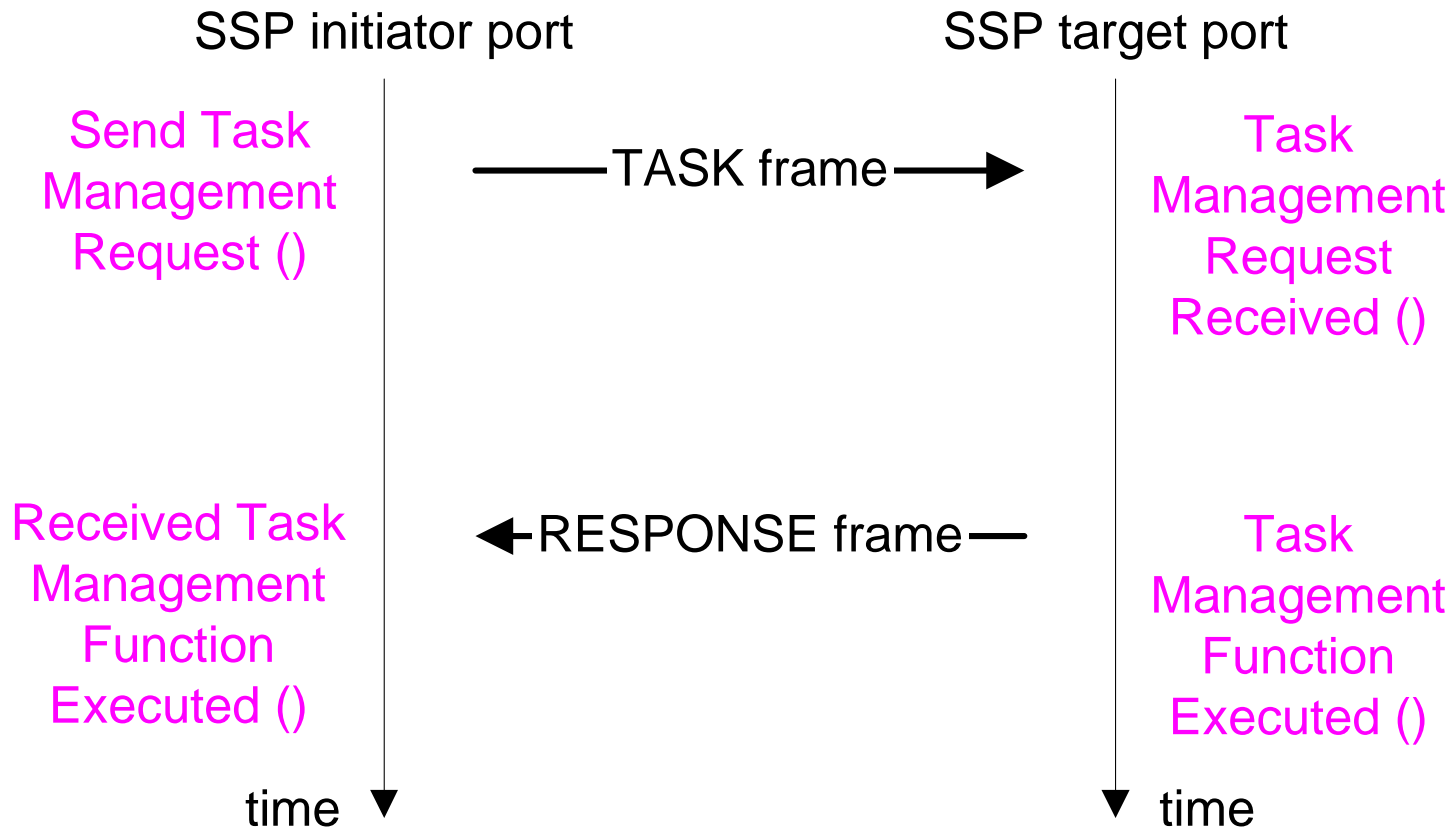
Byte	Field(s)		
0	Frame Type		
1 to 3	Hashed Destination SAS address		
4	Reserved		
5 to 7	Hashed Source SAS address		
8 to 9	Reserved		
10	Reserved	Retransmit	Rsvd
11	Reserved	Number of Fill Bytes	
12 to 15	Reserved		
16 to 17	Tag		
18 to 19	Target Port Transfer Tag		
20 to 23	Data Offset		
24 to m	Information Unit		
m to (n-3)	Fill bytes, if needed		
(n-3) to n	CRC		

SSP frame types

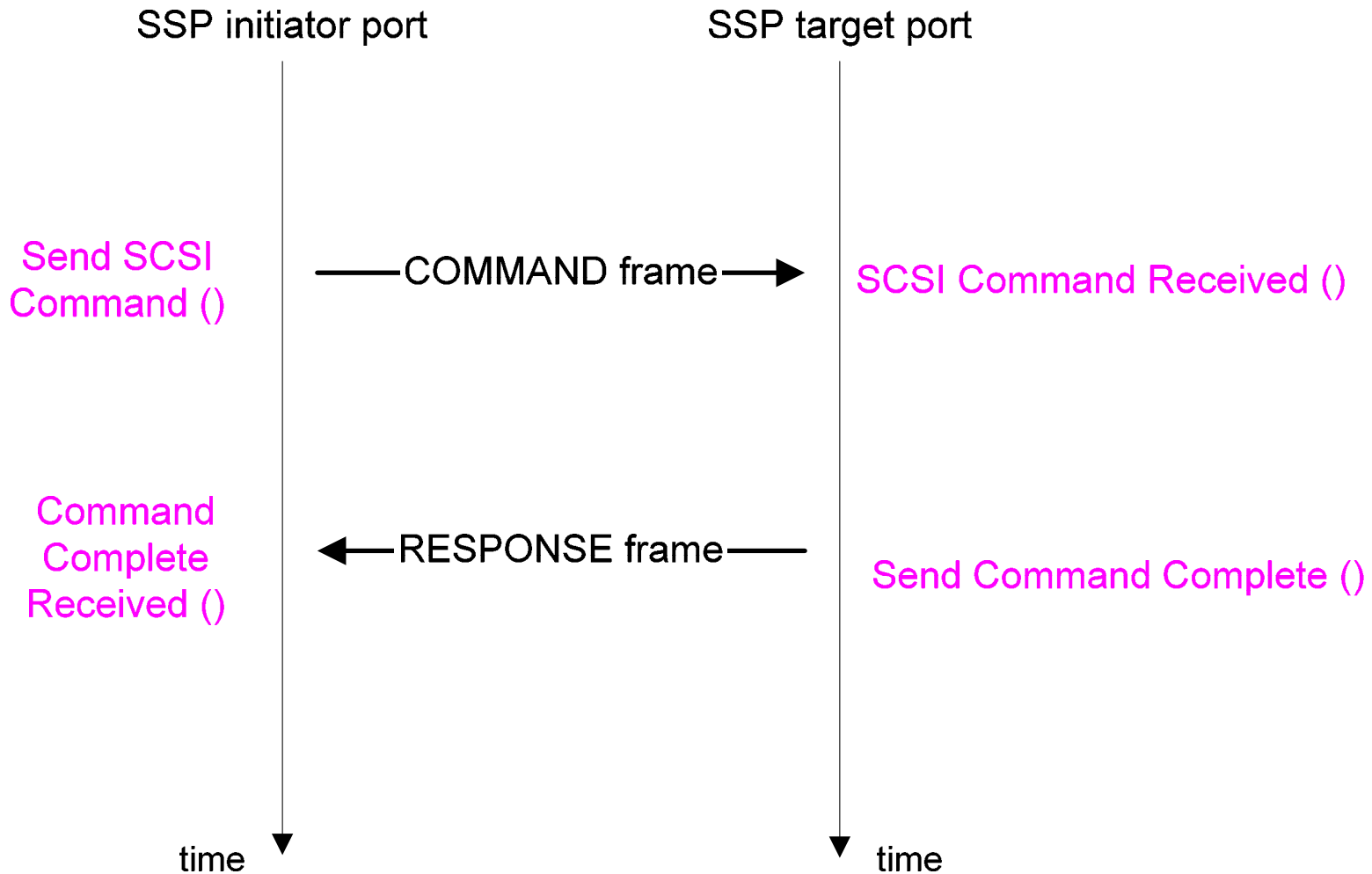


Command	Information Unit field size	Direction	Description
COMMAND	28 to 284	I to T	Send a command
TASK	28	I to T	Send a task management function
XFER_RDY	12	T to I	Request write data
DATA	1 to 1024	I to T or T to I	Write data (I to T) or read data (T to I)
RESPONSE	24 to 1024	T to I	Send SCSI status (for commands) or task management response (for task management functions)

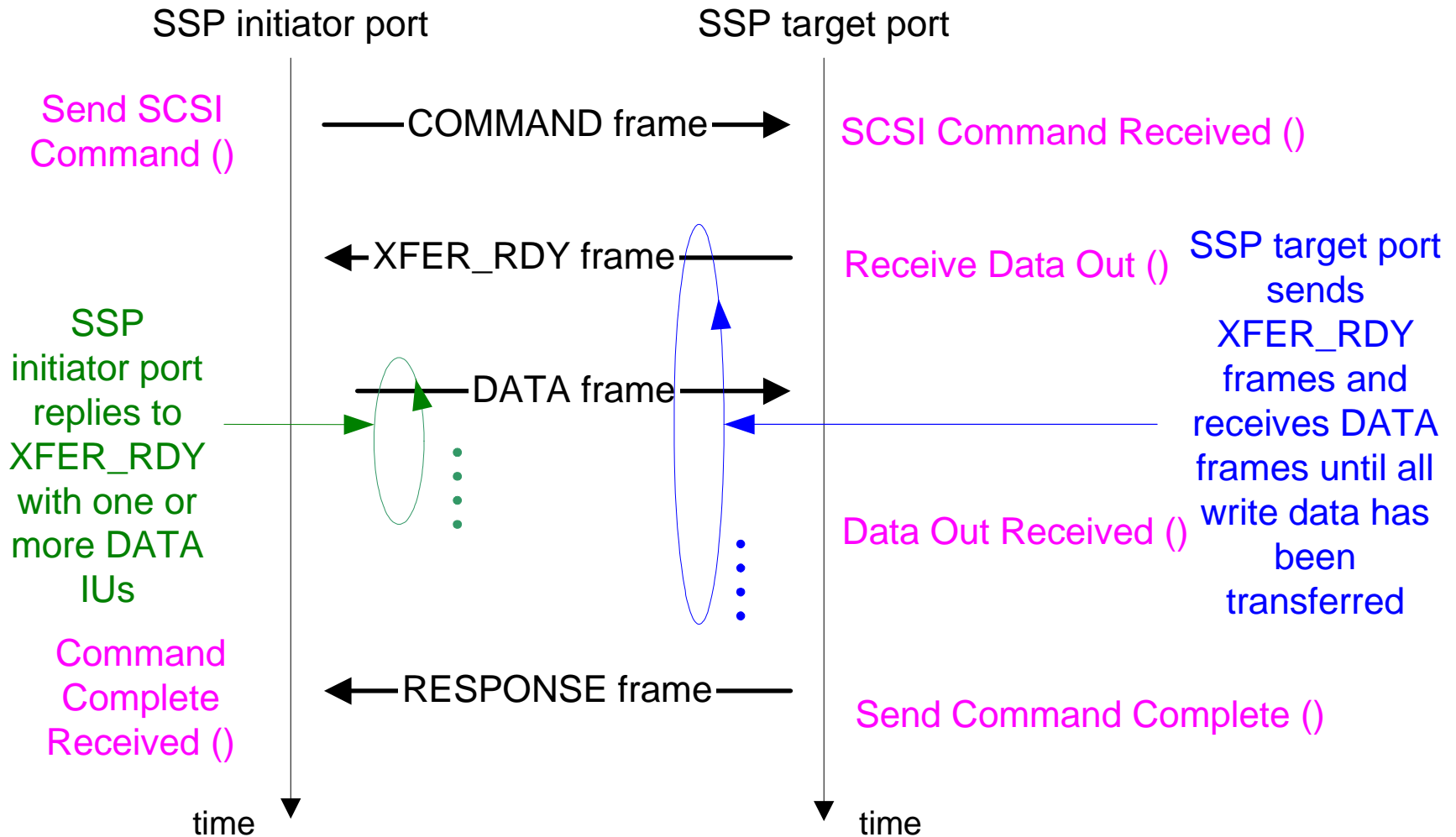
SSP task management frame sequence



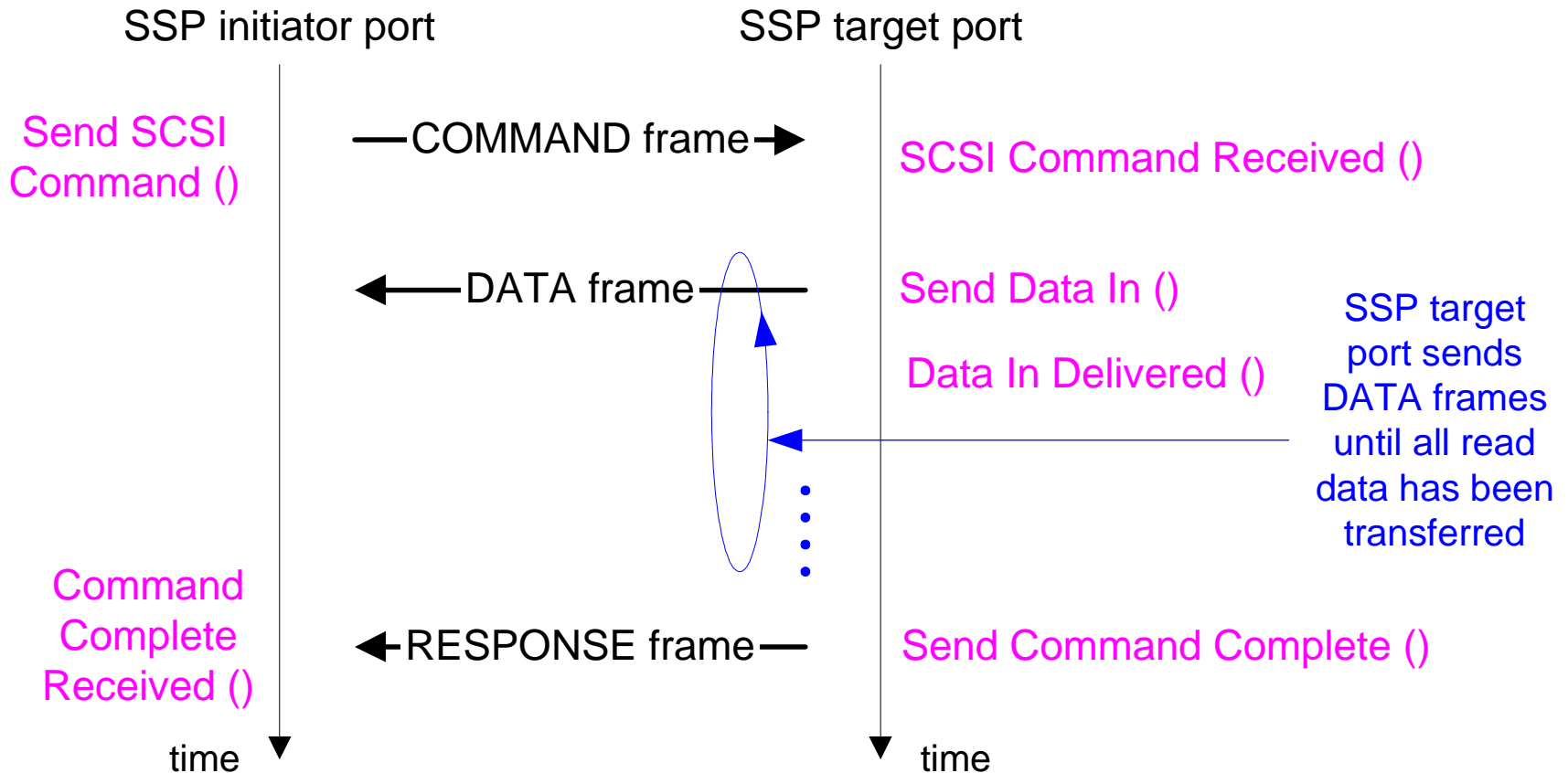
SSP non-data command sequence



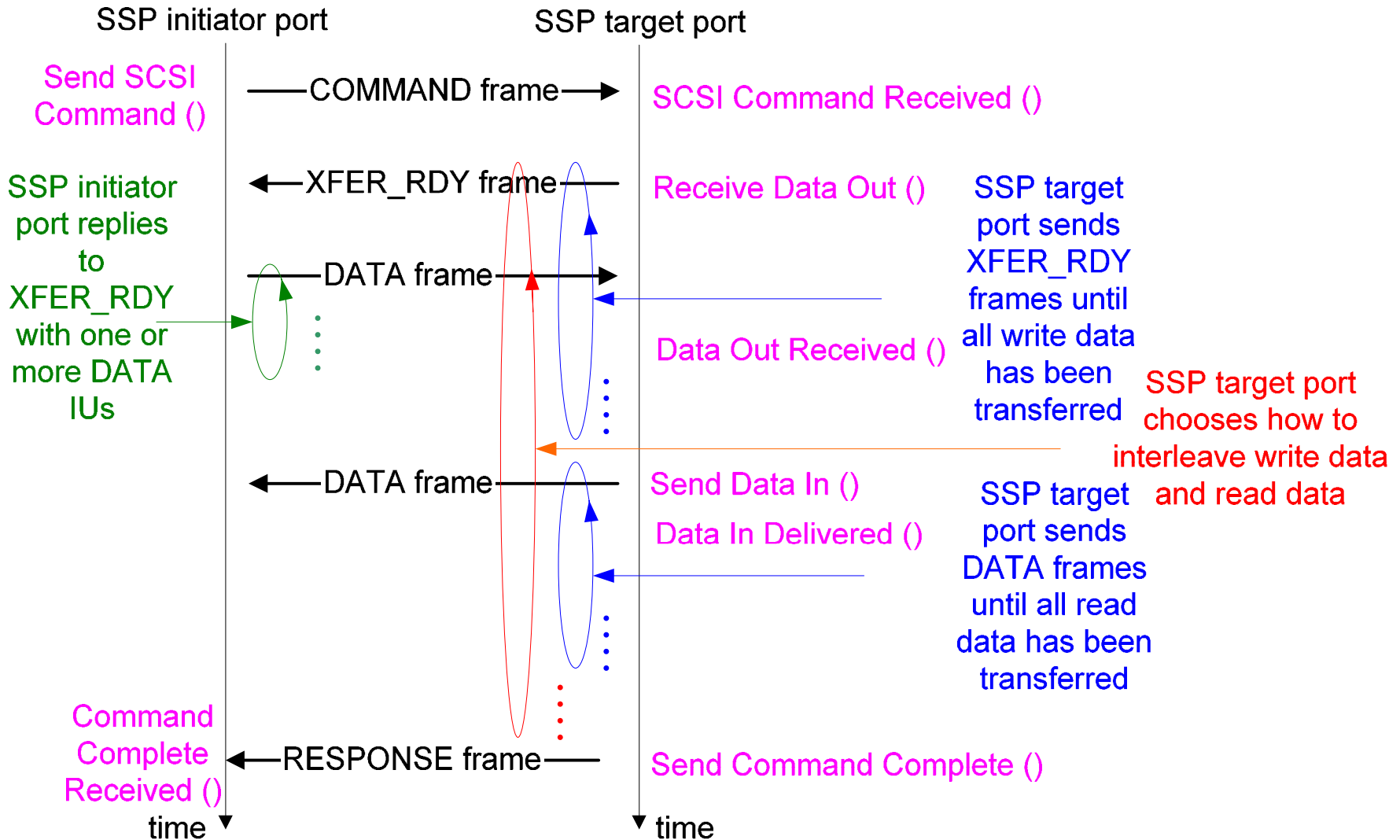
SSP write command sequence



SSP read command sequence



SSP bidirectional command sequence



SSP header – Hashed SAS addresses fields



- Hashed Source SAS Address and Hashed Destination SAS Address fields
 - 24-bit hashed versions of the source and destination SAS addresses
 - Used to double-check that the frame came from the proper source and arrived at the proper destination
 - Should match the addresses between which this connection was established
 - Some addresses will hash to the same value, allowing some incorrect values be accepted
 - Not required to check these fields

SSP header – Retransmit and Number of Fill Bytes fields



- **Retransmit** bit
 - only used for RESPONSE frames
 - If the target sends the frame but doesn't get an ACK or NAK, it doesn't know if the initiator got it or not
 - Set this bit to one and try again
 - If initiator did get the first one, it discards the repeat
- **Number of Fill Bytes** field
 - only used for DATA frames
 - If the Information Unit (e.g. the data) is not a multiple of 4, this indicates how many fill bytes follow it to keep the CRC 4-byte aligned
 - Only non-zero for the last DATA frame of a command

SSP header - Tag field



- Tag field
 - For frames related to a command, the SCSI tag
 - COMMAND, XFER_RDY, DATA
 - RESPONSE (with a tag from a COMMAND frame)
 - For frames related to a task management function, an “association” (like a tag)
 - TASK
 - RESPONSE (with a tag from a TASK frame)
 - I_T based
 - The target has to distinguish the same tag values from different initiators as different tags

SSP header - Target Port Transfer Tag field



- **Target Port Transfer Tag** field
 - only used for write DATA frames (from initiator to target)
 - Target sets it to a cookie in the XFER_RDY frame
 - For all DATA frames fulfilling that XFER_RDY, initiator returns the cookie
 - Lets the target do fast context lookup on incoming DATA frames
 - It could have XFER_RDYs outstanding for different commands (different I_T_L_Qs) at the same time

SSP COMMAND frame



- Used to send SCSI commands
- Initiator to target
- Variable frame length
 - Longer for variable-length CDBs (see SPC-3)
 - Minimum (and normal):
 $24+28+4=56$ bytes
 - Maximum:
 $24+284+4=312$ bytes

Byte	Field(s)
(24 bytes)	SSP frame header
0 to 7	Logical Unit Number
8	Reserved
9	Reserved Task Attribute
10	Reserved
11	Additional CDB Length
12 to 27	CDB
28 to m	Additional CDB bytes (if needed)
(0 bytes)	Fill bytes NOT needed
(4 bytes)	CRC

SSP COMMAND frame - Logical Unit Number and Task Attribute fields



- **Logical Unit Number** field
 - Contains the logical unit number (defined in SAM-3)
 - 64 bit field (subdivided into 4 16-bit fields)
 - All SCSI targets have a LUN 0

- **Task Attribute** field
 - Specifies information about SCSI queuing
 - Functionality defined in SAM-3; bits defined by SAS

Value	Task Attribute
000b	Simple
001b	Head of Queue
010b	Ordered
100b	ACA

SSP COMMAND frame - CDB fields



- **CDB field**
 - Contains the Command Descriptor Block (defined in SPC-3)
 - SCSI CDBs are 6, 10, 12, or 16 bytes long (or longer)
 - If not 16 bytes, the first (lowest) bytes contain the CDB and the rest are ignored
- **Additional CDB Length and Additional CDB Bytes field**
 - If **Additional CDB Length** is nonzero, the CDB is the concatenation of the **CDB field** and the **Additional CDB Bytes field**
 - Used for variable length CDBs
 - **Additional CDB Length** specifies how many more than 16 bytes are used
 - Units of **Additional CDB Length** are dwords

SSP TASK frame



- Used to send task management functions
- Initiator to target
- Fixed frame length
 - $24+28+4=56$ bytes

Byte	Field(s)
(24 bytes)	SSP frame header
0 to 7	Logical Unit Number
8 to 9	Reserved
10	Task Management Function
11	Reserved
12 to 13	Tag of Task to be Managed
14 to 27	Reserved
(0 bytes)	Fill bytes NOT needed
(4 bytes)	CRC

SSP TASK frame fields



- **Logical Unit Number** field
 - Contains the logical unit number (defined in SAM-3)
 - 64 bit field (subdivided into 4 16-bit fields)
 - All SCSI targets have a LUN 0
 - Used by all currently defined task management functions

- **Task Management Function** field
 - Specifies the task management function to run
 - Functionality defined in SAM-3

Value	Task Management Function
01h	Abort Task
02h	Abort Task Set
04h	Clear Task Set
08h	Logical Unit Reset
40h	Clear ACA
80h	Query Task

- **Tag of Task to be Managed** field
 - Used only for Abort Task and Query Task (which have I_T_L_Q nexus scope)

SSP XFER_RDY frame



- Request write data
- Target to initiator
- Fixed frame length
 - $24+12+4=40$ bytes
- Only one at a time per I_T_L_Q nexus
 - If multiple write commands are queued, can have one per write command

Byte	Field(s)
(24 bytes)	SSP frame header
0 to 3	Requested Offset
4 to 7	Write Data Length
8 to 11	Reserved
(0 bytes)	Fill bytes NOT needed
(4 bytes)	CRC

SSP XFER_RDY frame fields



- **Requested Offset** field
 - Offset of the write data to transfer in the application client's buffer
 - Each write data transfer must start at offset 0
 - Each XFER_RDY must request data in monotonically increasing order
 - No skipping forward or backwards
 - Must get all write data for one XFER_RDY before sending another (for the same I_T_L_Q)
- **Write Data Length** field
 - How many bytes to transfer from the specified application client buffer offset
 - Except for the last XFER_RDY for a command, must be 4 byte aligned

SSP DATA frame



- Transfer write data (initiator to target)
- Transfer read data (Target to initiator)
- Variable frame length
 - Maximum:
 $24 + 1024 + 4 = 1052$ bytes
- After the header, it's all data
- Data transferred from offset 0 on up
- **Data Offset** field in SSP header must increase for each read DATA frame in the I_T_L_Q nexus; no gaps or rewinds
- Data can be an odd amount only for the last data frame for a command
 - Only used for tape commands, INQUIRY, LOG SENSE/LOG SELECT, etc. that don't always transfer blocks
 - All data frames begin on 4 byte boundaries
 - Data frames other than the last one must be terminated on 4 byte boundaries

Byte	Field(s)
(24 bytes)	SSP frame header
0 to n	Data
(0 to 2 bytes)	Fill bytes, if needed
(4 bytes)	CRC

SSP RESPONSE frame



- Transfer SCSI status or task management response
- Variable frame length
 - Minimum:
24+24+4=52 bytes
 - Maximum:
24+1024+4=1052 bytes
- Depends on length of sense data and presence of response data
- Sense data is often 18 bytes

Byte	Field(s)
(24 bytes)	SSP frame header
0 to 9	Reserved
10	Reserved DataPres
11	Status
12 to 15	Reserved
16 to 19	Sense Data Length
20 to 23	Response Data Length
24 to (23+m)	Response Data
(24+m) to (23+m+n)	Sense Data
(0 bytes)	Fill bytes NOT needed
(4 bytes)	CRC

SSP RESPONSE frame - DataPres and Status fields



- **DataPres** field

- Indicates whether sense data or response data (or neither) is present

Value	Type of data present
00b	Neither (must be non-CHECK CONDITION status for a command)
01b	Response data
10b	Sense data

- **Status** field

- only used for responses to commands; ignored for task management responses
- Contains SCSI status for a command
- Values defined in SAM-3

Value	Status
00h	GOOD
02h	CHECK CONDITION
08h	BUSY
28h	TASK SET FULL
18h	RESERVATION CONFLICT

SSP RESPONSE frame - Sense Data fields



- **Sense Data Length** and **Sense Data** fields
 - Carries SCSI sense data
 - includes **Sense Key** and **Additional Sense Code** fields
 - Format defined in SPC-3
 - Only be present when **Status** is CHECK CONDITION
 - Most common fixed-length format is 18 bytes
 - Variable-length descriptor format required for SBC-2 disks > 2 TB, SSC-2 tape drives

SSP RESPONSE frame - Response Data fields



- Response Data Length and Response Data fields
 - Only valid lengths are 0 or 4 bytes
 - Contains 1 byte Response Code field

Value	Response Code	Description
00h	TASK MANAGEMENT FUNCTION COMPLETE	Task management function ran without problems (normal response)
02h	INVALID FRAME	A field in the COMMAND or TASK frame was bad
04h	TASK MANAGEMENT FUNCTION NOT SUPPORTED	The task management function sent in the TASK frame is not supported by the target
05h	TASK MANAGEMENT FUNCTION FAILED	The task management function encountered an error
08h	TASK MANAGEMENT FUNCTION SUCCEEDED	Task management function ran without problems and needs to return specific SUCCESS indication. (only used by QUERY TASK)
09h	INVALID LOGICAL UNIT NUMBER	The logical unit number is unknown. Only used for TASK frame problems; COMMAND frames with this problem result in Status of CHECK CONDITION and sense data

SSP transport layer state machines



- Interface between application layer and port layer
- Part of the port object
- Different state machines for initiator ports and target ports
 - ST_I – transport layer for SSP initiator ports
 - ST_ISF – initiator send frame
 - ST_IPD – initiator process data
 - ST_IPR – initiator process response
 - ST_IFR – initiator frame router
 - ST_T – transport layer for SSP target ports
 - ST_TFR – target frame router
 - ST_TTS – target transport server

SCSI application layer – mode pages

Mode and log pages



- Several mode pages have SAS specific content
- All mode pages and all fields within the mode pages are optional

Page code	Subpage code	Description
02h	00h	Disconnect-Reconnect mode page
19h	00h	Protocol-Specific Port mode page – short format
19h	01h	Protocol-Specific Port mode page – Phy Control and Discover subpage

- One SAS specific log page is also defined
- The log page and all fields within it are optional

Page code	Subpage code	Description
19h	00h	Protocol-Specific Port log page

Disconnect-Reconnect mode page



- Format and generic field definitions in SPC-3
- Each transport protocol defines which fields it supports and their specific meanings

Byte	Field(s)
0	Page Code (02h)
1	Page Length (0Eh)
2 to 3	Reserved
4 to 5	Bus Inactivity Time Limit
6 to 7	Reserved
8 to 9	Maximum Connect Time Limit
10 to 11	Maximum Burst Size
12 to 13	Reserved
14 to 15	First Burst Size

Disconnect-Reconnect mode page – Bus Inactivity Time Limit and Maximum Connect Time Limit fields



- **Bus Inactivity Time Limit** field
 - Units: 100 μ s
 - Maximum time a target can keep a connection open without transmitting a frame (read data, XFER_RDY)
 - Inbound frames don't count
 - Target must issue a DONE; whether the initiator replies is outside its control
- **Maximum Connect Time Limit** field
 - Units: 100 μ s
 - Maximum time a target can keep a connection open
 - Target must issue a DONE; whether the initiator does the same is outside its control

Disconnect-Reconnect mode page – Maximum Burst Size field



- **Maximum Burst Size** field
 - Units: 512 bytes
 - Reads
 - Maximum read data for one I_T_L_Q nexus without transferring at least one byte of data for some other I_T_L_Q nexus
 - Writes
 - Maximum amount of write data the target can request in one XFER_RDY
 - Useless field; should not be used
 - Initiators cannot be designed assuming they can turn this on

Disconnect-Reconnect mode page – First Burst Size field



- **First Burst Size** field
 - Units: 512 bytes
 - Creates an implicit XFER_RDY after receipt of each command
 - Allows the initiator to send write DATA frame(s) immediately after sending the COMMAND frame
 - Eliminates the wait for XFER_RDY
 - Does not eliminate the need for the link layer ACK
 - Target must be prepared to accept N bytes of data per command
 - Designed for high-latency environments like iSCSI; should not be in SAS
 - Dangerous; not recommended

Protocol-Specific mode page – short format



- Format and generic field definitions in SPC-3
- Each transport protocol defines which fields it supports and their specific meanings

Byte	Field(s)	
0	Page Code (19h)	
1	Page Length (06h)	
2	Reserved	Protocol Identifier (6h)
3	Reserved	
4 to 5	I_T Nexus Loss Time	
6 to 7	Initiator Response Timeout	



- I_T Nexus Loss Time field
 - Units: ms
 - Recommended default: 2 seconds
 - When target tries to OPEN an initiator and can't find it, allow this amount of time before giving up
 - Only select reasons allow retrying
 - OPEN_REJECT (NO DESTINATION)
 - OPEN timeout
 - Perhaps the physical link lost synchronization and is rerunning the link reset sequence
 - 0 ms (or field not implemented) means vendor-specific
 - FFFFh means there is no I_T nexus loss time
 - Target retries forever

Protocol Specific mode page – Initiator Response Timeout field



- Initiator Response Timeout field
 - Units: ms
 - Recommended default: none specified
 - After target sends an XFER_RDY frame, it might never see a write DATA frame in response
 - This ties up target resources
 - If no response within this time, abort that command
 - Timer runs across connections
 - Transport layer timer, not link layer
 - This times the initiator
 - It could still be doing useful work on another I_T_L_Q nexus (to the same or different target)
 - Use with caution
 - 0 ms (or field not implemented) means no timeout; wait forever

Protocol-Specific mode page – Phy Control and Discover subpage



- Provides access to the same functionality as the SMP PHY CONTROL and DISCOVER functions
- Allows targets to avoid implementing SMP

Byte	Field(s)		
0	PS	SPF (1h)	Page Code (19h)
1	Subpage Code (01h)		
2 to 3	Page Length		
4 to 6	Reserved		
7	Number of Phys		
8 to n	Phy mode descriptors		

- One phy mode descriptor for every phy in the device
 - Not just phys within the SAS port being accessed

Protocol-Specific mode page – Phy mode descriptor



Byte	Field(s)					
0	Reserved					
1	Phy Identifier					
2 to 3	Reserved					
4	Rsvd	Attached Device Type	Reserved			
5	Reserved		Negotiated Physical Link Rate			
6	Reserved		Att SSP I	Att STP I	Att SMP I	Reserved
7	Reserved		Att SSP T	Att STP T	Att SMP T	Reserved
8 to 15	SAS Address					
16 to 23	Attached SAS Address					
24	Attached Phy Identifier					
25 to 31	Reserved					
32	Programmed Minimum Physical Link Rate		Hardware Minimum Physical Link Rate			
33	Programmed Maximum Physical Link Rate		Hardware Maximum Physical Link Rate			
34 to 41	Reserved					
42 to 43	Vendor Specific					
44 to 47	Reserved					



Phy mode descriptor fields

- All fields are defined in the SMP DISCOVER and PHY CONTROL functions
- “Attached” fields refer to data received by the target during the identification sequence
 - E.g. they describe the expander to which it is attached

SCSI application layer – log pages

Protocol-Specific log page



- Provides access to the same functionality as the SMP REPORT PHY ERROR LOG function
- Allows targets to avoid implementing SMP
- One log parameter per target port
 - Numbered by relative target port identifier
- One phy mode descriptor for every phy in the port

Byte	Field(s)
0	Page Code (18h)
1	Reserved
2 to 3	Page Length
4 to n	Protocol-Specific log parameters

Byte	Field(s)
0 to 1	Parameter Code (relative target port identifier)
2	Parameter Control Bits
3	Parameter Length
4	Reserved Protocol Identifier (6h)
5 to 6	Reserved
7	Number of Phys
8 to n	Phy log descriptor

Protocol-Specific mode page – Phy log descriptor



Byte	Field(s)					
0	Reserved					
1	Phy Identifier					
2 to 3	Reserved					
4	Rsvd	Attached Device Type	Reserved			
5	Reserved		Negotiated Physical Link Rate			
6	Reserved		Att SSP I	Att STP I	Att SMP I	Reserved
7	Reserved		Att SSP T	Att STP T	Att SMP T	Reserved
8 to 15	SAS Address					
16 to 23	Attached SAS Address					
24	Attached Phy Identifier					
25 to 31	Reserved					
32 to 35	Invalid Dword Count					
36 to 39	Running Disparity Error Count					
40 to 43	Loss of Dword Synchronization					
44 to 47	Phy Reset Problem					



Phy log descriptor fields

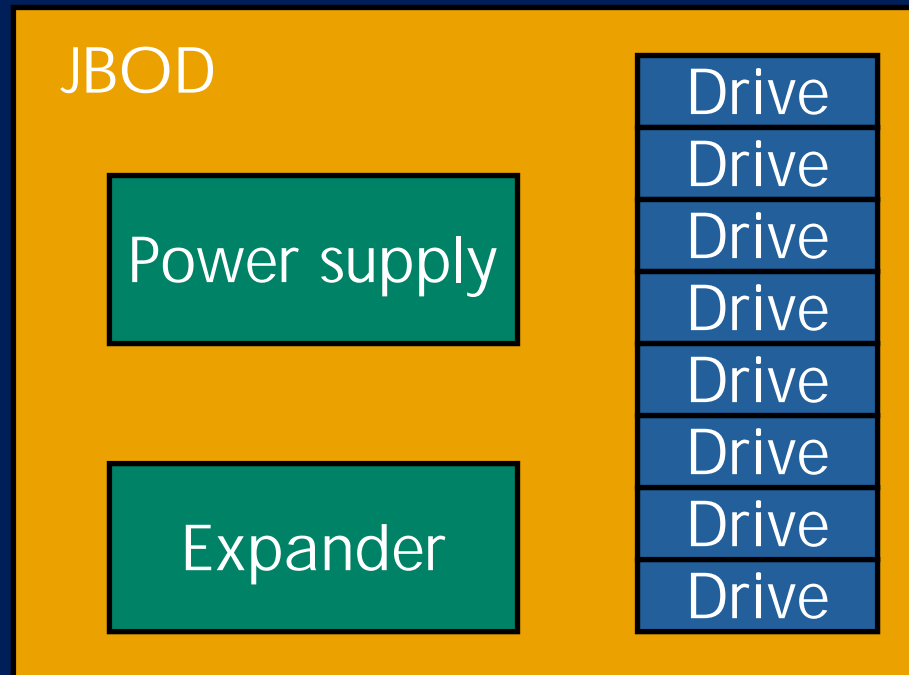
- All fields are defined in the SMP DISCOVER and PHY ERROR LOG functions
- “Attached” fields refer to data received by the target during the identification sequence
 - E.g. they describe the expander to which it is attached

SCSI application layer – power conditions

Spinup problem



- Disk drives consume a lot more power when spinning up than while running
 - e.g. 20 W while active, 50 W peak during spinup
- If all drives spin up simultaneously, can overload a power supply



Power conditions and spinup



- SATA drives spinup automatically after phy reset
- SAS drives do not spin up automatically
 - Wait for a NOTIFY (ENABLE SPINUP) primitive to arrive
 - Integrated into SCSI's already defined power condition states
 - START STOP UNIT command (defined in SBC-2)
 - Power Conditions mode page (defined in SPC-3)

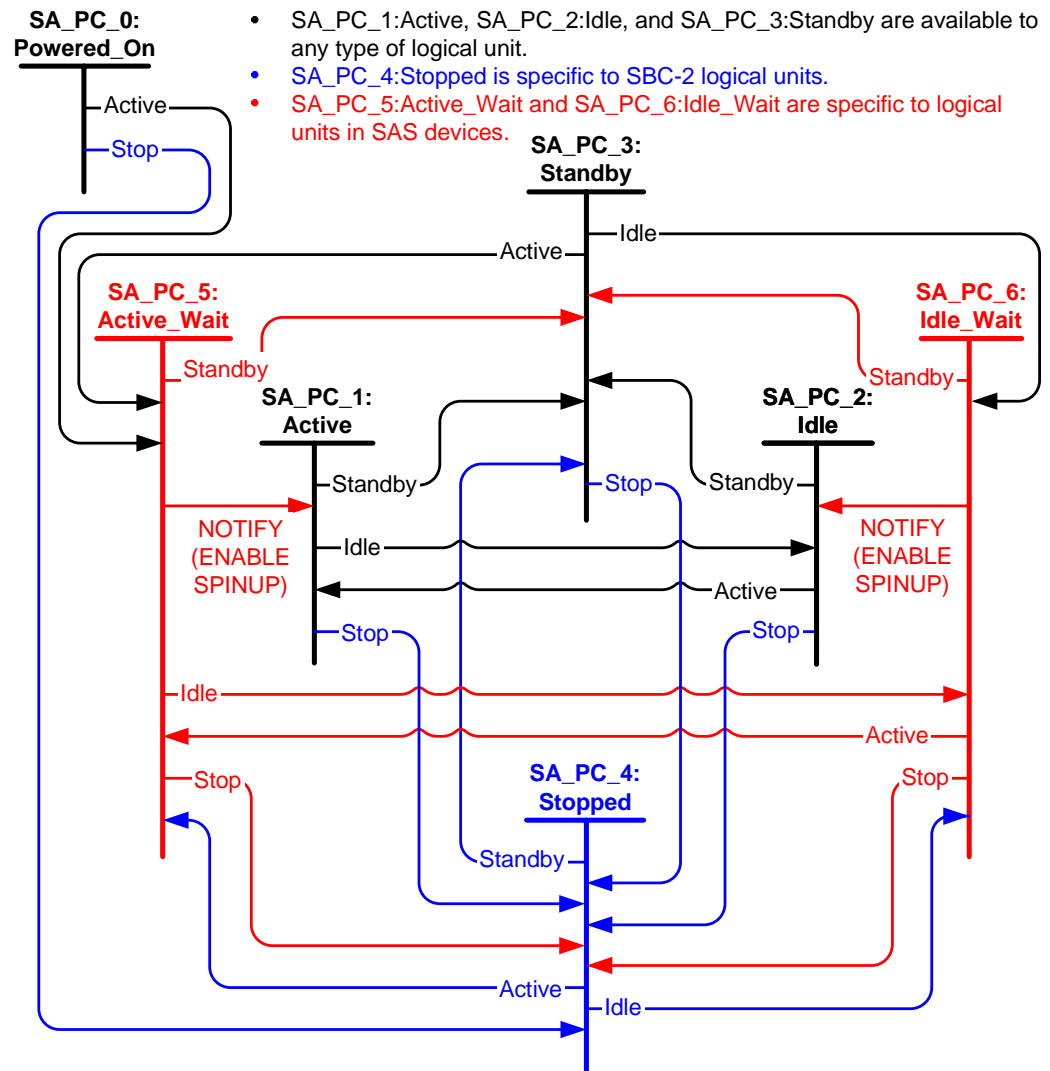
State	Description
Active	Fully operational – media is spinning
Idle	Operational - media is spinning. Longer command latency. Automatically transitions to Active as needed to process a command.
Standby	Media is stopped. Automatically transitions to Active or Idle as needed to process a command.
Stopped	Media is stopped. START STOP UNIT command required to restart.
Active_Wait	New for SAS. Waiting for a NOTIFY (ENABLE SPINUP) to enter Active state
Idle_Wait	New for SAS. Waiting for a NOTIFY (ENABLE SPINUP) to enter Idle state

Power conditions state machine



- Even START STOP UNIT command processing is delayed until NOTIFY (ENABLE SPINUP) arrives
- Expander or HBA must keep sending NOTIFY (ENABLE SPINUP) forever

SA_PC (SCSI application layer power condition) state machine



- SA_PC_1:Active, SA_PC_2:Idle, and SA_PC_3:Standby are available to any type of logical unit.
- SA_PC_4:Stopped is specific to SBC-2 logical units.
- SA_PC_5:Active_Wait and SA_PC_6:Idle_Wait are specific to logical units in SAS devices.



Wrap up

Serial Attached SCSI tutorials



- General overview (~2 hours)
- Detailed multi-part tutorial (~3 days to present):
 - Architecture
 - Physical layer
 - Phy layer
 - Link layer
 - Part 1) Primitives, address frames, connections
 - Part 2) Arbitration fairness, deadlocks and livelocks, rate matching, SSP, STP, and SMP frame transmission
 - Upper layers
 - Part 1) SCSI application and SSP transport layers
 - Part 2) ATA application and STP/SATA transport layers
 - Part 3) Management application and SMP transport layers, plus port layer
 - SAS SSP comparison with Fibre Channel FCP

Key SCSI standards



- Working drafts of **SCSI** standards are available on <http://www.t10.org>
- Published through <http://www.incits.org>
 - Serial Attached SCSI
 - SCSI Architecture Model – 3 (SAM-3)
 - SCSI Primary Commands – 3 (SPC-3)
 - SCSI Block Commands – 2 (SBC-2)
 - SCSI Stream Commands – 2 (SSC-2)
 - SCSI Enclosure Services – 2 (SES-2)
- **SAS connector** specifications are available on <http://www.sffcommittee.org>
 - SFF 8482 (internal backplane/drive)
 - SFF 8470 (external 4-wide)
 - SFF 8223, 8224, 8225 (2.5", 3.5", 5.25" form factors)
 - SFF 8484 (internal 4-wide)

Key ATA standards



- Working drafts of **ATA** standards are available on <http://www.t13.org>
 - Serial ATA 1.0a (output of private WG)
 - ATA/ATAPI-7 Volume 1 (architecture and commands)
 - ATA/ATAPI-7 Volume 3 (Serial ATA standard)
- **Serial ATA II** specifications are available on <http://www.t10.org> and <http://www.serialata.org>
 - Serial ATA II: Extensions to Serial ATA 1.0
 - Serial ATA II: Port Multiplier
 - Serial ATA II: Port Selector
 - Serial ATA II: Cables and Connectors Volume 1

For more information



- International Committee for Information Technology Standards
 - <http://www.incits.org>
- T10 (SCSI standards)
 - <http://www.t10.org>
 - Latest SAS working draft
 - T10 reflector for developers
- T13 (ATA standards)
 - <http://www.t13.org>
 - T13 reflector for developers
- T11 (Fibre Channel standards)
 - <http://www.t11.org>
- SFF (connectors)
 - <http://www.sffcommittee.org>
- SCSI Trade Association
 - <http://www.scsita.org>
- Serial ATA Working Group
 - <http://www.serialata.org>
- SNIA (Storage Networking Industry Association)
 - <http://www.snia.org>
- Industry news
 - <http://www.infostor.com>
 - <http://www.byteandswitch.com>
 - <http://www.wwpi.com>
 - <http://searchstorage.com>
- Training
 - <http://www.knowledgetek.com>



i n v e n t