Product Manual

Serial Attached SCSI (SAS) Interface Manual

Seagate

Product Manual

Serial Attached SCSI (SAS) Interface Manual

Seagate

Publication number: 100293071, Rev. B
May 2006

Revision status summary sheet

Revision	Date	Writers/Engineers	Notes
Rev. A	11/11/04	J. Coomes	Initial release.
Rev. B	05/07/06	C. Chalupa, J. Coomes, G. Houlder	All.

Contents

List of Figures

1.0 Interface requirements

1.1 Acknowledgements

The information contained in this publication was gathered from many sources. Portions of the text used to explain general SAS concepts were adapted in various forms, with permission, from the SCSI Trade Association, Hewett Packard Invent slides, and the T10/1760-D SAS-2 Interface Standard Draft 02.

1.2 How to use this interface manual

This manual provides a description of the Serial Attached SCSI (SAS) interface protocol and some general timing information as implemented by Seagate products. Each individual drive's Product Manual for the various SAS interface products contains additional and more detailed information on protocol, features supported, timing, and electrical/mechanical aspects of how the SAS interface is implemented by that product.

This manual provides a general, tutorial-type description of the ANSI SAS system. It is not intended to give all of the kinds of details needed to design/implement a SAS system or product.

For information about SAS interface details not included herein or in the individual drive product manuals, refer to the specifications listed in Section .

Note. The individual drive's product manual has tables that specify which SCSI features the drive implements, what the default parameters are for the various features they implement, which parameters are changeable, and which are not.

Note. SCSI commands are documented in the SCSI Commands Reference Manual, part number 100293068.

The combination of this specification together with the SCSI Commands Reference Manual and details in the individual drive's product manual, provides a description of how a particular product implements the SAS I/O system. This interface manual is intended to be used in conjunction with the individual drive's product manual and the SCSI Commands Reference Manual.

1.2.1 Scope

Figure 1 uses a representative set of specifications to show the functional partitions and the relationships among SCSI specifications applicable to drives covered by this product manual.

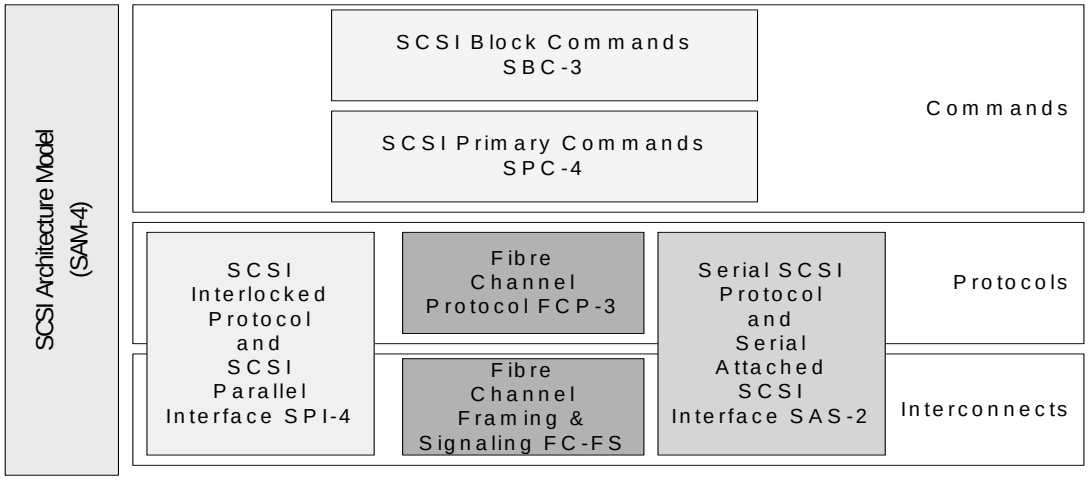


Figure 1. Functional scope

The functional areas define the scope of each as follows:

- SCSI Architecture Model: Defines the SCSI systems model, the functional partitioning of the SCSI set and requirements applicable to all SCSI implementations and implementation specifications.
- Commands: Implementation specifications which define classes including a device model for each class. These specifications specify the required commands and behavior that is common to all devices or unique to a given class of devices and prescribe the rules to be followed by a SCSI initiator port when sending commands to a device.
- Protocols: Implementation specifications which define the rules for exchanging information so that different SCSI devices can communicate.
- Interconnects: Implementation specifications which define the electrical and signaling rules essential for devices to interoperate over a given physical interconnect.

The diagram of Figure 1 shows how the specifications listed below fit within each category. The specifications included in the diagram are meant to serve as examples and may not reflect the full set of specifications currently in force.

1.2.2 Applicable specifications

The following ANSI specifications should be referenced for more details about SCSI system specifications of operation:

- SCSI Architecture Model-4 (SAM-4), T10/1683-D
- SCSI Enclosure Services - 2 (SES-2), T10/1559-D
- SCSI Block Commands - 3 (SBC-3), T10/xxxx-D
- SCSI Primary Commands-4 (SPC-4), T10/1731-D
- SCSI Serial Attached SCSI - 2 (SAS-2) T10/1760-D Rev 01
- Fibre Channel Protocol for SCSI, Third Version (FCP-3) T10/1560-D

1.2.3 Other references

For information on the current status of the listed documents, or regarding availability, contact the indicated organization.

- SFF-8223, *2.5" Drive Form Factor with Serial Connector*
- SFF-8323, *3.5" Drive Form Factor with Serial Connector*
- SFF-8523, *5.25" Drive Form Factor with Serial Connector*
- SFF-8410, *HSS Copper Testing and Performance Requirements*
- SFF-8460, *HSS Backplane Design Guidelines*
- SFF-8470, *Multi Lane Copper Connector*
- SFF-8482, *SAS Plug Connector*

1.3 General interface description

This SAS Interface Manual describes the Seagate Technology, Inc. subset of the Serial Attached SCSI (Small Computer Systems Interface) as implemented on the Seagate-built drives. The interface is compatible with the SCSI Interface Specifications listed in Section 1.2.2. The drives covered by this product manual are classified as “Intelligent” peripherals.

1.3.1 Introduction to Serial Attached SCSI Interface (SAS)

The SAS interface provides several advantages over Parallel SCSI. Parallel SCSI has reached a practical maximum transfer rate of 320 MB/sec. Parallel SCSI is limited to a maximum of 16 devices connected to the bus, one of which must be a host bus adapter. Fiber Channel (FC) allows SCSI to be transmitted in a serial manner using frames, rather than on a parallel bus. It allows up to 127 devices to be addressed on fibre optic cable, or copper conductors. FC devices are connected in a loop and arbitrate for control of the loop, using fibre optic cable devices may be physically separated by 10km of cable. Parallel ATA (PATA) is limited to a maximum of two devices per host adapter, lower data transfer rates, and is not considered intelligent, therefore, not well suited for enterprise environments. Serial ATA (SATA) increases the data transfer rate but is limited in addressing, and cable length, and intelligence. SATA uses small form cables and connectors to transfer data at up 300 MB/sec, with the standard allowing transmission of up to 600 MB/sec. Currently SATA is a point-to-point connection inside the computer’s system unit, with a maximum length of 18 inches. External SATA is in the development stage.

SAS combines the intelligence of SCSI with the physical transport layer of Serial ATA. This scheme allows the intelligence of SCSI to be transferred on a serial cable similar to SATA. Data is transfer in frames, like fiber channel. The initial data transfer rate for SAS was 300 MB/sec. with the SAS-2 standard allowing up to 600 MB/sec. SAS provides for full duplex operation, at 300 MB/sec. it is possible to attain 600 MB/sec. per pathway. SAS is a point to point connection as is SATA, Expanders are used to increase the number of devices that may be connected. There are two types of expanders; Edge Expanders that may address 128 physical devices and one other expander, and Fanout Expanders that may address 128 other devices or expanders. Theoretically, there may be 16,256 devices in a single SAS domain. Unlike Parallel SCSI and Fibre Channel SAS drives and Serial ATA drives may be attached to the same expander.

1.3.2 The SAS interface

The Seagate SAS interface described herein consists of a dual ported SAS bidirectional links. The SCSI interface supports multiple initiators, disconnect/reconnect, self-configuring host software, automatic features that relieve the host from the necessity of knowing the physical architecture of the target (logical block addressing is used), and some other miscellaneous features.

Unless specified otherwise in the individual drive's product manual, the drive is always a SAS target port, and never a SAS initiator port. For certain commands, which may or may not be supported by a particular drive model, the drive must act as a SAS initiator port, but does not otherwise do so. For purposes of this specification, "drive" may be substituted for the word "target" wherever "target" appears.

In the event of a conflict between this document and ANSI SCSI documents, the requirements of the ANSI documents shall apply.

In Figure 2, it can be seen that several "application clients" from a single initiator may have one or more tasks in queue with several "device servers" in a single target. A drive could be a SCSI target port or it could be one of the device servers as part of some larger entity. When reading the description, one needs to be able to put the drive of interest in the proper context in terms of what is shown in Figure 2. For a proper understanding of the operation of the SCSI protocol, the terms in the SCSI architectural model as described in ANSI specification T10/1683-D (SAM-4) should be well understood before reading operation descriptions in any SCSI document. The SAM-4 specification gives a more detailed understanding of some of the new SCSI terminology. Figure 2 shows the SCSI initiator connected to the SCSI target by the service delivery subsystem. The SAS interface provides this function.

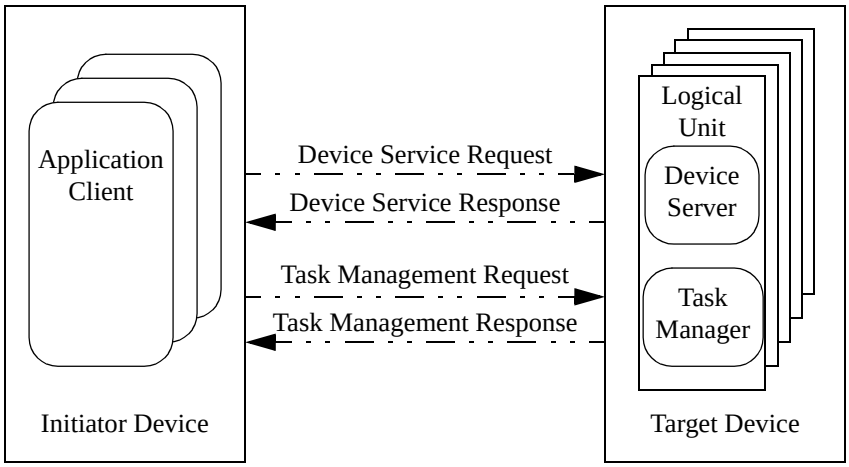


Figure 2. SCSI client-server model

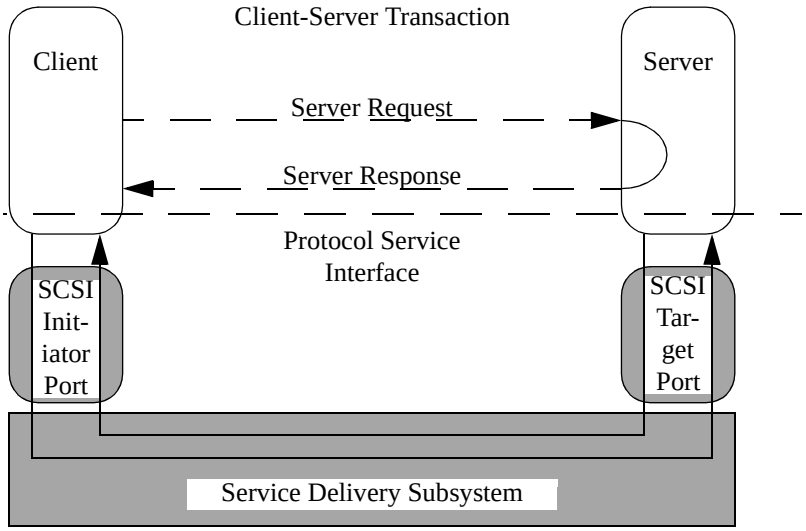


Figure 3. SCSI client-server model with service delivery subsystem

1.3.3 Glossary

8b 10b coding—A coding scheme that represents an 8-bit data byte as a 10-bit character.

aborted command—A SCSI command that has been ended by aborting the task created to execute it.

ACA—Auto Contingent Allegiance (see below).

ACA command—A command performed by a task with the ACA attribute. For additional information about the ACA command, refer to the Seagate SCSI Commands Reference Manual, part number 100293068.

ACK—Acknowledge primitive that specifies the positive acknowledgement of an SSP frame.

application client—An object that is the source of SCSI commands. An object in this sense is not a tangible piece of hardware, but may be a single numeric parameter, such as a logical unit number, or a complex entity that performs a set of operations or services on behalf of another object (see ANSI SAM-4, T10/1683-D).

auto contingent allegiance—One of the conditions of a task set following the return of a CHECK CONDITION status.

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

blocked (task state)—The state of a task that is prevented from completing due to an ACA condition.

broadcast primitive processor (BPP)—An object within an expander function that manages broadcast primitives.

burst time—The part of an OOB signal (see 3.4) where ALIGN primitives (see 4.2.5.2) are being transmitted.

byte—A sequence of eight contiguous bits considered as a unit.

call—The act of invoking a procedure.

character—A sequence of ten contiguous bits considered as a unit. A byte is encoded as a character using 8b10b coding.

client-server—A relationship established between a pair of distributed objects where one (the client) requests the other (the server) to perform some operation or unit of work on the client's behalf (see ANSI SAM-2, T10/1157D).

Client—An object that requests a service from a server.

command—A request describing a unit of work to be performed by a device server.

command descriptor block (CDB)—A structure used to communicate a command from a SCSI application client to a SCSI device server. See SAM-4.

completed command—A command that has ended by returning a status and service response of Task Complete or Linked Command Complete.

completed task—A task that has ended by returning a status and service response of Task Complete. The actual events comprising the Task Complete response are protocol specific.

configurable expander device—An expander device that contains an expander route table that is configured with expander route entries.

confirmation—A response returned to an object, which signals the completion of a service request.

confirmed service—A service available at the protocol service interface, which requires confirmation of completion. The confirmed service consists of the request and confirmation steps and optionally the indication and response steps.

connection—A temporary association between a SAS initiator port and a SAS target port.

connection rate—The effective rate of dwords through the pathway between a SAS initiator phy and a SAS target phy, established through the connection request.

control character (Kxx.y)—A character that does not represent a byte of data.

control mode page—The mode page that identifies the settings of several device server behaviors that may be of interest to an application client or may be changed by an application client. The complete definition of the Control mode page is found in the Seagate SCSI Commands Reference Manual, part number 100293068.

current task—A task that is in the process of sending messages, sending status, transferring data, or transferring command data to or from the initiator.

cyclic redundancy check (CRC)—An error detecting code used to detect the validity of data that has been transferred during the current data group.

data character (Dxx.y)—A character representing a byte of data.

data dword—A dword that starts with a Dxx.y (data character).

data field—The portion of a data group that contains data bytes.

data group—A sequence of data bytes and the four pCRC bytes during a DT DATA IN PHASE or a DT DATA OUT PHASE that starts at the first byte of the DT DATA phase or at the first byte after the last pCRC byte.

data group transfer—Parallel transfers that transfer data and pCRC information using only data groups. The last four bytes of a data group transfer contain CRC information over the whole data group.

D.C. idle—A differential signal level that is nominally 0 V(P-P).

deadlock—A condition in which two or more processes (e.g., connection requests) are waiting on each other to complete, resulting in none of the processes completing.

destination device—The SCSI device to which a service delivery transaction is addressed. *See* source device.

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

device name— A worldwide unique name for a device within a transport protocol.

device server—An object within a SAS target device that processes SCSI tasks (see SAM-3).

device service request—A request, submitted by an application client, conveying a SCSI command to a device server.

device service response—The response returned to an application client by a device server on completion of a SCSI command.

differential—A signalling alternative that employs differential (two complementary signals) drivers and receivers to improve signal-to-noise ratios and increase maximum cable lengths.

direct current (D.C.)—The non-A.C. component of a signal. In this, all frequency components below 100 kHz.

discover process—The algorithm used by a management application client to configure the SAS domain. See SAS.

disparity—The difference between the number of ones and zeros in a character.

domain—A SAS domain, a SCSI domain, or an ATA domain.

dword—A sequence of four contiguous bytes or four contiguous characters considered as a unit. The meaning depends on the context (e.g., when discussing the bits being transmitted over a physical link, dword represents four characters (i.e., 40 bits). When discussing the contents of a frame after 10b8b decoding (see 3.2), dword represents four bytes (i.e., 32 bits)).

dword synchronization—Detection of an incoming stream of dwords from a physical link by a phy.

edge expander device—An expander device that is part of a single edge expander device set.

edge expander device set—A group of one or more edge expander devices that may be attached to no more than one other edge expander device set or one fanout expander device.

enabled (task state)—The state of a task that may complete at any time. Alternatively, the state of a task that is waiting to receive the next command in a series of linked commands.

end device—A SAS device that is not contained within an expander device.

ended command—A command that has completed or aborted.

event notification—A message passed from the transport layer to the application layer notifying the application layer that a SCSI event has occurred. See SAM-4.

exception condition—Any event that causes a SCSI device to enter an auto contingent allegiance or contingent allegiance condition.

expander device—A device that is part of the service delivery subsystem and facilitates communication between SAS devices.

expander function—An object within an expander device that contains an expander connection manager, expander connection router, and broadcast primitive processor.

expander phy—A phy in an expander device that interfaces to a service delivery subsystem.

expander port—An expander device object that interfaces to the service delivery subsystem and to SAS ports in other devices.

expander route entry—A routed SAS address and an enable/disable bit in an expander route table.

expander route index—A value used in combination with a phy identifier to select an expander route entry in an expander route table.

fanout expander device—An expander device that is capable of being attached to two or more edge expander device sets.

faulted initiator—The initiator to which a Command Terminated or CHECK CONDITION status was returned.

faulted task set—A task set that contained a faulting task.

faulting command—A command that completed with a status of Check Condition or Command Terminated.

faulting task—A task that has completed with a status of Check Condition or Command Terminated.

field—A group of one or more contiguous bits.

frame—A sequence of data dwords between a start of frame primitive (e.g., SOF, SOAF, or SATA_SOF) and an end of frame primitive (e.g., EOF, EOAF, or SATA_EOF).

hard reset—A SAS device or expander device action in response to a reset event.

Hard reset sequence—A sequence that causes a hard reset.

hardware maximum physical link rate—The maximum physical link rate capability of a phy.

hardware minimum physical link rate—The minimum physical link rate capability of a phy.

hash function—A mathematical function that maps values from a larger set of values into a smaller set of values, reducing a long value into a shorter hashed value.

identification sequence—A sequence where phys exchange IDENTIFY address frames.

idle dword—A vendor-specific data dword that is scrambled and is transmitted outside a frame.

idle time—The part of an OOB signal where D.C. idle is being transmitted.

implementation-specific—A requirement or feature that is defined in a SCSI but whose implementation may be specified by the system integrator or vendor.

implementation option—An option whose actualization within an implementation is at the discretion of the implementor.

indication—A message passed from a lower layer state machine to a higher layer state machine, usually relaying a request (see) from a peer higher layer state machine.

information unit (IU)—The portion of an SSP frame that carries command, task management function, data, response, or transfer ready information.

information unit transfer—Parallel transfers that transfer data, status, commands, task attributes, task management information, acrid, and nexus information using only SPI information units.

initiator—A SCSI device containing application clients which originate device service and task management requests to be processed by a SCSI target port SCSI device.

invalid dword—A dword with an illegal character, with a control character in other than the first character position, with a control character other than K28.5 or K28.3 in the first character position, or with a running disparity error.

I/O operation—An operation defined by an unlinked SCSI command, a series of linked SCSI commands or a task management function.

I/O process—An I/O process consists of one initial connection or, if information units are enabled, the establishment of a nexus, and a zero or more physical or logical reconnection all pertaining to a single task or a group of tasks. An I/O process begins with the establishment of a nexus. If the SPI information unit transfers are disabled, an I/O process normally ends with a Command Complete message. If information unit transfers are enabled, an I/O process ends with a SPI L_Q information unit with the type field set to status and the Data Length field set to zero.

I_T nexus—A nexus that exists between a SCSI initiator port and a SCSI target port.

I_T nexus loss—A condition where a SAS port determines that another SAS port is no longer available.

I_T_L nexus—A nexus that exists between a SCSI initiator port, a SCSI target port, and a logical unit. This relationship extends the prior I_T nexus.

I_T_L_Q nexus—A nexus between a SCSI initiator port, a SCSI target port, a logical unit, and a tagged task. This relationship extends the prior I_T nexus or I_T_L nexus.

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

layer—A subdivision of the architecture constituted by subsystems of the same rank.

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

linked CDB—A CDB with the link bit in the control byte set to one.

linked command—One in a series of SCSI commands executed by a single task, which collectively make up a discrete I/O operation. In such a series, each command has the same task identifier, and all except the last have the link bit in the CDB control byte set to one.

link reset—Performing the link reset sequence

link reset sequence—For SATA, a phy reset sequence. For SAS, a phy reset sequence followed by an identification sequence, or a phy reset sequence followed by a hard reset sequence, another phy reset sequence, and an identification sequence.

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

livelock—A condition where two or more processes (e.g., connection requests) continually change their state in response to changes in other processes, resulting in none of the processes completing.

logical connect—Establishes an I_T_L_Q nexus using SPI L_Q information units.

logical disconnect—Reduces the current I_T_L_Q nexus to an I_T nexus.

logical reconnect—Reestablishes an I_T_L_Q nexus from an I_T nexus using SPI L_Q information units.

logical unit—a SCSI target port-resident entity which implements a device model and executes SCSI commands sent by an application client.

logical unit number—A 64-bit identifier for a logical unit.

logical unit option—An option pertaining to a logical unit, whose actualization is at the discretion of the logical unit implementor.

lower level protocol—A protocol used to carry the information representing upper level protocol transactions.

media—Particular elements comprising the interconnect including copper cables, printed circuit boards, and other transmission line materials.

media information—Information stored within a SCSI device which is non-volatile (retained through a power cycle) and accessible to a SCSI initiator port through the execution of SCSI commands.

message—Information sent between state machines.

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

multidrop—A characteristic of the SCSI bus that allows SCSI devices to be connected to the SCSI bus without disrupting the electrical path between the terminators.

multimode single-ended (MSE)—A signalling alternative for multimode SCSI devices that employs MSE drivers and receivers to allow multimode SCSI devices to operate when SE SCSI devices are present on the bus.

NAK—Specifies the negative acknowledgement of an SSP frame and the reason for doing so.

narrow link—A physical link that attaches a narrow port to another narrow port.

narrow port—A port that contains exactly one phy.

negotiated physical link rate—The current operational physical link rate established after speed negotiation between two phys.

nexus—A relationship between a SCSI initiator port and a SCSI target port, logical unit, or queue tag that begins with an initial connection and ends with the completion of the associated I/O process. This relationship is formed as the result of a task.

object—An architectural abstraction or “container” that encapsulates data types, services, or other objects that are related in some way.

OOB sequence—A sequence where two phys exchange OOB signals.

OOB signal—Pattern of ALIGNs and idle time used during the link reset sequence.

partial pathway—The set of physical links participating in a connection request which has not reached a SAS endpoint (i.e., the connection request has been transmitted by the source device and confirmed as received by at least one expander device with AIP).

pathway—A set of physical links between a SAS initiator phy and a SAS target phy being used by a connection.

pathway blocked count—The number of times the port has retried this connection request due to receiving OPEN_REJECT (PATHWAY BLOCKED).

pending task—A task that is not a current task.

phy—A device object that is used to interface to other devices.

phy reset sequence—An OOB sequence (see) followed by a speed negotiation sequence (see).

physical link—Two differential signal pairs, one pair in each direction, that connect two physical phys.

physical phy—A phy (see) that contains a transceiver and electrically interfaces to a physical link to communicate with another physical phy.

port—A SAS port or an expander port. Each port contains one or more phys.

potential pathway—A set of physical links between a SAS initiator phy and a SAS target phy.

power on—Power being applied.

primitive—A dword starting with K28.5 or K28.3 followed by three data characters.

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

procedure—An operation that can be invoked through an external calling interface.

programmed maximum physical link rate—The maximum operational physical link rate of a phy (e.g., as programmed via the SMP PHY CONTROL function (see SAS) or the Phy Control and Discover subpage.

programmed minimum physical link rate—The minimum operational physical link rate of a phy (e.g., as programmed via the SMP PHY CONTROL function (see SAS), the Phy Control and Discover subpage.

protocol—The rules governing the content and exchange of information passed between distributed objects through the service delivery subsystem.

protocol option—An option whose definition within a SCSI protocol is discretionary.

protocol service confirmation—A signal from the lower level protocol service layer notifying the upper layer that a protocol service request has completed.

protocol service indication—A signal from the lower level protocol service layer notifying the upper level that a protocol transaction has occurred.

protocol service request—A call to the lower level protocol service layer to begin a protocol service transaction.

protocol service response—A reply from the upper level protocol layer in response to a protocol service indication.

queue—The arrangement of tasks within a task set, usually according to the temporal order in which they were created. *See* task set.

queue tag—The parameter associated with a task that uniquely identifies it from other tagged tasks for a logical unit from the same initiator.

random jitter—Jitter that is assumed to have a Gaussian distribution.

rate—Data transfer rate of a physical link (e.g., 1,5 Gbps or 3,0 Gbps).

rate change delay time (RCDT)—The time between rates during the speed negotiation sequence.

receiver—The recipient of a signal.

reflection coefficient (r)—The reflection coefficient of the transmission media (i.e., the ratio of the reflected voltage divided by the voltage applied to the transmission media).

request—A message passed from a higher layer state machine to a lower layer state machine, usually to initiate some action.

request-response transaction—An interaction between a pair of distributed, cooperating objects, consisting of a request for service submitted to an object followed by a response conveying the result.

request-confirmation transaction—An interaction between a pair of cooperating objects, consisting of a request for service submitted to an object followed by a response for the object confirming request completion.

reset event—An event that triggers a hard reset from a SAS device.

response—A message passed from a higher layer state machine to a lower layer state machine, usually in response to an indication.

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

SAS address— worldwide unique name assigned to a SAS initiator port, SAS target port, expander device, SAS initiator device, or SAS target device (see 2.2.2).

SAS device—A SAS initiator device, SAS target device, or SAS target/initiator device.

SAS domain—The I/O system defined by this that may serve as an ATA domain and/or a SCSI domain.

SAS initiator device—A device containing SSP, STP, and/or SMP initiator ports in a SAS domain.

SAS initiator phy—A phy in a SAS initiator device.

SAS initiator port—An SSP initiator port, STP initiator port, and/or SMP initiator port in a SAS domain.

SAS phy—A phy in a SAS device that interfaces to a service delivery subsystem.

SAS port—A SAS initiator port, SAS target port, or SAS target/initiator port.

SAS target device—A device containing SSP, STP, and/or SMP target ports in a SAS domain.

SAS target phy—A phy in a SAS target device.

SAS target port—An SSP target port, STP target port, and/or SMP target port in a SAS domain.

SAS target/initiator device—A device that has all the characteristics of a SAS target device and a SAS initiator device.

SAS target/initiator port—A port that has all the characteristics of a SAS target port and a SAS initiator port in a SAS domain.

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

SCSI application layer—The protocols and procedures that implement or invoke SCSI commands and task management functions by using services provided by a SCSI protocol layer.

SCSI device—A device that contains one or more SCSI ports that are connected to a service delivery subsystem and supports a SCSI application protocol. See SAM-4.

SCSI domain—An I/O system consisting of a set of SCSI devices that communicate with one another by means of a service delivery subsystem. See SAM-4.

SCSI initiator device—A SCSI device containing application clients and SCSI initiator ports that originates device service and task management requests to be processed by a SCSI target device and receives device service and task management responses from SCSI target devices. See SAM-4.

SCSI initiator port—A SCSI initiator device object that acts as the connection between application clients and the service delivery subsystem through which indications and responses are routed. See SAM-4.

SCSI port—A SCSI initiator port or a SCSI target port. See SAM-4.

SCSI target device—A SCSI device containing logical units and SCSI target ports that receives device service and task management requests for processing and sends device service and task management responses to SCSI initiator devices. See SAM-4.

SCSI target port—A SCSI target device object that contains a task router and acts as the connection between device servers and task managers and the service delivery subsystem through which requests and confirmations are routed. See SAM-4.

SCSI target/initiator device—A device that has all the characteristics of a SCSI target device and a SCSI initiator device. See SAM-4.

SCSI target/initiator port—A SCSI target/initiator device object that has all the characteristics of a SCSI target port and a SCSI initiator port. See SAM-4.

sender—A client or server that originates a service delivery transaction.

Serial ATA Tunneled Protocol (STP)—The protocol defined in this used by STP initiator ports to communicate with STP target ports in a SAS domain.

Serial Attached SCSI (SAS)—The set of protocols and the interconnect defined by this manual.

Serial Management Protocol (SMP)—The protocol defined in this used by SAS devices to communicate management information with other SAS devices in a SAS domain.

Serial SCSI Protocol (SSP)—The protocol defined in this used by SCSI initiator ports to communicate with SCSI target ports in a SAS domain.

server—A SCSI object that performs a service on behalf of a client.

service—Any operation or function performed by a SCSI object, which can be invoked by other SCSI objects.

service delivery failure—Any non-recoverable error causing the corruption or loss of one or more service delivery transactions while in transit.

service delivery port—A device-resident interface used by the application client, device server or task manager to enter and retrieve requests and responses from the service delivery subsystem. Synonymous with “port.”

service delivery subsystem—The part of a SCSI I/O system that transmits information between a SCSI initiator port and a SCSI target port, or the part of an ATA I/O system that transmits information between an ATA host and an ATA device, or the part of a SAS I/O system that transmits information between a SAS initiator port and a SAS target port.

service delivery transaction—A request or response sent through the service delivery subsystem.

signal—(n) A detectable asynchronous event possibly accompanied by descriptive data and parameters. (v) The act of generating such an event.

single transition (ST)—The latching of data only on the assertion edge of the REQ or ACK signals.

SMP initiator phy—A SAS initiator phy in an SMP initiator port.

SMP initiator port—A SAS initiator device object in a SAS domain that interfaces to the service delivery subsystem with SMP.

SMP phy—A SAS phy in an SMP port.

SMP port—An SMP initiator port, SMP target port, or SMP target/initiator port.

SMP target phy—A SAS target phy in an SMP target port.

SMP target port—A SAS target device object in a SAS domain that interfaces to the service delivery subsystem with SMP.

SMP target/initiator port—A port that has all the characteristics of an SMP initiator port and an SMP target port.

source device—The SCSI device from which a service delivery transaction originates. *See* destination device.

speed negotiation lock time (SNLT)—The maximum time during a speed negotiation window for a transmitter to reply with ALIGN (1).

speed negotiation sequence—A sequence in which two phys negotiate the operational physical link rate.

speed negotiation transmit time (SNTT)—The time during which ALIGN (0) or ALIGN (1) is transmitted during the speed negotiation sequence.

spread spectrum clocking—The technique of modulating the operating frequency of a transmitted signal to reduce the measured peak amplitude of radiated emissions.

SSP initiator phy—A SAS initiator phy in an SSP initiator port.

SSP initiator port—A SCSI initiator port in a SAS domain that implements SSP.

SSP phy—A SAS phy in an SSP port.

SSP port—An SSP initiator port, SSP target port, or SSP target/initiator port.

SSP target phy—A SAS target phy in an SSP target port.

SSP target port—A SCSI target port in a SAS domain that implements SSP.

SSP target/initiator port—A port that has all the characteristics of an SSP initiator port and an SSP target port.

STP initiator phy—A SAS initiator phy in an STP initiator port.

STP initiator port—A SAS initiator device object in a SAS domain that interfaces to the service delivery subsystem with STP.

STP phy—A SAS phy in an STP port.

STP port—An STP initiator port, STP target port, or STP target/initiator port.

STP target phy—A SAS target phy in an STP target port.

STP target port—A SAS target device object in a SAS domain that interfaces to the service delivery subsystem with STP.

STP target/initiator port—A port that has all the characteristics of an STP initiator port and an STP target port.

STP/SATA bridge—An expander device object containing an STP target port, a SATA host port, and the functions required to forward information between the STP target port and SATA host port to enable STP initiator ports in a SAS domain to communicate with SATA devices in an ATA domain.

subsystem—An element in a hierarchically partitioned system which interacts directly only with elements in the next higher division or the next lower division of that system.

subtractive routing attribute—The attribute of an edge expander phy that indicates it may be used by the expander connection manager to route connection requests not resolved using the direct routing method or table routing method.

subtractive routing method—The method the expander connection manager uses to route connection requests not resolved using the direct routing method or table routing method to an expander device.

suspended information—Information stored within a logical unit that is not available to any pending tasks.

table routing attribute—The attribute of an expander phy that indicates it may be used by the expander connection manager to route connection requests using an expander route table.

table routing method—The method the expander connection manager uses to route connection requests to an expander device using an expander route table.

target—A SCSI device which receives SCSI commands and directs such commands to one or more logical units for execution.

task—An object within the logical unit representing the work associated with a command or group of linked commands. A task consists of one initial connection and zero or more physical or logical reconnections, all pertaining to the task.

task abort event—An event or condition indicating that the task has been aborted by means of a task management function.

task address—a SCSI initiator port identifies a task to a SCSI target port using a Task Address. The Task Address object represents either a Tagged Task Address or an Untagged Task Address without regard for the tagged or untagged nature of the Task. A Tagged Task Address is composed of a Logical Unit Identifier and a Tag. An Untagged Task Address is composed of a Logical Unit Identifier.

task completion event—An event or condition indicating that the task has ended with a service response of Task Complete.

task ended event—An event or condition indicating that the task has completed or aborted.

task management function—A task manager service which can be invoked by an application client to affect the execution of one or more tasks.

t**ask management request**—A request submitted by an application client, invoking a task management function to be executed by a task manager.

task management response—The response returned to an application client by a task manager on completion of a task management request.

task manager—A server within the target which executes task management functions.

task set—A group of tasks within a SCSI target port device, whose interaction is dependent on the queuing and auto contingent allegiance rules

task slot—Resources within the logical unit that may be used to contain a task.

task tags—A Tag is a field containing up to 64 bits that is a component of a Tagged Task Identifier. A SCSI initiator port assigns tag values in each Tagged Task Identifier in a way that ensures that the identifier uniqueness requirements stated in ANSI SAM-4, T10/1683-D, Section 4.9, are met.

third-party command—A SCSI command which requires a logical unit within the target device to assume the initiator role and send a SCSI command to a SCSI target port device.

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

transaction—A cooperative interaction between two objects, involving the exchange of information or the execution of some service by one object on behalf of the other.

transceiver—An object that contains both transmitter and receiver objects.

transmitter—The source or generator of a signal.

t**ransmitter compliance transfer function (TCTF)**—The mathematical statement of the transfer function through which the transmitter shall be capable of producing acceptable signals as defined by a receive mask.

transport protocol service confirmation—A message passed from the transport layer to the application layer (i.e., from the SSP initiator port to the SCSI application client) that notifies the application layer that a SCSI transport protocol service has completed.

transport protocol service indication—A message passed from the transport layer to the application layer notifying the application layer (i.e., from the SSP target port to the SCSI device server) to begin a SCSI transport protocol service.

transport protocol service request—A message passed from the SCSI application layer to the SSP transport layer (i.e., from the SCSI application client to the SCSI initiator port) to begin a SCSI transport protocol service.

transport protocol service response—A message passed from the application layer to the transport layer (i.e., from the SCSI device server to the SSP target port) that completes the SCSI transport protocol service.

unit interval (UI)—The time required to transmit one bit on a physical link (e.g., 666,$\overline{6}$ ps at 1,5 Gbps and 333,$\overline{3}$ ps at 3,0 Gbps).

unlinked command—A SCSI command having the link bit set to zero in the CDB control byte.

upper level protocol—An application-specific protocol executed through services provided by a lower level protocol.

valid dword—A dword that is not an invalid dword.

virtual phy—A phy (see) that interfaces to another virtual phy inside the same device.

wide link—A group of physical links that attaches a wide port to another wide port.

wide port—A port that contains more than one phy.

1.3.4 Keywords

Several keywords are used to differentiate between different levels of requirements and optionality, as follows:

vendor-specific—Specification of the referenced item is determined by the device vendor.

protocol-specific—Implementation of the referenced item is defined by a SCSI protocol (see Section .)

expected—A keyword used to describe the behavior of the models specified by this.

ignored—A keyword used to describe an unused bit, byte, word, field or code value. The contents or value of an ignored bit, byte, word, field or code value is not examined by the receiving SCSI device and may be set to any value by the transmitting SCSI device.

invalid—A keyword used to describe an illegal or unsupported bit, byte, word, field, or code value. Receipt of an invalid bit, byte, word, field, or code value shall be reported as an error.

mandatory—A keyword indicating items required to be implemented as defined by this.

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

may not—Keywords that indicates flexibility of choice with no implied preference (equivalent to “may or may not”).

obsolete—A keyword indicating items that were defined in prior SCSI specifications but have been removed from this.

option, optional—Keywords that describe features which are not required to be implemented by this. However, if any optional feature defined by the is implemented, it shall be implemented as defined by the.

reserved—A key word referring to bits, bytes, words, fields, and code values that are set aside for future ization. Their use and interpretation may be specified by future extensions to this or other specifications. A reserved bit, byte, word, or field shall be set to zero, or in accordance with a future extension to this. Recipients are not required to check reserved bits, bytes, words, or fields for zero values. Receipt of reserved code values in defined fields shall be treated as an error.

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

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

1.4 Physical interface characteristics

The physical interface characteristics (cables, connectors, electrical descriptions, etc.) for the drives covered by this Interface Manual are found in each individual drive's product manual since these features are not the same for all drives.

1.5 Bit and byte ordering

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

In a field in a table consisting of more than one byte that contains a single value (e.g., a number), the byte containing the MSB is stored at the lowest address and the byte containing the LSB is stored at the highest address (i.e., big-endian byte ordering). The MSB and LSB are labeled.

Note. SATA numbers bits within fields the same as this standard, but uses little-endian byte ordering.

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

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

A data dword consists of 32 bits. Table 1 shows a data dword containing a single value, where the MSB is on the left in bit 31 and the LSB is on the right in bit 0.

Table 1. Data dword containing a value

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB		Value																												LSB	

Table 2 shows a data dword containing four one-byte fields, where byte 0 (the first byte) is on the left and byte 3 (the fourth byte) is on the right. Each byte has an MSB on the left and an LSB on the right.

Table 2. Data dword containing four one-byte fields

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB		Byte 0 (First byte)				LSB		MSB		Byte 1 (Second byte)				LSB		MSB		Byte 2 (Third byte)				LSB		MSB		Byte 3 (Fourth byte)				LSB	

2.0 General

2.1 Architecture

2.1.1 Architecture overview

A SAS domain (see 2.1.7) contains two or more SAS devices and a service delivery subsystem. A SAS domain may support three transport protocols:

a. Serial SCSI Protocol (SSP): a mapping of SCSI supporting multiple initiators and targets;
b. Serial ATA Tunneled Protocol (STP): a mapping of Serial ATA expanded to support multiple initiators and targets; and
c. Serial Management Protocol (SMP): a management protocol.

Drives described in this manual support only SSP (see SAM-4). For a description of STP and SMP see the SAS standard.

A SAS device (see 2.1.4) contains one or more SAS ports (see 2.1.3). A SAS device may be a SCSI device (see SAM-4). The drives supported by this manual have two ports.

A SAS port (see 2.1.3) contains one or more phys (see 2.1.2). A SAS port may be a SCSI port (see SAM-4). The drives supported by this manual have one phy per port.

The service delivery subsystem (see 2.1.6) in a SAS domain may contain expander devices to increase the number of SAS devices supported (see 2.1.5).

Expander devices contain expander ports (see 2.1.3) and one SMP port. The SMP port is used for discovery and configuration of the expander.

An expander port contains one or more phys (see 2.1.2).

An expander device shares its phys with the SAS device(s) contained within the expander device.

2.1.2 Physical links and phys

A physical link is a set of four conductors used as two differential signal pairs. One differential signal transmits in one direction while the other differential signal transmits in the opposite direction. Data may be transmitted in both directions simultaneously.

A physical phy (Phy) contains a transceiver which electrically interfaces to a physical link, which attaches to another physical phy.

Phys are contained in ports (see 2.1.3). Phys interface to the service delivery subsystem (see 2.1.6).

Figure 4 shows two phys attached with a physical link.

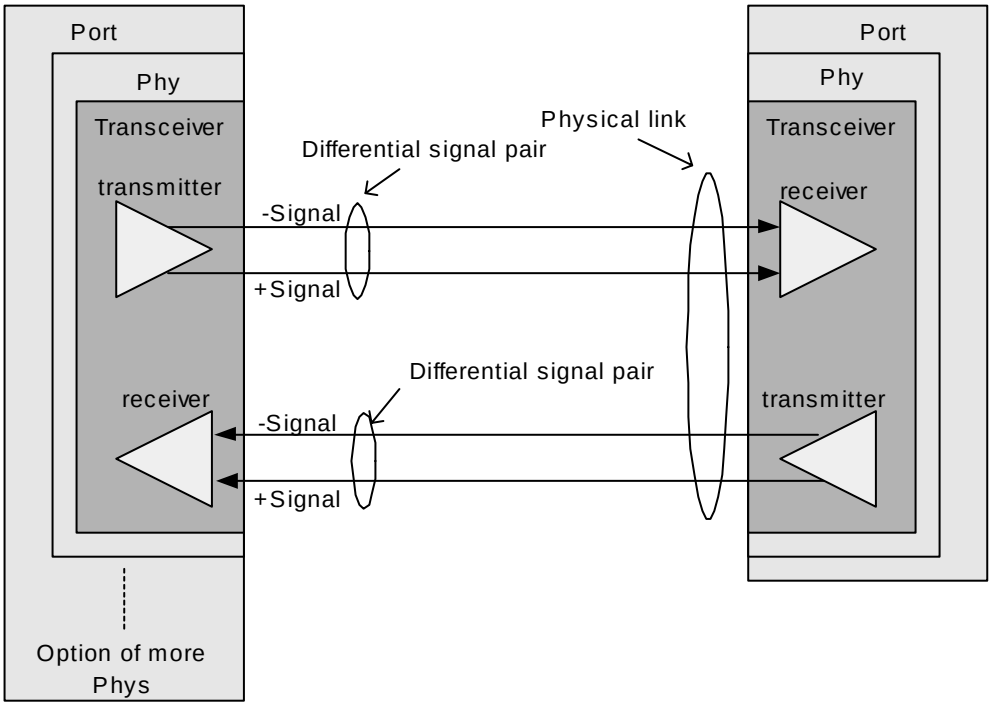


Figure 4. Physical links and phys

An attached phy is the phy to which a phy is attached over a physical link.

A device may contain one or more phys. Each phy is assigned a phy identifier which is unique within the device. Each phy has a SAS address, inherited from the SAS port or expander.

Phys transmit and receive bits at physical link rate. The physical link rates supported by a phy are specified or indicated by the NEGOTIATED PHYSICAL LINK RATE field, HARDWARE MINIMUM PHYSICAL LINK RATE field, the HARDWARE MAXIMUM PHYSICAL LINK RATE field, the PROGRAMMED MINIMUM PHYSICAL LINK RATE field, and the PROGRAMMED MAXIMUM PHYSICAL LINK RATE field in the SMP DISCOVER function, SMP PHY CONTROL function, and Phy Control and Discover subpage. The bits are part of dwords (4 bytes), each of which has been encoded using 8b10b coding into four 10-bit characters.

2.1.3 Ports (narrow ports and wide ports)

A port contains one or more phys. Ports in a device are associated with physical phys based on the identification sequence.

A port is created from a set of physical phys if one or more physical phys contained within a device:

a. transmit the same SAS address (see 2.2) during the identification sequence; and

b. receive the same SAS address during the identification sequence (i.e., the corresponding attached phy or phys transmit the same SAS address).

A wide port is created if there is more than one phy in the port. A narrow port is a port with only one phy. The drives supported by this manual implement narrow ports.

A wide link is the set of physical links that attach a wide port to another wide port. A narrow link is the physical link that attaches a narrow port to another narrow port.

Figure 5 shows examples of narrow ports and wide ports, with a representation of the SAS address transmitted during the identification sequence. Although several phys on the left transmit SAS addresses of B, only phys attached to the same SAS addresses become part of the same ports. The set of phys with SAS address B attached to the set of phys with SAS address Y become one port, while the set of phys with SAS address B attached to the set of phys with SAS address Z become another port.

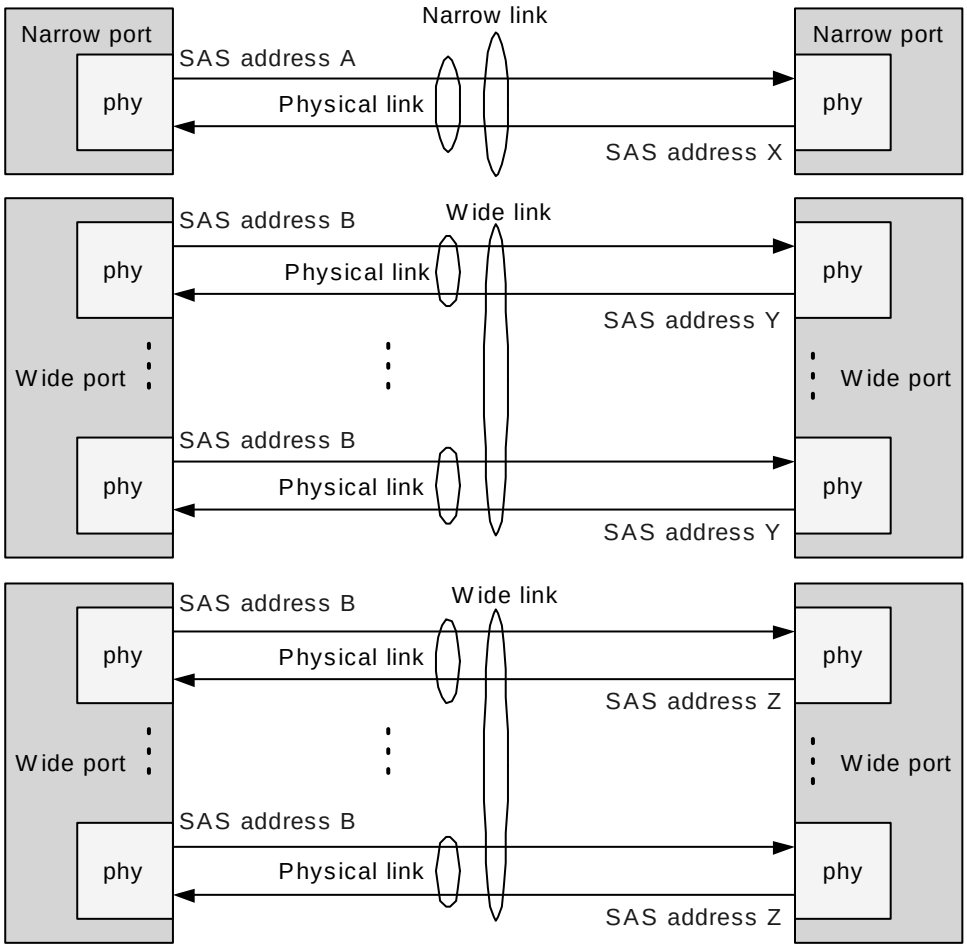


Figure 5. Ports (narrow ports and wide ports)

In figures in this manual that show ports but not phys, the phy level of detail is not shown; however, each port always contains one or more phys.

2.1.4 SAS devices

A SAS device contains one or more SAS ports, each containing one or more phys (i.e., a SAS port may be a narrow port or a wide port).

An end device is a device not contained in an expander.

Figure 6 shows examples of SAS devices with different port and phy configurations.

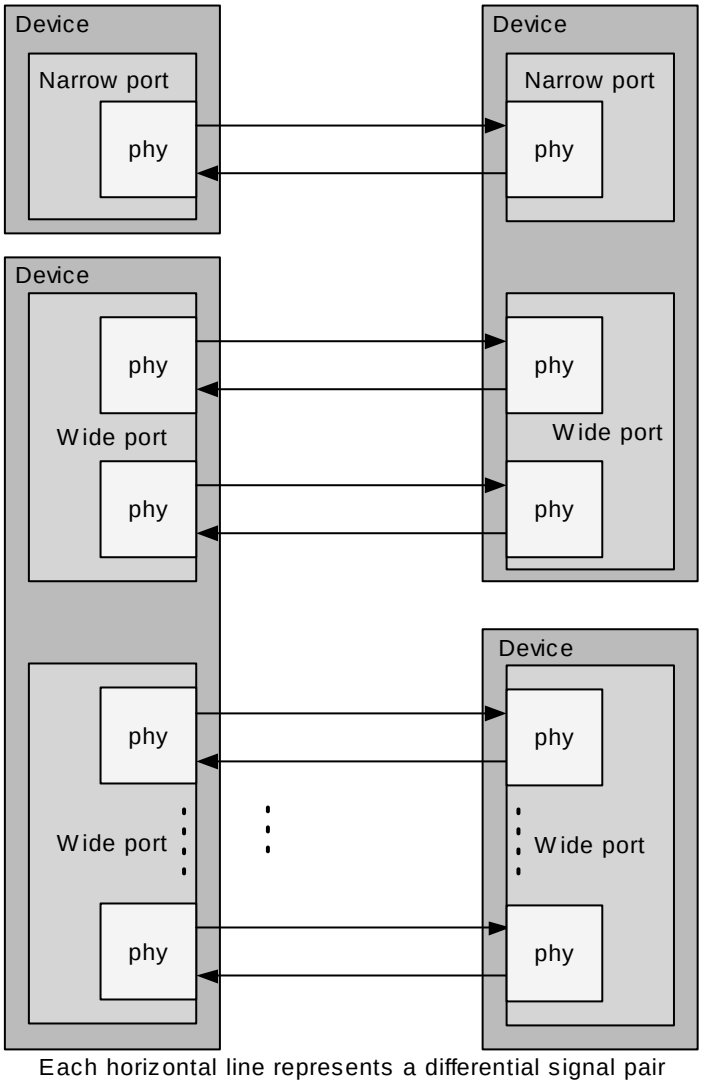


Figure 6. SAS devices

2.1.5 Expander devices (edge expander devices and fanout expander devices)

Expander devices are part of the service delivery subsystem and facilitate communication between multiple SAS devices. Expander devices contain two or more external expander ports. Each expander device contains at least one SMP target port for management and may contain SAS devices (e.g., an expander device may include an SSP target port for access to a SCSI enclosure services logical unit).

There are two types of expander devices:

a. edge expander devices; and

b. fanout expander devices.

An edge expander device is part of an edge expander device set (see 2.1.8.2). A fanout expander device is an expander device capable of being attached to two or more edge expander device sets.

Edge expander devices may contain phys with the subtractive routing attribute. A fanout expander device does not contain any phys with the subtractive routing attribute. Edge expander devices and fanout expander devices may also contain phys with direct routing and table routing attributes.

Expander devices with expander phys with the table routing attribute contain an expander route table. The expander route table may be configurable. An expander device implementation may either be self discover routing information or depend on a management application client within the SAS domain to use the discover process to configure the expander route table. An expander device with expander phys with the table routing attribute that does not have a configurable route table shall be self-configuring, and shall contain a management application client and SMP initiator port to perform the discover process to configure its own expander route table.

2.1.6 Service delivery subsystem

The service delivery subsystem is either:

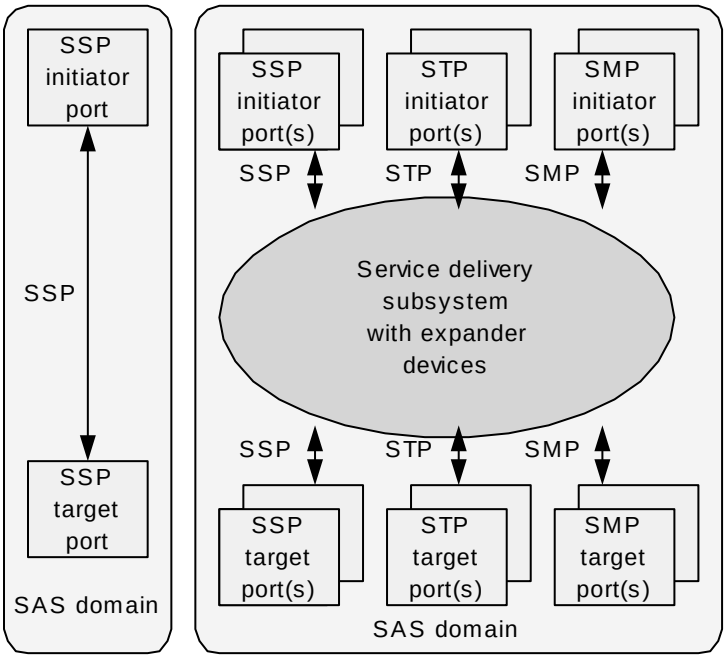
a. a set of physical links between a SAS initiator port and a SAS target port; or

b. a set of physical links and expander devices, supporting more than two SAS ports.

See 2.1.8 for rules on constructing service delivery subsystems from multiple expander devices.

2.1.7 Domains

Figure 7 shows examples of SAS domains.



Note: When expander devices are present, SAS target ports may be located in SAS devices contained in expander devices.

Figure 7. Domains

Figure 8 shows SAS initiator devices and SAS target devices with SAS ports in the same SAS domains and in different SAS domains. When a SAS device has ports in different SAS domains, the ports may have the same SAS address (see 2.2); when its ports are in the same SAS domain, they shall have different SAS addresses.

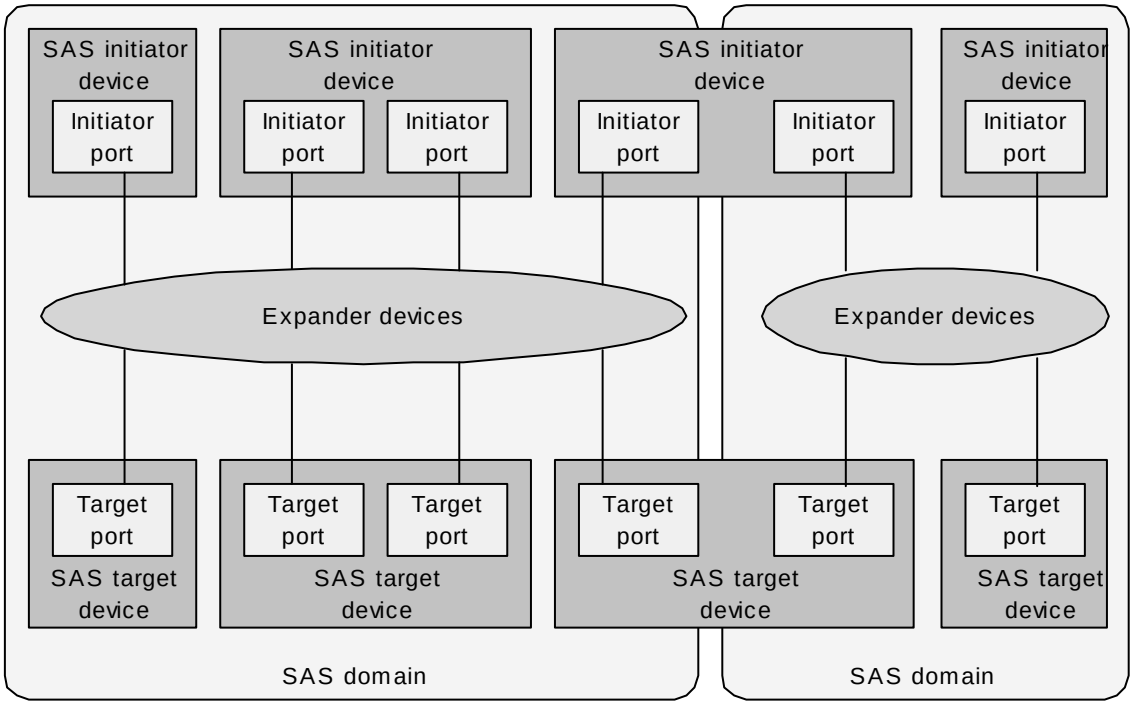


Figure 8. Devices spanning SAS domains

2.1.8 Expander device topologies

2.1.8.1 Expander device topology overview

More than one expander device may be part of a service delivery subsystem.

2.1.8.2 Edge expander device set

One or more edge expander devices may be grouped into an edge expander device set. The number of edge expander devices and the routing attributes of the expander phys in the edge expander devices within an edge expander device set shall be established when the edge expander device set is constructed. The method used to construct edge expander sets is outside the scope of this.

An edge expander device set is bounded by expander phys with the direct routing attribute that are either:

a. attached to end devices; or

b. not attached to any device,

and by expander phys with the subtractive routing attribute that are:

a. attached to expander phys of a fanout device;

b. attached to expander phys with subtractive routing attributes on another edge expander device set;

c. attached to an end device; or

d. not attached to any device.

Phys with table routing attributes within an edge expander device set are used to provide attachments between edge expander devices in the edge expander device set.

The number of SAS addresses used by an edge expander device set is limited to 128. This includes SAS addresses for internal and attached end devices.

To avoid an overflow of an edge expander route index during the discover process, an edge expander device set shall be constructed such that the number of edge expander route indexes available per phy identifier is greater than or equal to the sum of all SAS addresses addressable through the edge expander phy.

An edge expander device set shall not be attached to more than one fanout expander device.

An edge expander device set may be attached to one other edge expander device set if that is the only other edge expander device set in the SAS domain, they are attached using expander phys with subtractive routing attributes, and there are no fanout expander devices in the SAS domain.

Figure 9 shows an edge expander device set.

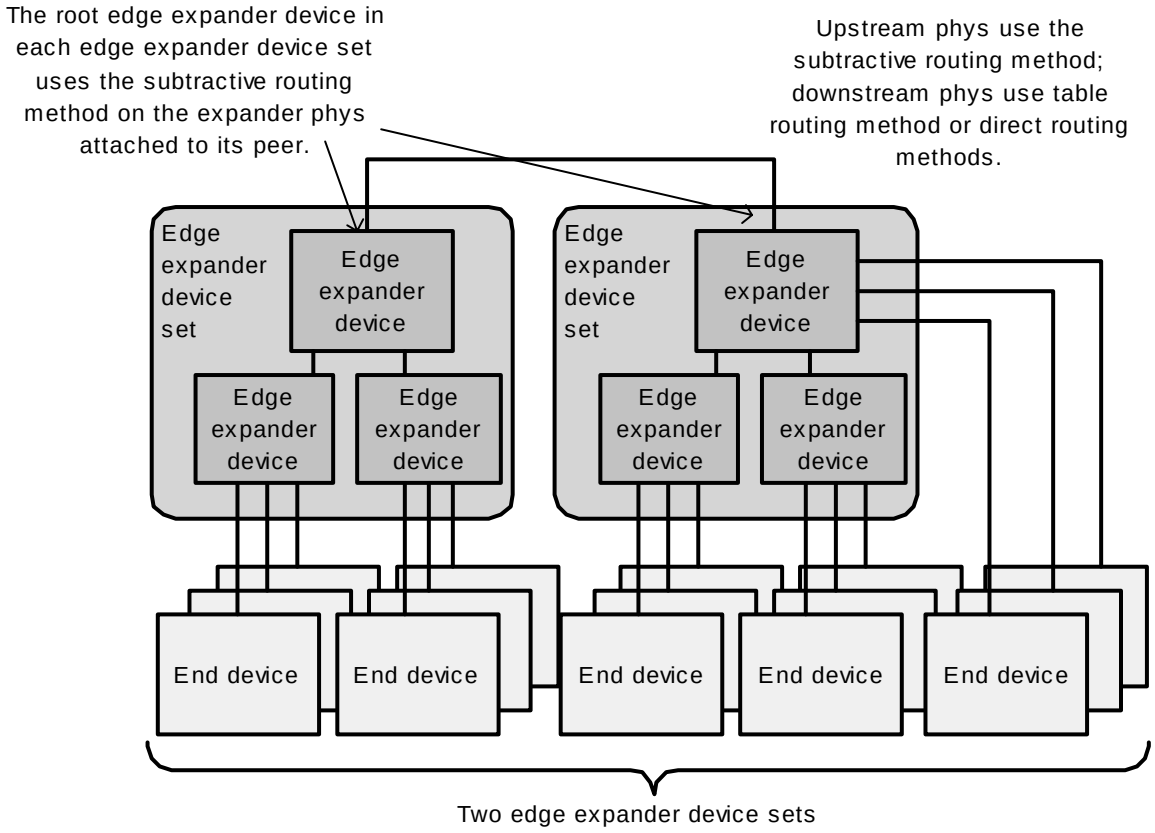


Figure 9. Edge expander device set

2.1.8.3 Expander device topologies

No more than one fanout expander device shall be included in a SAS domain. The fanout expander device may be attached to up to 128 edge expander device sets or SAS ports.

Figure 10 shows a SAS domain with the maximum number of expander device sets.

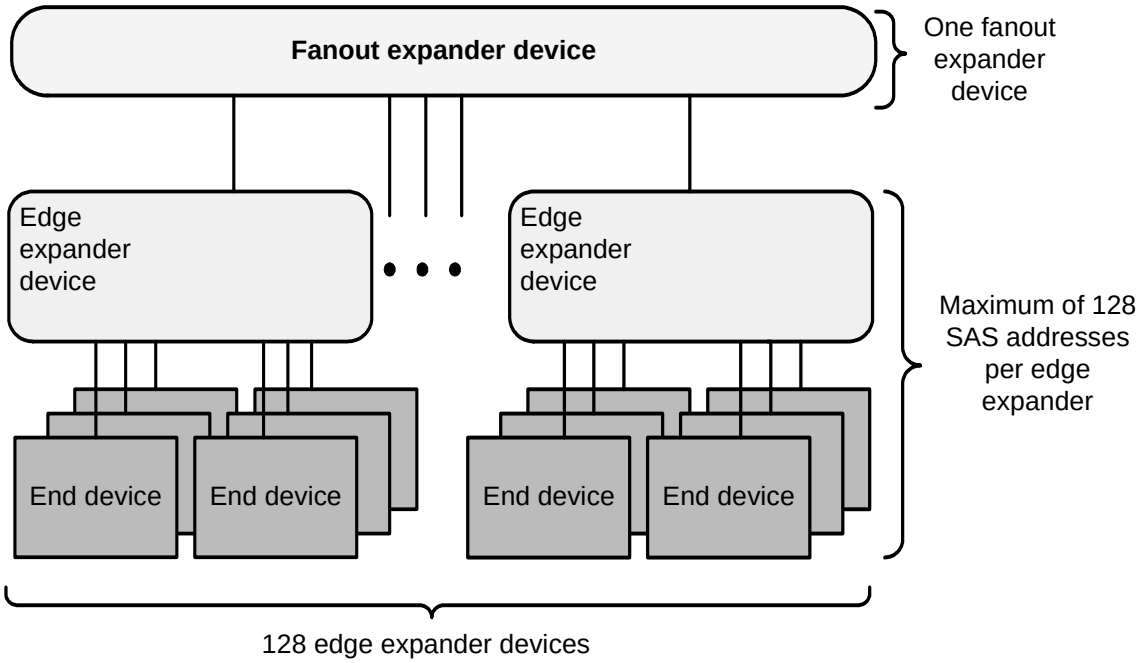


Figure 10. Maximum expander device set topology

End devices may be attached directly to the fanout expander device, as shown in figure 10.

2.1.9 Pathways

A potential pathway is a set of physical links between a SAS initiator phy and a SAS target phy. When a SAS initiator phy is directly attached to a SAS target phy, there is one potential pathway. When there are expander devices between a SAS initiator phy and a SAS target phy, there are multiple potential pathways, each consisting of a set of physical links between the SAS initiator phy and the SAS target phy. The physical links may or may not be using the same physical link rate.

A pathway is a set of physical links between a SAS initiator phy and a SAS target phy being used by a connection (see 2.1.10).

Figure 11 shows examples of potential pathways.

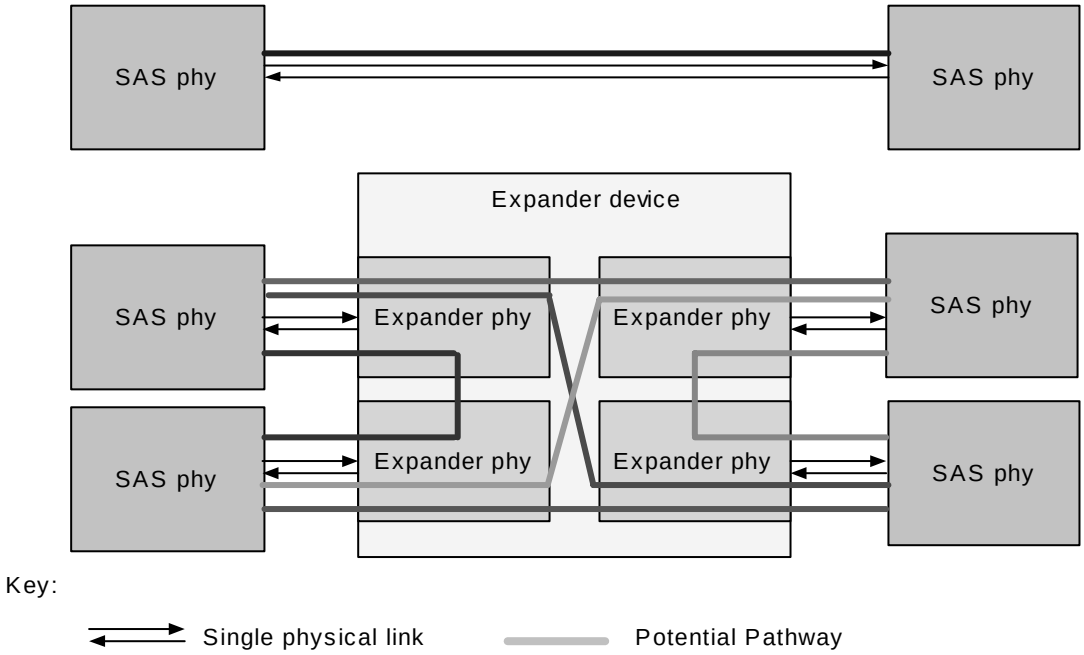


Figure 11. Potential pathways

A partial pathway is the set of physical links participating in a connection request that has not reached the destination phy (e.g., the OPEN address frame has been transmitted by the source phy but the OPEN address frame has not yet reached the destination phy).

A partial pathway is blocked when path resources it requires are held by either another connection or another partial pathway.

2.1.10 Connections

A connection is a temporary association between a SAS initiator port and a SAS target port. During a connection all dwords from the SAS initiator port are forwarded to the SAS target port, and all dwords from the SAS target port are forwarded to the SAS initiator port.

A connection is pending when an OPEN address frame has been delivered along a completed pathway to the destination phy but the destination phy has not yet responded to the connection request. A connection is established when an OPEN_ACCEPT is returned to the source phy.

A connection enables communication for one protocol: SSP, STP, or SMP. The drive rejects OPEN address frames requesting a STP or SMP connection. For SSP, connections may be opened and closed multiple times during the processing of a command.

The connection rate is the effective rate of dwords through the pathway between a SAS initiator phy and a SAS target phy, established through the connection request. Every phy shall support a 1.5 Gbps connection rate regardless of its physical link rate.

One connection may be active on a physical link at a time. For a SSP connection and there are no dwords to transmit associated with that connection, idle dwords are transmitted. If there is no connection on a physical link then idle dwords are transmitted.

The drive supports 1 connection per port.

If multiple potential pathways exist between the SAS initiator port(s) and the SAS target port(s), multiple connections may be established by a SAS port between the following:

a. one SAS initiator port to multiple SAS target ports;

b. one SAS target port to multiple SAS initiator ports; or

c. one SAS initiator port to one SAS target port.

Once a connection is established, the pathway used for that connection shall not be changed (i.e., all the physical links that make up the pathway remain dedicated to the connection until it is closed).

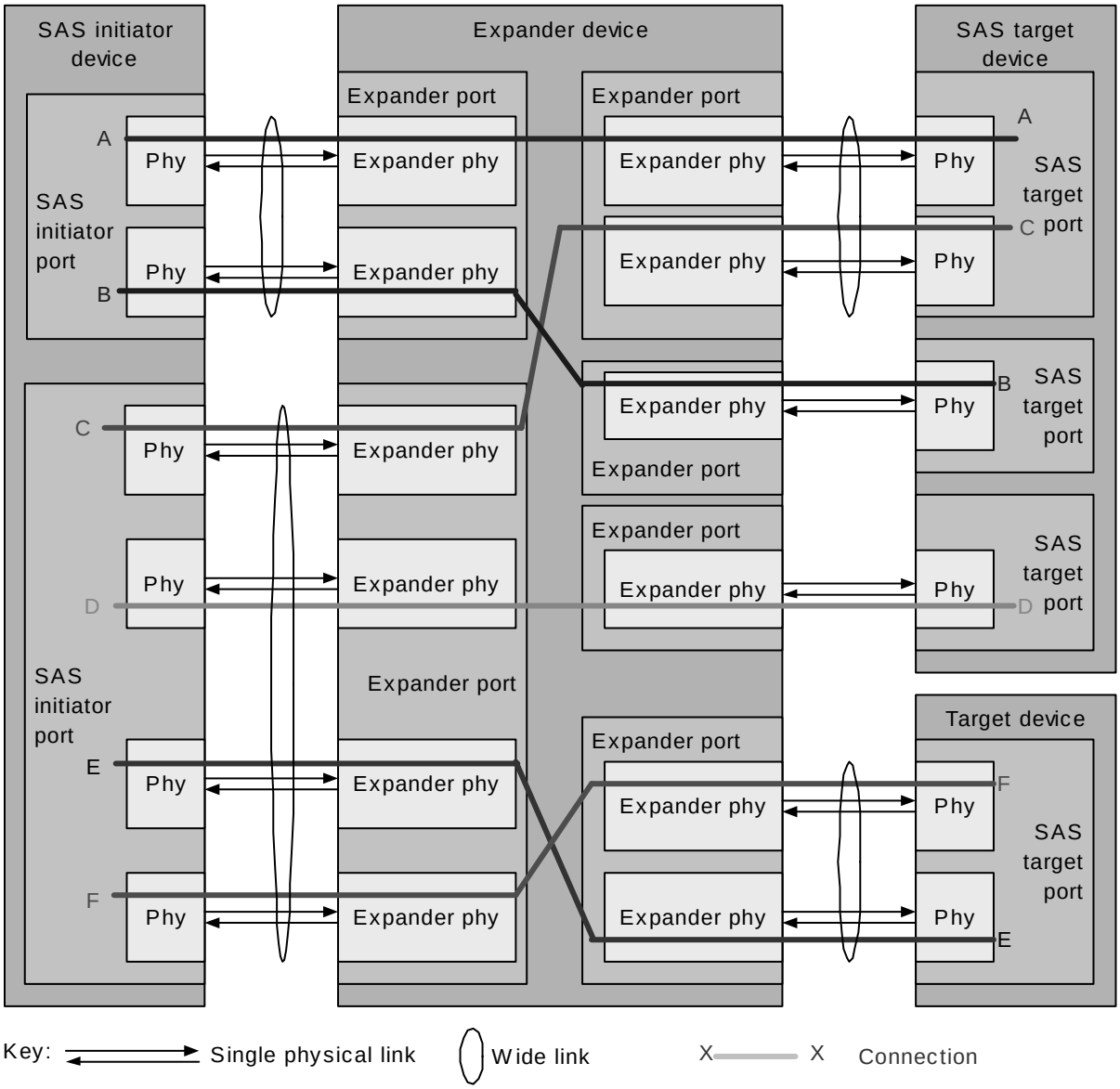
Figure 12 shows examples of connections between wide and narrow ports. All the connections shown may occur simultaneously. Additionally:

a. the connections labeled A and B are an example of one SAS initiator port with connections to multiple SAS target ports;

b. the connections labeled A and C are an example of one SAS target port with connections to multiple SAS initiator ports;

c. the connections labeled E and F are an example of multiple connections between one SAS initiator port and one SAS target port; and

d. the connections labeled C, D, E, and F are an example of one SAS initiator port with connections to multiple SAS target ports with one of those SAS target ports having multiple connections with that SAS initiator port.



Notes: The expander device has a unique SAS address. Each SAS initiator port and SAS target port has a unique SAS address. Connections E and F represent a wide SAS initiator port with two simultaneous connections to a wide SAS target port.

Figure 12. Multiple connections on wide ports

2.2 Names and identifiers

2.2.1 Names and identifiers

Four SAS addresses are allocated for each drive.

a. The drive has a general SAS address that does not refer to a port but to SAS device address. This is stored in inquiry vital product page 83h.

b. Port A has a unique SAS address. This is used for any identify or open address frames transmitted from this port.

c. Port B has a unique SAS address. This is used for any identify or open address frames transmitted from this port.

d. The drive has a SAS address for the single LUN supported in the drive.

2.2.2 SAS addresses

Table 3 defines the SAS address format. SAS addresses shall be compatible with the NAA (Name Address Authority) IEEE Registered format identification descriptor defined in SPC-4.

Table 3. SAS address format

Byte\Bit	7	6	5	4	3	2	1	0
0	NAA (5h)				(MSB)			
1	IEEE COMPANY ID							
2								
3				(LSB)	(MSB)			
4	VENDOR-SPECIFIC IDENTIFIER							
5								
6								
7								(LSB)

The NAA field contains 5h.

The IEEE COMPANY ID field contains a 24-bit canonical form company identifier assigned by the IEEE. Information about IEEE company identifiers may be obtained from the http://s.ieee.org/regauth/oui web site.

The VENDOR-SPECIFIC IDENTIFIER contains a 36-bit numeric value that is uniquely assigned by the organization associated with the company identifier in the IEEE COMPANY ID field.

Table 4 describes the way the drive SAS addresses are mapped into the address format.

Table 4. Drive SAS addresses

Object	Byte 7, Bits 1 &0
Device name	00b
Port A Identifier	01b
Port B Identifier	10b
LUN name	11b

2.2.3 Hashed SAS address

SSP frames include a hashed version of the SAS address to provide an additional level of verification of proper frame routing.

The generator polynomial for this code is:

$G(x) = (x^6 + x + 1) (x^6 + x^4 + x^2 + x + 1) (x^6 + x^5 + x^2 + x + 1) (x^6 + x^3 + 1)$

After multiplication of the factors, the generator polynomial is:

$G(x) = x^{24} + x^{23} + x^{22} + x^{20} + x^{19} + x^{17} + x^{16} + x^{13} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x + 1$

2.2.4 Device names

Each expander device, SAS initiator device, SAS target device, and SAS target/initiator device shall include a SAS address (see 2.2.2) as its device name. The selected SAS address shall be used for no other name or identifier.

Expander devices report their device names in the IDENTIFY address frame.

Logical units accessed through SSP target ports report SAS target device names through SCSI vital product data.

2.2.5 Port identifiers

Each SAS initiator port, SAS target port, and SAS target/initiator port has a SAS address (see 2.2.2) as its port identifier. The selected SAS address is unique.

SAS ports report their port identifiers in the IDENTIFY address frame. Port identifiers are used as source and destination addresses in the OPEN address frame.

The logical unit accessed through SSP target ports report drive identifiers through SCSI vital product data.

2.2.6 Phy identifiers

Each SAS phy and expander phy is assigned an identifier that is unique within the SAS device and/or expander device. The phy identifier is used for management functions.

Phy identifiers assigned in the drive are 0h for port A and 1h for port B.

2.3 Resets

2.3.1 Reset overview

Figure 13 describes the SAS reset terminology:

a. link reset sequence;
b. phy reset sequence;
c. SAS OOB sequence;
d. SAS speed negotiation sequence;
e. hard reset sequence; and
f. identification sequence.

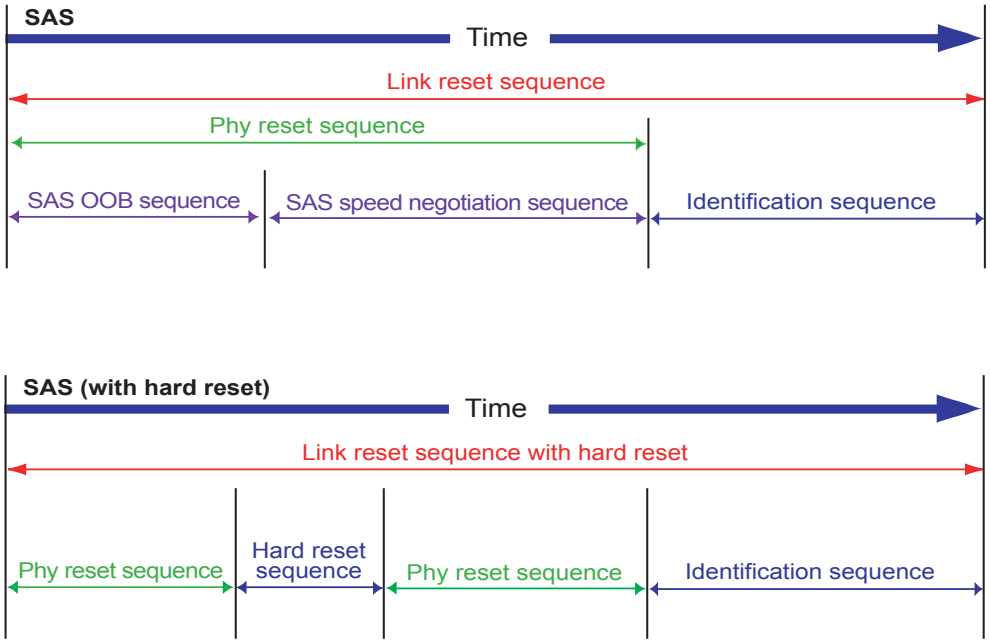


Figure 13. Reset terminology

The link reset sequence has no effect on the transport layer and application layer. HARD_RESET may be used during the identification sequence to initiate a hard reset.

2.3.2 Hard reset

When a HARD_RESET is received on a port phy, the drive:

a. stops transmitting dwords on the phys of that port,
b. terminates any existing connection on the other port,
c. abort all commands regardless of which port they were received,
d. performs an internal reset,
e. establishes a unit attention (SCSI bus reset occurred) for all initiators on both ports, and
f. initiates a link reset on the phy the HARD_RESET was received.

A link reset is not initiated on the phy port that did not receive the HARD_RESET.

2.4 I_T nexus loss

When the drive receives OPEN_REJECT (NO DESTINATION), OPEN_REJECT (PATHWAY BLOCKED), or an open connection time-out occurs in response to a connection request, it retries the connection request until:

a. the connection is established; or

b. for SSP target ports, the time indicated by the I_T NEXUS LOSS field in the Protocol-Specific Port Control mode page expires.

If the time expires, then the drive sends a Nexus Lost event notification to the SCSI application layer; the SCSI device shall perform the actions defined for I_T nexus loss in SAM-4.

3.0 Phy layer

3.1 Phy layer overview

The phy layer defines 8b10b coding and OOB signals. Phy layer interfaces between the link layer and the physical layer to perform the phy reset sequence and keep track of dword synchronization.

3.2 Encoding (8b10b)

3.2.1 Encoding overview

All data bytes transferred in SAS are encoded into 10-bit data characters using 8b10b coding. Additional characters not related to data bytes are called control characters.

Encoding is used to make the transfer of bits on the physical link more reliable.

All characters transferred in SAS are grouped into four-character sequences called dwords. A primitive is a dword whose first character is a control character and remaining three characters are data characters. A data dword is a dword starting with a data character.

Primitives are defined with both negative and positive starting running disparity. SAS defines primitives starting with the K28.3, K28.5 and K28.6 control characters. Table 5 shows special character usage. SSP only uses primitives starting with K28.5. Drives supported by this manual ignore primitives that do not start with K28.5.

Table 5. Special character usage

First character	Usage in SAS
K28.3	Primitives used only inside STP connections
K28.5	ALIGN and most primitives defined in this
K28.6	SATA_ERROR (used on SATA physical links)
Dxx.y	Data

Primitives are defined in Section.

Running disparity is maintained separately on each physical link. Expander devices convert incoming 10-bit characters to 8-bit bytes and generate the 10-bit character with correct disparity for the output physical link. Physical links may begin operation with either positive or negative disparity after the reset sequence.

3.2.2 8b10b coding introduction

Information to be transmitted across a physical link is encoded eight bits at a time into a 10-bit transmission character and then transmitted serially bit-by-bit across the physical link. Information received over the physical link is collected ten bits at a time, and those transmission characters that are used for data, called data characters, is decoded into the correct 8-bit codes. The 10-bit transmission code supports all 256 8-bit combinations and the control characters.

The encodings defined by the transmission code ensure that sufficient transitions are present in the serial bit stream to make clock recovery possible at the receiver. Such encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information. In addition, some of the special characters of the transmission code contain a distinct and easily recognizable bit pattern (a comma) which assists a receiver in achieving word alignment on the incoming bit stream.

3.3 Encoding and decoding

An unencoded eight bit character is represented as HGFEDCBA where H is the most significant bit. The hex to decimal value translation is accomplished as shown in figure 14.

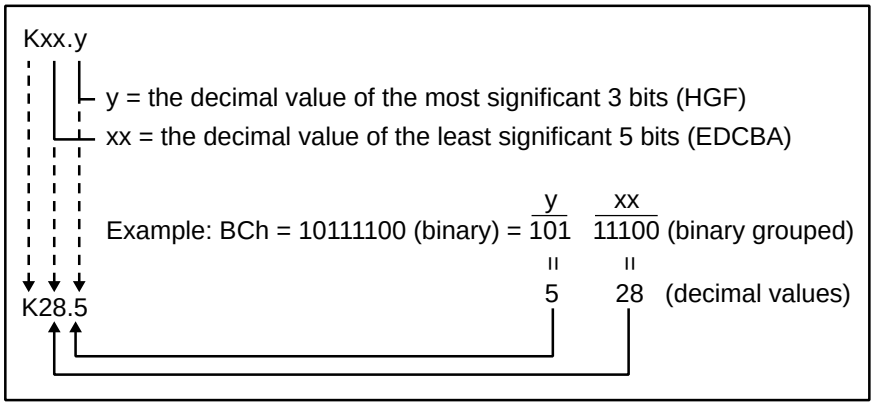


Figure 14. Decimal value translation

A decimal value is assigned to each bit combination with the range of 0 to 31 for xx and 0 to 7 for y. This means the range of valid data characters using this naming convention is D00.0 through D31.7.

Serial transmission delivers 10-bit characters which represent encoded data. Of the 1,024 characters possible with the 10-bit space, 256 8-bit byte data characters are mapped, along with several control characters. This mapping process is called 8B/10B encoding. This encoding method involves selecting encoded 10-bit characters to maintain a run-length-limited serial stream of bits. To prevent too many ones or zeros on the serial interface from causing a DC electrical shift of the serial media, the encoder monitors the number of ones in the encoded character and selects the option of the 10-bit encode character that will shift to balance the total number of zeros and ones. This balancing is called running disparity.

A 10-bit character is actually made up of 6- and 4-bit sub-blocks. The 6-bit sub-block shifts out first followed by the 4-bit sub-block. Running disparity is set positive at the end of the sub-block as follows:

- If the number of ones in a sub-block is greater that the number of zeros, the 6-bit sub-block is 000111b, or the 4-bit sub-block is 0011b. Running disparity is set positive at the end of the sub-block.
- If the number of zeros in a sub-block is greater than the number of ones, the 6-bit sub-block is 111000b, or the 4-bit sub-block is 1100b. Running disparity is set negative at the end of the sub-block.
- If the number of ones and zeros in a sub-block are equal, running disparity is neutral and the value of running disparity at the end of the sub-block remains the same as the preceding character even if it is separated by neutral characters.

Note. The rules of running disparity prohibit consecutive positive or consecutive negative characters even if they are separated by neutral disparity characters. In other words, the negative and positive disparity

characters must alternate, even if separated by a neutral disparity character. See Table 6.

Table 6. Running Disparity Examples

Sub-blocks					Valid?
1	2	3	4	5	
+	N	–	+	–	Yes
+	N	+			No
+	+				No
–	–				No

For a complete list of the valid encoded values, see SAS.

3.4 Out of band (OOB) signals

Out of band (OOB) signals are low-speed signal patterns detected by the phy that do not appear in normal data streams. They consist of defined amounts of idle time followed by defined amounts of burst time. During the idle time, D.C. idle is transmitted. During the burst time, ALIGN (0) primitives are transmitted repeatedly. The transmitter output levels during burst time and idle time are described in Section **TBD**. The signals are differentiated by the length of idle time between the burst times.

SATA defines two OOB signals: COMINIT/COMRESET and COMWAKE. COMINIT and COMRESET are used in this manual interchangeably. Phys compliant with this identify themselves with an additional SAS-specific OOB signal called COMSAS.

Table 7 defines the timing specifications for OOB signals.

Table 7. OOB signal timing specifications

Parameter	Minimum	Nominal	Maximum	Comments
COMSAS detect timeout	13.65 µs			The minimum time a receiver shall allow to detect COMSAS after transmitting COMSAS. Derived from: OOBI x 512 x 40

Table 8 describes the OOB signal transmitter requirements for the burst time, idle time, and negation times that comprise each OOB signal.

Table 8. OOB signal transmitter requirements

Signal	Burst time (ns)	Idle time (ns)	Negation time (ns)
COMWAKE	106.65 min 106.67 nom 106.68 max	106.65 min 106.67 nom 106.68 max	186.65 min 186.67 nom 186.68 max
COMINIT/RESET	106.65 min 106.67 nom 106.68 max	319.97 min 320.00 nom 320.03 max	533.28 min 533.33nom 533.37 max
COMSAS	106.65 min 106.67 nom 106.68 max	959.90 min 960.00 nom 960.10 max	1599.84 min 1600.00 nom 1600.16 max

To transmit an OOB signal, a transmitter repeats these steps six times:

1. transmit D.C. idle for an idle time; and

2. transmit an ALIGN burst for a burst time.

It then transmits D.C. idle for an OOB signal negation time.

The drive sends the ALIGNs for the burst portion of the OOB signal at generation 1 (G1) physical link rates (i.e., 1.5 Gbps).

Figure 15 describes OOB signal transmission by the SP transmitter.

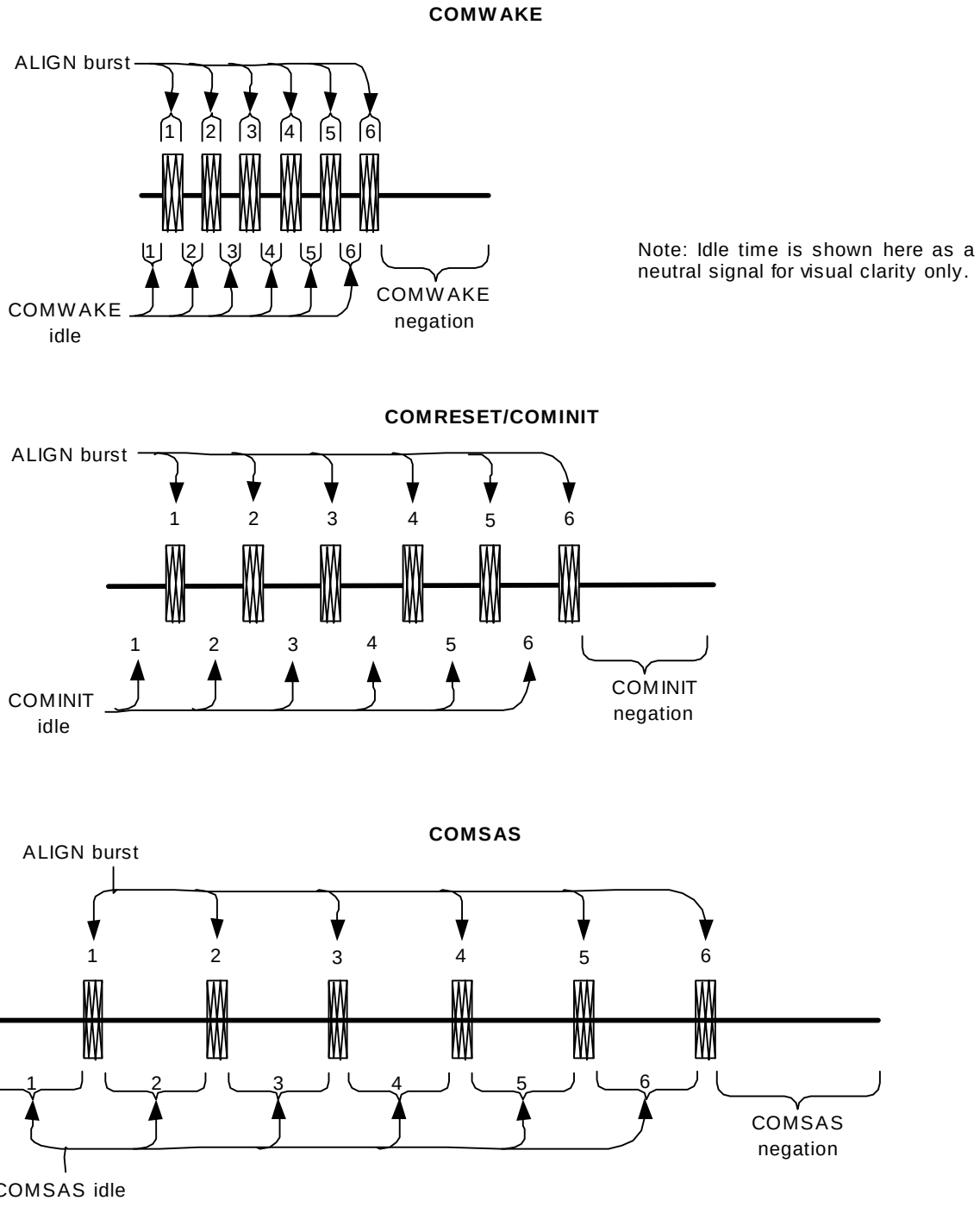


Figure 15. OOB signal transmission

Table 9 describes the OOB signal receiver requirements for detecting burst times, assuming T_{burst} is the length of the detected burst time. The burst time is not used to distinguish between signals.

Table 9. OOB signal receiver burst time detection requirements

Signal	May detect	Shall detect
COMWAKE	$T_{burst} \leq 100$ ns	$T_{burst} > 100$ ns
COMINIT/COMRESET	$T_{burst} \leq 100$ ns	$T_{burst} > 100$ ns
COMSAS	$T_{burst} \leq 100$ ns	$T_{burst} > 100$ ns

Table 10 describes the OOB signal receiver requirements for detecting idle times, assuming T_{idle} is the length of the detected idle time.

Table 10. OOB signal receiver idle time detection requirements

Signal	May detect	Shall detect	Shall not detect
COMWAKE	55 ns $\leq T_{idle} <$ 175 ns	101,3 ns $\leq T_{idle} \leq$ 112 ns	$T_{idle} <$ 55 ns or $T_{idle} \geq$ 175 ns
COMINIT/COMRESET	175 ns $\leq T_{idle} <$ 525 ns	304 ns $\leq T_{idle} \leq$ 336 ns	$T_{idle} <$ 175 ns or $T_{idle} \geq$ 525 ns
COMSAS	525 ns $\leq T_{idle} <$ 1,575 ns	911,7 ns $\leq T_{idle} \leq$ 1,008 ns	$T_{idle} <$ 525 ns or $T_{idle} \geq$ 1,575 ns

Table 11 describes the OOB signal receiver requirements for detecting negation times, assuming T_{idle} is the length of the detected idle time.

Table 11. OOB signal receiver negation time detection requirements

Signal	Shall detect
COMWAKE	$T_{idle} >$ 175 ns
COMINIT/COMRESET	$T_{idle} >$ 525 ns
COMSAS	$T_{idle} >$ 1,575 ns

A receiver detects an OOB signal after receiving four consecutive idle time/burst time pairs (see figure 16). It is not an error to receive more than four idle time/burst time pairs. A receiver must not detect the same OOB signal again until it has detected the corresponding negation time (i.e., a COMINIT negation time for a COMINIT) or has detected a different OOB signal (e.g., if a COMINIT was previously detected, then four sets of COMWAKE idle times followed by burst times are detected, a COMWAKE is detected; another COMINIT may follow).

A SAS receiver detects OOB signals comprised of ALIGNs transmitted at any rate up to its highest supported physical link rate. This includes physical link rates below its lowest supported physical link rate (e.g., a SAS receiver supporting only 3.0 Gbps needs to detect 1.5 Gbps based ALIGNs to interoperate with a SAS transmitter supporting both 1.5 Gbps and 3.0 Gbps).

Figure 16 describes SAS OOB signal detection by the receiver.

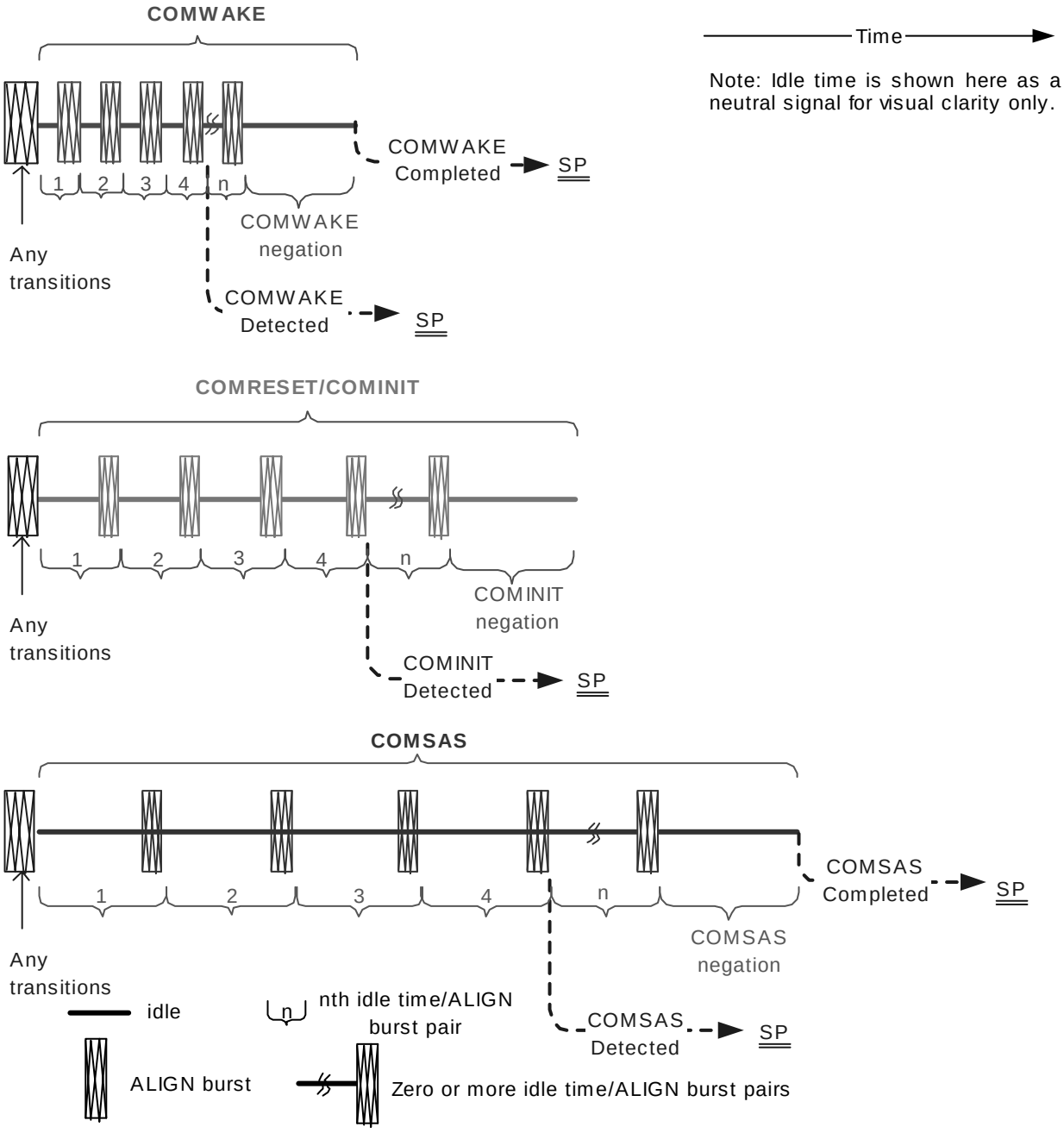


Figure 16. OOB signal detection

Expander devices shall not forward OOB signals. An expander device shall run the link reset sequence independently on each physical link.

3.5 Phy reset sequences

3.5.1 Phy reset sequences overview

The phy reset sequence consists of an OOB sequence and a speed negotiation sequence.

The phy reset sequence only affects the phy, not the port or device containing the phy or other phys in the same port or device.

The following are reasons to shall originate a phy reset sequence:

a. power on;
b. hard reset (i.e., receiving a HARD_RESET);
c. management application layer request;
d. losing dword synchronization; and
e. for expander phys, after a hot-plug timeout.

An expander SAS phy retries an unsuccessful phy reset sequence after a hot-plug timeout (see 3.5.3).

Phys shall not originate a phy reset sequence until 10 ms have elapsed since the previous attempt at running a phy reset sequence (e.g., if a reply to COMINIT is not detected in an OOB sequence, or after a speed negotiation sequence fails).

Table 12 defines the value for the hot-plug timeout

Table 12. hot-plug timing specification

Parameter	Time	Comments
Hot-plug timeout	500 ms	The maximum time after which an expander phy shall retry an unsuccessful phy reset sequence (see 3.5.3).

Note. The drive does not implement Hot Plug Timeout. This is an initiator/expander function.

The drive originates a phy reset sequence after power on and hard reset (i.e., receiving a HARD_RESET).The drive originates a phy reset sequence within 250 ms after receiving a HARD_RESET.

The drive also originates a phy reset if it looses sync or if it does not receive IDENTIFY within 1 ms after completion of the phy reset sequence.

3.5.2 SAS to SAS phy reset sequence

3.5.2.1 SAS OOB sequence

To initiate a SAS OOB sequence a phy transmits a COMINIT.

On receipt of a COMINIT a phy either:

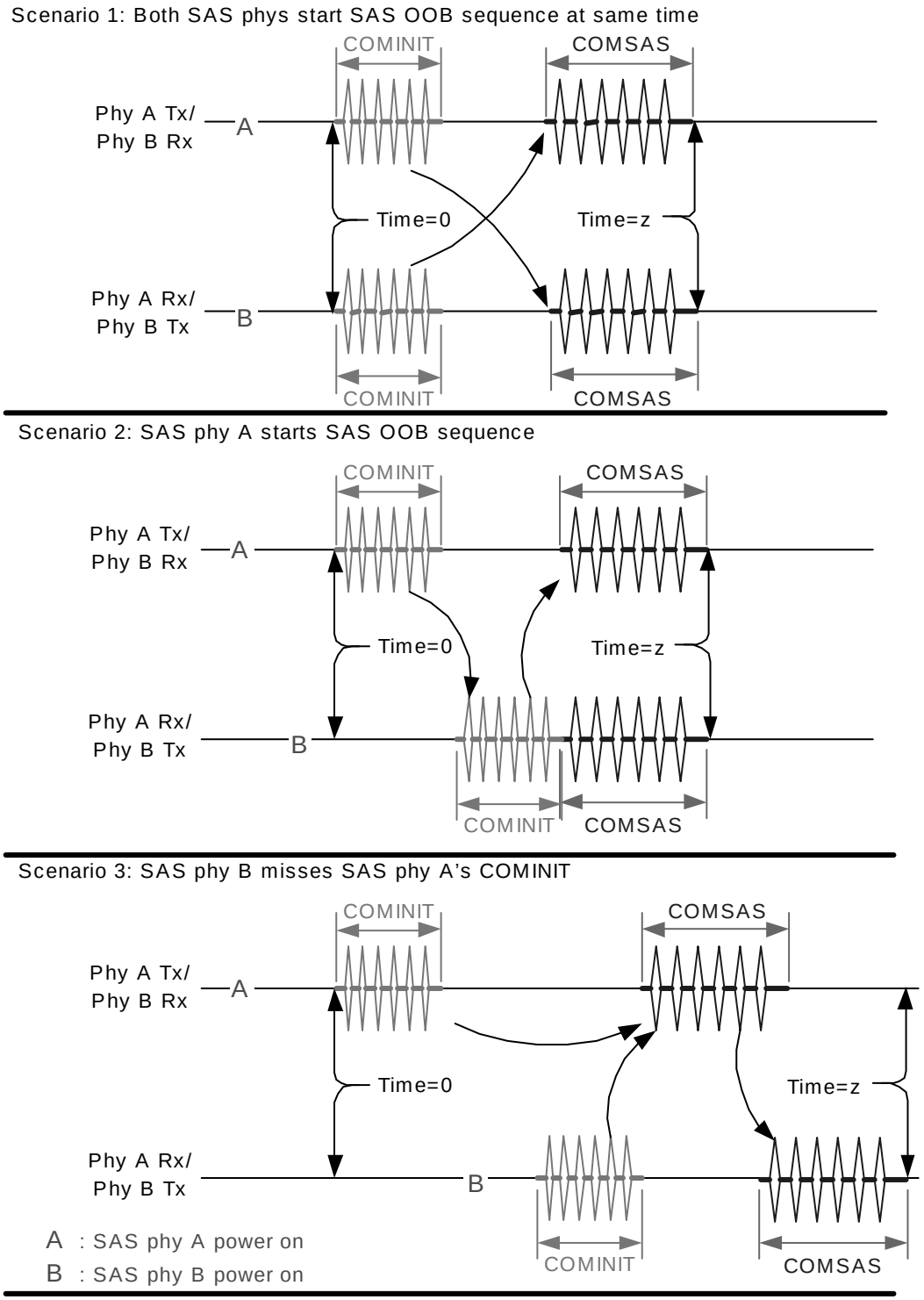
a. if the receiving phy has not yet transmitted a COMINIT, transmit a COMINIT followed by a COMSAS; or
b. if the receiving phy has transmitted a COMINIT, transmit a COMSAS.

On receipt of a COMSAS, if the receiving phy has not yet transmitted a COMSAS, the phy transmits a COMSAS.

After completing the transmission of a COMSAS and the successful receipt a COMSAS the SAS OOB sequence is complete and the SAS speed negotiation sequence begins.

A phy detecting COMINIT and COMSAS continues with the SAS speed negotiation sequence after completing the SAS OOB sequence.

Figure 17 shows several different SAS OOB sequences between phy A and phy B, with phy A starting the SAS OOB sequence at the same time as phy B, before phy B, and before phy B powers on.



Time 0: SAS phy reset sequence begins

Time z: SAS speed negotiation sequence begins

Figure 17. SAS to SAS OOB sequence

3.5.2.2 SAS speed negotiation sequence

The SAS speed negotiation sequence is a peer-to-peer negotiation technique that does not assume initiator and target (i.e., host and device) roles. The sequence consists of a set of speed negotiation windows for each physical link rate, starting with 1.5 Gbps, then 3.0 Gbps, then the next rate. The length of the speed negotiation sequence is determined by the number of physical link rates supported by the phys.

Figure 18 defines the speed negotiation window, including:

a. speed negotiation window time;
b. rate change delay time (RCDT);
c. speed negotiation transmit time (SNTT); and
d. speed negotiation lock time (SNLT).

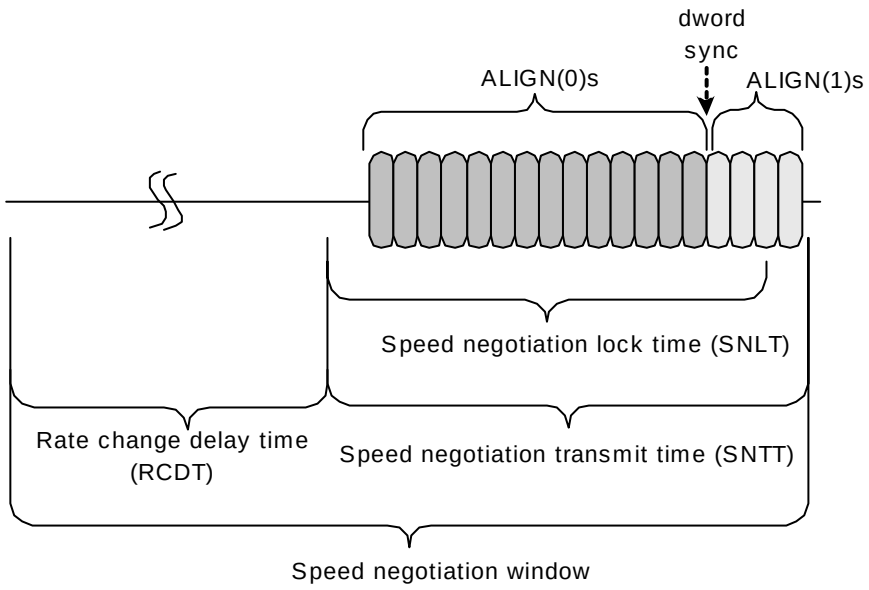


Figure 18. SAS speed negotiation window

Table 13 defines the timing specifications for the SAS speed negotiation sequence.

Table 13. SAS speed negotiation sequence timing specifications

Parameter	Time µS	Comments
Rate change delay time (RCDT)	499.95 min 500.00 nom 500.05 max	The time the transmitter shall transmit D.C. idle between rates during speed negotiation.
Speed negotiation transmit time (SNTT)	109.216 min 109.23 nom 109.238 max	The time during which ALIGN (0) or ALIGN (1) is transmitted at each physical link rate during the speed negotiation sequence. Derived from: OOBI x 4 096 x 40.
Speed negotiation lock time (SNLT)	102.39 min 102.40 nom 102.41max	The maximum time during the speed negotiation window for a transmitter to reply with ALIGN (1). Derived from: OOBI x 3 840 x 40
Speed negotiation window time	609.166 min 609.227 nom 609.287max	The duration of a speed negotiation window. Derived from: RCDT + SNTT.

The speed negotiation window shall consist of the following transmission sequence for each speed negotiation window:

1. a transmission of D.C. idle for an RCDT; and
2. if the phy supports the physical link rate, a transmission of ALIGNs at that physical link rate for the remainder of the entire speed negotiation window. If the phy does not support the physical link rate, transmission of D.C. idle for the remainder of the entire speed negotiation window.

If the phy supports the physical link rate, it shall attempt to synchronize on an incoming series of dwords at that rate for the SNLT. The received dwords may be ALIGN (0) or ALIGN (1) primitives. If the phy achieves dword synchronization within the SNLT, it shall change from transmitting ALIGN (0) primitives to transmitting ALIGN (1) primitives for the remainder of the SNTT (i.e., the remainder of the speed negotiation window). If the phy does not achieve dword synchronization within the SNLT, it shall continue transmitting ALIGN(0)s for the remainder of the SNTT (i.e., the remainder of the speed negotiation window).

At the end of the SNTT, if a phy is both transmitting and receiving ALIGN (1) primitives, it shall consider that physical link rate valid. The phy shall then proceed to the next speed negotiation window. A phy shall participate in all speed negotiation windows:

a. up to its highest supported physical link rate plus one; or
b. until it runs a speed negotiation window that does not detect a valid physical link rate after having detected a valid physical link rate in a previous speed negotiation window.

If the phy has detected a valid physical link rate in the previous speed negotiation window, it shall enter the final speed negotiation window using the highest previously successful link rate.

Figure 19 shows speed negotiation between a phy A that supports G1 through G3 link rates and a phy B that only supports the G2 link rate. Both phys run:

1. the G1 speed negotiation window, supported by phy A but not by phy B;
2. the G2 speed negotiation window, supported by both phys; and
3. the G3 speed negotiation window, supported by phy A but not by phy B.

Both phys then select G2 for the final speed negotiation window to establish the negotiated physical link rate.

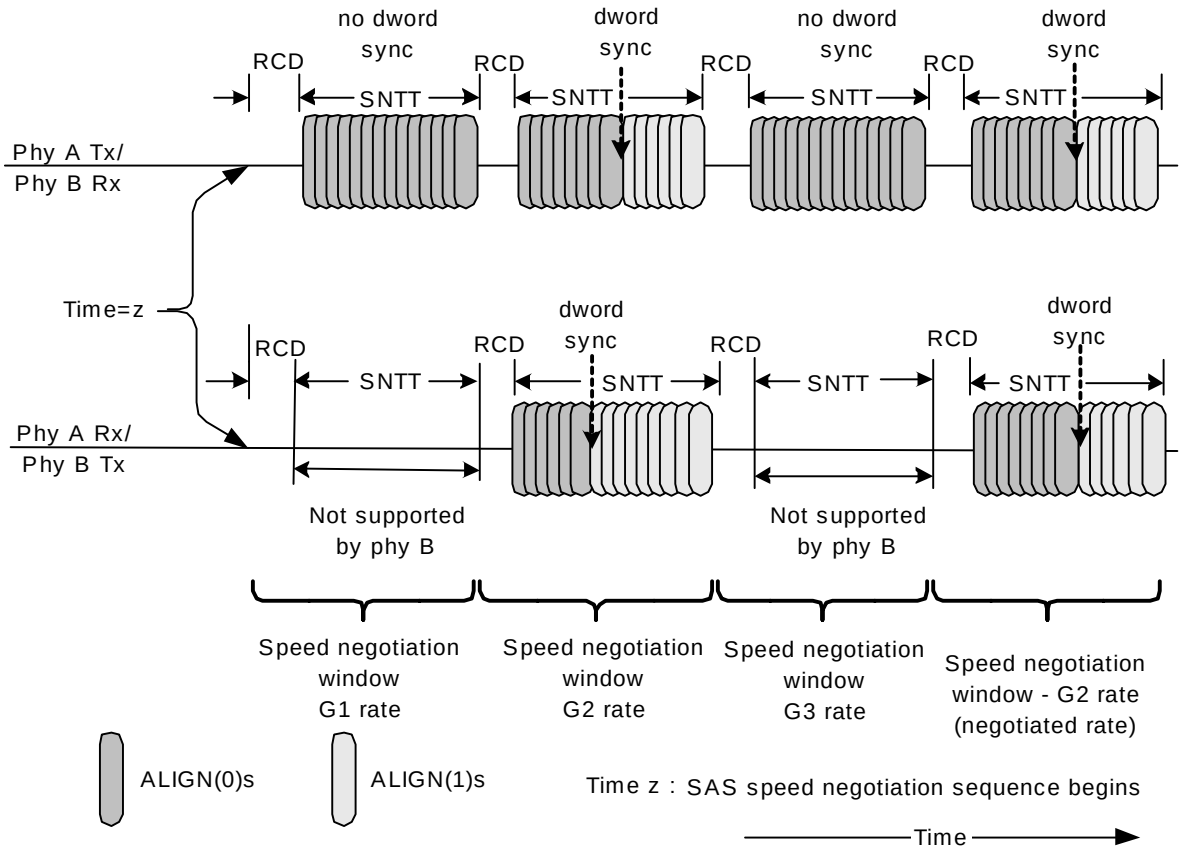


Figure 19. SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G2 only)

If the phy does not obtain dword synchronization during the final speed negotiation window the SAS speed negotiation sequence fails. This may be counted and reported in the PHY RESET PROBLEM field in the SMP REPORT PHY ERROR LOG page (see SAS) and the REPORT PHY ERROR LOG log page.

Figure 20 shows the same speed negotiation sequence as in figure 19 when phy B does not obtain dword synchronization during the final speed negotiation window. If this occurs, the handshake is not complete and the OOB sequence shall be retried starting with COMINIT, forcing the phy to retry the whole reset sequence.

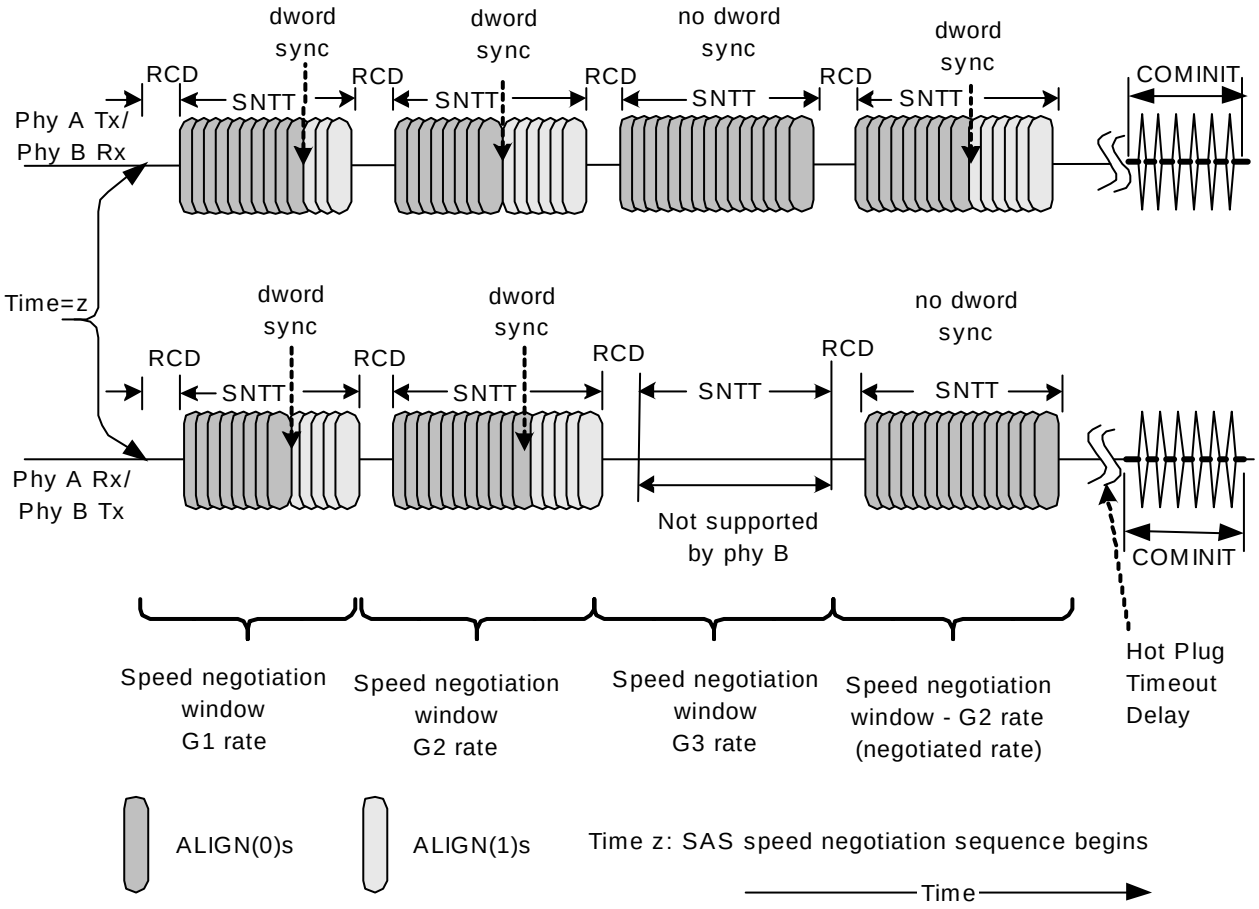


Figure 20. SAS speed negotiation sequence (phy A: G1, G2, G3, phy B: G1, G2) that fails

For more examples of speed negotiations between phys that support various speeds.

3.5.3 Phy reset sequence after devices are attached

Since SATA and SAS signal cable connectors do not include power lines, it is not possible to detect the physical insertion of the signal cable connector onto a plug. Non-cabled environments may similarly not have a way to detect physical insertion of a device. As a result, every time a phy reset sequence is originated:

a) expander phys that are enabled but not active originate a new phy reset sequence repeatedly, with no more than a hot-plug timeout (see table 12) between each attempt, until a speed negotiation sequence completes successfully;
b) SAS initiator phys should originate a new phy reset sequence after every hot-plug timeout; and
c) drive does not originate a new phy reset sequence after their first attempt.

Figure 21 shows how two phys complete the phy reset sequence if the phys are not attached at power on. In this example, phy A and phy B are attached some time before phy B's second hot-plug timeout occurs. Phy B's OOB detection circuitry detects a COMINIT after the attachment, and therefore phy B transmits COMSAS, since it has both transmitted and received a COMINIT. Upon receiving COMSAS, phy A transmits its own COMSAS. The SAS speed negotiation sequence follows.

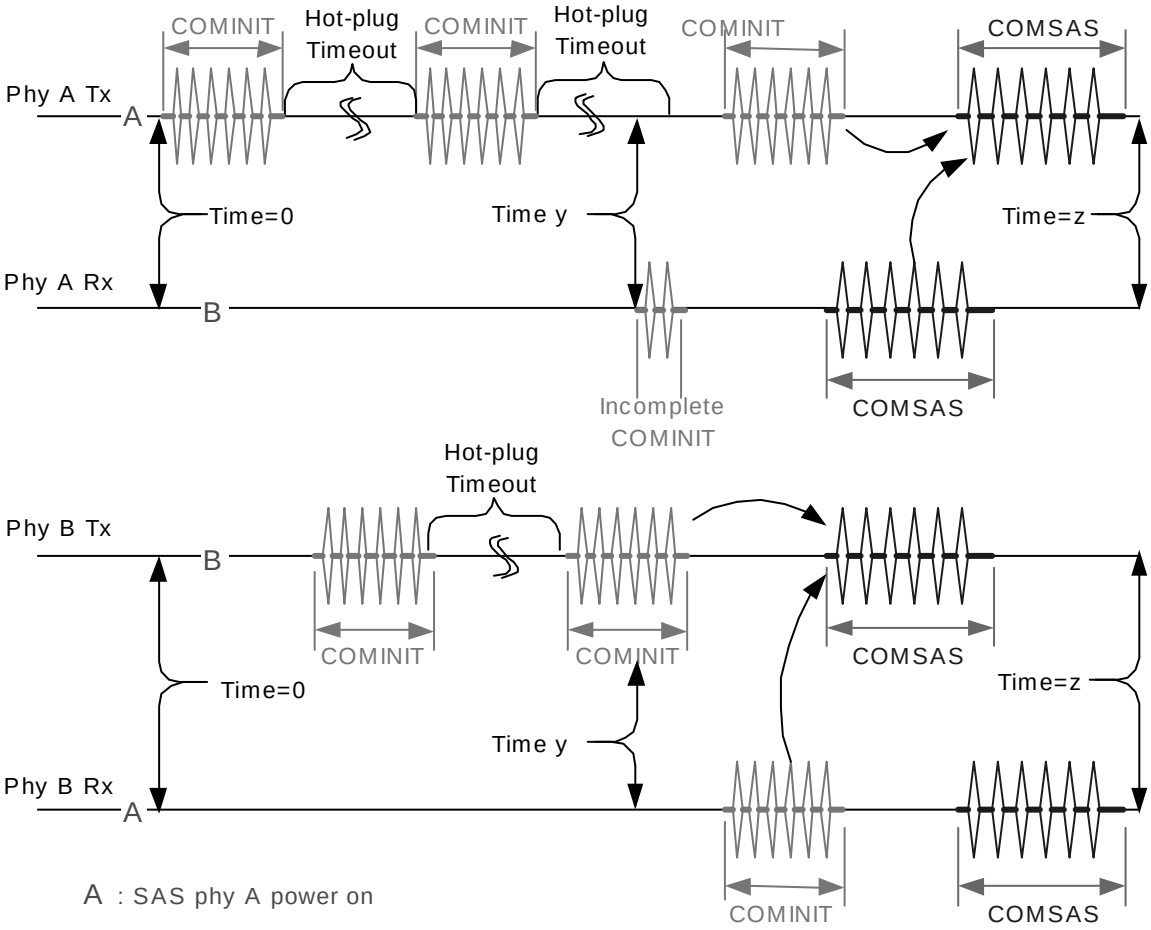


Figure 21. Hot-plug and the phy reset sequence

3.6 DWS (dword synchronization)

The phy establishes the same dword boundaries at the receiver as at the attached transmitter by searching for control characters. A receiver in the phy monitors and decodes the incoming data stream and forces K28.5 characters into the first character position to effectively perform dword alignment. The receiver continues to reestablish dword alignment by forcing received K28.5 characters into the first character position until a valid primitive is detected. The resultant primitives, dwords and valid dword indicators (e.g., encoding error indicators) are processed to determine if the phy is in dword synchronization.

While dword synchronization is lost, the data stream received is invalid and dwords are not be passed to the link layer.

3.6.1 Acquiring DWS

To acquire DWS, three valid primitives starting with K28.5 must be received without any intervening invalid dwords. The drive continuously tries to acquire DWS. Only dwords received while in dword sync are processed.

3.6.2 Losing DWS

After DWS is acquired, when an invalid dword is detected, it requires two valid dwords to nullify its effect. When four invalid dwords are detected without nullification, dword synchronization is considered lost.

3.7 Spin-up

The drive implements SAS power conditions. This states that the Power Condition mode page and START STOP UNIT command interact with the NOTIFY(ENABLE SPINUP) primitive to control temporary consumption of additional power as described below: (Reference SAS Spec section 10.2.8)

The drive uses NOTIFY(ENABLE SPINUP) to:

a. automatically spin–up after power on; and

b. delay spin-up requested by START STOP UNIT commands.

The drive enters the Active Wait state, as defined in the SAS specification, after power-on. In this state the drive will spin up upon receiving NOTIFY (ENABLE SPINUP). If the drive receives a START/STOP UNIT command with the Start bit set to zero at any time, it will spin down and enter the STOPPED state. In this state, NOTIFY (ENABLE SPINUP) is ignored. A START/STOP UNIT with the Start bit set to 1 will return the drive to the Active Wait state, where it will wait for a new NOTIFY (ENABLE SPINUP). If the START/STOP UNIT command has the IMMED bit set to zero (meaning that the spinup must complete before returning status), the drive will wait up to five seconds for NOTIFY (ENABLE SPINUP). If not received within this time, the drive will terminate the command with a sense key of NOT READY and an ASC/ASCQ of NOTIFY (ENABLE SPINUP) REQUIRED (02/0411). The drive will remain in the Active Wait state so a subsequent NOTIFY (ENABLE SPINUP) will cause the drive to spin up.

Most host bus adapters and expanders will send NOTIFY to drives at intervals of a few seconds so timeouts should not occur. However, if the attached device must be given special instructions to send NOTIFY, a delay of at least 2 ms is recommended between the START/STOP unit command and the NOTIFY. This is because the drive must decode and begin processing the START UNIT command before it enters the Active Wait state and begins waiting for NOTIFY.

4.0 Link layer

4.1 Link layer overview

The link layer defines primitives, address frames, and connections. Link layer state machines interface to the port layer and the phy layer and perform the identification and hard reset sequences, connection management, and SSP, STP, and SMP specific frame transmission and reception.

4.2 Primitives

4.2.1 Primitives overview

Primitives are dwords whose first character is a K28.3, K28.5, or K28.6 control character. Primitives are not considered big-endian or little-endian; they are just interpreted as first, second, third, and last characters. Table 14 defines the primitive format. The drive ignores all primitives that do not start with K28.5.

Table 14. Primitive format

Character	Description
First	K28.5 control character (for primitives defined in this manual), or K28.3 control character (for primitives defined by SATA)
Second	Constant data character.
Third	Constant data character.
Last	Constant data character.

4.2.2 Primitive summary

Table 15 defines the primitives not specific to the type of connection.

Table 15. Primitives not specific to type of connection (Sheet 1 of 3)

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [d]
		I	E	T	I	E	T	
AIP (NORMAL)	NoConn		E		I	E	T	Single
AIP (RESERVED 0)	NoConn				I	E	T	Single
AIP (RESERVED 1)	NoConn				I	E	T	Single
AIP (RESERVED 2)	NoConn				I	E	T	Single
AIP (RESERVED WAITING ON PARTIAL)	NoConn				I	E	T	Single

Table 15. Primitives not specific to type of connection (Sheet 2 of 3)

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [d]
		I	E	T	I	E	T	
AIP (WAITING ON CONNECTION)	NoConn		E		I	E	T	Single
AIP (WAITING ON DEVICE)	NoConn		E		I	E	T	Single
AIP (WAITING ON PARTIAL)	NoConn		E		I	E	T	Single
ALIGN (0)	All	I	E	T	I	E	T	Single
ALIGN (1)	All	I	E	T	I	E	T	Single
ALIGN (2)	All	I	E	T	I	E	T	Single
ALIGN (3)	All	I	E	T	I	E	T	Single
BREAK	All	I	E	T	I	E	T	Redundant
BROADCAST (CHANGE)	NoConn	I	E		I	E	T	Redundant
BROADCAST (SES)	NoConn			T	I	E	T	Redundant
BROADCAST (EXPANDER)	NoConn		E		I	E	T	Redundant
BROADCAST (RESERVED 2)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 3)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED 4)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED CHANGE 0)	NoConn				I	E	T	Redundant
BROADCAST (RESERVED CHANGE 1)	NoConn				I	E	T	Redundant
CLOSE (CLEAR AFFILIATION)	STP	I					T	Triple
CLOSE (NORMAL)	Conn	I		T	I		T	Triple
CLOSE (RESERVED 0)	Conn				I		T	Triple
CLOSE (RESERVED 1)	Conn				I		T	Triple
EOAF	NoConn	I	E	T	I	E	T	Single
ERROR	All		E		I	E	T	Single
HARD_RESET	NoConn	I	E		I	E	T	Redundant
NOTIFY (ENABLE SPINUP)	All	I	E				T	Single
NOTIFY (RESERVED 0)	All				I	E	T	Single
NOTIFY (RESERVED 1)	All				I	E	T	Single
NOTIFY (RESERVED 2)	All				I	E	T	Single
OPEN_ACCEPT	NoConn	I		T	I		T	Single
OPEN_REJECT (BAD DESTINATION)	NoConn		E		I		T	Single
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	NoConn	I	E	T	I		T	Single
OPEN_REJECT (NO DESTINATION)	NoConn		E		I		T	Single

Table 15. Primitives not specific to type of connection (Sheet 3 of 3)

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [d]
		I	E	T	I	E	T	
OPEN_REJECT (PATHWAY BLOCKED)	NoConn		E		I		T	Single
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	NoConn	I		T	I		T	Single
OPEN_REJECT (RESERVED ABANDON 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 2)	NoConn				I		T	Single
OPEN_REJECT (RESERVED ABANDON 3)	NoConn				I		T	Single
OPEN_REJECT (RESERVED CONTINUE 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED CONTINUE 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED INITIALIZE 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED INITIALIZE 1)	NoConn				I		T	Single
OPEN_REJECT (RESERVED STOP 0)	NoConn				I		T	Single
OPEN_REJECT (RESERVED STOP 1)	NoConn				I		T	Single
OPEN_REJECT (RETRY)	NoConn	I		T	I		T	Single
OPEN_REJECT (STP RESOURCES BUSY)	NoConn		E	T	I			Single
OPEN_REJECT (WRONG DESTINATION)	NoConn	I		T	I		T	Single
SOAF	NoConn	I	E	T	I	E	T	Single

[a] The Use column indicates when the primitive is used:
  a) NoConn: SAS physical links, outside connections;
  b) Conn: SAS physical links, inside connections;
  c) All: SAS physical links, both outside connections or inside any type of connection; or
  d) STP: SAS physical links, inside STP connections.

[b] The From and To columns indicate the type of ports that originate each primitive or are the intended destinations of each primitive:
  a) I for SAS initiator ports;
  b) E for expander ports; and
  c) T for SAS target ports.

[c] Expander ports are not considered originators of primitives that are passing through from expander port to expander port.

[d] The Primitive sequence type columns indicate whether the primitive is sent as a single primitive sequence, a repeated primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 4.2.4).

Table 16 defines the primitives used only inside SSP and SMP connections.

Table 16. Primitives used only inside SSP and SMP connections

Primitive	Use [a]	From [b]			To [b]			Primitive sequence type [d]
		I	E	T	I	E	T	
ACK	SSP	I		T	I		T	Single
CREDIT_BLOCKED	SSP	I		T	I		T	Single
DONE (ACK/NAK TIMEOUT)	SSP	I		T	I		T	Single
DONE (CREDIT TIMEOUT)	SSP	I		T	I		T	Single
DONE (NORMAL)	SSP	I		T	I		T	Single
DONE (RESERVED 0)	SSP				I		T	Single
DONE (RESERVED 1)	SSP				I		T	Single
DONE (RESERVED TIMEOUT 0)	SSP				I		T	Single
DONE (RESERVED TIMEOUT 1)	SSP				I		T	Single
EOF	SSP, SMP	I		T	I		T	Single
NAK (CRC ERROR)	SSP	I		T	I		T	Single
NAK (RESERVED 0)	SSP				I		T	Single
NAK (RESERVED 1)	SSP				I		T	Single
NAK (RESERVED 2)	SSP				I		T	Single
RRDY (NORMAL)	SSP	I		T	I		T	Single
RRDY (RESERVED 0)	SSP				I		T	Single
RRDY (RESERVED 1)	SSP				I		T	Single
SOF	SSP, SMP	I		T	I		T	Single

[a] The Use column indicates when the primitive is used:
a) SSP: SAS physical links, inside SSP connections; or
b) SMP: SAS physical links, inside SMP connections.

[b] The From and To columns indicate the type of ports that originate each primitive or are the intended destinations of each primitive:
a) I for SSP initiator ports and SMP initiator ports;
b) E for expander ports; and
c) T for SSP target ports and SMP target ports.

[c] Expander ports are not considered originators of primitives that are passing through from expander port to expander port.

[d] The Primitive sequence type columns indicate whether the primitive is sent as a single primitive sequence, a repeated primitive sequence, a triple primitive sequence, or a redundant primitive sequence (see 4.2.4).

4.2.3 Primitive encodings

Table 17 defines the primitive encoding for primitives not specific to type of connection.

Table 17. Primitive encoding for primitives not specific to type of connection (Sheet 1 of 2)

Primitive	Character			
	1st	2nd	3rd	4th (last)
AIP (NORMAL)	K28.5	D27.4	D27.4	D27.4
AIP (RESERVED 0)	K28.5	D27.4	D31.4	D16.7
AIP (RESERVED 1)	K28.5	D27.4	D16.7	D30.0
AIP (RESERVED 2)	K28.5	D27.4	D29.7	D01.4
AIP (RESERVED WAITING ON PARTIAL)	K28.5	D27.4	D01.4	D07.3
AIP (WAITING ON CONNECTION)	K28.5	D27.4	D07.3	D24.0
AIP (WAITING ON DEVICE)	K28.5	D27.4	D30.0	D29.7
AIP (WAITING ON PARTIAL)	K28.5	D27.4	D24.0	D04.7
ALIGN (0)	K28.5	D10.2	D10.2	D27.3
ALIGN (1)	K28.5	D07.0	D07.0	D07.0
ALIGN (2)	K28.5	D01.3	D01.3	D01.3
ALIGN (3)	K28.5	D27.3	D27.3	D27.3
BREAK	K28.5	D02.0	D24.0	D07.3
BROADCAST (CHANGE)	K28.5	D04.7	D02.0	D01.4
BROADCAST (SES)	K28.5	D04.7	D07.3	D29.7
BROADCAST (EXPANDER)	K28.5	D04.7	D01.4	D24.0
BROADCAST (RESERVED 2)	K28.5	D04.7	D04.7	D04.7
BROADCAST (RESERVED 3)	K28.5	D04.7	D16.7	D02.0
BROADCAST (RESERVED 4)	K28.5	D04.7	D29.7	D30.0
BROADCAST (RESERVED CHANGE 0)	K28.5	D04.7	D24.0	D31.4
BROADCAST (RESERVED CHANGE 1)	K28.5	D04.7	D27.4	D07.3
CLOSE (CLEAR AFFILIATION)	K28.5	D02.0	D07.3	D04.7
CLOSE (NORMAL)	K28.5	D02.0	D30.0	D27.4
CLOSE (RESERVED 0)	K28.5	D02.0	D31.4	D30.0
CLOSE (RESERVED 1)	K28.5	D02.0	D04.7	D01.4
EOAF	K28.5	D24.0	D07.3	D31.4
ERROR	K28.5	D02.0	D01.4	D29.7
HARD_RESET	K28.5	D02.0	D02.0	D02.0
NOTIFY (ENABLE SPINUP)	K28.5	D31.3	D31.3	D31.3

Table 17. Primitive encoding for primitives not specific to type of connection (Sheet 2 of 2)

Primitive	Character			
	1st	2nd	3rd	4th (last)
NOTIFY (RESERVED 0)	K28.5	D31.3	D07.0	D01.3
NOTIFY (RESERVED 1)	K28.5	D31.3	D01.3	D07.0
NOTIFY (RESERVED 2)	K28.5	D31.3	D10.2	D10.2
OPEN_ACCEPT	K28.5	D16.7	D16.7	D16.7
OPEN_REJECT (BAD DESTINATION)	K28.5	D31.4	D31.4	D31.4
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	K28.5	D31.4	D04.7	D29.7
OPEN_REJECT (NO DESTINATION)	K28.5	D29.7	D29.7	D29.7
OPEN_REJECT (PATHWAY BLOCKED)	K28.5	D29.7	D16.7	D04.7
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	K28.5	D31.4	D29.7	D07.3
OPEN_REJECT (RESERVED ABANDON 0)	K28.5	D31.4	D02.0	D27.4
OPEN_REJECT (RESERVED ABANDON 1)	K28.5	D31.4	D30.0	D16.7
OPEN_REJECT (RESERVED ABANDON 2)	K28.5	D31.4	D07.3	D02.0
OPEN_REJECT (RESERVED ABANDON 3)	K28.5	D31.4	D01.4	D30.0
OPEN_REJECT (RESERVED CONTINUE 0)	K28.5	D29.7	D02.0	D30.0
OPEN_REJECT (RESERVED CONTINUE 1)	K28.5	D29.7	D24.0	D01.4
OPEN_REJECT (RESERVED INITIALIZE 0)	K28.5	D29.7	D30.0	D31.4
OPEN_REJECT (RESERVED INITIALIZE 1)	K28.5	D29.7	D07.3	D16.7
OPEN_REJECT (RESERVED STOP 0)	K28.5	D29.7	D31.4	D07.3
OPEN_REJECT (RESERVED STOP 1)	K28.5	D29.7	D04.7	D27.4
OPEN_REJECT (RETRY)	K28.5	D29.7	D27.4	D24.0
OPEN_REJECT (STP RESOURCES BUSY)	K28.5	D31.4	D27.4	D01.4
OPEN_REJECT (WRONG DESTINATION)	K28.5	D31.4	D16.7	D24.0
SOAF	K28.5	D24.0	D30.0	D01.4

Table 18 defines the primitive encodings for primitives used only inside SSP connections.

Table 18. Primitive encoding for primitives used only inside SSP connections

Primitive	Character			
	1st	2nd	3rd	4th (last)
ACK	K28.5	D01.4	D01.4	D01.4
CREDIT_BLOCKED	K28.5	D01.4	D07.3	D30.0
DONE (ACK/NAK TIMEOUT)	K28.5	D30.0	D01.4	D04.7
DONE (CREDIT TIMEOUT)	K28.5	D30.0	D07.3	D27.4

Table 18. Primitive encoding for primitives used only inside SSP connections

DONE (NORMAL)	K28.5	D30.0	D30.0	D30.0
DONE (RESERVED 0)	K28.5	D30.0	D16.7	D01.4
DONE (RESERVED 1)	K28.5	D30.0	D29.7	D31.4
DONE (RESERVED TIMEOUT 0)	K28.5	D30.0	D27.4	D29.7
DONE (RESERVED TIMEOUT 1)	K28.5	D30.0	D31.4	D24.0
EOF	K28.5	D24.0	D16.7	D27.4
NAK (CRC ERROR)	K28.5	D01.4	D27.4	D04.7
NAK (RESERVED 0)	K28.5	D01.4	D31.4	D29.7
NAK (RESERVED 1)	K28.5	D01.4	D04.7	D24.0
NAK (RESERVED 2)	K28.5	D01.4	D16.7	D07.3
RRDY (NORMAL)	K28.5	D01.4	D24.0	D16.7
RRDY (RESERVED 0)	K28.5	D01.4	D02.0	D31.4
RRDY (RESERVED 1)	K28.5	D01.4	D30.0	D02.0
SOF	K28.5	D24.0	D04.7	D07.3

4.2.4 Primitive sequences

4.2.4.1 Primitive sequences overview

Table 19 summarizes the types of primitive sequences.

Table 19. Primitive sequences

Primitive sequence type	Number of times to transmit	Number of times received to detect
Single	1	1
Repeated	2	1
Triple	3	3
Redundant	6	3

Any number of ALIGNs and NOTIFYs may be sent inside primitive sequences without affecting the count or breaking the consecutiveness requirements. Rate matching ALIGNs and NOTIFYs shall be sent inside primitive sequences inside of connections if rate matching is enabled (see 4.9.2).

4.2.4.2 Single primitive sequence

Primitives labeled as single primitive sequences are sent one time.

4.2.4.3 Triple primitive sequence

Primitives that form triple primitive sequences (e.g., CLOSE (NORMAL)) are sent three times consecutively. ALIGNs and NOTIFYs may be sent inside primitive sequences as described in 4.2.4.1.

Receivers detect a triple primitive sequence after the identical primitive is received in three consecutive dwords. After receiving a triple primitive sequence, a receiver is not detect a second instance of the same triple primitive sequence until it has received three consecutive dwords that are not any of the following:

a. the original primitive; or

b. an ALIGN or NOTIFY.

Figure 22 shows examples of triple primitive sequences.

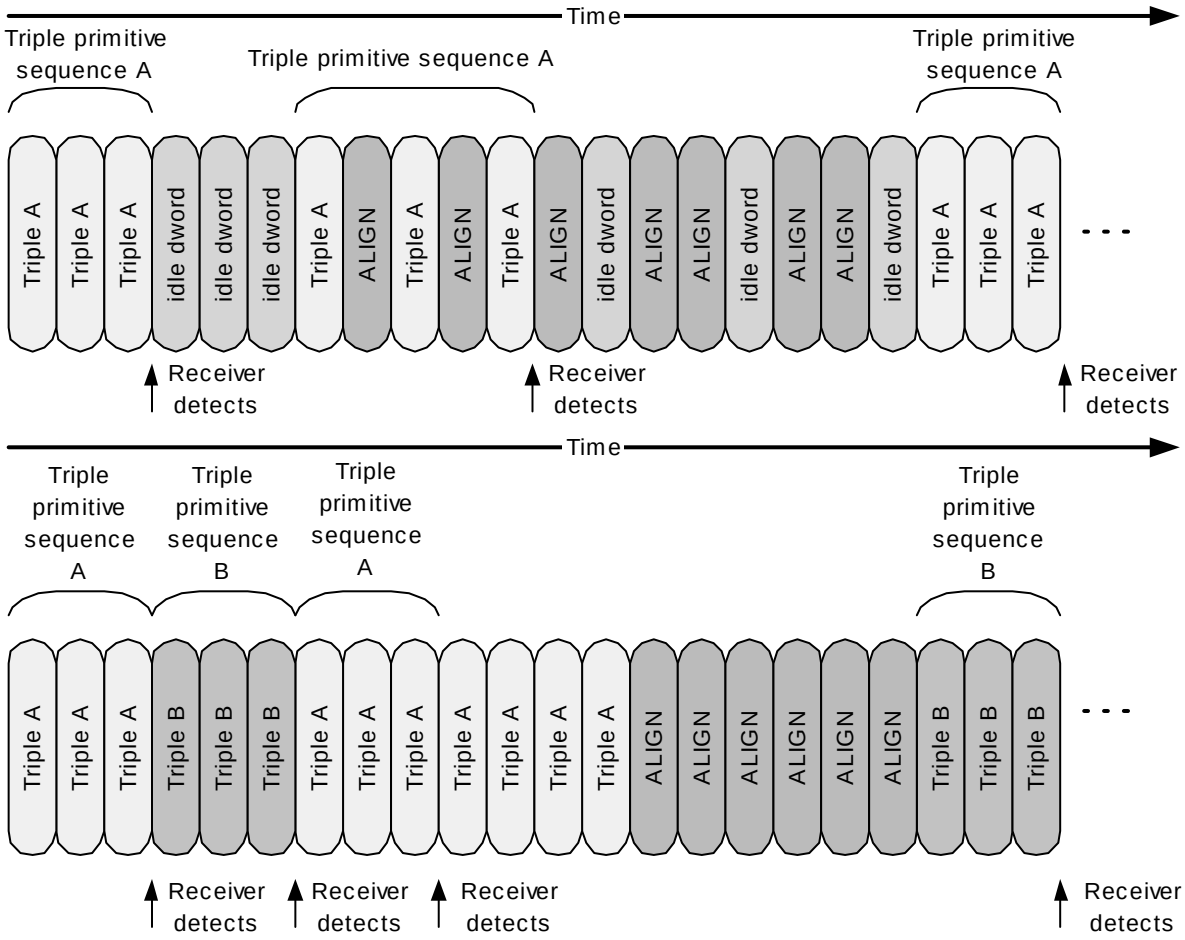


Figure 22. Triple primitive sequence

4.2.4.4 Redundant primitive sequence

Primitives that form redundant primitive sequences (e.g., BROADCAST (CHANGE)) are sent six times consecutively. ALIGNs and NOTIFYs may be sent inside primitive sequences as described in 4.2.4.1.

A receiver detects a redundant primitive sequence after the identical primitive is received in any three of six consecutive dwords. After receiving a redundant primitive sequence, a receiver does not detect a second instance of the same redundant primitive sequence until it has received six consecutive dwords that are not any of the following:

a. the original primitive; or

b. an ALIGN or NOTIFY.

Figure 23 shows examples of redundant primitive sequences.

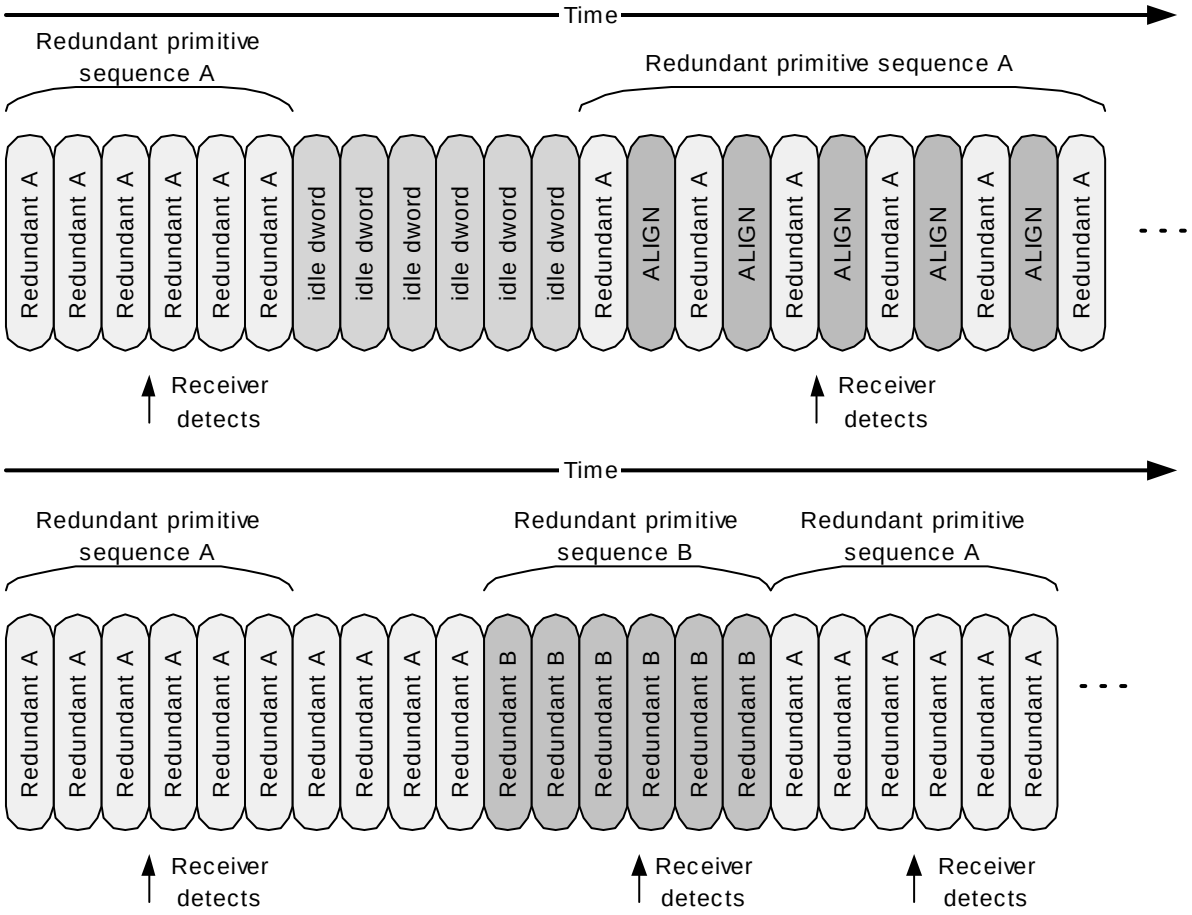


Figure 23. Redundant primitive sequence

4.2.5 Primitives not specific to type of connections

4.2.5.1 AIP (Arbitration in progress)

AIP is sent by an expander device after a connection request to indicate that the connection request is being processed and indicate the status of the connection request.

The versions of AIP representing different statuses are defined in table 20.

Table 20. AIP primitives

Primitive	Description
AIP (NORMAL)	Expander device has just accepted the connection request.
AIP (RESERVED 0)	Reserved. Processed the same as AIP (NORMAL).
AIP (RESERVED 1)	Reserved. Processed the same as AIP (NORMAL).
AIP (RESERVED 2)	Reserved. Processed the same as AIP (NORMAL).
AIP (WAITING ON CONNECTION)	Expander device has determined the routing for the connection request, but the destination phys are all being used for connections.
AIP (WAITING ON DEVICE)	Expander device has determined the routing for the connection request and forwarded it to the output physical link.
AIP (WAITING ON PARTIAL)	Expander device has determined the routing for the connection request, but the destination phys are all busy with other partial pathways (i.e., connection requests that have not reached the destination phy).
AIP (RESERVED WAITING ON PARTIAL)	Reserved. Processed the same as AIP (WAITING ON PARTIAL).

See 4.9.1 for details on connections.

4.2.5.2 ALIGN

ALIGNs are used for:

a. OOB signals;
b. character and dword alignment during the speed negotiation sequence;
c. clock skew management after the phy reset sequence; and
d. rate matching during connections.

Table 21 defines the different versions of ALIGN primitives.

Table 21. ALIGN primitives

Primitive	Description
ALIGN (0)	Used for OOB signals, the speed negotiation sequence, clock skew management and rate matching.
ALIGN (1)	Used for the speed negotiation sequence, clock skew management and rate matching.
ALIGN (2)	Used for clock skew management and rate matching.
ALIGN (3)	Used for clock skew management and rate matching.

Phys shall use ALIGN(0) to construct OOB signals as described in 3.4. Phys shall use ALIGN(0) and ALIGN(1) during the speed negotiation sequence as described in 3.5.2.2. Phys shall rotate through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3) for all ALIGNs sent after the phy reset sequence for clock skew management (see 4.3) and rate matching (see 4.9.2).

Phys receiving ALIGNs after the phy reset sequence shall not verify the rotation and shall accept any of the ALIGNs at any time.

Phys shall only detect an ALIGN after decoding all four characters in the primitive.

For clock skew management and rate matching, ALIGNs may be replaced by NOTIFYs (see 4.2.5.9).

4.2.5.3 BREAK

BREAK is used to abort a connection request or break a connection.

See 4.9.1.6 and 4.9.1.8 for details on breaking connections.

4.2.5.4 BROADCAST

BROADCASTs are used to notify all SAS ports in a domain of an event.

The versions of BROADCAST representing different reasons are defined in table 22.

Table 22. BROADCAST primitives

Primitive	Description
BROADCAST (CHANGE)	Notification of a configuration change.
BROADCAST (RESERVED CHANGE 0)	Reserved. Processed the same as BROADCAST (CHANGE) by SAS ports (i.e, SAS initiator ports and SAS target ports).
BROADCAST (RESERVED CHANGE 1)	Reserved. Processed the same as BROADCAST (CHANGE) by SAS ports (i.e., SAS initiator ports and SAS target ports).
BROADCAST (SES))	Notification of an asynchronous event from a logical unit with a peripheral device type set to 0Dh (i.e., enclosure services device) (see SPC-3 and SES-2) in the SAS domain.
BROADCAST (EXPANDER)	Notification of an expander event, including: a) a phy event information peak value detector reaching its threshold value; and b) a phy event information peak value detector being cleared. These expander events do not include SAS domain changes.
BROADCAST (RESERVED 2)	Reserved.
BROADCAST (RESERVED 3)	Reserved.
BROADCAST (RESERVED 4)	Reserved.

When an expander port receives a BROADCAST it transmits the same BROADCAST on at least one phy in all other expander ports. BROADCAST is only sent outside of connections after the phy reset sequence has completed.

An expander device is not required to queue multiple identical BROADCASTs for the same expander port. If a second identical BROADCAST is requested before the first BROADCAST has been transmitted, the second BROADCAST may be ignored.

BROADCAST (CHANGE) is sent by an expander device to notify SAS initiator ports and other expander devices that a configuration change has occurred. BROADCAST (CHANGE) may also be transmitted by SAS initiator ports. BROADCAST (CHANGE) are ignored by SAS target ports.

BROADCAST (RESERVED CHANGE 0) and BROADCAST (RESERVED CHANGE 1) are processed the same as BROADCAST (CHANGE) by SAS ports. BROADCAST (RESERVED 0), BROADCAST (RESERVED 1), BROADCAST (RESERVED 2), BROADCAST (RESERVED 3), and BROADCAST (RESERVED 4) shall be ignored by SAS ports.

BROADCAST (SES) is sent by a SAS target port to notify SAS initiator ports that an asynchronous event has occurred in an enclosure. BROADCAST (SES) is ignored by SAS target ports.

BROADCAST (EXPANDER) is sent by an expander device to notify SAS initiator ports that an expander event has occurred, including:

a) one or more of its phy event information peak value detectors has reached its threshold value; and

b) one or more of its phy event information peak value detectors has been cleared by an SMP CONFIGURE PHY EVENT INFORMATION function.

BROADCAST (RESERVED CHANGE 0) and BROADCAST (RESERVED CHANGE 1) shall be processed the same as BROADCAST (CHANGE) by SAS ports. BROADCAST (RESERVED 2), BROADCAST (RESERVED 3), and BROADCAST (RESERVED 4) shall be ignored by SAS ports.

See 4.9 for details on SAS domain changes. See SAS for details on counting BROADCAST (CHANGE) generation in an expander device.

4.2.5.5 CLOSE

CLOSE is used to close a connection. This primitive may be originated by a SAS initiator port or a SAS target port.

The versions of CLOSE representing different reasons are defined in table 23.

Table 23. CLOSE primitives

Primitive	Description
CLOSE (CLEAR AFFILIATION)	Close an open STP connection and clear the affiliation. Processed the same as CLOSE (NORMAL) if: a. the connection is not an STP connection; b. the connection is an STP connection, but affiliations are not implemented by the STP target port; or c. the connection is an STP connection, but an affiliation is not present.
CLOSE (NORMAL)	Close a connection.
CLOSE (RESERVED 0)	Reserved. Processed the same as CLOSE (NORMAL).
CLOSE (RESERVED 1)	Reserved. Processed the same as CLOSE (NORMAL).

See 4.9.1.7 for details on closing connections.

4.2.5.6 EOAF (End of address frame)

EOAF indicates the end of an address frame.

See 4.7 for details on address frames.

4.2.5.7 ERROR

An expander device substitutes the ERROR primitive for an invalid dword when it is forwarding dwords from a SAS physical link.

4.2.5.8 HARD_RESET

HARD_RESET is used to force the drive to perform a hard reset. This primitive is only valid after the phy reset sequence without an intervening identification sequence (see 2.3) and is ignored at other times.

4.2.5.9 NOTIFY

NOTIFY may be transmitted in place of any ALIGN being transmitted for clock skew management (see 4.3) or rate matching (see 4.9.2). Substitution of a NOTIFY may or may not affect the ALIGN sequencing (i.e., the NOTIFY may take the place of one of the ALIGNs in the rotation through ALIGN (0), ALIGN (1), ALIGN (2), or ALIGN (3) or it may delay the rotation). A specific NOTIFY can not be transmitted a second time until at least three ALIGNs or different NOTIFYs have been transmitted.

NOTIFY is not forwarded through expander devices. Expander devices substitute an ALIGN for a NOTIFY.

The versions of NOTIFY representing different reasons are defined in table 24.

Table 24. NOTIFY primitives

Primitive	Description
NOTIFY (ENABLE SPINUP)	Specify to an SAS target device that it may temporarily consume additional power while transitioning into the active or idle power condition state.
NOTIFY (RESERVED 0)	Reserved.
NOTIFY (RESERVED 1)	Reserved.
NOTIFY (RESERVED 2)	Reserved.

NOTIFY (ENABLE SPINUP) is transmitted by a SAS initiator port for point to point configurations or expander port and is used to signal the drive when to spinup. The drive's reaction to a NOTIFY (ENABLE SPINUP) is controlled by the Power Conditions mode page and/or the START STOP UNIT command.

NOTIFY (ENABLE SPINUP) is transmitted after power on when the enclosure is ready for initial spin-up. After the initial NOTIFY (ENABLE SPINUP), is periodically transmitted.

I_T nexus loss, logical unit reset, and hard reset do not cause the drive to spinup automatically on receipt of NOTIFY (ENABLE SPINUP).

The drive treats a NOTIFY (ENABLE SPINUP) received on either port as signal to proceed with spinup as selected by the Power Conditions mode page and/or the START STOP UNIT command

NOTIFY (RESERVED 0), NOTIFY (RESERVED 1), and NOTIFY (RESERVED 2) are ignored by the drive.

NOTIFY (POWER LOSS EXPECTED) is transmitted by a SAS initiator port or expander port and is used to specify to a SAS target device that power loss may occur within the time specified in the POWER LOSS TIMEOUT field in the Protocol-Specific Logical Unit mode page (see 6.1.3.3).

NOTIFY (POWER LOSS EXPECTED) shall be transmitted at least three times by the SAS initiator port or expander port.

If a SAS target device supports NOTIFY (POWER LOSS EXPECTED) and receives NOTIFY (POWER LOSS EXPECTED) on an SSP target port, then the device server for each logical unit to which the SSP target port has access shall, within 1 ms:

1. stop writing data to the media on a block boundary (e.g., all write activity shall continue until a block boundary is reached then all writing shall stop); and
2. clear all task sets (i.e., the device server acts as if it has received a CLEAR TASK SET task management function (see SAM-4) for each task set); and
3. establish a unit attention condition for the initiator port associated with every I_T nexus with the additional sense code set to COMMANDS CLEARED BY POWER LOSS NOTIFICATION.

If a SAS target device supports NOTIFY (POWER LOSS EXPECTED) and receives NOTIFY (POWER LOSS EXPECTED) on an SSP target port, then each SAS phy within the target device shall:

1. if there is an SSP connection, then transmit a BREAK on that connection; and
2. respond to SSP connection requests with OPEN_REJECT (RETRY) until the power loss timeout timer expires or power is lost.

If any frames are received by the SAS target device after receiving NOTIFY (POWER LOSS EXPECTED) before a connection is closed, then the SAS target device shall discard the received frames.

After power on, the power loss timeout timer shall be initialized and stopped until a NOTIFY (POWER LOSS EXPECTED) is received.

4.2.5.10 OPEN_ACCEPT

OPEN_ACCEPT indicates the acceptance of a connection request.

See 4.9.1 for details on connection requests.

4.2.5.11 OPEN_REJECT

OPEN_REJECT indicates that a connection request has been rejected and indicates the reason for the rejection. The result of some OPEN_REJECTs is to abandon (i.e., not retry) the connection request and the result of other OPEN_REJECTs is to retry the connection request.

All of the OPEN_REJECT versions defined in table 25 result in the originating device abandoning the connection request.

Table 25. OPEN_REJECT abandon primitives

Primitive	Originator	Description
OPEN_REJECT (BAD DESTINATION)	Expander phy	An expander device receives a request in which the destination SAS address equals the source SAS address, or a connection request arrives through an expander phy using the direct routing or table routing method and the expander device determines the connection request would have to be routed to the same expander port as the expander port through which the connection request arrived.
OPEN_REJECT (CONNECTION RATE NOT SUPPORTED)	Any phy	The requested connection rate is not supported on some physical link on the pathway between the source phy and destination phy. When a SAS initiator phy is directly attached to a SAS target phy, the requested connection rate is not supported by the destination phy. The connection request may be modified and reattempted as described in 4.9.1.2.2.

Table 25. OPEN_REJECT abandon primitives

Primitive	Originator	Description
OPEN_REJECT (PROTOCOL NOT SUPPORTED)	Destination phy	Device with destination SAS address exists but the destination device does not support the requested initiator/ target role, protocol, initiator connection tag, or features (i.e., the values in the INITIATOR PORT bit, the PROTOCOL field, the INITIATOR CONNECTION TAG field, and/or the FEATURES field in the OPEN address frame are not supported).
OPEN_REJECT (RESERVED ABANDON 0)	Unknown	Reserved. Process the same as OPEN_REJECT (WRONG DESTINATION).
OPEN_REJECT (RESERVED ABANDON 1)	Unknown	Reserved. Process the same as OPEN_REJECT (WRONG DESTINATION).
OPEN_REJECT (RESERVED ABANDON 2)	Unknown	Reserved. Process the same as OPEN_REJECT (WRONG DESTINATION).
OPEN_REJECT (RESERVED ABANDON 3)	Unknown	Reserved. Process the same as OPEN_REJECT (WRONG DESTINATION).
OPEN_REJECT (STP RESOURCES BUSY)	Destination phy	STP target port with destination SAS address exists but the STP target port has an affiliation with another STP initiator port or all of the available task file registers have been allocated to other STP initiator ports. Process the same as OPEN_REJECT (WRONG DESTINATION) for non-STP connection requests.
OPEN_REJECT (WRONG DESTI-NATION)	Destination phy	The destination SAS address does not match the SAS address of the SAS port to which the connection request was delivered.

All of the OPEN_REJECT versions defined in table 26 result in the originating device retrying the connection request.

Table 26. OPEN_REJECT retry primitives

Primitive	Originator	Description
OPEN_REJECT (NO DESTINA-TION) [c]	Expander phy	Either: a) No such destination device; b) a connection request arrives through an expander phy using the subtractive routing method and the expander device determines the connection request would have to be routed to the same expander port as the expander port through which the connection request arrived; or c) the SAS address is valid for an STP target port in an STP/SATA bridge, but the initial Register - Device to Host FIS has not been successfully received.
OPEN_REJECT (PATHWAY BLOCKED) [b]	Expander phy	An expander device determined the pathway was blocked by higher priority connection requests.

Table 26. OPEN_REJECT retry primitives

Primitive	Originator	Description
OPEN_REJECT (RESERVED CONTINUE 0) [a]	Unknown	Reserved. Process the same as OPEN_REJECT (RETRY).
OPEN_REJECT (RESERVED CONTINUE 1) [a]	Unknown	Reserved. Process the same as OPEN_REJECT (RETRY).
OPEN_REJECT (RESERVED INITIALIZE 0) [c]	Unknown	Reserved. Process the same as OPEN_REJECT (NO DESTINATION).
OPEN_REJECT (RESERVED INITIALIZE 1) [c]	Unknown	Reserved. Process the same as OPEN_REJECT (NO DESTINATION).
OPEN_REJECT (RESERVED STOP 0) [b]	Unknown	Reserved. Process the same as OPEN_REJECT (PATHWAY BLOCKED)
OPEN_REJECT (RESERVED STOP 1) [b]	Unknown	Reserved. Process the same as OPEN_REJECT (PATHWAY BLOCKED).
OPEN_REJECT (RETRY) [a]	Destination phy	Device with destination SAS address exists but is not able to accept connections.

[a] If the I_T Nexus Loss timer is already running, it is stopped.
[b] If the I_T Nexus Loss timer is already running, it continues running. Stop retrying the connection request if the I_T Nexus Loss timer expires.
[c] If the I_T Nexus Loss timer is already running, it continues running; if it is not already running, it is initialized and started. Stop retrying the connection request if the I_T Nexus Loss timer expires.

When a destination device detects more than one reason to transmit an OPEN_REJECT, the device transmits only one OPEN_REJECT and shall select the primitive using the following priority:

1. OPEN_REJECT (WRONG DESTINATION) (highest priority selection);
2. OPEN_REJECT (PROTOCOL NOT SUPPORTED);
3. OPEN_REJECT (CONNECTION RATE NOT SUPPORTED);
4. OPEN_REJECT (STP RESOURCES BUSY); or
5. OPEN_REJECT (RETRY) (lowest priority selection).

When an expander device detects more than one reason to transmit an OPEN_REJECT, the expander transmits only one OPEN_REJECT primitive and selects that primitive using the following priority:

1. OPEN_REJECT (BAD DESTINATION) or OPEN_REJECT (NO DESTINATION) (highest priority selection);
2. OPEN_REJECT (CONNECTION RATE NOT SUPPORTED); or
3. OPEN_REJECT (STP RESOURCES BUSY) or OPEN_REJECT (PATHWAY BLOCKED) (lowest priority selection).

See 4.9.1 for details on connection requests.

4.2.5.12 SOAF (Start of address frame)

SOAF indicates the start of an address frame.

See 4.7 for details on address frames.

4.2.6 Primitives used only inside SSP

4.2.6.1 ACK (Acknowledge)

ACK indicates the positive acknowledgement of an SSP frame.

See 4.9.3.3 for details on SSP frame transmission.

4.2.6.2 CREDIT_BLOCKED

CREDIT_BLOCKED indicates that no more credit is going to be sent during this connection.

See 4.9.3.4 for details on SSP flow control.

4.2.6.3 DONE

DONE is used to start closing an SSP connection and indicate a reason for doing so. This primitive may be originated by an SSP initiator port or an SSP target port.

The versions of DONE representing different reasons are defined in table 27.).

Table 27. DONE primitives

Primitive	Description
DONE (ACK/NAK TIMEOUT)	A timed out occurred waiting for an ACK or NAK and the transmitter is going to transmit BREAK in 1 ms unless DONE is received within 1 ms of transmitting the DONE (ACK/NAK TIMEOUT).
DONE (RESERVED TIMEOUT 0)	Reserved. Processed the same as DONE (ACK/NAK TIMEOUT).
DONE (RESERVED TIMEOUT 1)	Reserved. Processed the same as DONE (ACK/NAK TIMEOUT).
DONE (NORMAL)	Finished transmitting all frames.
DONE (RESERVED 0)	Reserved. Processed the same as DONE (NORMAL).
DONE (RESERVED 1)	Reserved. Processed the same as DONE (NORMAL).
DONE (CREDIT TIMEOUT)	A timed out occurred waiting for an RRDY or received a CREDIT BLOCKED and the transmitter is going to transmit BREAK if credit is extended for 1 ms without receiving a frame or a DONE.

See 4.9.3.6 for details on closing SSP connections.

4.2.6.4 EOF (End of frame)

EOF indicates the end of an SSP or SMP frame. See 4.9.3.3 for details on SSP frame transmission.

4.2.6.5 NAK (Negative acknowledgement)

NAK indicates the negative acknowledgement of an SSP frame and the reason for doing so. The versions of NAK representing different reasons are defined in table 28.

Table 28. NAK primitives

Primitive	Description
NAK (CRC ERROR)	The frame had a bad CRC.

Table 28. NAK primitives

Primitive	Description
NAK (RESERVED 0)	Reserved. Processed the same as NAK (CRC ERROR).
NAK (RESERVED 1)	Reserved. Processed the same as NAK (CRC ERROR).
NAK (RESERVED 2)	Reserved. Processed the same as NAK (CRC ERROR).

See 4.9.3.3 for details on SSP frame transmission.

4.2.6.6 RRDY (Receiver ready)

RRDY is used to increase SSP frame credit.

The versions of RRDY representing different reasons are defined in table 28.

Table 29. RRDY primitives

Primitive	Description
RRDY (NORMAL)	Increase transmit frame credit by one.
RRDY (RESERVED 0)	Reserved. Processed the same as RRDY (NORMAL).
RRDY (RESERVED 1)	Reserved. Processed the same as RRDY (NORMAL).

See 4.9.3.4 for details on SSP flow control.

4.2.6.7 SOF (Start of frame)

SOF indicates the start of an SSP. See 4.9.3.3 for details on SSP frame transmission.

4.3 Clock skew management

The internal clock for a device is typically based on a PLL with its own clock generator and is used when transmitting dwords on the physical link. When receiving, however, dwords need to be latched based on a clock derived from the input bit stream itself. Although the input clock is nominally a fixed frequency, it may differ slightly from the internal clock frequency due to accepted manufacturing tolerance. Over time, if the input clock is faster than the internal clock, the device may receive a dword and not be able to forward it to an internal buffer; this is called an overrun. If the input clock is slower than the internal clock, the device may not have a dword when needed in an internal buffer; this is called an underrun.

To solve this problem, transmitting devices insert ALIGNs or NOTIFYs in the dword stream. Receivers may pass ALIGNs and NOTIFYs through to their internal buffers, or may strip them out when an overrun occurs. Receivers add ALIGNs or NOTIFYs when an underrun occurs. The internal logic shall ignore all ALIGNs and NOTIFYs that arrive in the internal buffers.

Elasticity buffer circuitry, as shown in figure 24, is required to absorb the slight differences in frequencies between the SAS initiator phy, SAS target phy, and expander phys. The frequency tolerance for a phy is specified in SAS-2 clause 5.3.3.

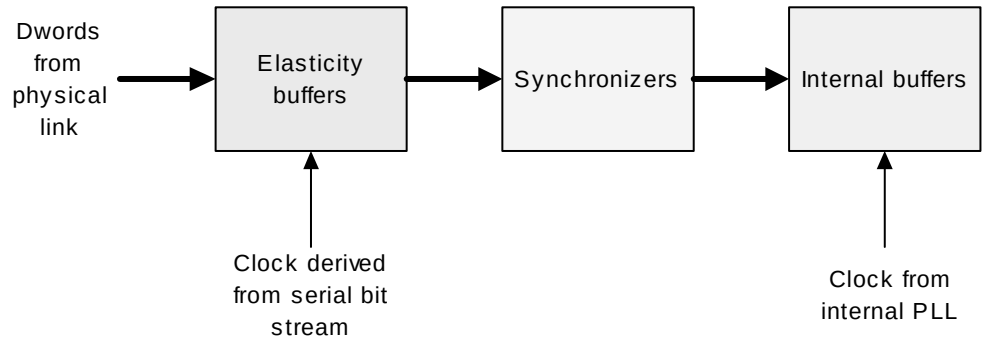


Figure 24. Elasticity buffers

The drive sends an ALIGN every 2,048 dwords whether in or outside a connection.

See 4.2.5.2 for details on rotating through ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3). NOTIFYs may also be used in place of ALIGNs (see 4.2.5.9) on SAS physical links.

An expander device that is forwarding dwords (i.e., is not the original source) is allowed to insert or delete as many ALIGNs or NOTIFYs as required to match the transmit and receive connection rates (e.g., it is not required to ensure that it transmits one ALIGN or NOTIFY within every 2,048 dwords when forwarding to a SAS physical link).

4.4 Idle physical links

Idle dwords are vendor-specific data dwords.

Phys shall transmit idle dwords if there are no other dwords to transmit and:

a. no connection is open; or

b. an SSP connection is open.

Idle dwords are scrambled (see 4.6).

4.5 CRC

4.5.1 CRC overview

All frames include cyclic redundancy check (CRC) values to help detect transmission errors.

4.5.1.1 CRC field

The Cyclic Redundancy Check (CRC) is a 4-byte field that follows the payload field. The CRC is used to verify the integrity of the frame header and payload fields. This helps detect errors in a frame. All data dwords following the SOF and before the CRC field are included in the CRC calculation.

The algorithm used to calculate the CRC field value is the same as that used in the Fiber Distributed Data Interface (FDDI). The polynomial for the CRC is:

$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

4.6 Scrambling

Scrambling is used to reduce the probability of long strings of repeated patterns appearing on the physical link.

All data dwords are scrambled. Table 30 lists the scrambling for different types of data dwords.

Table 30. Scrambling for different data dword types

Connection state	Data dword type	Description of scrambling
Outside connections	SAS idle dword	When a connection is not open and there are no other dwords to transmit, vendor-specific scrambled data dwords shall be transmitted.
	Address frame	After an SOAF, all data dwords shall be scrambled until the EOAF.
Inside SSP connection	SSP frame	After an SOF, all data dwords shall be scrambled until the EOF.
	SSP idle dword	When there are no other dwords to transmit, vendor-specific scrambled data dwords shall be transmitted.
Inside SMP connection	SMP frame	After an SOF, all data dwords shall be scrambled until the EOF.
	SMP idle dword	When there are no other dwords to transmit, vendor-specific scrambled data dwords shall be transmitted.
Inside STP connection	STP frame	After a SATA_SOF, all data dwords shall be scrambled until the SATA_EOF.
	Repeated SATA primitive	After a SATA_CONT, vendor-specific scrambled data dwords shall be sent until a primitive other than ALIGN or NOTIFY is transmitted.

Data dwords being transmitted are XORed with a defined pattern to produce a scrambled value encoded and transmitted on the physical link. Received data dwords are XORed with the same pattern after decoding to produce the original data dword value, provided there are no transmission errors.

The pattern that is XORed with the data dwords is defined by the output of a linear feedback shift register implemented with the following polynomial:

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

The output of the pattern generator is 16 bits wide. For each data dword the output of the generator is applied to the lower 16 bits (i.e., bits 15 through 0) of the 32-bit data dword being transmitted or received; the next output of the generator is applied to the upper 16 bits (i.e., bits 31 through 16).

The value of the linear feedback shift register is initialized at each SOF and SOAF to FFFFh. The sequence of values XORed with the data being transmitted is constant for a given displacement from the last SOF or SOAF

4.7 Address frames

4.7.1 Address frames overview

Address frames are used for the identification sequence and for connection requests. The address frame follows an SOAF and ends with an EOAF. Address frames are only sent outside connections. Address frames are not terminated early. All data dwords in an address frame are scrambled.

Table 31 defines the address frame format.

Table 31. Address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0					ADDRESS FRAME TYPE			
1	Frame type dependent bytes							
27								
28	(MSB)	CRC						
31								(LSB)

The ADDRESS FRAME TYPE field indicates the type of address frame and is defined in table 32. This field determines the definition of the frame type dependent bytes.

Table 32. Address frame types

Code	Frame type	Description
0h	Identify	Identification sequence
1h	Open	Connection request
All others	Reserved	

The CRC field contains a CRC value (see 4.5) that is computed over the entire address frame prior to the CRC field.

Address frames with unknown address frame types, incorrect lengths, or CRC errors are ignored by the drive.

4.7.2 IDENTIFY address frame

Table 33 defines the IDENTIFY address frame format used for the identification sequence. The IDENTIFY address frame is sent after the phy reset sequence completes if the physical link is a SAS physical link.

Table 33. IDENTIFY address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	Restricted (for OPEN address frame)	DEVICE TYPE			ADDRESS FRAME TYPE (0h)			
1	Restricted (for OPEN address frame)							
2	Reserved				SSP INITIATOR PORT	STP INITIATOR PORT	SMP INITIATOR PORT	Restricted (for OPEN address frame)
3	Reserved				SSP TARGET PORT	STP TARGET PORT	SMP TARGET PORT	Restricted (for OPEN address frame)
4	DEVICE NAME							
11								
12	SAS ADDRESS							
19								
20	PHY IDENTIFIER							
21	Reserved							
27								
28	(MSB) CRC							
31								(LSB)

The DEVICE TYPE field indicates the type of device containing the phy, and is defined in table 34.

Table 34. Device types

Code	Description
001b	End device
010b	Edge expander device
011b	Fanout expander device
All others	Reserved

The ADDRESS FRAME TYPE field shall be set to 0h.

An SSP INITIATOR PORT bit set to one indicates the presence of an SSP initiator port. An SSP INITIATOR PORT bit set to zero indicates an SSP initiator port is not present. Expander devices set the SSP INITIATOR PORT bit to zero. The drive sets this bit to a zero in IDENTIFY frames it originates.

An STP INITIATOR PORT bit set to one indicates the presence of an STP initiator port. An STP INITIATOR PORT bit set to zero indicates an STP initiator port is not present. Expander devices set the STP INITIATOR PORT bit to zero. The drive sets this bit to a zero in IDENTIFY frames it originates.

An SMP INITIATOR PORT bit set to one indicates the presence of an SMP initiator port. An SMP INITIATOR PORT bit set to zero indicates an SMP initiator port is not present. Expander devices may set the SMP INITIATOR PORT bit to one. The drive sets this bit to a zero in IDENTIFY frames it originates.

An SSP TARGET PORT bit set to one indicates the presence of an SSP target port. An SSP TARGET PORT bit set to zero indicates an SSP target port is not present. Expander devices set the SSP TARGET PORT bit to zero. The drive sets this bit to a one in IDENTIFY frames it originates.

An STP TARGET PORT bit set to one indicates the presence of an STP target port. An STP TARGET PORT bit set to zero indicates an STP target port is not present. Expander devices set the STP TARGET PORT bit to zero. The drive sets this bit to a zero in IDENTIFY frames it originates.

An SMP TARGET PORT bit set to one indicates the presence of an SMP target port. An SMP TARGET PORT bit set to zero indicates an SMP target port is not present. Expander devices set the SMP TARGET PORT bit to one. The drive sets this bit to a zero in IDENTIFY frames it originates.

For SAS ports, the SAS ADDRESS field indicates the port identifier of the SAS port transmitting the IDENTIFY address frame. For expander ports, the SAS ADDRESS field indicates the device name of the expander device transmitting the IDENTIFY address frame.

The DEVICE NAME field specifies the device name (see 2.2.4) of the SAS device or expander device transmitting the IDENTIFY address frame. A DEVICE NAME field set to 00000000 00000000h specifies the device name is not provided in this field.

Note. In expander devices, the DEVICE NAME field, if not set to 00000000 00000000h, contains the same value as the SAS ADDRESS field.

For SAS ports, the SAS ADDRESS field specifies the port identifier (see 2.2.5) of the SAS port transmitting the IDENTIFY address frame. For expander ports, the SAS ADDRESS field specifies the device name (see 2.2.4) of the expander device transmitting the IDENTIFY address frame.

The PHY IDENTIFIER field indicates the phy identifier of the phy transmitting the IDENTIFY address frame.

The CRC field is defined in 4.7.1.

4.7.3 OPEN address frame

Table 35 defines the OPEN address frame format used for connection requests.

Table 35. OPEN address frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	INITIATOR PORT	PROTOCOL			ADDRESS FRAME TYPE (1h)			
1	FEATURES				CONNECTION RATE			
2	(MSB)	INITIATOR CONNECTION TAG						
3								(LSB)
4		DESTINATION SAS ADDRESS						
11								
12		SOURCE SAS ADDRESS						
19								
20	COMPATIBLE FEATURES							
21	PATHWAY BLOCKED COUNT							
22	(MSB)	ARBITRATION WAIT TIME						
23								(LSB)
24		MORE COMPATIBLE FEATURES						
27								
28	(MSB)	CRC						
31								(LSB)

An INITIATOR PORT bit set to one indicates the source port is acting as a SAS initiator port. An INITIATOR PORT bit set to zero indicates the source port is acting as a SAS target port. If a SAS target/initiator port sets the INITIATOR PORT bit to one, it operates only in its initiator role during the connection. If a target/initiator port sets the INITIATOR PORT bit to zero, it operates only in its target role during the connection.

If a SAS target/initiator port accepts an OPEN address frame with the INITIATOR PORT bit set to one, it operates only in its target role during the connection. If a SAS target/initiator port accepts an OPEN address frame with the INITIATOR PORT bit set to zero, it operates only in its initiator role during the connection.

The PROTOCOL field indicates the protocol for the connection being requested and is defined in table 36.

Table 36. Protocol

Code	Description
000b	SMP
001b	SSP
010b	STP
All others	Reserved

The ADDRESS FRAME TYPE field is set to 1h.

The FEATURES field is set to zero.

The CONNECTION RATE field indicates the connection rate (see 2.1.10) being requested between the source and destination, and is defined in table 37.

Table 37. Connection rate

Code	Description
8h	1.5 Gbps
9h	3.0 Gbps
All others	Reserved

When requesting a connection to a SAS target port, a SAS initiator port sets the CONNECTION RATE field to a value supported by at least one potential pathway.

When requesting an SSP connection to an SSP initiator port, the drive sets the CONNECTION RATE field to the connection rate in effect when the command was received unless it has received an OPEN_REJECT (CONNECTION RATE NOT SUPPORTED). When the drive receives an OPEN_REJECT (CONNECTION RATE NOTE SUPPORTED), it will abort the command. See 4.9.1.2.2 for details on handling OPEN_REJECT (CONNECTION RATE NOT SUPPORTED).

When the drive determines it has frames to sends to a SAS initiator port and a connection already exists to that initiator, the drive sends the frames at the current connection rate.

The INITIATOR CONNECTION TAG field is used for SSP connection requests to provide a SAS initiator port an alternative to using the SAS target port's SAS address for context lookup when the SAS target port originates a connection request. SSP initiator ports set the INITIATOR CONNECTION TAG field to FFFFh if they do not require the field be provided by the SAS target port. If they do require the field to be provided, an SSP initiator port should set the INITIATOR CONNECTION TAG field to a unique value per SAS target port. A SAS initiator port is required to use the same INITIATOR CONNECTION TAG field value for all connection requests to the same SAS target port, and only change the INITIATOR CONNECTION TAG field value when it has no commands outstanding to that SAS target port. The drive does not to check the consistency of the INITIATOR CONNECTION TAG field in different connection requests from the same SAS initiator port. When requesting a connection to a SAS initiator port, the drive sets the INITIATOR CONNECTION TAG field in the connection request to the INITIATOR CONNECTION TAG in affect when the command or task management request was received from the SAS initiator port. If more than one command or task management response is sent in a connection, the INITIATOR CONNECTION TAG in the connection request may be from any of the commands or task management requests if they were inconsistent from the SAS initiator.

The DESTINATION SAS ADDRESS field indicates the port identifier of the SAS port to which a connection is being requested.

The SOURCE SAS ADDRESS field indicates the port identifier of the SAS port that originated the OPEN address frame.

The drive sets the COMPATIBLE FEATURES field In OPEN address frames it originates zero. The drive ignores the COMPATIBLE FEATURES field in received OPEN address frames.

The PATHWAY BLOCKED COUNT field indicates the number of times the port has retried this connection request due to receiving OPEN_REJECT (PATHWAY BLOCKED). The drive does not increment the PATHWAY BLOCKED COUNT value past FFh. If the drive changes the destination of a connection request, it sets the PATHWAY BLOCKED COUNT of 00h.

The ARBITRATION WAIT TIME field indicates how long the port transmitting the OPEN address frame has been waiting for a connection request to be accepted. This time is maintained by the port layer in an Arbitration Wait Time timer. For values from 0000h to 7FFFh, the Arbitration Wait Time timer increments in one microsecond steps. For values from 8000h to FFFFh, the Arbitration Wait Time timer increments in one millisecond steps. The maximum value represents 32,767 ms + 32,768 µs. Table 38 describes several values of the ARBITRATION WAIT TIME field. See 4.9.1.3 for details on arbitration fairness.

Table 38. Arbitration wait time

Code	Description
0000h	0 µs
0001h	1 µs
...	...
7FFFh	32,767 µs
8000h	0 ms + 32,768 µs
8001h	1 ms + 32,768 µs
...	...
FFFFh	32,767 ms + 3,768 µs

The drive sets the MORE COMPATIBLE FEATURES field in OPEN address frames it originates to zero. The drive ignores the MORE COMPATIBLE FEATURES field.

The CRC field is defined in 4.7.1.

4.8 Identification and hard reset sequence

4.8.1 Identification and hard reset sequence overview

After the phy reset sequence has been completed indicating the physical link is using SAS, each phy transmits either:

a. an IDENTIFY address frame (see 4.7.2); or

b. a HARD_RESET.

Each phy receives an IDENTIFY address frame or a HARD_RESET from the phy to which it is attached. The combination of a phy reset sequence, an optional hard reset sequence, and an identification sequence is called a link reset sequence (see 2.3.1).

The drive transmits the different SAS address on the phy of each port.

The drive considers the attached phys as part of different ports regardless of the SAS addresses received in an IDENTIFY address frame.

The drive will restart the phy reset sequence if it does not receive a valid IDENTIFY address frame within 1 ms of completing the phy reset sequence.

If the drive receives an additional IDENTIFY address frame after receiving the first one, without an intervening phy reset sequence, it ignores the additional IDENTIFY address frame.

If the drive receives a HARD_RESET on a port, it performs a reset (see 2.3.2). The drive will process HARD RESET only if it is received between the end of Phy Reset and before receiving an IDENTIFY address frame.

4.8.2 SAS initiator device rules

When a discover process is performed after a link reset sequence, the management application client discovers all the devices in the SAS domain. When a discover process is performed after a BROADCAST (CHANGE), the management application client determines which devices have been added to or removed from the SAS domain.

The discover information may be used to select connection rates for connection requests.

4.8.3 Fanout expander device rules

After completing the identification sequence on a phy and completing internal initialization, the ECM within a fanout expander device is capable of routing connection requests through that phy. The expander device may return OPEN_REJECT (NO DESTINATION) until it is ready for connection requests.

After a link reset sequence, or after receiving a BROADCAST (CHANGE), the management application client behind an SMP initiator port in a fanout expander device that does not have a configurable expander route table follows the SAS initiator device rules to perform a discover process. Reference the SAS ANSI for a description of this process.

The ECM of a fanout expander device that has a configurable expander route table is dependent on the completion of the discover process for routing connection requests using the table routing method.

4.8.4 Edge expander device rules

After completing the identification sequence on a phy and completing internal initialization, the ECM within an edge expander device is capable of routing connection requests through that phy. The expander device may return OPEN_REJECT (NO DESTINATION) until it is ready for connection requests.

The ECM of an edge expander device that has a configurable expander route table is dependent on the completion of the discover process for routing connection requests using the table routing method. Reference the SAS ANSI for a description of this process.

4.9 SAS domain changes

After power on or receiving BROADCAST (CHANGE), an application client in each SAS initiator port should scan the SAS domain using the discover process to search for SAS initiator devices, SAS target devices, and expander devices.

The expander device transmits BROADCAST (CHANGE) from at least one phy in each expander port other than the expander port that is the cause for transmitting BROADCAST (CHANGE).

Expander devices transmit BROADCAST (CHANGE) for the following reasons:

a. after an expander phy has lost dword synchronization;

b. after the link reset sequence completes; and

c. after the expander device receives BROADCAST (CHANGE).

BROADCAST (CHANGE) may be sent by SAS initiator ports to force other SAS initiator ports and expander ports to re-run the discover process.

A SAS initiator port that detects BROADCAST (CHANGE) follows the SAS initiator device rules to discover the topology.

A fanout expander device that detects BROADCAST (CHANGE) follows the fanout device rules to discover the topology.

An edge expander device that detects BROADCAST (CHANGE) follows the edge device rules.

4.9.1 Connections

4.9.1.1 Connections overview

A connection is opened between a SAS initiator port and a SAS target port before communication begins. A connection is established between one SAS initiator phy in the SAS initiator port and one SAS target phy in the SAS target port.

SSP initiator ports open SSP connections to transmit SCSI commands, task management functions, or transfer data. SSP target ports open SSP connections to transfer data or transmit status.

SMP initiator ports open SMP connections to transmit SMP requests and receive SMP responses. Reference the SAS ANSI for a description of SMP.

STP initiator ports and STP target ports open STP connections to transmit SATA frames. An STP target port in an expander device opens STP connections on behalf of SATA devices. Reference the SAS ANSI for a description of STP.

The OPEN address frame is used to request that a connection be opened. AIP, OPEN_ACCEPT and OPEN_REJECT are the responses to an OPEN address frame. BREAK is used to abort connection requests and to unilaterally break a connection. CLOSE is used for orderly closing a connection.

Connections use a single pathway from the SAS initiator phy to the SAS target phy. While a connection is open, only one pathway is used for that connection.

A wide port may have separate connections on each of its phys.

4.9.1.2 Opening a connection

4.9.1.2.1 Connection request

The OPEN address frame (see 4.7.3) is used to open a connection from a source port to a destination port using one source phy and one destination phy.

To make a connection request, the source port transmits an OPEN address frame through an available phy. The source phy transmits idle dwords and ALIGNs required for clock skew management and rate matching after the OPEN address frame until it receives a response or aborts the connection request with BREAK.

After transmitting an OPEN address frame, the source phy initializes and starts a 1 ms Open Timeout timer. Whenever an AIP is received, the source phy reinitializes and restarts the Open Timeout timer. Source phys are not required to enforce a limit on the number of AIPs received before aborting the connection request. When any connection response is received, the source phy reinitializes the Open Timeout timer. If the Open Timeout timer expires before a connection response is received, the source phy may assume the destination port does not exist and transmit BREAK to abort the connection request.

The OPEN address frame flows through expander onto intermediate physical links. If an expander on the pathway is unable to forward the connect request because none of the prospective physical links support the requested connection rate, the expander device returns OPEN_REJECT (CONNECTION RATE NOT SUPPORTED). If the OPEN address frame reaches the destination, the destination returns either OPEN_ACCEPT or OPEN_REJECT. Rate matching is used on any physical links in the pathway with negotiated physical link rates that are faster than the requested connection rate (see 4.9.2).

4.9.1.2.2 Connection responses

If the drive receives one of the following OPEN REJECT types at any time in the process of opening a connection, the drive shall abandon the connection request and abort the associated command:

- OPEN REJECT (BAD DESTINATION)
- OPEN REJECT (PROTOCOL NOT SUPPORTED)
- OPEN REJECT (RESERVED ABANDON n)
- OPEN REJECT (STP RESOURCES BUSY)
- OPEN REJECT (WRONG DESTINATION)
- OPEN REJECT (CONNECTION RATE NOT SUPPORTED) The connection request should be tried at a different rate per connection responses in SAS 1.1. A link from the initiator to the expander may be a wide link running at different speeds or just re negotiated to a different rate. See Table 39.

Table 39. Reaction to CONNECTION RATE NOT SUPPORTED

Physical rate	Rate in hosts's Open Address frame with command	Reaction
1.5	1.5	Internally abort command.
1.5	3.0	Retry one time at 1.5. Note: This can only happen if OOB occurs while the command is in the queue. If the drive receives an Open Address frame for 3.0 when it is physically at 1.5, it will send OPEN REJECT (CONNECTION RATE NOT SUPPORTED).
3.0	1.5	Retry one time at 3.0.
3.0	3.0	Retry one time at 1.5.

The I_T Nexus Loss Timer allows the drive to monitor the progress of a connection request. If the drive receives a connection response causing it to start the I_T Nexus Loss Timer, and is unable to establish a connection before the timer expires, the drive shall abort all commands for the host which could not be opened, and establish a UNIT ATTENTION condition (Sense Key=06), with ASC/ASCQ = I_T NEXUS LOSS OCCURRED (29/07) for that host. The I_T Nexus Loss time is defined by the I_T NEXUS LOSS TIME field in Mode Page 19 (Protocol Specific Port Control). If I_T NEXUS LOSS TIME is FFFFh the drive shall not run the I_T Nexus Loss timer when opening connections. If I_T NEXUS LOSS TIME is 0000h, the drive shall use a value of 2 seconds.

The drive shall start the I_T Nexus Loss Timer (if its value is not FFFFh) if it receives any of the following responses to a connection attempt,

- OPEN_REJECT (NO DESTINATION)
- OPEN_REJECT (RESERVED INITIALIZE 0)
- OPEN_REJECT (RESERVED INITIALIZE 1)
- Open connection timeout

Open connection timeout is detected when the drive sends an Open Address frame and does not receive OPEN ACCEPT, OPEN REJECT, or a higher priority Open Address frame within 1 ms.

While receiving OPEN REJECT(RETRY), the drive shall retry the connection until one of the following occurs:

- Hard Reset received
- Connection established
- OPEN_ACCEPT received in response to an open connection attempt
- Port accepts connection from the SAS address it is attempting to open
- OPEN_REJECT(RESERVED CONTINUE 0) received in response to an open connection attempt
- OPEN_REJECT(RESERVED CONTINUE 1) received in response to an open connection attempt
- One of the Open Reject “Abandon” primitive types is received (listed above)

While receiving OPEN REJECT(PATHWAY BLOCKED) or OPEN REJECT (RESERVED STOP n) the drive shall retry the connection indefinitely unless the I_T Nexus Loss Timer has previously started.

The drive shall continue attempting to open the connection until one of the following occurs:

- I_T Nexus Loss Timer expires
- Hard Reset received
- Connection established
- OPEN_ACCEPT received in response to an open connection attempt
- Port accepts connection from the SAS address it is attempting to open
- OPEN_REJECT(RETRY) received in response to an open connection attempt
- OPEN_REJECT(RESERVED CONTINUE 0) received in response to an open connection attempt
- OPEN_REJECT(RESERVED CONTINUE 1) received in response to an open connection attempt
- One of the Open Reject “Abandon” primitive types is received (listed above)

4.9.1.3 Arbitration fairness

SAS supports least-recently used arbitration fairness.

Each SAS port and expander port implement an Arbitration Wait Time timer which counts the time from when the port makes a connection request until its request is granted. The Arbitration Wait Time timer counts in microseconds from 0 ms to 32,767 ms and in milliseconds from 32,768 ms to 32,767 ms + 32,768 ms. The Arbitration Wait Time timer stops incrementing when its value reaches 32,767 ms + 32,768 ms.

SAS ports (i.e., SAS initiator ports and SAS target ports) start the Arbitration Wait Time timer when they transmit the first OPEN address frame (see 4.7.3) for the connection request. When the SAS port retransmits the OPEN address frame (e.g., after losing arbitration and handling an inbound OPEN address frame), it sets the ARBITRATION WAIT TIME field to the current value of the Arbitration Wait Time timer.

Drive ports set the Arbitration Wait Time timer to zero when they transmit the first OPEN address frame for the connection request. A SAS initiator port may be unfair by setting the ARBITRATION WAIT TIME field in the OPEN address frame (see 4.7.3) to a higher value than its Arbitration Wait Time timer indicates. Unfair SAS ports are not permitted to set the ARBITRATION WAIT TIME field to a value greater than or equal to 8000h; this limits the amount of unfairness and helps prevent livelocks.

The expander port that receives an OPEN address frame sets the Arbitration Wait Time timer to the value of the incoming ARBITRATION WAIT TIME field and start the Arbitration Wait Time timer as it arbitrates for internal access to the outgoing expander port. When the expander transmits the OPEN address frame out another expander port, it sets the outgoing ARBITRATION WAIT TIME field to the current value of the Arbitration Wait Time timer maintained by the incoming expander port.

A port stops the Arbitration Wait Time timer and sets it to zero when it wins arbitration (i.e., it receives either OPEN_ACCEPT or OPEN_REJECT from the destination SAS port rather than from an intermediate expander device). A port stops the Arbitration Wait Time timer when it loses arbitration to a connection request that satisfies its arbitration request (i.e., it receives an OPEN address frame from the destination SAS port with the INITIATOR PORT bit set to the opposite value and a matching PROTOCOL field).

When arbitrating for access to an outgoing expander port, the expander device selects the connection request from the expander port with the largest Arbitration Wait Time timer value. If the largest Arbitration Wait timer values are identical, then the connection request with the largest SOURCE SAS ADDRESS wins arbitration.

If two connection requests pass on a physical link, the winner shall be determined by comparing OPEN address frame field contents using the arbitration priority described in table 40.

Table 40. Arbitration priority for OPENs passing on a physical link

Bits 79-64 (79 is MSB)	Bits 63-0 (0 is LSB)
ARBITRATION WAIT TIME field value	SOURCE SAS ADDRESS field value

See 4.7.3 for details on the OPEN address frame and the ARBITRATION WAIT TIME field.

4.9.1.4 Arbitration and resource management in an expander device

4.9.1.4.1 Arbitration overview

The ECM arbitrates and assigns or denies path resources for connection attempts requested by each expander phy in response to receiving valid OPEN address frames.

Arbitration includes adherence to the SAS arbitration fairness algorithm and path recovery. Path recovery is used to avoid potential deadlock scenarios within the SAS topology by deterministically choosing which partial pathway(s) to tear down to allow at least one connection to complete.

The ECM responds to connection requests by returning arbitration won, lost, and reject confirmations to the requesting expander phy.

Each path request contains the Arbitration Wait Time and the Source SAS Address arguments from the received OPEN address frame.

If two or more path requests contend, the winner shall be selected by comparing the OPEN address frame contents using the arbitration priority described in table 41.

Table 41. Arbitration priority for contending path requests in the ECM

Bits 83-68 (83 is MSB)	Bits 67-5	Bits 3-0 (0 is LSB)
ARBITRATION WAIT TIME field value	SOURCE SAS ADDRESS field value	CONNECTION RATE field value

The ECM generates the arbitration reject confirmation when any of the following conditions are met:

a. the connection request does not map to a valid expander phy;
b. the connection request specifies an unsupported connection rate; or
c. the connection request specifies a destination port that contains at least one partial pathway and pathway recovery rules require this connection request to release path resources.

The ECM generates the arbitration lost confirmation when all of the following conditions are met:

a. the connection request maps to an available expander phy at a supported connection rate; and
b. the destination expander phy of this connection request has received a higher priority OPEN address frame with this expander phy as its destination (i.e., when two expander phys both receive an OPEN address frame destined for each other, the ECM shall provide arbitration lost confirmation to the expander phy that received the lowest priority OPEN address frame).

The ECM generates the arbitration won confirmation when all of the following conditions are met:

a. the connection request maps to an available expander phy at a supported connection rate; and
b. no higher priority connection requests are present with this expander phy as the destination.

4.9.1.4.2 Arbitration status

Arbitration status is conveyed between expander devices and by expander devices to SAS endpoints using AIP primitives. This status is used to monitor the progress of connection attempts and to facilitate pathway recovery as part of deadlock recovery.

The arbitration status of an expander phy is set to the last type of AIP received.

4.9.1.4.3 Partial Pathway Timeout timer

Each expander phy maintains a Partial Pathway Timeout timer. This timer is used to identify potential deadlock conditions and to request resolution by the ECM. An expander phy initializes the Partial Pathway Timeout timer to the partial pathway timeout value it reports in the SMP DISCOVER function and run the Partial Pathway Timeout timer whenever the ECM provides confirmation to an expander phy that all expander phys within the requested destination port are blocked waiting on partial pathways.

When the Partial Pathway Timeout timer is not running, an expander phy initializes and starts the Partial Pathway Timeout timer when all of the following conditions are met:

a. there are no unallocated expander phys within a requested destination port available to complete the connection; and
b. at least one expander phy within the requested destination port contains a blocked partial pathway.

When one of the conditions above are not met, the expander phy stops the Partial Pathway Timeout timer. If the timer expires, pathway recovery occurs (see 4.9.1.4.4).

4.9.1.4.4 Pathway recovery

Pathway recovery provides a means to abort connection requests in order to prevent deadlock using pathway recovery priority comparisons. Pathway recovery priority compares the OPEN address frames of the blocked connection requests as described in table 42.

Table 42. Pathway recovery priority

Bits 75-68 (75 is MSB)	Bits 67-5	Bits 3-0 (0 is LSB)
PATHWAY BLOCKED COUNT field value	SOURCE SAS ADDRESS field value	CONNECTION RATE field value

When the Partial Pathway Timeout timer for an arbitrating expander phy expires (i.e., reaches a value of zero), the ECM determines whether to continue the connection request or to abort the connection request.

The ECM instructs the arbitrating expander phy to reject the connection request by transmitting OPEN_REJECT (PATHWAY_BLOCKED) when the Partial Pathway Timeout timer expires and the pathway recovery priority of the arbitrating expander phy (i.e., the expander phy requesting the connection) is less than the pathway recovery priority of all expander phys within the destination port with an arbitration status of WAITING_ON_PARTIAL.

4.9.1.5 Expander devices and connection requests

4.9.1.5.1 All expander devices

Before an expander device transmits AIP, it may have transmitted an OPEN address frame on the same physical link. Arbitration fairness dictates which OPEN address frame wins (see 4.9.1.3).

After an expander device transmits an AIP, it does not forward an OPEN address frame unless it has higher arbitration priority than the incoming connection request.

Expander devices transmit no more than three consecutive AIPs without transmitting an idle dword. Expander devices transmit at least one AIP every 128 dwords.

Expander devices transmit an AIP within 128 dwords of receiving an OPEN address frame.

4.9.1.5.2 Edge expander devices

When an edge expander device receives a connection request, it compares the destination SAS address to the SAS addresses of the devices to which each of its phys is attached. For all phys which have table routing attributes and are attached to edge expander devices, it compares the destination SAS address to all the enabled routed SAS addresses in the expander route table.

If it finds a match in one or more phys, then the expander device arbitrates for access to one of the matching phys and forward the connection request.

If it does not find a match, but at least one phy has the subtractive routing attribute and is attached to an expander device (either an edge expander device or a fanout expander device), and the request did not come from that expander device, the connection request forwarded to the expander device through any of the subtractive routing phys.

If it does not find a match and no subtractive routing phy is available, the edge expander device replies with OPEN_REJECT (NO DESTINATION).

If the destination phy is in the same expander port as the source phy and the source phys are using the subtractive routing method, the edge expander device replies with OPEN_REJECT (NO DESTINATION). If the source phys are not using subtractive routing, the edge expander device shall reply with OPEN_REJECT (BAD DESTINATION).

4.9.1.5.3 Fanout expander devices

When a fanout expander device receives a connection request, it compares the destination SAS address to the SAS addresses of the devices to which each of its phys is attached. For all phys that are attached to edge expander devices, the fanout expander device compares the destination SAS addresses to all the enabled SAS addresses in the expander route table.

If the fanout expander device finds a match in one or more phys, it arbitrates for access to one of the matching phys and forward the connection request.

If the fanout expander device does not find a match, it replies with OPEN_REJECT (NO DESTINATION). If the destination phy is in the same expander port as the source phy, it replies with OPEN_REJECT (BAD DESTINATION).

4.9.1.6 Aborting a connection request

BREAK may be used to abort a connection request. The source phy transmits a BREAK after the Open Time-out timer expires or if it chooses to abort its request for any other reason.

After transmitting BREAK, the source phy initializes a Break Timeout timer to 1 ms and start the Break Timeout timer.

Table 43 lists the responses to a BREAK being transmitted before a connection response has been received.

Table 43. Abort connection responses

Response	Description
BREAK	This confirms that the connection request has been aborted.
Break Timeout timer expires	The originating phy shall assume the connection request has been aborted.

When a phy sourcing a BREAK is attached to an expander, the BREAK response to the source phy is generated by the expander phy to which the source phy is attached, not the other SAS phy in the connection. If the expander has transmitted a connection request to the destination, it transmits BREAK to the destination. If the expander device has not transmitted a connection request to the destination, it shall not transmit BREAK to the destination. After transmitting BREAK back to the originating phy, the expander does not forward dwords from the destination. Figure 25 shows an example of BREAK usage.

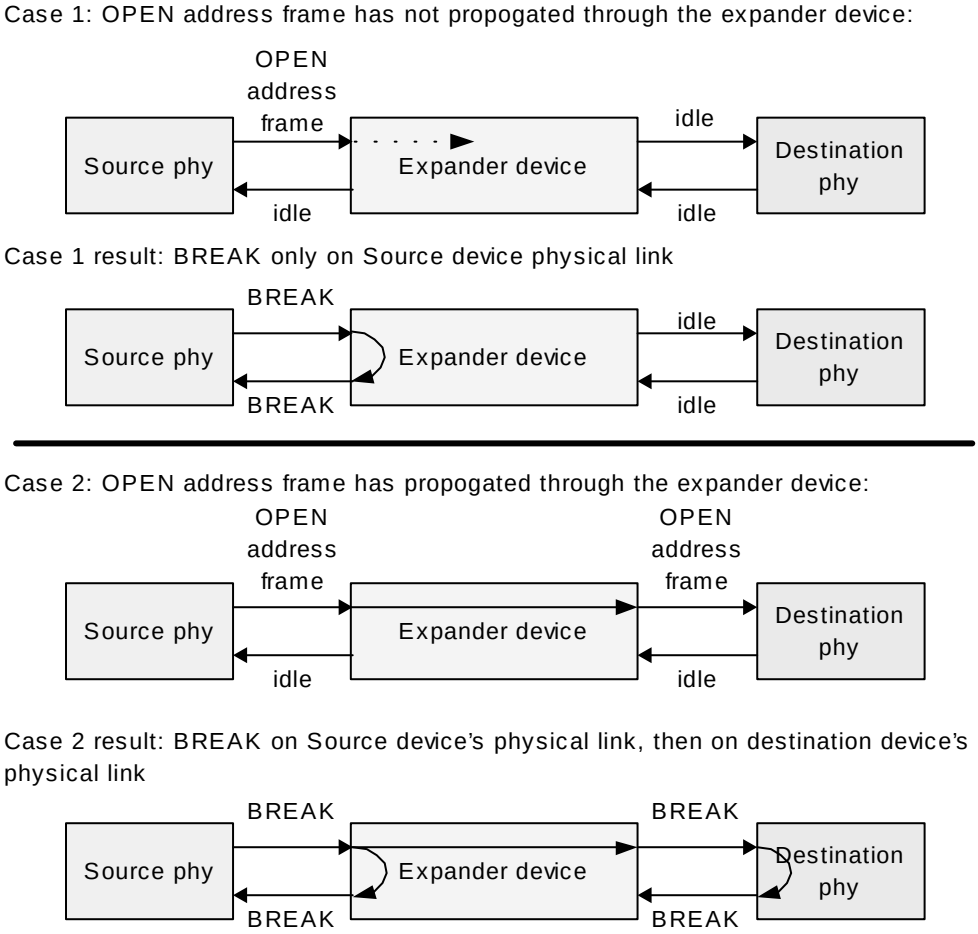


Figure 25. Aborting a connection request with BREAK

Figure 26 shows the sequence for a connection request where the Open Timeout timer expires.

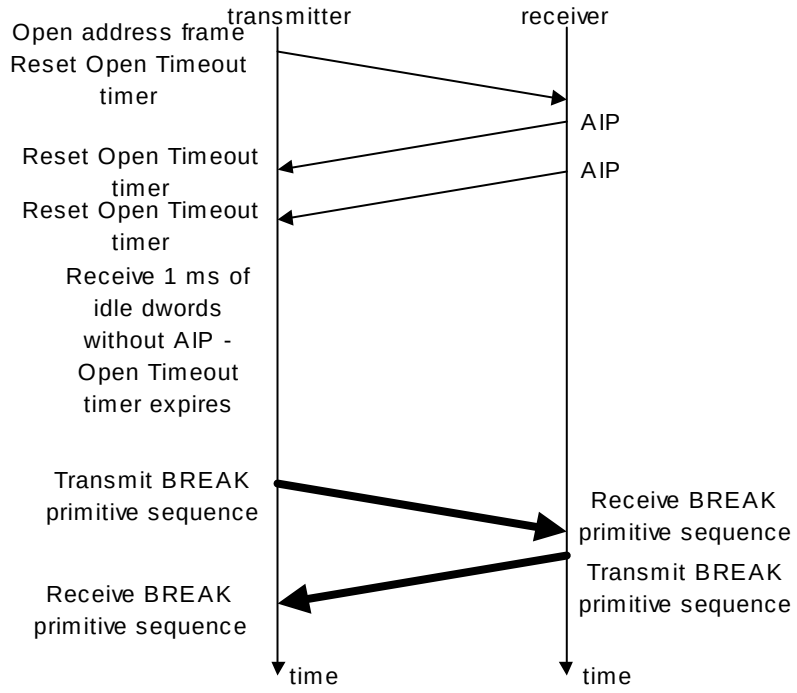


Figure 26. Connection request timeout example

4.9.1.7 Closing a connection

CLOSE is used to close a connection. See 4.9.3.6 for details on closing SSP connections.

After transmitting CLOSE, the source phy initializes a Close Timeout timer to 1 ms and start the Close Timeout timer.

Table 44 lists the responses to a CLOSE being transmitted.

Table 44. Close connection responses

Response	Description
Receive CLOSE	This confirms that the connection has been closed.
Close Timeout timer expires	The originating phy attempts to break the connection (see 4.9.1.8).

No additional dwords for the connection follow the CLOSE. Expanders closes the full-duplex connection upon detecting a CLOSE in each direction.

When a phy has both transmitted and received CLOSE, it considers the connection closed.

Figure 27 shows example sequences for closing a connection.

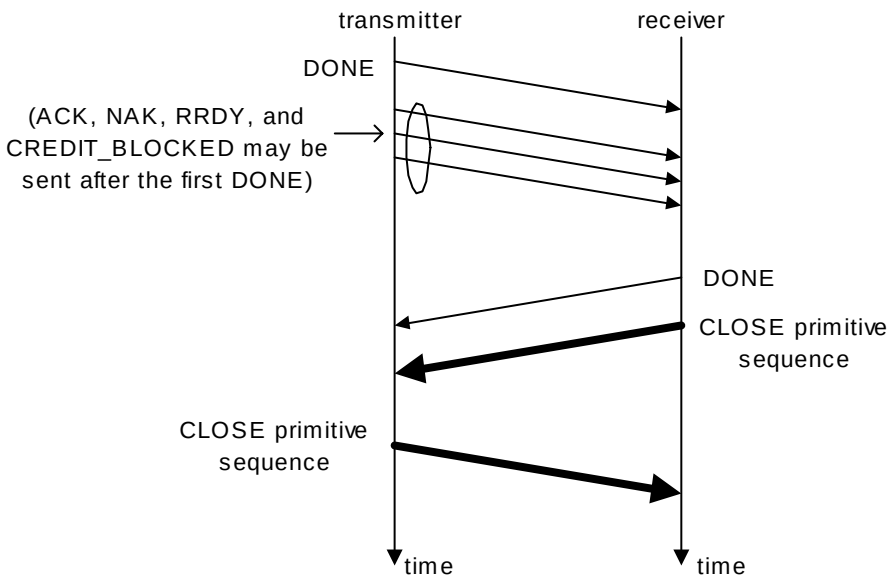


Figure 27. Closing a connection example

4.9.1.8 Breaking a connection

In addition to aborting a connection request, BREAK may also be used to break a connection, in cases where CLOSE is not available. After transmitting BREAK, the originating phy ignores all incoming dwords except for BREAKs.

After transmitting BREAK, the source phy initializes a Break Timeout timer to 1 ms and start the Break Timeout timer.

Table 45 lists the responses to a BREAK being transmitted after a connection has been established.

Table 45. Break connection responses

Response	Description
Receive BREAK	This confirms that the connection has been broken.
Break Timeout timer expires	The originating phy assumes the connection has been broken. The originating phy may perform a link reset sequence.

In addition to a BREAK, a connection is considered broken due to loss of dword synchronization (see 3.6).

In addition to the actions described in this subclause and in 4.9.1.6, the following are the responses by an SSP phy to a broken connection:

a. Received frames having no CRC error may be considered valid regardless of whether an ACK has been transmitted in response to the frame prior to the broken connection;

b. Transmitted frames for which an ACK has been received prior to a broken connection are considered successfully transmitted; and

c. Transmitted frames for which an ACK or NAK has not been received prior to a broken connection are considered not successfully transmitted.

4.9.2 Rate matching

Each successful connection request contains the connection rate (see 2.1.10) of the pathway.

Every phy in the physical link inserts ALIGNs or NOTIFYs between dwords to match the connection rate. Phys receiving ALIGNs and NOTIFYs delete them regardless of whether the ALIGNs and NOTIFYs were inserted for clock skew management (see 4.3) or for rate matching.

The faster phy rotates between ALIGN (0), ALIGN (1), ALIGN (2), and ALIGN (3) to reduce long strings of repeated patterns appearing on the physical link. NOTIFYs may be used to replace ALIGNs (see 4.2.5.9).

Figure 28 shows an example of rate matching between a 3.0 Gbps source phy and a 3.0 Gbps destination phy, with an intermediate 1.5 Gbps physical link in between.

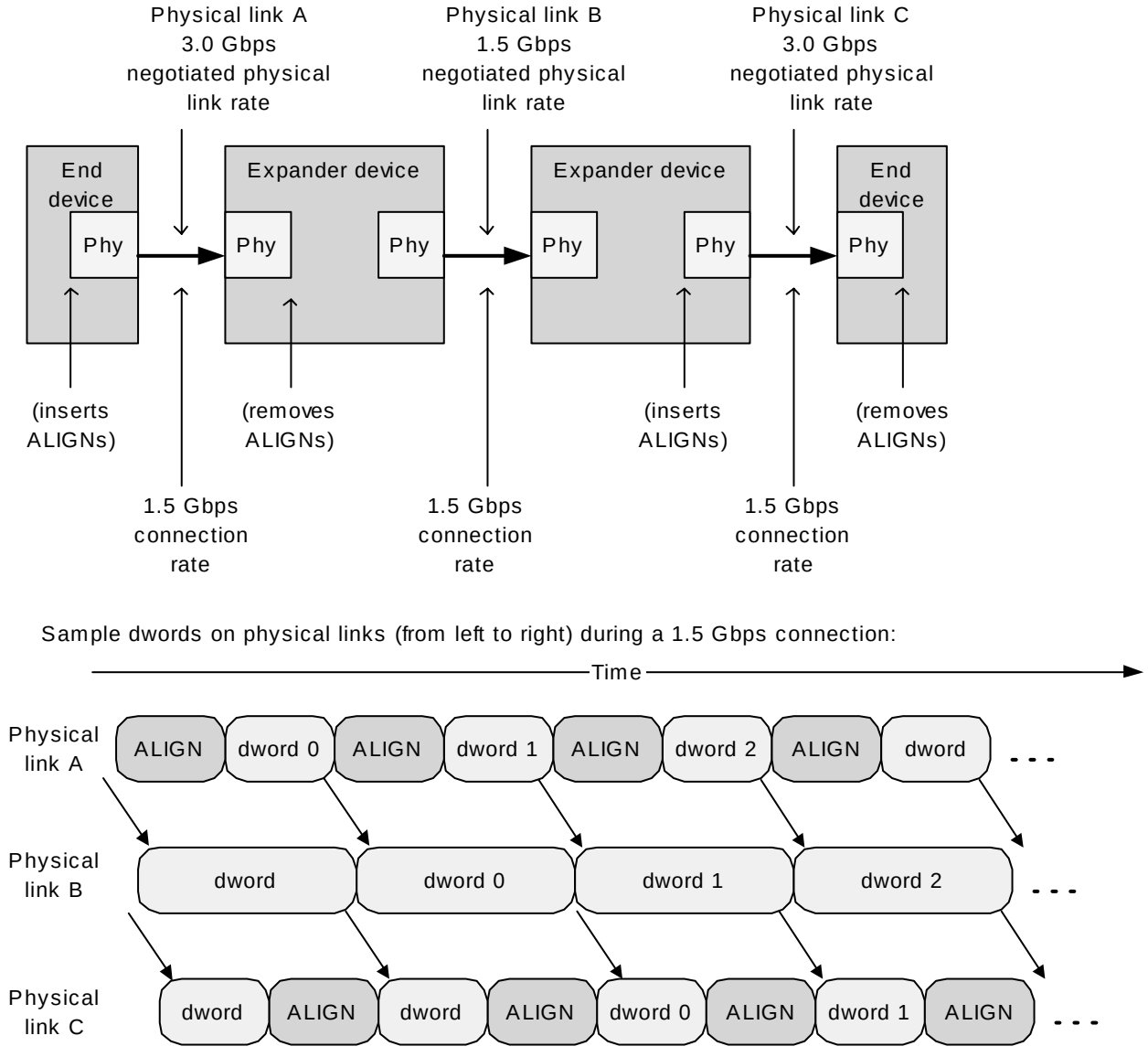


Figure 28. Rate matching example

A phy starts inserting ALIGNs and NOTIFYs for rate matching at the selected connection rate with the first dword following:

a. transmitting the EOAF for an OPEN address frame; or

b. transmitting an OPEN_ACCEPT.

The source phy transmits idle dwords including ALIGNs and NOTIFYs at the selected connection rate while waiting for the connection response. This enables each expander device to start forwarding dwords from the source phy to the destination phy after forwarding an OPEN_ACCEPT.

A phy stops inserting ALIGNs and NOTIFYs for rate matching after:

a. transmitting the first dword in a CLOSE;

b. transmitting the first dword in a BREAK;

c. receiving an OPEN_REJECT for a connection request; or

d. losing arbitration to a received OPEN address frame.

4.9.3 SSP link layer

4.9.3.1 Opening an SSP connection

When a drive accepts an OPEN address frame, it transmit at least one RRDY in that connection within 1 ms of transmitting an OPEN_ACCEPT. If the drive is not able to grant credit, it responds with OPEN_REJECT (RETRY) and does not accept the connection request.

4.9.3.2 Full duplex

SSP is a full duplex protocol. An SSP phy may receive an SSP frame or primitive in a connection the same time it is transmitting an SSP frame or primitive in the same connection. A wide SSP port may send and/or receive SSP frames or primitives concurrently on different connections (i.e., on different phys).

When a connection is open and an SSP phy has no more SSP frames to transmit on that connection, it transmits a DONE to start closing the connection. The other direction may still be active, so the DONE may be followed by one or more CREDIT_BLOCKED, RRDY, ACK, or NAKs.

4.9.3.3 SSP frame transmission and reception

During an SSP connection, SSP frames are preceded by SOF and followed by EOF as shown in figure 29.

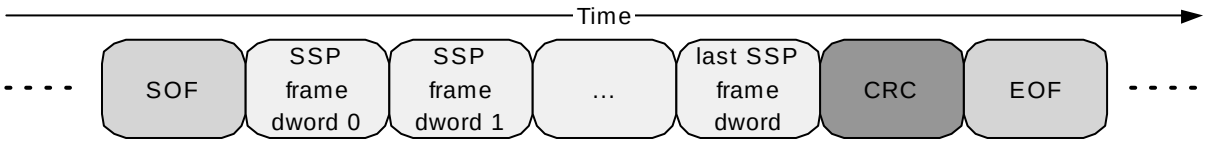


Figure 29. SSP frame transmission

The last data dword after the SOF prior to the EOF always contains a CRC (see 4.5). The drive checks received frames for being too short, too long and that the CRC is valid. Frames that are less than 7 data dwords long or greater than 263 data dwords long are discarded.

The drive acknowledges received SSP frames within 1 ms if not discarded with either a positive acknowledgement (ACK) or a negative acknowledgement (NAK). ACK means the SSP frame was received into a frame buffer without errors. NAK (CRC ERROR) means the SSP frame was received with a CRC error.

If the drive receives NAK in response to any Data or XFR_RDY frame, it will terminate the command associated with the NAKed frame with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED. If the drive receives NAK for a Response frame, it will retry sending the frame one time, with the RETRANSMIT bit set to zero.

4.9.3.4 SSP flow control

SSP phys grant credit for permission to transmit frames with RRDY. Each RRDY increments credit by one frame. Frame transmission decrements credit by one frame. Credit of zero frames is established at the beginning of each connection.

The drive port sends a maximum of 2RRDYs. The drive port accepts up to 255 RRDYs.

To prevent deadlocks where an SSP initiator port and SSP target port are both waiting on each other to provide credit, an SSP initiator port is required to send at least one RRDY if it enters a connection either by sending OPEN_ACCEPT or receiving an OPEN_ACCEPT for a connection request it originated. It may refuse to provide credit by rejecting connection requests for other reasons (e.g., temporary buffer full conditions).

An SSP target port may refuse to provide credit for any reason, including because it needs to transmit a frame itself.

4.9.3.5 Interlocked frames

Table 46 shows indicates which SSP frames are interlocked and which are non-interlocked.

Table 46. SSP frame interlock requirements

SSP frame type	Interlock requirement
COMMAND	INTERLOCKED
TASK	INTERLOCKED
XFER_RDY	INTERLOCKED
DATA	NON-INTERLOCKED
RESPONSE	INTERLOCKED

Before transmitting an interlocked frame, an SSP phy waits for all SSP frames to be acknowledged with ACK or NAK, even if credit is available. After transmitting an interlocked frame, an SSP phy does not transmit another SSP frame until it has been acknowledged with ACK or NAK, even if credit is available.

Before transmitting a non-interlocked frame, an SSP phy waits for:

a. all non-interlocked frames with different tags; and

b. all interlocked frames;

to be acknowledged with ACK or NAK, even if credit is available.

After transmitting a non-interlocked frame, an SSP phy may transmit another non-interlocked frame with the same tag if credit is available. The phy does not transmit:

a. a non-interlocked frame with a different tag; or

b. an interlocked frame;

until all SSP frames have been acknowledged with ACK or NAK, even if credit is available.

Interlocking does not prevent transmitting and receiving interlocked frames simultaneously (e.g., an SSP initiator phy could be transmitting a COMMAND frame while receiving XFER_RDY, DATA, or RESPONSE frames for a different command).

An SSP phy may transmit primitives responding to traffic it is receiving (e.g. an ACK or NAK to acknowledge an SSP frame, an RRDY to grant more receive credit, or a CREDIT_BLOCKED to indicate credit is not forthcoming) while waiting for an interlocked frame it transmitted to be acknowledged. These primitives may also be interspersed within an SSP frame.

Figure 30 shows an example of interlocked frame transmission.

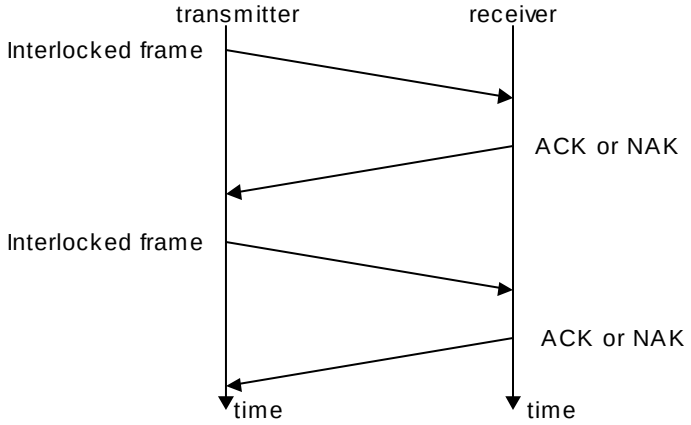


Figure 30. Interlocked frames

Figure 31 shows an example of non-interlocked frame transmission with the same tags.

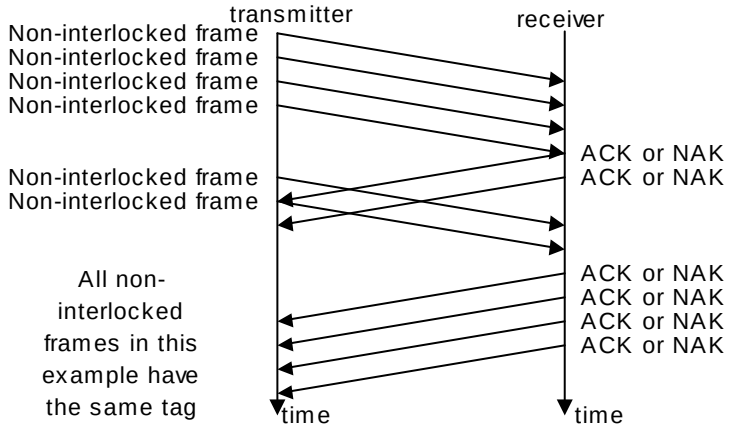


Figure 31. Non-interlocked frames with the same tag

Figure 32 shows an example of non-interlocked frame transmission with different tags.

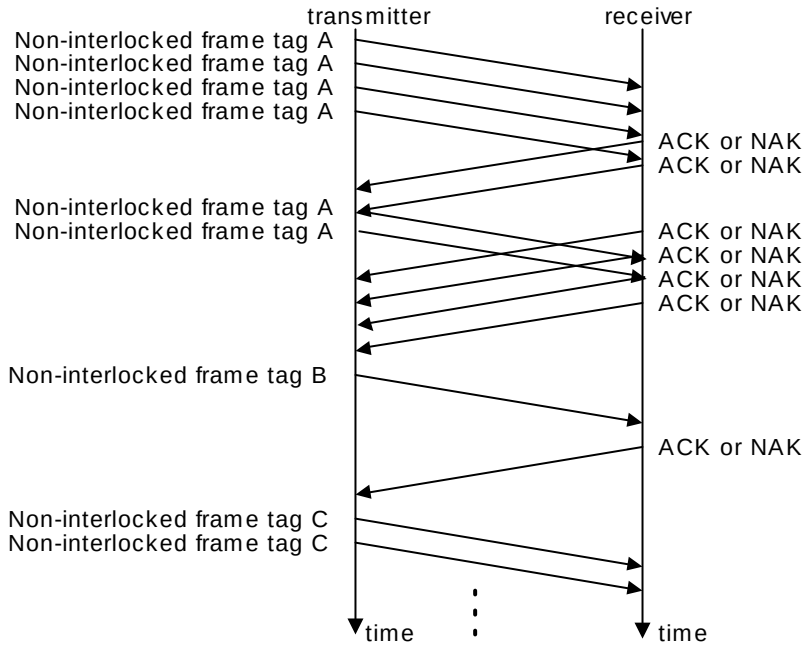


Figure 32. Non-interlocked frames with different tags

4.9.3.6 Closing an SSP connection

DONE is exchanged prior to closing an SSP connection. There are several versions of the DONE primitive indicating additional information about why the SSP connection is being closed:

a. DONE (NORMAL) indicates normal completion; the transmitter has no more SSP frames to transmit;

b. DONE (CREDIT TIMEOUT) indicates the transmitter still has SSP frames to transmit, but did not receive an RRDY granting frame credit within 1 ms; and

c. DONE (ACK/NAK TIMEOUT) indicates the transmitter transmitted an SSP frame but did not receive the corresponding ACK or NAK within 1 ms. As a result, the ACK/NAK count is not balanced and the transmitter is going to transmit a BREAK in 1 ms unless the recipient replies with DONE and the connection is closed.

After transmitting DONE, the transmitting phy initializes and starts a 1 ms DONE Timeout timer.

After transmitting DONE, the transmitting phy shall not transmit any additional frames during this connection. However, the phy may transmit ACK, NAK, RRDY, and CREDIT_BLOCKED after transmitting DONE if the other phy is still transmitting SSP frames in the reverse direction. Once an SSP phy has both transmitted and received DONE, it closes the connection by transmitting CLOSE (NORMAL) (see 4.9.1.7).

Figure 33 shows the sequence for a closing an SSP connection.

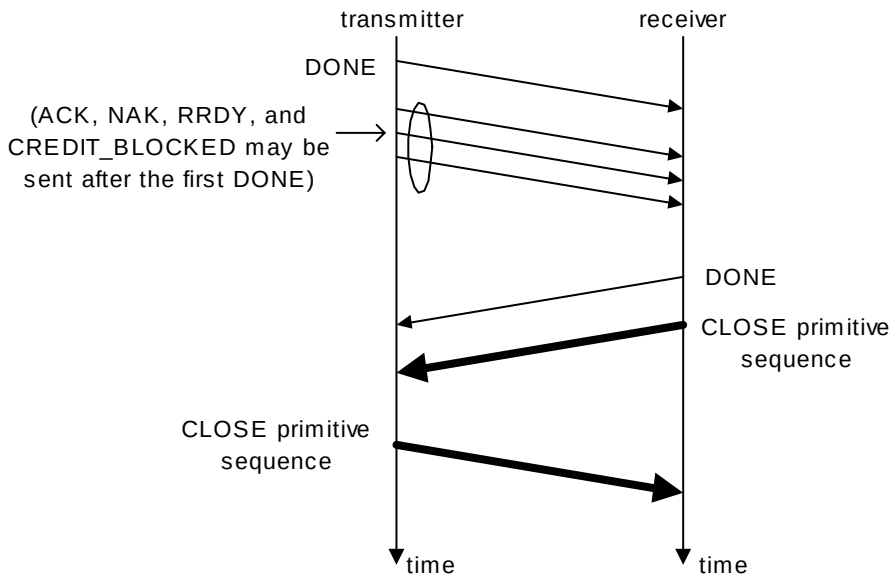


Figure 33. Closing an SSP connection example

5.0 Transport layer

5.1 Transport layer overview

The transport layer defines frame formats. Transport layer interfaces to the application layer and port layer and construct and parses frame contents. For SSP, the transport layer only receives frames from the port layer that are going to be ACKed by the link layer.

5.2 SSP transport layer

5.2.1 SSP frame format

Table 47 defines the SSP frame format.

Table 47. SSP frame format

Byte\Bit	7	6	5	4	3	2	1	0
0	FRAME TYPE							
1	(MSB)	HASHED DESTINATION SAS ADDRESS						
3								(LSB)
4	Reserved							
5	(MSB)	HASHED SOURCE SAS ADDRESS						
7								(LSB)
8	Reserved							
9	Reserved							
10	Reserved					RETRY DATA FRAMES	RETRANSMIT	CHANGING DATA POINTER
11	Reserved						NUMBER OF FILL BYTES	
12	Reserved							
13	Reserved							
15								

Table 47. SSP frame format

Byte\Bit	7	6	5	4	3	2	1	0
16	(MSB)			TAG				
17								(LSB)
18	(MSB)			TARGET PORT TRANSFER TAG				
19								(LSB)
20	(MSB)			DATA OFFSET				
23								(LSB)
24				INFORMATION UNIT				
m								
	Fill bytes, if needed							
n - 3	(MSB)			CRC				
n								(LSB)

Table 48 defines the FRAME TYPE field, which defines the format of the INFORMATION UNIT field.

Table 48. FRAME TYPE **field**

Code	Name of frame	Information unit	Originator	Information unit size (bytes)	Reference
01h	DATA frame	Data	SSP initiator port or SSP target port	1 to 1 024	5.2.2.4
05h	XFER_RDY frame	Transfer ready	SSP target port	12	5.2.4.3
06h	COMMAND frame	Command	SSP initiator port	28 to 284	5.2.4.1
07h	RESPONSE frame	Response	SSP target port	24 to 1 024	5.2.4.5
16h	TASK frame	Task management function	SSP initiator port	28	5.2.4.2
F0h - FFh	Vendor specific				
All others	Reserved				

The HASHED DESTINATION SAS ADDRESS field contains the hashed value (see 2.2.3) of the destination SAS address. The drive port compares the HASHED DESTINATION SAS ADDRESS field in received frames with its hashed value of the port's Port Identifier. If the compare fails, the drive discards the frame.

The HASHED SOURCE SAS ADDRESS field contains the hashed value of the source SAS address. The drive port compares the HASHED SOURCE SAS ADDRESS field in received frames with the SOURCE SAS ADDRESS received in the OPEN address frame for the connection. If the compare fails, the drive discards the frame. If the HASHED SOURCE SAS ADDRESS field doesn't match for a Command or Task frame, the drive will send a Response frame with DATAPRES=RESONSE DATA, and a Response code of INVALID FRAME.

The RETRY DATA FRAMES bit is set to one for XFER_RDY frames under the conditions defined in 5.2.4 and shall be set to zero for all other frame types. When set to one this bit specifies that the SSP initiator port may retry write DATA frames that fail.

The RETRANSMIT bit is set to one for RESPONSE frames under certain conditions (see 5.2.4.5) and shall be set to zero for all other frame types. This bit indicates the frame is a retransmission after the SSP target port failed in its previous attempt to transmit the frame.

The CHANGING DATA POINTER bit is set to one for DATA frames under the conditions defined in 5.2.4 and shall be set to zero for all other frame types. When set to one this bit specifies that the frame is a retransmission after the SSP target port failed in its previous attempt to transmit the frame or a subsequent frame and the DATA OFFSET field of the frame may not be sequentially increased from that of the previous frame.

The NUMBER OF FILL BYTES field indicates the number of fill bytes between the INFORMATION UNIT field and the CRC field. The NUMBER OF FILL BYTES field shall be set to zero for all frame types except DATA frames (i.e., all other frame types are already four-byte aligned).

The TAG field contains a value that allows the SSP initiator port to establish a context for commands and task management functions.

For COMMAND and TASK frames, the SSP initiator port shall set the TAG field to a value that is unique for the I_T nexus established by the connection (see 4.9.1). An SSP initiator port shall not reuse the same tag when transmitting COMMAND or TASK frames to different LUNs in the same SSP target port; it may reuse a tag when transmitting frames to different SSP target ports. The TAG field in a COMMAND frame contains the tag defined in SAM-4. The TAG field in a TASK frame does not correspond to a SAM-4 tag, but corresponds to an SAM-4 association. The tag space used in the TAG fields is shared across COMMAND and TASK frames (e.g., if a tag is used for a COMMAND frame, it is not simultaneously used for a TASK frame).

For DATA, XFER_RDY, and RESPONSE frames, the drive sets the TAG field to the tag of the command or task management function to which the frame pertains.

The TARGET PORT TRANSFER TAG field is used by the drive port to establish a write data context when receiving DATA frames. The drive port sets the field in every XFER_RDY frame to a value that is unique for the L_Q portion of the I_T_L_Q nexus. The drive may set the Target Port Transfer Tag to any value when transmitting other frames.

SSP initiator ports set the TARGET PORT TRANSFER TAG field as follows:

a. For each DATA frame that is sent in response to a XFER_RDY frame, the SSP initiator port sets the TARGET PORT TRANSFER TAG field to the value that was in the corresponding XFER_RDY frame;
b. For each DATA frame that is sent containing first burst data (see 5.2.2.4), the SSP initiator port sets the TARGET PORT TRANSFER TAG field to FFFFh; and
c. For frames other than DATA frames, the SSP initiator port sets the TARGET PORT TRANSFER TAG field to FFFFh.

For DATA frames, the DATA OFFSET field is described in 5.2.2.4. For all other frame types, the DATA OFFSET field is ignored.

The INFORMATION UNIT field contains the information unit, the format of which is defined by the FRAME TYPE field. The maximum size of the INFORMATION UNIT field is 1,024 bytes, making the maximum size of the frame 1,052 bytes (1,024 bytes of data + 24 bytes of header + 4 bytes of CRC).

Fill bytes shall be included after the INFORMATION UNIT field so the CRC field is aligned on a four byte boundary. The number of fill bytes are indicated by the NUMBER OF FILL BYTES field. The contents of the fill bytes are vendor specific.

The CRC field contains a CRC value (see 4.5) that is computed over the entire SSP frame prior to the CRC field including the fill bytes (i.e., all data dwords between the SOF and EOF). The CRC field is checked by the link layer (see 4.9.3), not the transport layer.

5.2.2 Information units

5.2.2.1 COMMAND information unit

Table 49 defines the command IU. The COMMAND frame is sent by an SSP initiator port to request that a command be processed by a device server in a logical unit.

Table 49. COMMAND information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)							
7				LOGICAL UNIT NUMBER				(LSB)
8	Reserved							
9	ENABLE FIRST BURST	TASK PRIORITY				TASK ATTRIBUTE		
10	Reserved							
11	ADDITIONAL CDB LENGTH (**n dwords**)						Reserved	
12								
27				CDB				
28								
27+n×4				ADDITIONAL CDB BYTES				

The LOGICAL UNIT NUMBER field contains the address of the logical unit. Drives supported by this manual implement only logical unit zero. Except for Inquiry, Request Sense, and Reports LUNs, the drive returns a check condition, Illegal Request LUN Not Supported if the LOGICAL UNIT NUMBER field is not zero.

The TASK ATTRIBUTE field is defined in table 50. The drive processes all commands as tagged.

Table 50. TASK ATTRIBUTE **field**

Code	Task attribute	Description
000b	SIMPLE	Requests that the task be managed according to the rules for a simple task attribute (see SAM-4).
001b	HEAD OF QUEUE	Requests that the task be managed according to the rules for a head of queue task attribute (see SAM-4).
010b	ORDERED	Requests that the task be managed according to the rules for an ordered task attribute (see SAM-4).
011b	Reserved	
100b	ACA	Requests that the task be managed according to the rules for an automatic contingent allegiance task attribute (see SAM-4).
101b-111b	Reserved	

The ADDITIONAL CDB LENGTH field contains the length in dwords (four bytes) of the ADDITIONAL CDB field.

The CDB and ADDITIONAL CDB BYTES fields together contain the CDB to be interpreted by the addressed logical unit. Any bytes between the end of the CDB and the end of the two fields are ignored (e.g., a six-byte CDB occupies the first six bytes of the CDB field; the remaining ten bytes are ignored; and the ADDITIONAL CDB BYTES field is not present).

The contents of the CDB are defined in the SCSI command specifications (e.g., SPC-4).

5.2.2.2 TASK information unit

Table 51 defines the task management function IU. The TASK frame is sent by an SSP initiator port to request that a task management function be processed by a task manager in a logical unit.

Table 51. TASK information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)							
7				LOGICAL UNIT NUMBER				(LSB)
8				Reserved				
9				Reserved				
10				TASK MANAGEMENT FUNCTION				
11				Reserved				
12	(MSB)							
13				TAG OF TASK TO BE MANAGED				(LSB)
14								
27				Reserved				

The LOGICAL UNIT NUMBER field contains the address of the logical unit. The structure of the logical unit number field shall be as defined in SAM-4. If the addressed logical unit does not exist, the task manager returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and its RESPONSE CODE field set to INVALID LOGICAL UNIT.

Table 52 defines the TASK MANAGEMENT FUNCTION field.

Table 52. Task management functions

Code	Task management function	Uses LOGICAL UNIT NUMBER field	Uses TAG OF TASK TO BE MANAGED field	Description
01h	ABORT TASK	yes	yes	The task manager shall perform the ABORT TASK task management function with L set to the value of the LOGICAL UNIT NUMBER field and Q set to the value of the TAG OF TASK TO BE MANAGED field (see SAM-4).
02h	ABORT TASK SET	yes	no	The task manager shall perform the ABORT TASK SET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-4).

Table 52. Task management functions

Code	Task management function	Uses LOGICAL UNIT NUMBER field	Uses TAG OF TASK TO BE MANAGED field	Description
04h	CLEAR TASK SET	yes	no	The task manager shall perform the CLEAR TASK SET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-4).
08h	LOGICAL UNIT RESET	yes	no	The task manager shall perform the LOGICAL UNIT RESET task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-4).
20h	Reserved [a]			
40h	CLEAR ACA	yes	no	The task manager shall perform the CLEAR ACA task management function with L set to the value of the LOGICAL UNIT NUMBER field (see SAM-4).
80h	QUERY TASK	yes	yes	The task manager shall perform the QUERY TASK task management function with L set to the value of the LOGICAL UNIT NUMBER field and Q set to the value of the TAG OF TASK TO BE MANAGED field (see SAM-4).
All others	Reserved			

[a] The TARGET RESET task management function defined in SAM-4 is not supported.

If TASK MANAGEMENT FUNCTION contains a reserved or unsupported value, the task manager returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and its RESPONSE CODE field set to TASK MANAGEMENT FUNCTION NOT SUPPORTED.

If TASK MANAGEMENT FUNCTION is set to ABORT TASK or QUERY TASK, the TAG OF TASK TO BE MANAGED field specifies the TAG value from the COMMAND frame that contained the task to be aborted or checked. For all other task management functions, the TAG OF TASK TO BE MANAGED field is ignored.

5.2.2.3 XFER_RDY information unit

Table 53 defines the transfer ready IU. The XFER_RDY frame is sent by an SSP target port to request write data from the SSP initiator port.

Table 53. XFER_RDY information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)	REQUESTED OFFSET						
3								(LSB)
4	(MSB)	WRITE DATA LENGTH						
7								(LSB)
8		Reserved						
11								

The REQUESTED OFFSET field contains the application client buffer offset of the segment of write data the SSP initiator port may transmit to the logical unit (using DATA frames). The requested offset is a multiple of four (i.e., each DATA frame shall begin transferring data on a dword boundary). The drive sets the REQUESTED OFFSET field to zero for the first XFER_RDY frame of a command.

The WRITE DATA LENGTH field contains the number of bytes of write data the SSP initiator port may transmit to the logical unit (using DATA frames) from the application client buffer starting at the requested offset. The drive sets the WRITE DATA LENGTH field to a value greater than or equal to 00000001h. If the value in the MAXIMUM BURST SIZE field in the Disconnect-Reconnect mode page is not zero, the drive sets the WRITE DATA LENGTH field to a value less than or equal to the value in the MAXIMUM BURST SIZE field.

Only the last XFER_RDY frame for a command may contain a WRITE DATA LENGTH field that is not divisible by four.

5.2.2.4 DATA information unit

Table 54 defines the data IU. The DATA frame is sent by an SSP initiator port to deliver write data and is sent by an SSP target port to deliver read data. The maximum size of the data IU is the maximum size of any IU in an SSP frame (see 5.2.1). The minimum size of the data IU is one byte.

Table 54. DATA information unit

Byte\Bit	7	6	5	4	3	2	1	0
0		DATA						
n								

The DATA field contains the read or write data.

An SSP initiator port shall only transmit a DATA frame to the drive in response to an XFER_RDY frame. The drive discards received data frames without an associated outstanding XFER_RDY.

If the value in the MAXIMUM BURST SIZE field on the Disconnect-Reconnect mode page is not zero, the maximum amount of data that is transferred at one time by the drive per I_T_L_Q nexus is limited by the value in the MAXIMUM BURST SIZE field.

The DATA frame only contains write data for a single XFER_RDY frame.

Only the last read DATA frame for a command may contain fill bytes.

An SSP initiator port may set the NUMBER OF FILL BYTES field to a non-zero value in the last DATA frame that it transmits in response to a XFER_RDY. An SSP initiator port sets the NUMBER OF FILL BYTES field in the frame header (see 5.2.1) to zero in all other DATA frames that it transmits.

An SSP initiator port does not transmit a DATA frame for a given I_T_L_Q nexus after it has sent a TASK frame that terminates that task (e.g., an ABORT TASK).

The DATA OFFSET field in the frame header (see 5.2.1) contains the application client buffer offset as described by SAM-4. The data offset shall be a multiple of four (i.e., each DATA frame shall transfer data beginning on a dword boundary).

The initial read DATA frame for a given command is set the DATA OFFSET field to zero. If any additional read DATA frames are required, the DATA OFFSET field is set to the value of the previous read DATA frame's data offset plus the previous read DATA frame's data length.

The initial write DATA frame for a given command is set the DATA OFFSET field to zero. If any additional write DATA frames are required, the DATA OFFSET field is set to the value of the previous write DATA frame's data offset plus the previous write DATA frame's data length.

5.2.2.5 RESPONSE information unit

5.2.2.5.1 RESPONSE information unit overview

Table 55 defines the response IU. The RESPONSE frame is sent by an SSP target port to deliver SCSI status (e.g., GOOD or CHECK CONDITION) and sense data, or to deliver SSP-specific status (e.g., illegal frame format). The maximum size of the RESPONSE frame is the maximum size of any IU in an SSP frame (see 5.2.1).

Table 55. RESPONSE information unit

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
7								
8	(MSB)	RETRY DELAY TIMER						
9								(LSB)
10	Reserved							DATAPRES
11	STATUS							
12	Reserved							
15								
16	(MSB)	SENSE DATA LENGTH (**n bytes**)						
19								(LSB)
20	(MSB)	RESPONSE DATA LENGTH (**m bytes**)						
23								(LSB)

Table 55. RESPONSE information unit

Byte\Bit	7	6	5	4	3	2	1	0
24	RESPONSE DATA (if any)							
23+m								
24+m	SENSE DATA (if any)							
23+m+n								

The RETRY DELAY TIMER field contains the retry delay timer code (see SAM-3).

Table 56 defines the DATAPRES field, which indicates the format and content of the STATUS field, SENSE DATA LENGTH field, RESPONSE DATA LENGTH field, RESPONSE DATA field, and SENSE DATA field.

Table 56. DATAPRES field

Code	Name	Description	Reference
00b	NO DATA	No data present	5.2.2.5.2
01b	RESPONSE_DATA	Response data present	5.2.2.5.3
10b	SENSE_DATA	Sense data present	5.2.2.5.4
11b	Reserved		

The drive returns a RESPONSE frame with the DATAPRES field set to NO_DATA if a command completes without sense data to return.

The drive returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA in response to every TASK frame and in response to errors that occur while the transport layer is processing a COMMAND frame.

The drive returns a RESPONSE frame with the DATAPRES field set to SENSE_DATA if a command completes with sense data to return (e.g., CHECK CONDITION status).

If the DATAPRES field is set to a reserved value, then the SSP initiator port discards the RESPONSE frame.

5.2.2.5.2 RESPONSE information unit NO_DATA format

If the DATAPRES field is set to NO_DATA, then:

a. the STATUS field contains the status code for a command that has ended (see SAM-4 for a list of status codes);

b. the SENSE DATA LENGTH field and the RESPONSE DATA LENGTH field are set to zero and ignored by the SSP initiator port; and

c. the SENSE DATA field and the RESPONSE DATA field are not present.

5.2.2.5.3 RESPONSE information unit RESPONSE_DATA format

If the DATAPRES field is set to RESPONSE_DATA, then:

a. the STATUS field and the SENSE DATA LENGTH field are set to zero and ignored by the SSP initiator port;

b. the SENSE DATA field is not present;

c. the RESPONSE DATA LENGTH field is set to four; and

d. the RESPONSE DATA field is present.

Table 57 defines the RESPONSE DATA field, which contains information describing protocol failures detected during processing of a request received by the drive. The RESPONSE DATA field is present if the drive detects any of the conditions described by a non-zero RESPONSE CODE value and is present for a RESPONSE frame sent in response to a TASK frame.

Table 57. RESPONSE DATA **field**

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved							
3	RESPONSE CODE							

Table 58 defines the RESPONSE CODE field, which indicates the error condition or the completion status of a task management function.

Table 58. RESPONSE CODE **field**

Code	Description
00h	TASK MANAGEMENT FUNCTION COMPLETE [a]
02h	INVALID FRAME
04h	TASK MANAGEMENT FUNCTION NOT SUPPORTED [a]
05h	TASK MANAGEMENT FUNCTION FAILED [a]
08h	TASK MANAGEMENT FUNCTION SUCCEEDED [a]
09h	INVALID LOGICAL UNIT NUMBER [a]
0Ah	OVERLAPPED TAG ATTEMPTED [b]
All others	Reserved

[a] Only valid when responding to a TASK frame
[b] Returned in case of command/task management function or task management function/task management function tag conflicts.

5.2.2.5.4 RESPONSE information unit SENSE_DATA format

If the DATAPRES field is set to SENSE_DATA, then:

a. the STATUS field contains the status code for a command that has ended (see SAM-4 for a list of status codes);

b. the RESPONSE DATA LENGTH field is set to zero and ignored by the initiator;

c. the RESPONSE DATA field is not be present;

d. the SENSE DATA LENGTH field is set to a non-zero value indicating the number of bytes in the SENSE DATA field. The maximum SENSE DATA LENGTH is 1,000 (see table 48); and

e. the SENSE DATA field contains sense data (see SAM-4).

5.2.3 Sequences of SSP frames

Figure 34, figure 35, figure 36, and figure 37 show examples of the sequences of frames for single task management functions and commands. Frames may be interleaved in any order when multiple commands and/or task management functions are outstanding. Frames may be transmitted during one or more connections (e.g., the COMMAND frame could be transmitted in a connection originated by the SSP initiator port, and the DATA frames and RESPONSE frame transmitted in one or more connections originated by the SSP target port). RESPONSE frames may be returned in any order (i.e., the order in which TASK frames and COMMAND frames are sent has no effect on the order that RESPONSE frames are returned).

Figure 34 shows the sequence of SSP frames for a task management function, including the transport protocol services invoked by the SCSI application layer.

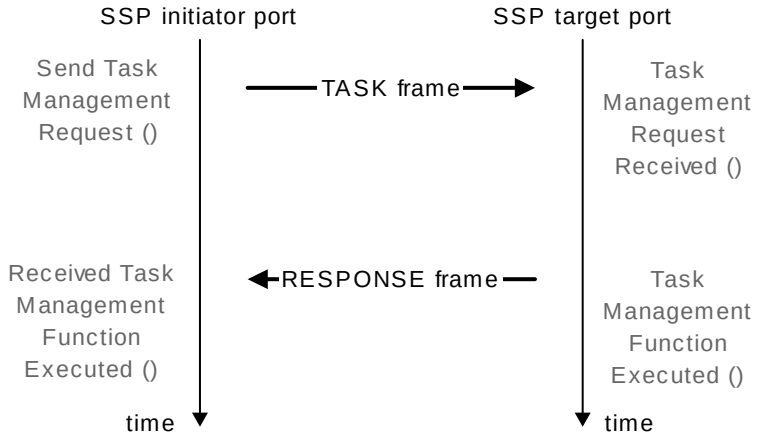


Figure 34. Task management function sequence of SSP frames

Figure 35 shows the sequence of SSP frames for a write command, including the transport protocol services invoked by the SCSI application layer.

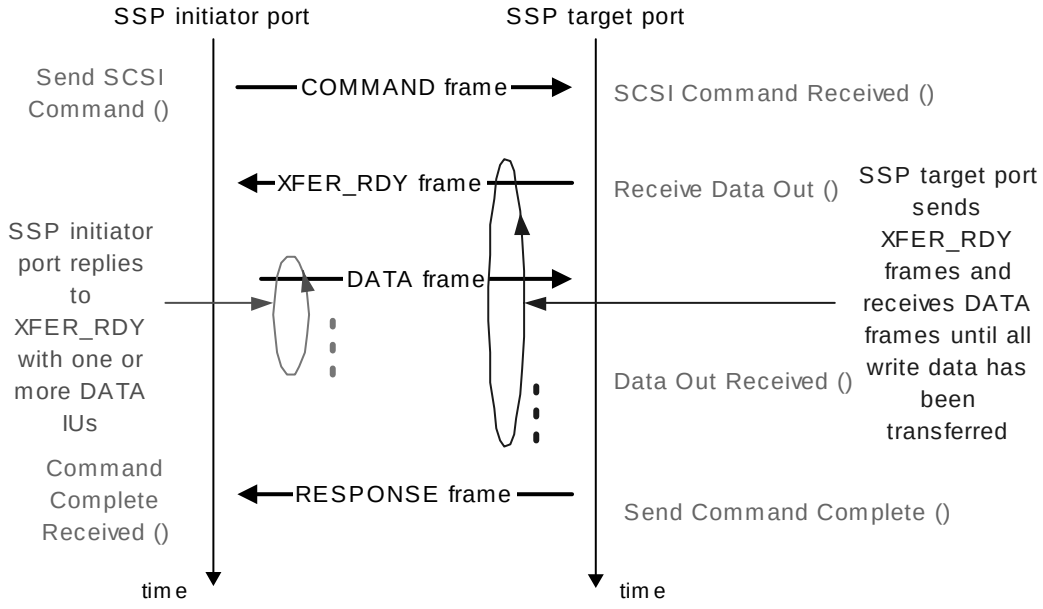


Figure 35. Write command sequence of SSP frames

Figure 36 shows the sequence of SSP frames for a read command, including the transport protocol services invoked by the SCSI application layer.

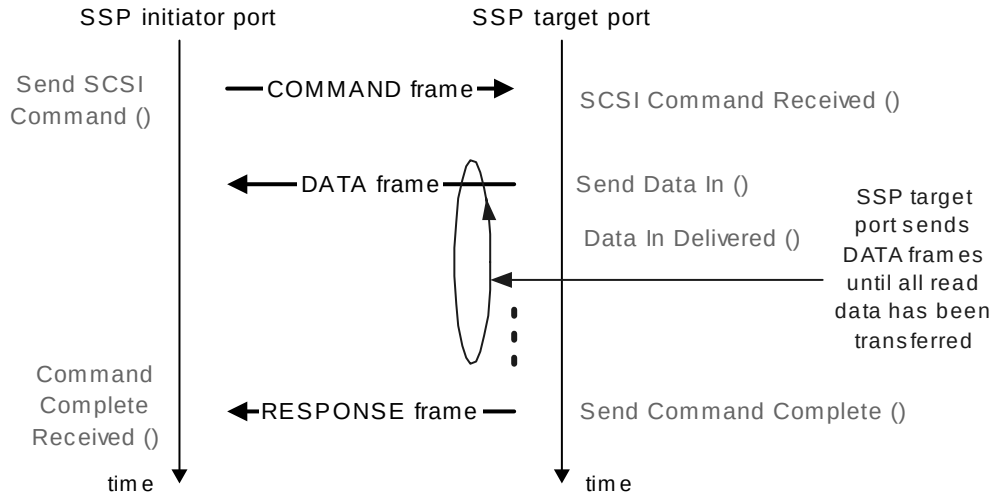


Figure 36. Read command sequence of SSP frames

Figure 37 shows the sequence of SSP frames for a bidirectional command, including the transport protocol services invoked by the SCSI application layer.

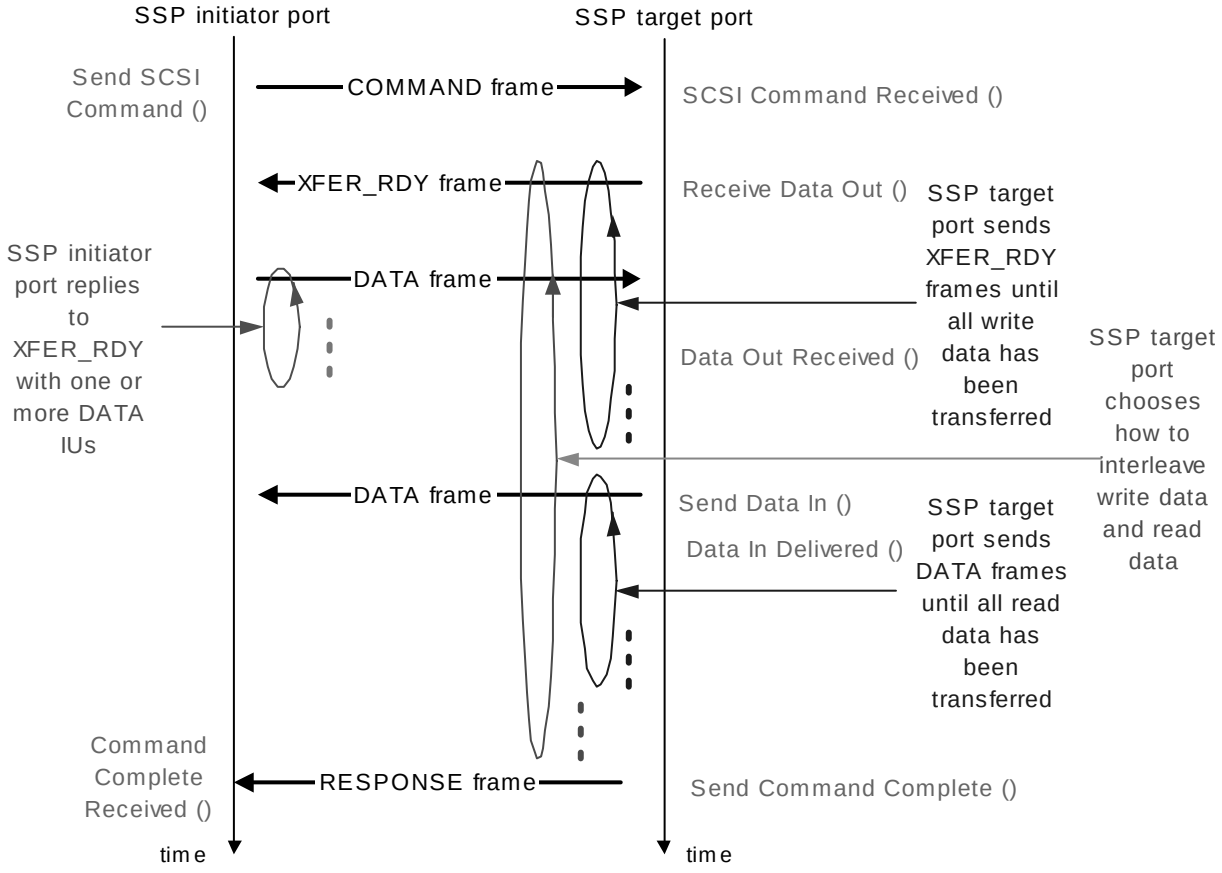


Figure 37. Bidirectional command sequence of SSP frames

5.2.4 SSP transport layer handling of link layer errors

5.2.4.1 COMMAND frame

If an SSP initiator port transmits a COMMAND frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken) it transmits a QUERY TASK task management function in the next connection to determine whether the command was received. If QUERY TASK returns a TASK MANAGEMENT FUNCTION SUCCEEDED response, the SSP initiator port assumes the COMMAND frame was ACKed. If QUERY TASK returns a TASK MANAGEMENT FUNCTION COMPLETE response, and a RESPONSE frame has not yet been received for that I_T_L_Q nexus, the SSP initiator port assumes the command was NAKed or lost and may reuse the tag.

5.2.4.2 TASK frame

If an SSP initiator port transmits a TASK frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken) it retransmits the TASK frame in a new connection.

5.2.4.3 XFER_RDY frame

If the drive transmits an XFER_RDY frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it closes the connection with DONE (ACK/NAK TIMEOUT) and returns a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of ACK/NAK TIMEOUT.

If an SSP target port transmits an XFER_RDY frame and receives a NAK, it shall return a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED.

5.2.4.4 DATA frame

If the drive transmits a DATA frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it closes the connection with DONE (ACK/NAK TIMEOUT) and returns a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of ACK/NAK TIMEOUT.

If the drive transmits a DATA frame and receives a NAK, it returns a CHECK CONDITION status for that command with a sense key of ABORTED COMMAND and an additional sense code of NAK RECEIVED.

If an SSP initiator port transmits a DATA frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it aborts the command.

If an SSP initiator port transmits a DATA frame and receives a NAK, it aborts the command.

5.2.4.5 RESPONSE frame

If the drive transmits a RESPONSE frame and does not receive an ACK or NAK (e.g., times out, or the connection is broken), it tries transmitting the RESPONSE frame again in a new connection. It retries one time. The RETRANSMIT bit is set to one.

If the drive transmits a RESPONSE frame and receives a NAK, it retries transmitting the RESPONSE frame one time. The RETRANSMIT bit is set to zero.

If an SSP initiator port receives a RESPONSE frame with a RETRANSMIT bit set to one, and it has previously received a RESPONSE frame for the same I_T_L_Q nexus, it discards the extra RESPONSE frame. If it has not previously received the RESPONSE frame, it treats it as the valid RESPONSE frame.

5.2.5 SSP transport layer error handling

5.2.5.1 SSP target port error handling

If drive receives an XFER_RDY frame or an unsupported frame type, it discards the frame.

If drive receives a COMMAND frame and:

a. the frame is too short to contain a LUN field;

b. the frames ADDITIONAL CDB LENGTH field indicates the frame should be a different length,

the drive returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID FRAME.

If drive receives a TASK frame that is too short, it returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID FRAME.

If the drive receives a COMMAND frame with a tag that is already in use, it returns a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of OVERLAPPED COMMANDS DETECTED.

If the drive receives a TASK frame with a tag that is already in use, it returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID FRAME.

If the drive receives a DATA frame with an unknown tag, it discards the frame.

If the drive receives a DATA frame that does not contain first burst data and for which there is no XFER_RDY frame outstanding, it discards the frame.

If the drive receives a TASK frame with an unknown logical unit number, it returns a RESPONSE frame with the DATAPRES field set to RESPONSE_DATA and the RESPONSE CODE set to INVALID LOGICAL UNIT.

The drive will ignore the Target Port Transfer Tag in COMMAND or TASK frames.

The drive will set the Target Port Transfer Tag in XFER_RDY frames. If the drive receives a DATA frame for which the Target Port Transfer Tag does not match that sent in the XFER_RDY, it will discard the frame.

If the drive receives a DATA frame with a data offset that was not expected, it discards that frame and any subsequent DATA frames received for that command and, shall terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of DATA OFFSET ERROR.

If the drive receives a DATA frame with more write data than expected (i.e., the length of the DATA frame extends past the end of the expected write data length), it discards the frame and terminates the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of TOO MUCH WRITE DATA.

If the drive receives a zero length DATA frame, it discards the frame and terminates the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code of INFORMATION UNIT TOO SHORT.

5.2.5.2 SSP initiator port error handling

If an SSP initiator port receives a COMMAND or TASK frame or an unsupported frame type, it discards the frame.

If an SSP initiator port receives a DATA, XFER_RDY, or RESPONSE frame with an unknown TAG field value (including a tag for which it has sent a COMMAND or TASK frame but not yet received an ACK), it discards the frame. It may then abort the command with that tag.

If an SSP initiator port receives an XFER_RDY frame that is not 12 bytes long, it discards the frame. It may then abort the command.

If an SSP initiator port receives an XFER_RDY frame in response to a command with no write data, it discards the frame. It then aborts the command.

If an SSP initiator port receives an XFER_RDY frame requesting more write data than expected, it aborts the command.

If an SSP initiator port receives an XFER_RDY frame requesting zero bytes, it aborts the command.

If an SSP initiator port receives an XFER_RDY frame with a requested offset that was not expected, it aborts the command.

If an SSP initiator port receives a DATA frame with more read data than expected, it discards the frame and abort the command. It may receive a RESPONSE for the command before being able to abort the command.

If an SSP initiator port receives a DATA frame with zero bytes, it discards the frame and aborts the command. It may receive a RESPONSE for the command before being able to abort the command.

If an SSP initiator port receives a DATA frame with a data offset that was not expected, it discards that frame and any subsequent DATA frames received for that command and abort the command. It may receive a RESPONSE for the command before being able to abort the command.

6.0 Application layer

6.1 SCSI application layer

6.1.1 SCSI transport protocol services

6.1.1.1 SCSI transport protocol services overview

An application client requests the processing of a SCSI command.

6.1.2 Device server error handling

If a device server calls Receive Data-Out () and receives a Delivery Result set to a value in table 59, it shall terminate the command with a CHECK CONDITION status with a sense key of ABORTED COMMAND and an additional sense code as indicated by table 59.

Table 59. Delivery Result to additional sense code mapping

Delivery Result	Additional sense code
DELIVERY FAILURE - DATA OFFSET ERROR	DATA OFFSET ERROR
DELIVERY FAILURE - TOO MUCH WRITE DATA	TOO MUCH WRITE DATA
DELIVERY FAILURE - INFORMATION UNIT TOO SHORT	INFORMATION UNIT TOO SHORT
DELIVERY FAILURE - ACK/NAK TIMEOUT	ACK/NAK TIMEOUT
DELIVERY FAILURE - NAK RECEIVED	NAK RECEIVED
DELIVERY FAILURE - INITIATOR RESPONSE TIMEOUT	INITIATOR RESPONSE TIMEOUT

6.1.3 SCSI mode parameters

6.1.3.1 Disconnect-Reconnect mode page

6.1.3.1.1 Disconnect-Reconnect mode page overview

The Disconnect-Reconnect mode page (see SPC-4) provides the application client the means to tune the performance of the service delivery subsystem. Table 60 defines the parameters which are applicable to SSP. If any field in the Disconnect-Reconnect mode page is not implemented, the value assumed for the functionality of the field shall be zero (i.e., as if the field in the mode page is implemented and the field is set to zero).

The application client sends the values in the fields to be used by the device server to control the SSP connections by means of a MODE SELECT command. The device server shall then communicate the field values to the SSP target port. The field values are communicated from the device server to the SSP target port in a vendor-specific manner.

SAS devices shall only use the parameter fields defined below in this subclause. If any other fields within the Disconnect-Reconnect mode page of the MODE SELECT command contain a non-zero value, the device server shall return CHECK CONDITION status for that MODE SELECT command. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ILLEGAL FIELD IN PARAMETER LIST.

Table 60. Disconnect-Reconnect mode page for SSP

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (02h)					
1	PAGE LENGTH (0Eh)							
2	Reserved							
3	Reserved							
4	(MSB)	BUS INACTIVITY TIME LIMIT						
5								(LSB)
6		Reserved						
7								
8	(MSB)	MAXIMUM CONNECT TIME LIMIT						
9								(LSB)
10	(MSB)	MAXIMUM BURST SIZE						
11								(LSB)
12	Reserved							
13	Reserved							
14	(MSB)	FIRST BURST SIZE						
15								(LSB)

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SUBPAGE FORMAT (SPF) bit shall be set to zero for access to the short format mode page.

The PAGE CODE (PS) field shall be set to 02h.

The PAGE LENGTH field shall be set to 0Eh.

The BUS INACTIVITY TIME LIMIT field is defined in 6.1.3.1.2.

The MAXIMUM CONNECT TIME LIMIT field is defined in 6.1.3.1.3.

The MAXIMUM BURST SIZE field is defined in 6.1.3.1.4.

The FIRST BURST SIZE field is defined in 6.1.3.1.5.

6.1.3.1.2 BUS INACTIVITY TIME LIMIT field

The value in the BUS INACTIVITY TIME LIMIT field contains the maximum period that an SSP target port is permitted to maintain a connection (see 2.1.10) without transferring a frame to the SSP initiator port. This value shall be the number of 100 ms increments between frames that the SSP target port transmits during a connection. When this number is exceeded, the SSP target port shall prepare to close the connection (i.e., by requesting to have the link layer transmit DONE). This value may be rounded as defined in SPC-4. A value of zero in this field shall specify that there is no bus inactivity time limit. The bus inactivity time limit is enforced by the port layer.

6.1.3.1.3 MAXIMUM CONNECT TIME LIMIT field

The value in the MAXIMUM CONNECT TIME LIMIT field contains the maximum duration of a connection (see 2.1.10). This value shall be the number of 100 ms increments that an SSP target port transmits during a connection after which the SSP target port shall prepare to close the connection (e.g., a value of one in this field means that the time is less than or equal to 100 ms and a value of two in this field means that the time is less than or equal to 200 ms). If an SSP target port is transferring a frame when the maximum connection time limit is exceeded, the SSP target port shall complete transfer of the frame before preparing to close the connection. A value of zero in this field shall specify that there is no maximum connection time limit. The maximum connection time limit is enforced by the port layer.

6.1.3.1.4 MAXIMUM BURST SIZE field

For read data, the value in the MAXIMUM BURST SIZE field contains the maximum amount of data that is transferred during a connection by an SSP target port per I_T_L_Q nexus without transferring at least one frame for a different I_T_L_Q nexus. If the SSP target port:

a. has read data to transfer for only one I_T_L_Q nexus, and

b. has no requests to transfer write data for any I_T_L_Q nexus;

the SSP target port shall prepare to close the connection after the amount of data specified by the MAXIMUM BURST SIZE field is transferred to the SSP initiator port.

For write data, the value shall specify the maximum amount of data that an SSP target port requests via a single XFER_RDY frame (see 5.2.2.3).

This value shall be specified in 512-byte increments (e.g., a value of one in this field means that the number of bytes transferred to the SSP initiator port for the nexus is less than or equal to 512 and a value of two in this field means that the number of bytes transferred to the SSP initiator port for the nexus is less than or equal to 1 024). The device server may round this value down as defined in SPC-4. A value of zero in this field shall specify that there is no maximum burst size.

In terms of the SCSI transport protocol services (see 6.1.1), the device server shall limit the Request Byte Count argument to the Receive Data-Out () protocol service and the Send Data-In () protocol service to the amount specified in this field.

6.1.3.1.5 FIRST BURST SIZE field

The value in the FIRST BURST SIZE field contains the maximum amount of write data in 512-byte increments that may be sent by the SSP initiator port to the SSP target port without having to receive an XFER_RDY frame (see 5.2.2.3) from the SSP target port (e.g., a value of one in this field means that the number of bytes transferred by the SSP initiator port is less than or equal to 512 and a value of two in this field means that the number of bytes transferred by the SSP initiator port is less than or equal to 1 024).

Specifying a non-zero value in the FIRST BURST SIZE field is equivalent to an implicit XFER_RDY frame for each command requiring write data where the WRITE DATA LENGTH field of the XFER_RDY frame is set to the 512 times the value of the FIRST BURST SIZE field.

The rules for data transferred using the value in the FIRST BURST SIZE field are the same as those used for data transferred for an XFER_RDY frame (i.e., the number of bytes transferred using the value in the FIRST BURST SIZE field is as if that number of bytes was requested by an XFER_RDY frame).

If the amount of data to be transferred for the command is less than the amount of data specified by the FIRST BURST SIZE field, the SSP target port shall not transmit an XFER_RDY frame for the command. If the amount of data to be transferred for the command is greater than the amount of data specified by the FIRST BURST SIZE field, the SSP target port shall transmit an XFER_RDY frame after it has received all of the data specified by the FIRST BURST SIZE field from the initiator. All data for the command is not required to be transferred during the same connection in which the command is transferred.

A value of zero in this field shall specify that there is no first burst size (i.e., an SSP initiator port shall transmit no data frames to the SSP target port before receiving an XFER_RDY frame).

The first burst size is handled by the SCSI transport protocol services (see 6.1.1) and the SSP transport layer.

6.1.3.2 Protocol-Specific Port mode page

6.1.3.2.1 Protocol-Specific Port mode page overview

The Protocol-Specific Port mode page (see SPC-4) contains parameters that affect SSP target port operation. If the mode page is implemented, all logical units in SCSI target devices in SAS domains supporting the MODE SELECT or MODE SENSE commands shall implement the page, and there shall be one copy of the mode page shared by all SSP initiator ports.

If a SAS target device has multiple SSP target ports, changes in the short page parameters for one SSP target port should not affect other SSP target ports.

Table 62 defines the subpages of this mode page.

Table 61. Protocol-Specific Port Control mode page subpages

Subpage	Description	Reference
Short page	Short format	6.1.3.2.2
Long page 00h	Not allowed	
Long page 01h	Phy Control And Discover subpage	6.1.3.2.3
Long page E0h - FEh	Vendor specific	
Long page FFh	Return all subpages for the Port Control mode page	SPC-4
All others	Reserved	

6.1.3.2.2 Protocol-Specific Port mode page - short format

Parameters in this page shall affect all phys in the SSP target port, and may affect all SSP target ports in the SAS target device.

Table 62 defines the format of the page for SAS SSP.

Table 62. Protocol-Specific Port Control mode page for SAS SSP - short format

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (19h)					
1	PAGE LENGTH (06h)							
2	Reserved				PROTOCOL IDENTIFIER (6h)			

Table 62. Protocol-Specific Port Control mode page for SAS SSP - short format

Byte\Bit	7	6	5	4	3	2	1	0
3	Reserved							
4	(MSB)	I_T NEXUS LOSS TIME						
5								(LSB)
6	(MSB)	INITIATOR RESPONSE TIMEOUT						
7								(LSB)

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SPF field shall be set to zero for access to the short format mode page.

The PAGE CODE field shall be set to 19h.

The PAGE LENGTH field shall be set to 06h.

The PROTOCOL IDENTIFIER field shall be set to 6h indicating this is a SAS SSP specific mode page.

The I_T NEXUS LOSS TIME field contains the time that the SSP target port shall retry connection requests to an SSP initiator port that are rejected with responses indicating the SSP initiator port may no longer be present before recognizing an I_T nexus loss (see 2.4). Table 63 defines the values of the I_T NEXUS LOSS TIME field. If this mode page is not implemented, the I_T nexus loss time is vendor specific. This value is enforced by the port layer.

Table 63. I_T nexus loss time

Code	Description
0000h	Vendor-specific amount of time.
0001h to FFFEh	Time in milliseconds.
FFFFh	The SSP target port shall never recognize an I_T nexus loss (i.e., it shall retry the connection requests forever).

Note. If this mode page is implemented, the default value of the I_T NEXUS LOSS TIME field should be non-zero. It is recommended that this value be 2 000 ms.

The INITIATOR RESPONSE TIMEOUT field contains the time in milliseconds that the SSP target port shall wait for the receipt of a frame (e.g., a write DATA frame) before aborting the command associated with that frame. An INITIATOR RESPONSE TIMEOUT field value of zero indicates that the SSP target port shall disable the initiator response timeout timer. If this mode page is not implemented, the logical unit shall not implement an initiator response timeout timer. This value is enforced by the transport layer.

6.1.3.2.3 Protocol-Specific Port mode page - Phy Control And Discover subpage

The Phy Control And Discover subpage contains phy-specific parameters. Parameters in this page shall affect only the referenced phy.

Table 64 defines the format of the subpage for SAS SSP.

Table 64. Protocol-Specific Port Control mode page for SAS SSP - Phy Control And Discover subpage

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (19h)					
1	SUBPAGE CODE (01h)							
2	(MSB)	PAGE LENGTH (n - 3)						
3								(LSB)
4	Reserved							
5	Reserved				PROTOCOL IDENTIFIER (6h)			
6	Reserved							
7	NUMBER OF PHYS							
SAS phy mode descriptors								
8	First SAS phy mode descriptor							
...	...							
n	Last SAS phy mode descriptor							

The PARAMETERS SAVEABLE (PS) bit is defined in SPC-4.

The SPF field shall be set to one to access the long format mode pages.

The PAGE CODE field shall be set to 19h.

The SUBPAGE CODE field shall be set to 01h.

The PAGE LENGTH field shall be set to (4 + (the value of the NUMBER OF PHYS field) ¥ (the length in bytes of the SAS phy mode descriptor)).

The PROTOCOL IDENTIFIER field shall be set to 6h indicating this is a SAS SSP specific mode page.

The NUMBER OF PHYS field contains the number of phys in the SAS target device and indicates the number of SAS phy mode descriptors that follow. This field shall not be changeable.

A SAS phy mode descriptor shall be included for each phy in the SAS target device, starting with the lowest numbered phy and ending with the highest numbered phy.

Table 65 defines the SAS phy mode descriptor.

Table 65. SAS phy mode descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	PHY IDENTIFIER							
2	Reserved							
3								
4	Reserved	ATTACHED DEVICE TYPE			Reserved			
5	Reserved				NEGOTIATED PHYSICAL LINK RATE			
6	Reserved				ATTACHED SSP INITIATOR PORT	ATTACHED STP INITIATOR PORT	ATTACHED SMP INITIATOR PORT	Reserved
7	Reserved				ATTACHED SSP TARGET PORT	ATTACHED STP TARGET PORT	ATTACHED SMP TARGET PORT	Reserved
8	SAS ADDRESS							
15								
16	ATTACHED SAS ADDRESS							
23								
24	ATTACHED PHY IDENTIFIER							
25	Reserved							
31								
32	PROGRAMMED MINIMUM PHYSICAL LINK RATE				HARDWARE MINIMUM PHYSICAL LINK RATE			
33	PROGRAMMED MAXIMUM PHYSICAL LINK RATE				HARDWARE MAXIMUM PHYSICAL LINK RATE			
34	Reserved							
41								
42	Vendor specific							
43								
44	Reserved							
47								

The PHY IDENTIFIER field, ATTACHED DEVICE TYPE field, NEGOTIATED PHYSICAL LINK RATE field, ATTACHED SSP INITIATOR PORT bit, ATTACHED STP INITIATOR PORT bit, ATTACHED SMP INITIATOR PORT bit, ATTACHED SSP TARGET PORT bit, ATTACHED STP TARGET PORT bit, ATTACHED SMP TARGET PORT bit, SAS ADDRESS field, ATTACHED SAS ADDRESS field, ATTACHED PHY IDENTIFIER, HARDWARE MINIMUM PHYSICAL LINK RATE field, and HARDWARE MAXIMUM PHYSICAL LINK RATE field are defined in the SMP DISCOVER function (see SAS). These fields shall not be changeable.

The PROGRAMMED MINIMUM PHYSICAL LINK RATE field and PROGRAMMED MAXIMUM PHYSICAL LINK RATE field are defined in the SMP PHY CONTROL function (see SAS).

6.1.3.3 Protocol-Specific Logical Unit mode page

SSP logical units shall not include the Protocol-Specific Logical Unit mode page (see SPC-4).

6.1.4 SCSI log parameters

6.1.4.1 Protocol-Specific log page

The Protocol Specific log page for SAS defined in table 66 is used to report errors that have occurred on the SAS target device's phy(s).

Table 66. Protocol-Specific log page for SAS

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved		PAGE CODE (18h)					
1	Reserved							
2	(MSB)		PAGE LENGTH (m - 3)					
3								(LSB)
Protocol-specific log parameters								
4			First protocol-specific log parameter					
...	...							
			n^{th} protocol-specific log parameter					
m								

The PAGE CODE field shall be set to 18h.

The PAGE LENGTH field shall be set to the total length in bytes of the log parameters.

Table 67 defines the format for a SAS log parameter.

Table 67. Protocol-Specific log parameter format for SAS

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)							
1	PARAMETER CODE (relative target port identifier)							(LSB)
2	DU	DS	TSD	ETC	TMC		LBIN	LP
3	PARAMETER LENGTH (y - 3)							
4	Reserved				PROTOCOL IDENTIFIER (6h)			
5	Reserved							
6								
7	NUMBER OF PHYS							
SAS phy log descriptors								
8	First SAS phy log descriptor							
	...							
	Last SAS phy log descriptor							
y								

The PARAMETER CODE field contains the relative target port identifier (see SPC-4) of the SSP target port that this log parameter describes.

Table 68 defines the values for the log parameter control bits for this log parameter.

Table 68. Parameter control bits for SAS log parameters

Bit	Value for LOG SENSE	Value for LOG SELECT	Description
DU	0	0 or 1	The value is provided by the device server.
DS	0	0 or 1	The device server supports saving of the parameter.
TSD	0	0 or 1	The device server manages saving of the parameter.
ETC	0	0 or 1	No threshold comparison is made on this value.
TMC	0	any	This field is ignored when the ETC bit is 0.
LBIN	1	1	The parameter is in binary format.
LP	1	1	The parameter is a list parameter.

The PARAMETER LENGTH field is set to the length of the log parameter minus three.

The PROTOCOL IDENTIFIER field is set to 6h.

The NUMBER OF PHYS field contains the number of SAS phy log descriptors that follow.

Table 69 defines the SAS phy log descriptor. Each SAS phy log descriptor is the same length.

Table 69. SAS phy log descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	PHY IDENTIFIER							
2	Reserved							
3	SAS PHY LOG DESCRIPTOR LENGTH (m-3)							
4	Reserved	ATTACHED DEVICE TYPE			Reserved			
5	Reserved				NEGOTIATED PHYSICAL LINK RATE			
6	Reserved				ATTACHED SSP INITIATOR PORT	ATTACHED STP INITIATOR PORT	ATTACHED SMP INITIATOR PORT	Reserved
7	Reserved				ATTACHED SSP TARGET PORT	ATTACHED STP TARGET PORT	ATTACHED SMP TARGET PORT	Reserved
8	SAS ADDRESS							
15								
16	ATTACHED SAS ADDRESS							
23								
24	ATTACHED PHY IDENTIFIER							
25	Reserved							
31								
32	(MSB)	INVALID DWORD COUNT						
35								(LSB)
36	(MSB)	RUNNING DISPARITY ERROR COUNT						
39								(LSB)
40	(MSB)	LOSS OF DWORD SYNCHRONIZATION						
43								(LSB)

Table 69. SAS phy log descriptor

Byte\Bit	7	6	5	4	3	2	1	0
44	(MSB)							
47				PHY RESET PROBLEM				(LSB)
48								
50				Reserved				
51				NUMBER OF PHY EVENT DESCRIPTORS				
52								
m				PHY EVENT DESCRIPTOR(S)				

The PHY IDENTIFIER field, ATTACHED DEVICE TYPE field, NEGOTIATED PHYSICAL LINK RATE field, ATTACHED SSP INITIATOR PORT bit, ATTACHED STP INITIATOR PORT bit, ATTACHED SMP INITIATOR PORT bit, ATTACHED SSP TARGET PORT bit, ATTACHED STP TARGET PORT bit, ATTACHED SMP TARGET PORT bit, SAS ADDRESS field, ATTACHED SAS ADDRESS field, and ATTACHED PHY IDENTIFIER field are defined in the SMP DISCOVER function (see SAS).

The SAS PHY LOG DESCRIPTOR LENGTH field indicates the number of bytes that follow in the SAS phy log descriptor. A SAS PHY LOG DESCRIPTOR LENGTH field set to zero indicates there are 44 additional bytes.

The INVALID DWORD COUNT field, RUNNING DISPARITY ERROR COUNT field, LOSS OF DWORD SYNCHRONIZATION field, and PHY RESET PROBLEM COUNT field are each defined in the SMP REPORT PHY ERROR LOG response data (see SAS-2).

The number of phy event descriptors field indicates how many phy event descriptors follow.

Each phy event descriptor is 8 bytes long and follows the format defined for the SMP REPORT PHY EVENT INFORMATION function.

6.1.5 SCSI vital product data (VPD)

Each logical unit in a SAS target device that is a SCSI target device includes the identification descriptors listed in table 70 in the Device Identification VPD page (83h) returned by the INQUIRY command (see SPC-4).

Table 70. Device Identification VPD page required identification descriptors

Field in identification descriptor	Identification descriptor			
	Logical unit name	**Target port identifier**	**Relative target port identifier**	**Target device name**
IDENTIFIER TYPE	3h (i.e., NAA)	3h (i.e., NAA)	4h (i.e., relative target port identifier)	3h (i.e., NAA)
ASSOCIATION	0h (i.e., logical unit)	1h (i.e., SCSI target port)	1h (i.e., SCSI target port)	2h (i.e., SCSI target device)
CODE SET	1h (i.e., binary)	1h (i.e., binary)	1h (i.e., binary)	1h (i.e., binary)
IDENTIFIER LENGTH	8 or 16	8	4	8

Table 70. Device Identification VPD page required identification descriptors

Field in identification descriptor	Identification descriptor			
	Logical unit name	**Target port identifier**	**Relative target port identifier**	**Target device name**
piv (protocol identifier valid)	0	1	0	1
PROTOCOL IDENTIFIER	Any	6h (i.e., SAS)	Any	6h (i.e., SAS)
IDENTIFIER	NAA IEEE Registered format or NAA IEEE Registered Extended format	NAA IEEE Registered format	SCSI target ports shall be numbered sequentially starting with 00000001h	NAA IEEE Registered format
The IDENTIFIER field contains an NAA field set to 5h (i.e., IEEE Registered); the IDENTIFIER LENGTH field is set to 8. The IDENTIFIER field contains an NAA field set to 6h (i.e., IEEE Registered Extended); the IDENTIFIER LENGTH field is set to 16. The IDENTIFIER field contains the SAS address of the drive port through which the INQUIRY command was received. The IDENTIFIER field contains the relative drive port identifier of the drive port through which the INQUIRY command was received.				

Seagate Technology LLC
920 Disc Drive, Scotts Valley, California 95066-4544, USA
Publication Number: 100293071, Rev. B, Printed in USA