Product Manual

Parallel SCSI Interface

Ultra 160

Ultra 320

Seagate

Product Manual

Parallel SCSI Interface

Ultra 160

Ultra 320

Seagate

Publication number: 100293069, Rev. A
March 2006

Revision status summary sheet

Revision	Date	Writer/Engineer	Sheets Affected
Rev. A	03/27/06	C. Chalupa/G. Houlder	All.

Table of Contents

List of Figures

1.0 Interface requirements

1.1 How to use this interface manual

This manual provides a description of the SCSI[1] interface protocol and some general timing information as implemented by Seagate products. The features described in this manual are typically referred to as “Ultra160 SCSI” or “Ultra320 SCSI” features. Individual drive’s Product Manual, for the various SCSI interface products, contains additional and more detailed information on protocol, features supported, timing, and electrical/ mechanical aspects of how the SCSI interface is implemented by that product.

This manual provides a general, tutorial-type description of the ANSI SCSI (formerly called SCSI-3) system. It is not intended to give all of the kinds of details needed to design/implement a SCSI system or product. For information about SCSI interface details not included herein, refer to the standards listed in Section 1.1.1.

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.

The combination of this specification together with the details in the individual drive’s Product Manual, provides a description of how a particular product implements the SCSI I/O system. This specification is Volume 2 of a set of manuals that is made up of an individual drive’s Product Manual, and this manual. The older Ultra2 SCSI Interface Manual, part number 77738479, applies to Seagate products that implement older versions of the SCSI interface (SCSI-1/SCSI-2). This new Parallel SCSI Interface Manual, part number 100293069, is referenced by newer individual drive’s Product Manuals, representing Seagate products that support Ultra160 or Ultra320 SCSI features and other new features, such as packetized information transfer (SPI information units), data group transfers, paced transfers, increased CRC protection, etc.

1Unless required for clarity, “SCSI” is now used instead of “SCSI-3.”

1.1.1 Scope of SCSI standards

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

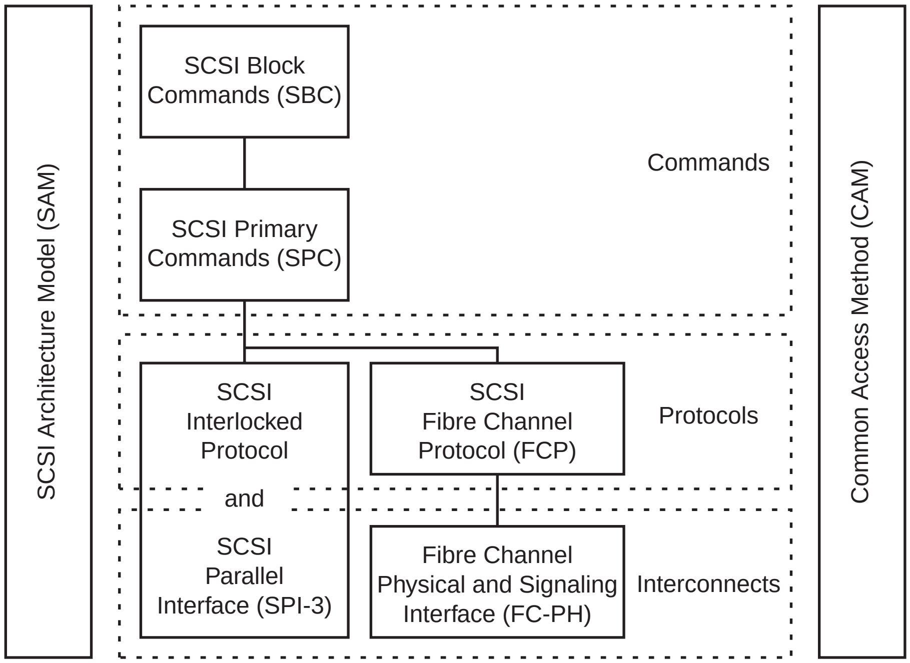


Figure 1. Functional scope of SCSI[1] standards

The functional areas define the scope of each standard as follows:

- SCSI Architecture Model: Defines the SCSI systems model, the functional partitioning of the SCSI standard set and requirements applicable to all SCSI implementations and implementation standards.
- Commands: Implementation standards which define classes including a device model for each class. These standards 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.
- Common Access Method: Implementation standard which defines a host architecture and set of services for device access.
- Protocols: Implementation standards which define the rules for exchanging information so that different SCSI devices can communicate.
- Interconnects: Implementation standards which define the electrical and signaling rules essential for devices to interoperate over a given physical interconnect.

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

1.1.2 Applicable standards

The following ANSI standards should be referenced for more details about SCSI system standards 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), T10/1215-D
- SCSI Primary Commands - 4 (SPC-4), T10/1731-D
- SCSI Enhanced Parallel Interface (EPI), T10/1143-DT
- SCSI Parallel Interface (SPI-5), T10/1525D
- SCSI Medium Changer Commands - 3 (SMC-3), T10/1730-D
- SCSI Controller Command Set - 2 (SCC-2), T10/1225D
- SCSI Stream Command - 3 (SSC-3), T10/1611-D

1.2 General interface description

This Parallel SCSI Interface Manual describes the Seagate Technology LLC. subset of the SCSI (Small Computer Systems Interface) as implemented on the Seagate-built drives. The interface is compatible with the SCSI Interface Specifications listed in Section 1.1.2. The drives covered by this manual are classified as “Intelligent” peripherals.

The Seagate SCSI interface described herein consists of a 9 or 18 bit bidirectional data bus (includes bits for parity checking and enabling CRC protection), plus 9 control signals. 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.

The SCSI physical interface uses either single-ended drivers and receivers or low voltage differential drivers and receivers and uses asynchronous or synchronous communication protocols. The bus interface transfer rate for asynchronous or synchronous is given in individual drive’s Product Manual. The bus protocol supports multiple initiators, disconnect/reconnect, additional messages plus 6-byte, 10-byte, 12-byte, 16-byte and variable length Command Descriptor Blocks.

Unless specified otherwise in the individual drive’s Product Manual, the drive is always a SCSI target port, and never a SCSI initiator port. For certain commands, which may or may not be supported by a particular drive model, the drive must act as a SCSI 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.

Note. In this revision, some new terminology is introduced as taken from the ANSI specifications. In many instances, the broader scope terms such as “initiator” and “target” are not used, but rather the more specific terms “Application Client” and “Device Server” appear. 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. Although a Glossary of terms is provided herein, the definitions may not be adequate for some. The SAM-4 specification gives a more detailed understanding of some of the new SCSI terminology

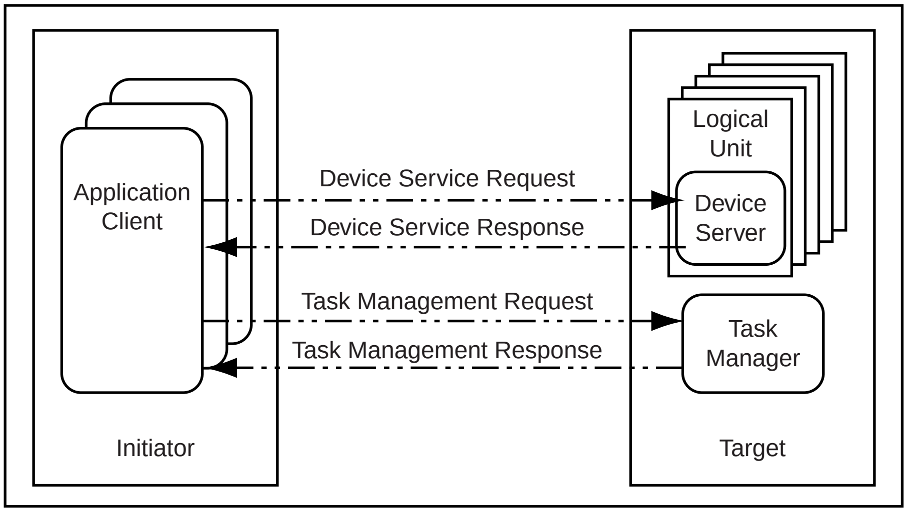


Figure 2. SCSI client-server model

1.2.1 Glossary

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

ACA—Auto Contingent Allegiance (see below).

additional sense code—a combination of the ADDITIONAL SENSE CODE and ADDITONAL SENSE CODE QUALIFIER in the sense data (see SPC-4)

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).

asynchronous event notification—A procedure used by targets to notify initiators of events that occur when a pending task does not exist for that initiator.

asynchronous transfer—An information transfer that uses the REQ/ACK handshake with an offset of zero.

auto contingent allegiance (ACA)—One of the conditions of a task set following the return of a CHECK CONDITION status. See Section 4.4.2.

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

blocking boundary—A task set boundary denoting a set of conditions that inhibit tasks outside the boundary from entering the Enabled state.

byte—An 8-bit construct.

call—The act of invoking a procedure.

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 SAM-4).

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—A structure used to communicate a command from an application client to a device server. Command structures of 6, 10, 12, or 16 bytes are used, but a new variable length command structure has recently been introduced.

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.

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.

contingent allegiance (CA)—An optional condition of a task set following the return of a CHECK CONDITION status. A detailed definition of contingent allegiance may be found in Section 4.4.2. (CA declared obsolete by SAM-4, ACA supported by SAM-4)

control mode page—The mode page that identifies the settings, and provides control, 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 Command Reference Manual, Part number 100293068, or SPC-4.

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 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.

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

device server—An object within the logical unit which executes SCSI tasks according to the rules for task management described in clause 7 of ANSI SAM-4 document, T10/1683-D.

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.

disconnect—The action that occurs when a SCSI device releases control of the SCSI bus, allowing it to go to the BUS FREE PHASE.

domain—An I/O system consisting of a set of SCSI devices that interact with one another by means of a service delivery subsystem.

dormant (task state)—The state of a task that is prevented from starting execution due to the presence of certain other tasks in the task set.

double transition (DT)—The latching of data on both the assertion edge and the negated edge of the REQ or ACK signals.

driver—The circuitry used to control the state of the bus.

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.

ended command—A command that has completed or aborted.

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

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

faulted I_T nexus: The I_T nexus on which a CHECK CONDITION status was returned that resulted in the establishment of an ACA. The faulted I_T nexus condition is cleared when the ACA condition is cleared.

faulted task set: A task set that contains a faulting task. The faulted task set condition is cleared when the ACA condition resulting from the CHECK CONDITION status is cleared.

faulting command: A command that completed with a status of CHECK CONDITION that resulted in the establishment of an ACA.

faulting task: A task that has completed with a status of CHECK CONDITION that resulted in the establishment of an ACA.

function complete—A logical unit response indicating that a task management function has finished. The actual events comprising this response are protocol specific.

hard reset—a SCSI target port response to a reset event or a SCSI target port Reset in which the target performs the operations described in Section 7.6.7.

implementation—The physical realization of an object.

implementation-specific—A requirement or feature that is defined in a SCSI standard 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—The second step in a four-step confirmed service reply to a request.

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

initial connection—The result of a physical connect. It exists from the assertion of the BSY signal in a SELECTION PHASE until the next BUS FREE PHASE or the next QAS REQUEST message.

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.

interconnect—The electrical media (including connectors and passive loads) used to connect the TERMPWR, terminators, and SCSI devices in a SCSI bus.

interconnect subsystem—One or more physical interconnects which appear as a single path for the transfer of information between SCSI devices in a domain.

intersymbol interference (ISI)—The effect of adjacent symbols on the symbol currently being received.

in transit—Information that has been sent to a remote object but not yet received.

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_L nexus—A nexus that exists between a SCSI initiator port, a SCSI target port, and a logical unit. This relationship replaces 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 queue tag following the successful receipt of one of the queue tag messages. This relationship replaces the prior I T L nexus.

iuCRC protection—The use of CRC to detect DT DATA PHASE data transmission errors during parallel transfers. Contains CRC information covering all bytes transmitted in a SPI information unit.

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

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.

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.

logical unit reset— A logical unit action in response to a logical unit reset event in which the logical unit performs the operations described in SCSI Architecture Model-4.

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

mandatory—The referenced item is required to claim compliance with a standard.

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.

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 LVD SCSI devices that combines LVD SCSI and single-ended SCSI (see SPI-5, SCSI parallel interface electrical characteristics) drivers and receivers to allow operation when SE SCSI devices are present on the bus.

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.

odd parity—Odd logical parity, where the parity bit is driven and verified to be that value that makes the number of assertions on the associated data byte plus the parity bit equal to an odd number (1, 3, 5, 7, or 9). *See* parity bit. If an even number of asserted bits are detected at the receiver, a parity error occurs.

paced transfer—Parallel transfers that transfer information using pacing.

pacing—Use of the ACK or REQ signal as a continuously running clock in combination with the P1 signal to indicate when data is valid.

packetized—A method of transferring information using SPI information units. *See* object.

pad field—The portion of a data group that contains pad information.

parallel protocol request—Messages used to negotiate a synchronous data transfer agreement, a wide data transfer agreement, and set the protocol options between two SCSI devices.

parity bit—A bit associated with a byte that is used to detect the presence of an odd number of asserted bits within the byte. The parity bit is driven such that the number of logical ones in the byte plus the parity bit is odd.

pCRC field—The portion of a data group that contains pCRC information.

pCRC protection—The use of pCRC to detect DT DATA PHASE.

peer-to-peer protocol service—A service used by an upper level protocol implementation to exchange information with its peer.

peer entities—Entities within the same (protocol) layer.

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

physical interconnect—A single physical pathway for the transfer of information between SCSI devices in a domain.

physical reconnect—The act of resuming a nexus to continue a task. A SCSI target port initiates a physical reconnect when conditions are appropriate for the physical bus to transfer data associated with a nexus between a SCSI initiator port and a SCSI target port.

physical reconnection—The result of a physical reconnect that exists from the assertion of the BSY signal in a SELECTION or RESELECTION PHASE. A physical reconnection ends with the BUS FREE PHASE (see Section 3.1.1) or a QAS REQUEST message (see Section 4.3.13).

port—Synonymous with “service delivery port.” A single attachment to a SCSI bus from a SCSI device.

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

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 standard 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.

quick arbitration and selection process (QAS)—Quicker than the normal arbitration and selection process. Implementation is optional for SCSI devices.

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.

receiver—A client or server that is the recipient of a service delivery transaction.

reference model—A standard model used to specify system requirements in an implementation-independent manner.

request—A transaction invoking a service.

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—A protocol-specific event which may trigger a hard reset response from a SCSI device as described in Section 5.3.

response—A transaction conveying the result of a request.

SCSI application layer (SAL)—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 at least one SCSI port and the means to connect its drivers and receivers to the bus.

SCSI device identifier—An address by which a SCSI device is referenced within a domain.

SCSI I/O system—An I/O system, consisting of two or more SCSI devices, a SCSI interconnect and a SCSI protocol, which collectively interact to perform SCSI I/O operations.

SCSI protocol layer—The protocol and services used by a SCSI application layer to transport data representing a SCSI application protocol transaction.

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

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—That part of a SCSI I/O system which transmits service requests to a logical unit or target and returns logical unit or target responses to a SCSI initiator 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.

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

SPI information unit—Data structures that encapsulate data, status, command, task attributes, iuCRC, and nexus information into various formats.

SPI L_Q information unit—The SPI L_Q information unit (see Section 6.2.2, tables 49 and 50) contains L_Q nexus (Logical unit—Q tag relationship) information for the information unit that follows, the type of information unit that follows, and the length of information unit that follows. A SPI L_Q information unit shall precede all SPI command information units, SPI multiple command information units, SPI data information units, SPI status information units, and the first of an uninterrupted sequence of SPI data stream information units.

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.

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

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 of Section 7.6.1.

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, 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.

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.

transfer period—The negotiated time between edges of REQ or ACK that latch data. For ST, the transfer period is measured from assertion edge of the REQ or ACK signal to the next assertion edge of the signal. For DT, the transfer period is measured from a transition edge of the REQ or ACK signal to the next transition edge of the signal.

unconfirmed protocol service—A service available at the protocol service interface, which does not result in a completion confirmation.

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.

1.2.2 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 standard (see Section 1.1.1.)

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

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 standard.

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 standards but have been removed from this standard.

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

reserved—A key word referring to bits, bytes, words, fields, and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word, or field shall be set to zero, or in accordance with a future extension to this standard. 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 standard conformant products.

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

1.3 Physical interface characteristics

The physical interface characteristics (cables, connectors, electrical descriptions, termination requirements, 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.

1.4 Summary of SCSI messages

Following is an alphabetical summary of the SCSI messages described in this manual. Details are given in Section 4.

Message Name	Hex Code	Page number
ABORT TASK	06	105
CLEAR QUEUE	0E	106
DISCONNECT	04	87
EXTENDED MESSAGE	01	84
IDENTIFY	80-FF	87
IGNORE WIDE RESIDUE	23	88
INITIATE RECOVERY	0F	Not described in this manual
INITIATOR DETECTED ERROR	05	89
MESSAGE PARITY ERROR	09	89
MESSAGE REJECT	07	89
MODIFY DATA POINTER	01, 05, 00	89 (extended message)
NO OPERATION	08	91
PARALLEL PROTOCOL REQUEST	01, 06, 04	92
QAS REQUEST	55	96
RELEASE RECOVERY	10	Not described in this manual
RESTORE POINTERS	03	96
SAVE DATA POINTERS	02	96
SYNCHRONOUS DATA TRANSFER REQUEST	01, 03, 01	97 (extended message)
TASK ATTRIBUTE MESSAGES		102
ACA (AUTO CONTINGENT ALLEGIANCE)	24	103
HEAD OF QUEUE TAG	21	104
LINKED COMMAND COMPLETE	0A	89
ORDERED QUEUE TAG	22	104
SIMPLE QUEUE TAG	20	104
TASK COMPLETE	00	99
TASK MANAGEMENT MESSAGES		105
ABORT TASK	00	105
ABORT TASK SET	06	106
CLEAR ACA	16	106
CLEAR TASK SET	0E	106
LOGICAL UNIT RESET	17	106
TARGET RESET	0C	106
WIDE DATA TRANSFER REQUEST	01, 03	100 (extended message)

2.0 SCSI bus

This manual discusses only the “logical” and timing characteristics of the SCSI system and interface. A general description of the SCSI bus physical characteristics (voltages, connector configurations, pinouts, etc.) are given in the individual drive’s Product Manual, in the “Interface requirements” section, which covers the interface requirements and SCSI features supported by the drive described in the particular Product Manual being referenced. For all of the physical details of the SCSI interface, consult the ANSI standards referenced in Section 1.1.2.

Communication on the SCSI Bus is allowed between only two SCSI devices at a time. Seagate drives support systems with a maximum of 16 SCSI devices including the host computer(s) connected to the SCSI bus. Each SCSI device has a SCSI ID bit assigned as shown in Table 1. The SCSI ID is assigned by installing 0-3 jumper plugs onto a connector in a binary coded configuration during system configuration. Some drive models have an interface that includes the SCSI bus ID lines, so that the host can set the drive ID over the interface (see individual drive’s Product Manual, “Option/configuration headers” section).

When two SCSI devices communicate on the SCSI Bus, one acts as a SCSI initiator port and the other acts as a SCSI target port. The initiator (typically a host computer) originates an I/O process and the target performs whatever operations/tasks are called for by the I/O process. Devices arbitrate to get control of the bus to perform whatever communications are required by the current I/O process. As part of the arbitration process, devices on the SCSI bus assert their bus ID (one of the DB lines as shown in Table 1). The arbitration process is discussed in more detail later. Devices supported by this interface specification always operate as targets, unless otherwise specified (i.e., if certain commands are supported) in the individual drive’s Product Manual, Volume 1.

Table 1: SCSI IDs and associated SCSI bus arbitration priorities

SCSI address	DB 15							DB 8	DB 7							DB 0	Priority
7	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	**1**
6	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	**2**
5	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	**3**
4	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	**4**
3	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	**5**
2	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	**6**
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	**7**
0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	**8**
15	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**9**
14	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**10**
13	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	**11**
12	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	**12**
11	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	**13**
10	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	**14**
9	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	**15**
8	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	**16**
Key: - = a logical 0 bit resulting from the data bus bit being released																	

The Host Adapter/Initiator must be identified by one of the 16 SCSI Device IDs (usually 7, which is highest arbitration priority). Make sure that none of the devices on the SCSI bus have duplicate IDs.

Certain SCSI bus functions are assigned to the initiator and certain SCSI bus functions are assigned to the target. The initiator will select a particular target. The target will request the transfer of Command, Data, Status, or other information on the data bus.

Under SCSI-2 protocol, information transfers on the data bus are interlocked and follow a defined REQ/ACK Handshake protocol. One byte of information will be transferred with each handshake. Synchronous data transfers do not require a one-for-one interlocking of REQ/ACK signals, but the total number of REQ pulses in a particular data transfer event must equal the total number of ACK pulses. The synchronous data transfer option is described in Paragraphs 3.1.5.2 and 3.5.3.2 of the Ultra2 SCSI Interface Manual, part number 77738479.

This Ultra160/Ultra320 Parallel SCSI Interface Manual, discusses SCSI-3 protocol (now called only "SCSI"). There are now different ways of latching data from the REQ and ACK signals, depending on whether ST DATA phases, DT DATA phases, or paced transfers are being used for information transfers. This is described in Section 2.1.2.

The drive supports single initiator, single target; single initiator, multiple target; multiple initiator, single target; or multiple initiator, multiple target bus configurations.

2.1 SCSI bus signals overview

Information transfer on the SCSI bus is allowed between only two SCSI devices at any given time except during MESSAGE IN PHASE when QAS is enabled. All SCSI devices that have QAS enabled are required to monitor messages during a MESSAGE IN PHASE for a QAS REQUEST MESSAGE. The maximum number of SCSI devices is determined by the width of the data path implemented. The SCSI devices may be any combination of SCSI initiator ports (commonly called “initiators”) and SCSI target ports (commonly called “targets”), provided there is at least one of each.

Each SCSI device has a SCSI address and a corresponding SCSI ID bit assigned to it. When two SCSI devices communicate on the SCSI bus, one acts as the initiator and the other acts as the target. The initiator originates an I/O process and the target receives the I/O process.

Some drive models have a single 80-pin I/O connector that contains additional interface lines that carry drive configuration select signals. These are peculiar to certain drives and are not SCSI standard signals. These are described in the individual drive’s product manual, Volume 1.

The 28 SCSI standard signals are described as follows:

BSY (Busy)—An “OR-tied” signal to indicate the bus is being used.

SEL (Select)—An “OR-tied” signal used by a SCSI initiator port to select a SCSI target port, or by a SCSI target port to reselect a SCSI initiator port.

RST (Reset)—An “OR-tied” signal that indicates the bus reset condition (see Section 5.2).

C/D (Control/Data)—A signal sourced by a SCSI target port that indicates whether CONTROL or DATA PHASE information is on the data bus. Assertion indicates Control (i.e., COMMAND, STATUS, and MESSAGE phases).

I/O (Input/Output)—A signal sourced by a SCSI target port to control the direction of data movement on the Data Bus with respect to a SCSI initiator port. Assertion indicates input to the initiator. This signal also distinguishes between SELECTION and RESELECTION phases.

MSG (Message)—A signal sourced by a SCSI target port to indicate the MESSAGE phase or a DT DATA phase depending on whether C/D is true or false. Asserted indicates MESSAGE or DT DATA.

REQ (Request)—A signal sourced by a SCSI target port to indicate a request for an information transfer on the SCSI bus.

ACK (Acknowledge)—A signal sourced by a SCSI initiator port to respond with an acknowledgment of an information transfer on the SCSI bus.

ATN (Attention)—A signal sourced by a SCSI initiator port to indicate the Attention condition.

DIFFSENS (Differential Sense)/Multimode—SE or LVD alternative—“LW” and “LC” models have I/O circuits that can operate either in single-ended (SE) or low voltage differential (LVD) mode. When the interface DIFFSENS line is between -0.35 V and +0.5 V, the drive interface circuits operate single-ended. When DIFFSENS is between +0.7 V and +1.9 V, the drive interface circuits operate low voltage differential. This arrangement is not intended to allow dynamically changing transmission modes, but rather to prevent incompatible devices from attempting to interoperate. Drives must operate only in the mode for which the installation and interface cabling is designed. Multimode I/O circuits used by “LW” and “LC” devices do not operate at high voltage differential levels and should never be exposed to high voltage differential environments unless the command mode voltages in the environment are controlled to safe levels for single-ended and low voltage differential devices (see the ANSI SPI-5 specification). High Voltage Differential (HVD) is now an obsolete ANSI standard.

P_CRCA (Parity/CRC Available)—A signal identifying either parity or CRC available based on bus phase and negotiated settings.

During the SELECTION PHASE, RESELECTION PHASE, ST DATA PHASE, COMMAND PHASE, MESSAGE PHASE, and STATUS PHASE, this signal is referred to as DB(P_CRCA) and is sourced by the SCSI device port driving the Data Bus. The DB(P_CRCA) signal is associated with the DB(7-0) signals and is used to detect the presence of an odd number of bit errors within the byte. The DB(P_CRCA) bit is driven such that the number of logical ones in the byte plus the parity bit is odd.

Data group transfers are enabled (see Section 4.3.12) when this signal is referred to as P_CRCA and is sourced by the target to control whether a data group field is a pad field, pCRC field, or data field (see Section 2.11.1). When asserted, the data group field shall be pad or pCRC fields that shall not be transferred to the application client. When negated, the data group field shall be a data field that shall be transferred to the application client.

During DT DATA phases when information unit transfers are enabled, this signal is referred to as P_CRCA and sourced by the SCSI target. Depending on the negotiated condition of read streaming and write flow control, the SCSI initiator and target usage for P_CRCA is different. When information unit transfers are enabled, the SCSI target and initiator shall use the P_CRCA signal as indicated in Table 2.

Table 2: P_CRC signal usage requirements

Write flow control	Read streaming	DT Data phase	SCSI initiator response to P_CRCA	SCSI target usage of P_CRCA
Disabled	Disabled	All	Ignore	Continuously negated.
Enabled	Disabled	DT DATA IN	Ignore	Continuously negated.
		DT DATA OUT	Monitor	Asserts to indicate when the current SPI data stream information unit is the last SPI data stream information unit of the current write stream.
Disabled	Enabled	DT DATA IN	Monitor	Asserts to indicate when the current SPI data stream information unit is the last SPI data stream information unit of the current read stream.
		DT DATA OUT	Ignore	Continuously negated.
Enabled	Enabled	DT DATA IN	Monitor	Asserts to indicate when the current SPI data stream information unit is the last SPI data stream information unit of the current read stream.
		DT DATA OUT	Monitor	Asserts to indicate when the current SPI data stream information unit is the last SPI data stream information unit of the current read stream.
A SCSI device is not required to use read streaming even if it is enabled A SCSI device is not required to use write flow control even if it is enabled				

P1 (Parity 1)—A signal normally sourced by the SCSI device driving the Data Bus. The P1 signal is associated with the DB(15–8) signals and is used to detect the presence of an odd number of bit errors within the byte The P1 bit is driven such that the number of logical ones in the byte plus the P1 bit is odd.

During the ST DATA PHASE with transfer length set for 8-bit transfers, COMMAND PHASE, MESSAGE PHASE, and STATUS phase, the P1 signal shall not be driven by any SCSI device.

During the SELECTION phase and the RESELECTION phase on a 16-bit wide bus segment the P1 signal shall be sourced by the SCSI device driving the DATA BUS.

When data group transfers are enabled (see Section 4.3.12), the P1 signal shall be continuously negated by the SCSI device driving the DB(15-0) signals and shall be ignored by the SCSI device receiving the DB(15-0) signals during DT DATA phases.

When information unit transfers are enabled, the P1 signal shall be continuously negated by the SCSI device driving the DB(15-0) signals and shall be ignored by the SCSI device receiving the DB(15-0) signals during DT DATA phases.

During DT DATA phases when information unit transfers and paced transfers are enabled the P1 signal shall be sourced by the SCSI device driving the DATA BUS. The P1 signal is used to indicate the data valid or data invalid state during paced transfers.

DB(7–0) (8-bit data bus)—Each data bit that forms the 8-bit data bus. Bit significance and priority during arbitration are shown in Table 1.

DB(15–0) (16-bit data bus)—16 data bit signals that form the 16-bit Data Bus. Bit significance and priority during arbitration are shown in Table 1.

Greater detail on each of the SCSI bus signals is found in the following sections.

2.1.1 Drive select

For SCSI ID selection, install drive select jumpers as shown in configuration selection figure in the individual drive's Product Manual. Refer to the "Physical interface" section of the individual drive's Product Manual for the location of the drive select header. Drives using the 8-bit data interface can have one of eight ID bits selected by installing 0 through 2 (3) jumpers in a binary coded configuration on the drive select header. Drives using the 16-bit data interface can have one of 16 ID bits selected by installing 0 through 3 (4) jumpers in a binary coded configuration on the drive select header. "LC" model drives (80-pin direct connect I/O connector) can be assigned their bus ID over the SCSI interface.

2.1.2 Signal values

Signals may assume true or false values. There are two methods of driving these signals. In both cases, the signal shall be actively driven true, or asserted. In the case of OR-tied drivers, the driver does not drive the signal to the false state, rather the bias circuitry of the bus terminators pulls the signal false whenever it is released by the drivers at every SCSI device. If any driver is asserted, then the signal is true. In the case of non-OR-tied drivers, the signal may be negated. Negated means that the signal may be actively driven false, or may be simply released (in which case the bias circuitry pulls it false), at the option of the implementor.

2.2 Signal states

2.2.1 SE signals

Signals may be in a true (asserted) or false (negated) state. Signals that are asserted are actively driven to the true state. Signals that are negated may either be actively driven to the false state or released to the false state. A signal that is released goes to the false state because the bias of the terminator pulls the signal false. OR-tied signals shall not be actively driven false.

Note. The advantage of actively negating signals false during information transfer is that the noise margin is higher than if the signal is simply released. This facilitates reliable data transfer at high transfer rates.

Bits of the data bus are defined as one when the signal is true and defined as zero when the signal is false.

2.2.2 LVD signals

Figure 3 defines the voltage and current definitions. A signal that is released goes to the false state because the bias of the terminator pulls the signal false.

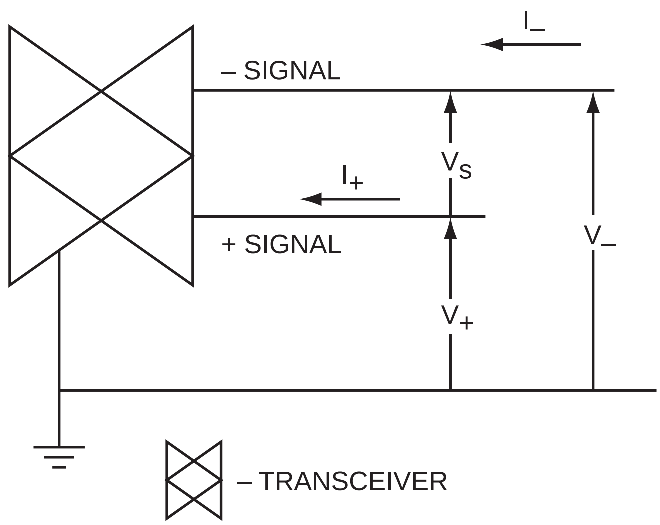


Figure 3. Voltage and current definitions

Figure 4 defines the signaling sense of the voltages appearing on the – signal and + signal lines as follows:

a. The – signal terminal of the driver shall be negative with respect to the + signal terminal for an asserted state.
b. The – signal terminal of the driver shall be positive with respect to the + signal terminal for a negated state.

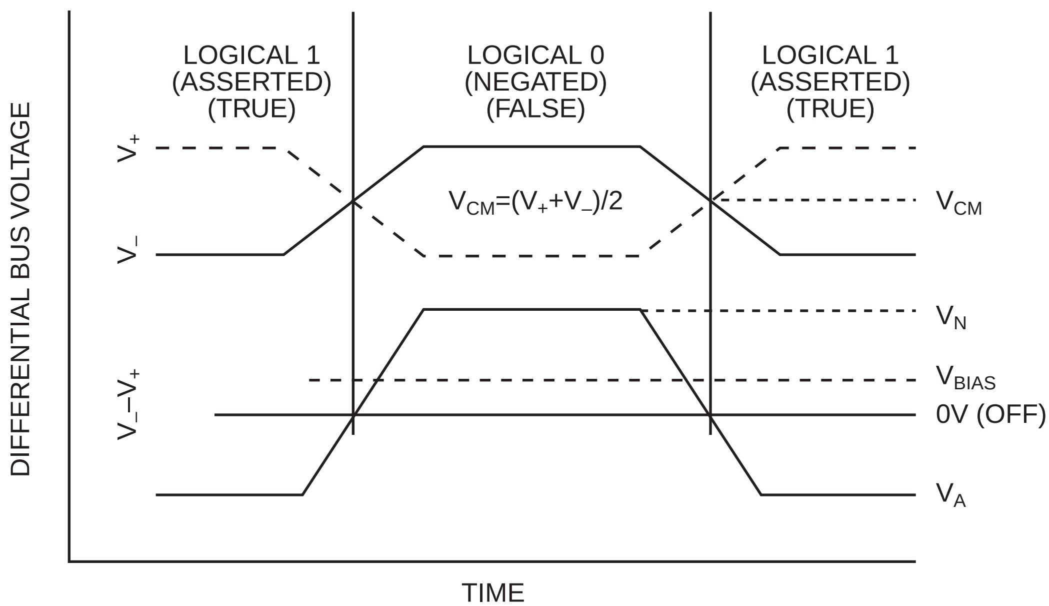


Figure 4. LVD Signaling sense

Note. For a description of V_{BIAS} see Section 7.3.1 of ANSI specification (SPI-5), T10/1525D.

2.3 OR-tied signals

The BSY, SEL, and RST signals shall be OR-tied. BSY and RST signals may be simultaneously driven true by several SCSI devices. No signals other than BSY, SEL, RST, DB(P_CRCA), and DB(P1) are simultaneously driven by two or more SCSI devices. DB(P_CRCA) and DB(P1) shall not be driven false during the ARBITRATION PHASE but may be driven false in other phases.

2.4 Signal sources

Table 3 indicates the type of SCSI device allowed to source each signal. No attempt is made to show if the source is driving asserted, driving negated, or is released. All SCSI device drivers that are not active sources shall be in the high-impedance state. The RST signal may be asserted by any SCSI device at any time.

Table 3: Signal sources

	16 data bit (P) cable						
	8 data bit (A) cable						
SCSI bus phase	**BSY**	**SEL**	**C/D, I/O, MSG, REQ**	**ACK, ATN**	**DB(7-0)**	**P_CRCA**	**DB(15-8), DB(P1)**
BUS FREE	None	None	None	None	None	None	None
ARBITRATION (NORMAL)	All	Win	None	None	S ID	S ID	S ID
QAS ARBITRATION	PT	Win	None	None	S ID	S ID	S ID
SELECTION	I & T	Init	None	Init	Init	Init	Init
RESELECTION	I & T	Targ	Targ	Init	Targ	Targ	Targ
COMMAND	Targ	None	Targ	Init	Init	Init	None
ST DATA IN	Targ	None	Targ	Init	Targ	Targ	Targ
ST DATA OUT	Targ	None	Targ	Init	Init	Init	Init
DT DATA IN	Targ	Targ	Targ	Init	Targ	Targ	Targ
DT DATA OUT	Targ	Targ	Targ	Init	Init	Targ	Init
STATUS	Targ	None	Targ	Init	Targ	Targ	None
MESSAGE IN	Targ	None	Targ	Init	Targ	Targ	None
MESSAGE OUT	Targ	None	Targ	Init	Init	Init	None

Table abbreviations are defined as follows:

All: The signal shall be driven by all SCSI devices that are actively arbitrating.

S ID: A unique data bit (the SCSI ID) shall be driven by each SCSI device that is actively arbitrating; the other data bits shall be released (i.e., not driven) by this SCSI device. The P_CRCA and DB(P1) bit(s) may be released or driven to the true state, but shall not be driven to the false state during this phase.

I&T: The signal shall be driven by the initiator, target, or both, as specified in the SELECTION PHASE and RESELECTION PHASE.

Init: If driven, this signal shall be driven only by the active initiator.

None: The signal shall be released; that is, not driven by any SCSI device. The bias circuitry of the bus terminators pulls the signal to the false state.

Win: The signal shall be driven by the one SCSI device that wins arbitration.

Targ: If the signal is driven, it shall be driven only by the active target.

PT: Target that initiated the QAS arbitration.

2.5 SCSI bus timing

Unless otherwise indicated, the delay time measurements for each SCSI device, defined in paragraphs 2.5.1 through 2.5.60 shall be calculated from signal conditions existing at that SCSI device's own SCSI bus connection. Thus, these measurements (except skew delay) can be made without considering delays in the cable. Refer to the tables below for the actual timing values for these delays.

Table 4: SCSI bus control timing values in nanoseconds

Timing description	Reference	Type	Timing value (ns unless noted)
Arbitration delay	2.5.1	minimum	2.4 µs
Bus clear delay	2.5.4	maximum	800
Bus free delay	2.5.5	minimum	800
Bus set delay	2.5.6	maximum	1.6 µs
Bus settle delay	2.5.7	minimum	400
Cable skew [1]	2.5.8	maximum	4
Data release delay	2.5.21	maximum	400
DIFFSENS voltage filter time	2.5.22	minimum	100 ms
Physical disconnection delay	2.5.24	minimum	200 µs
Power on to selection [2]	2.5.25	maximum	10 s
QAS arbitration delay	2.5.26	minimum	1000
QAS assertion delay	2.5.27	maximum	200
QAS release delay	2.5.28	maximum	200
QAS non-data phase REQ(ACK) period	2.5.29	minimum	50
Reset delay	2.5.42	minimum	200
Reset hold time	2.5.43	minimum	25 µs
Reset to selection [2]	2.5.44	maximum	250 ms
Selection abort time	2.5.46	maximum	200 µs
Selection time-out delay [2]	2.5.47	minimum	250 ms
System deskew delay	2.5.51	minimum	45

[1] Cable skew is measured at each SCSI device connection with the transmitted skew subtracted from the received skew.

[2] This is a recommended time. It is not mandatory.

Table 5: SCSI bus data and information phase ST timing values

Timing description	Reference	Type	Timing values for negotiated transfer rate in nanoseconds unless otherwise noted [1]				
			Asynch	Fast-5	Fast-10	Fast-20	Fast-40
ATN transmit setup time	2.5.2	min	90	33	33	21.5	19.25
ATN receive setup time	2.5.3	min	45	17	17	8.5	6.75
Cable skew [2]	2.5.8	max	4.0	4.0	4.0	3.0	2.5
Receive assertion period [3]	2.5.30	min	N/A	70	22	11	6.5
Receive hold time [3] [4]	2.5.31	min	N/A	25	25	11.5	4.75
Receive negation period [3]	2.5.34	min	N/A	70	22	11	6.5
Receive setup time [3] [4]	2.5.35	min	N/A	15	15	6.5	4.75
Receive REQ(ACK) period tolerance	2.5.36	min	N/A	1.1	1.1	1.1	1.1
Signal timing skew	2.5.48	max	8.0	8.0	8.0	5.0	4.5
REQ(ACK) period	2.5.41	nom	N/A	200	100	50	25
Transmit assertion period [2]	2.5.55	min	N/A	80	30	15	8.0
Transmit hold time [3] [4]	2.5.56	min	N/A	53	33	16.5	9.25
Transmit negation period [3]	2.5.58	min	N/A	80	30	15	8.0
Transmit setup time [3] [4]	2.5.59	min	N/A	23	23	11.5	9.25
Transmit REQ(ACK) period tolerance	2.5.60	max	N/A	1.0	1.0	1.0	1.0

[1] SCSI bus timing values specified by the maximum transfer rate for the given range shall apply even if a slower transfer rate within the given range is negotiated.
[2] Cable skew is measured at each SCSI device connection within the same bus segment with the transmitted skew subtracted from the received skew.
[3] See Section 2.6 for measurement points for the timing specifications.
[4] See Section 9.6 in the ANSI SCSI Parallel Interface-5 (SPI-5) specification for examples of how to calculate setup and hold timing.

Table 6: Miscellaneous SCSI bus data and information phase DT timing values

Timing description	Reference	Type	Timing values for negotiated transfer rate in nanoseconds unless otherwise noted [1]					
			Fast-10	Fast-20	Fast-40	Fast-80	Fast-160	Fast-320
Cable skew [2]	2.5.8	max	4.0	3.0	2.5	2.5	2.5	2.5
REQ(ACK) period	2.5.41	nom	200	100	50	25	12.5	6.25
Residual skew error [3]	2.5.45	max	N/A	N/A	N/A	N/A	±0.15	± 0.20
Skew correction range [4]	2.5.49	min	N/A	N/A	N/A	N/A	±3.45 [5]	± 3.45 [5]
Signal timing skew	2.5.48	max	26.8	13.4	6.7	3.35	4.85	4.85
Strobe offset tolerance	2.5.50	max	N/A	N/A	N/A	N/A	±0.125	±0.125

[1] SCSI bus timing values specified by the maximum transfer rate for the given range shall apply even if a slower transfer rate within the given range is negotiated.

[2] Cable skew is measured at each SCSI device connection within the same bus segment with the transmitted skew subtracted from the received skew.
[3] Calculated assuming timing budget shown in Table 9.
[4] Measured at the receiver terminal using clean input signals with 500 mV peak amplitude and 1 ns rise and fall time between 20% and 80% of the signal.
[5] Relative to the REQ(ACK) clocking signal.

Note. Fast-160 SCSI devices shall not change timing parameters between training (see Section 3.5.3.1) or reset events (see Section 5.4).

Table 7: Transmit SCSI bus data and information phase DT timing values

Timing description	Reference	Type	Timing values for negotiated transfer rate in nanoseconds unless otherwise noted [1]					
			Fast-10	Fast-20	Fast-40	Fast-80	Fast-160	Fast-320
ATN transmit setup time	2.5.2	min	48.4	29.2	19.6	14.8	14	14
Flow control transmit hold time	2.5.15	min	38.4	19.2	9.6	4.8	14	14
Flow control transmit setup time	2.5.16	min	48.4	29.2	19.6	14.8	14	14
pCRC transmit hold time	2.5.19	min	38.4	19.2	9.6	4.8	N/A	N/A
pCRC transmit setup time	2.5.20	min	48.4	29.2	19.6	14.8	N/A	N/A
Transmit assertion period [2]	2.5.55	min	92	46	23	11.5	5.69	2.565
Transmit hold time [2] [3]	2.5.56	min	38.4	19.2	9.6	4.8	4.77	0.365
Transmit ISI compensation	2.5.57	max					1.0	0
Transmit negation period [2]	2.5.58	min	92	46	23	11.5	5.69	2.565
Transmit REQ(ACK) period tolerance	2.5.60	max	0.6	0.6	0.6	0.6	0.06	0.06
Transmit REQ assertion period with P_CRCA transitioning	2.5.61	min	97.5	54	35.5	24	N/A	N/A
Transmit setup time [2] [3]	2.5.59	min	38.4	19.2	9.6	4.8	1.48	-2.76
Transmitter skew	2.5.63	max	N/A	N/A	N/A	N/A	± 0.75	± 0.75
Transmitter time asymmetry	2.5.64	max	N/A	N/A	N/A	N/A	± 0.25	± 0.25

[1] SCSI bus timing values specified by the maximum transfer rate for the given range shall apply even if a slower transfer rate within the given range is negotiated.

[2] See Section 2.6 for measurement points for the timing specifications.

[3] See Section 9.6 in the ANSI SCSI Parallel Interface-5 (SPI-5) specification for examples of how to calculate setup and hold timing.

Note. Fast-160 and fast 320 SCSI devices shall not change timing parameters between training or reset events.

Table 8: Receive SCSI bus data and information phase DT timing values

Timing description	Reference	Type	Timing values for negotiated transfer rate in nanoseconds unless otherwise noted [1]					
			Fast-10	Fast-20	Fast-40	Fast-80	Fast-160	Fast-320
ATN receive setup time	2.5.3	min	13.6	7.8	4.9	3.45	3.0	3.0
Flow control receive hold time	2.5.13	min	11.6	5.8	2.9	1.45	3.0	3.0
Flow control receive setup time	2.5.14	min	18.6	12.8	9.9	8.45	3.0	3.0
pCRC receive hold time	2.5.17	min	11.6	5.8	2.9	1.45	N/A	N/A
pCRC receive setup time	2.5.18	min	18.6	12.8	9.9	8.45	N/A	N/A
Receive assertion period [2]	2.5.30	min	80	40	20	8.5	4.74	1.615
Receive hold time [2] [3]	2.5.31	min	11.6	5.8	2.9	1.45	-0.08	-6.635
Receive negation period [2]	2.5.34	min	80	40	20	8.5	4.74	1.615
Receive setup time [2] [3]	2.5.35	min	11.6	5.8	2.9	1.45	-6.33	-9.76
Receive REQ(ACK) period tolerance	2.5.36	min	0.7	0.7	0.7	0.7	0.06	0.06
Receive REQ assertion period with P_CRCA transitioning	2.5.37	min	85.5	48	32.5	21	N/A	N/A
Receive REQ negation period with P_CRCA transitioning	2.5.38	min	85.5	48	32.5	21	N/A	N/A
Receive skew compensation	2.5.39	max	N/A	N/A	N/A	N/A	N/A	N/A
Receive internal hold time [4]	2.5.32	min	N/A	N/A	N/A	N/A	0.345	0.032
Receive internal setup time [4]	2.5.33	min	N/A	N/A	N/A	N/A	0.345	0.032
Receiver de-skewed data valid window [4]	2.5.12	min	N/A	N/A	N/A	N/A	±2.1	± 1.1

[1] SCSI bus timing values specified by the maximum transfer rate for the given range shall apply even if a slower transfer rate within the given range is negotiated.
[2] See Section 2.6 for measurement points for the timing specifications.
[3] See Section 9.6 in the ANSI SCSI Parallel Interface-5 (SPI-5) specification for examples of how to calculate setup and hold timing.
[4] Calculated assuming timing budget shown in Table 9.

Note. Fast-160 and fast 320 SCSI devices shall not change timing parameters between training or reset events.

Table 9: SCSI Fast-160 and fast 320 non-compensatable timing budget in nanoseconds

Item	Fast-160		Fast-320		Comments
REQ(ACK) period	12.5		6.25		From Table 7
Transfer period	6.25		3.125		REQ(ACK) period / 2
Transmitter and receiver errors:	**Total Errors**	**Post Compensation**	**Total Errors**	**Post Compensation**	**Worst case total of + and - time shift except where noted**
Transmitter errors:					
REQ(ACK) period tolerance / 2	0.06	0.06	0.06	0.06	Tolerance of transmitter plus measurement error [b]
Clock Jitter	0.25	0.25	0.25	0.50	
System noise at transmitter	0.25	0.25	0.2	0.2	Time impact
Transmitter chip skew	0.75	0	0.75	0	
Transmitter trace skew	0.2	0	0.2	0	
Transmit time asymmetry	0.50	0.50	0.50	0	Compensated for on fast-320 SCSI devices
Total transmitter error budget:	**2.01**	**1.06**	**1.96**	**0.51**	
Receiver errors:					
System noise at receiver	0.25	0.25	0.2	0.2	Time impact
Chip noise in receiver	0.2	0.2	0.2	0.2	Time impact
Receiver chip skew	0.75	0	0.75	0	
Receiver trace skew	0.2	0	0.2	0	
Receiver time asymmetry	0.35	0.35	0.35	0	
Residual skew error	0	0.3	0	0.2	After skew compensation
Strobe offset tolerance	0	0.5	0	0.2	Accuracy of centering strobe
Offset induced time asymmetry	N/A [a]	N/A [a]	0.8	0.2	Time impact from cumulative D.C. offsets at receiver
Receiver amplitude time skew	0.2	0.2	0.2	0.05	
Total receiver error budget:	**1.95**	**1.8**	**2.7**	**1.05**	
Total Transmitter + receiver error budget:	**3.96**	**2.86**	**4.44**	**1.53**	
Total timing error budget for interconnect and system margin:	**2.29**	**3.39**	**-1.535**	**1.565**	Transfer period - (total transmitter + total receiver error budget)

a Timing budgets in some previous standards neglected asymmetry & detection ambiguity and lumps chip noise, clock jitter, crosstalk time shift, noise, ISI and receiver amplitude skew into other terms (e.g., signal distortion skew) and/or ignores the effects.

b Tolerance adjusted for half cycle (i.e., transfer period)

Table 10: SCSI fast-160 and fast-320 interconnect timing budget in nanoseconds

Item		Fast-160		Fast-320		Comments
Nominals:						
	REQ(ACK) period	12.5		6.25		from table 6
	Transfer period	6.25		3.125		REQ(ACK) period/2
Transmitter and receiver errors:		**Total Errors**	**Post Compensation**	**Total Errors**	**Post Compensation**	**Worst case total of + and - time shift except where noted**
Interconnect errors:						
	Cable skew [a b]	2.5	0	2.5	0	
	Crosstalk time shift	0.7	0.7	0.5	0.5	Time impact
	ISI of data	4.0	2.0	4.0	1.0	Worse case pattern
Total interconnect budget		**7.2**	**2.7**	**7.0**	**1.5**	

a **See 2.5.8.**

b **The residual deskew error is included in the receiver error budget.**

Note. For more information on interconnect errors see the SCSI Passive Interconnect Performance standard.

2.5.1 Arbitration delay

The minimum time a SCSI device shall wait from asserting BSY for arbitration until the data bus can be examined to see if arbitration has been won (see Section 3.1.2). There is no maximum time.

2.5.2 ATN transmit setup time

When information unit transfers are not being used, the ATN transmit setup time is the minimum time provided by the transmitter between the assertion of the ATN signal and the last negation of the ACK signal in any phase.

When information unit transfers are being used with synchronous transfers, the ATN transmit setup time is the minimum time provided by the transmitter between the assertion of the ATN signal and the negation of the ACK signal corresponding to the last iuCRC transfer of an information unit.

When information unit transfers are being used with paced transfers, the ATN transmit setup time is the minimum time provided by the transmitter between the assertion of the ATN signal and the assertion of the ACK signal corresponding to the last iuCRC transfer of an information unit.

Specified to provide the increased ATN receive setup time, subject to intersymbol interference, cable skew, and other distortions.

2.5.3 ATN receive setup time

When information unit transfers are not being used, the ATN receive setup time is the minimum time required at the receiver between the assertion of the ATN signal and the last negation of the ACK signal in any phase to recognize the assertion of an attention condition.

When information unit transfers are being used with synchronous transfers, the ATN receive setup time is the minimum time required at the receiver between the assertion of the ATN signal and the negation of the ACK signal corresponding to the last iuCRC transfer of an information unit to recognize the assertion of an attention condition.

When information unit transfers are being used with paced transfers, the ATN receive setup time is the minimum time required at the receiver between the assertion of the ATN signal and the assertion of the ACK signal corresponding to the last iuCRC transfer of an information unit to recognize the assertion of an attention condition.

2.5.4 Bus clear delay

The maximum time for a SCSI device to stop driving all bus signals after:

1. The BUS FREE phase is detected (i.e., the BSY and SEL signals are both false for a bus settle delay).
2. The SEL signal is received from another SCSI device during the ARBITRATION phase.
3. The transition of the RST signal to true.

For item 1 above, the maximum time for a SCSI device to release all SCSI bus signals is 1200 ns from the BSY and SEL signals first becoming both false. If a SCSI device requires more than a bus settle delay to detect BUS FREE phase, it shall release all SCSI bus signals within a bus clear delay minus the excess time.

2.5.5 Bus free delay

The minimum time that a SCSI device shall wait from its detection of the BUS FREE phase (i.e., BSY and SEL both false for a bus settle delay) until its assertion of the BSY signal in preparation for entering the ARBITRATION phase.

2.5.6 Bus set delay

The maximum time for a SCSI device to assert the BSY signal and its SCSI ID after it detects a BUS FREE phase for the purpose of entering the ARBITRATION phase.

2.5.7 Bus settle delay

The minimum time to wait for the bus to settle after changing certain control signals as specified in the protocol definitions. Provides time for a signal transition to propagate from the driver to the terminator and back to the driver.

2.5.8 Cable skew delay

The maximum difference in propagation time allowed between any two SCSI bus signals when measured between any two SCSI devices excluding any signal distortion skew delays.

2.5.9 Chip noise in receiver

The maximum transition to transition time shift due to the internal physics of the receiving SCSI device circuitry.

2.5.10 Clock jitter

The maximum transition to transition time shift of SCSI bus signals caused by short term variations in the transmitting SCSI device's clock.

2.5.11 Crosstalk time shift

The peak-to-peak timeshift error on DB(0-15), P_CRCA, or DB(P1) caused by transitions on all other DB(0-15), P_CRCA, or DB(P1) signals.

2.5.12 De-skewed data valid window

The minimum difference in time allowed between the rising or falling edge of a “1010...” pattern on the DAT BUS or DB(P1) and its clocking signal on the ACK or REQ signal as measured at their zero-crossing points after skew compensation is applied by the receiver without allowing any error in the received data. The de-skewed data valid window shall be equal to:

+/– [(data transfer period) – (residual skew error) – (strobe offset tolerance) – (clock jitter) – (receiver amplitude skew) – (ship noise) – (system noise at receiver) – (receiver asymmetry)] / 2.

2.5.13 Flow control receive hold time

The maximum time required by the initiator between the assertion of the REQ signal corresponding to the last iuCRC transfer of a SPI data streaming information unit and the changing of the P_CRCA signal.

2.5.14 Flow control receive setup time

The maximum time required by the initiator between the assertion of the P_CRCA signal and the assertion of the REQ signal corresponding to the last iuCRC transfer of a SPI data streaming information unit. Also, the maximum time required by the initiator between the negation of the P_CRCA signal and the assertion of the REQ signal corresponding to any valid data transfer of a SPI L_Q information unit.

2.5.15 Flow control transmit hold time

The minimum time provided by the target between the assertion of the REQ signal corresponding to the last iuCRC transfer of a SPI data stream information unit and the changing of the P_CRCA signal. Specified to provide the increased P_CRCA receive setup time, subject to intersymbol interference, cable skew, and other distortions.

2.5.16 Flow control transmit setup time

The minimum time provided by the target between the assertion of the P_CRCA signal and the assertion of the REQ signal corresponding to the last iuCRC transfer of a SPI data streaming information unit. Also, the minimum time provided by the target between the negation of the P_CRCA signal and the assertion of the REQ signal corresponding to any valid data transfer of a SPI L_Q information unit. Specified to provide the increased P_CRCA receive setup time, subject to intersymbol interference, cable skew, and other distortions.

2.5.17 pCRC receive hold time

The minimum time required at the receiver between the transition of the REQ signal and the transition of the P_CRCA signal during data group transfers.

2.5.18 pCRC receive setup time

The minimum time required at the receiver between the transition of the P_CRCA signal and the transition of the REQ signal during data group transfers. Specified to ease receiver timing requirements and ensure that this signal, that is outside CRC protection, is received correctly.

2.5.19 pCRC transmit hold time

The minimum time provided by the transmitter between the transition of the REQ signal and the transition of the P_CRCA signal during data group transfers.

2.5.20 pCRC transmit setup time

The minimum time provided by the transmitter between the transition of the P_CRCA signal and the transition of the REQ signal during data group transfers. Specified to provide the increased receive setup time, subject to intersymbol interference, cable skew, and other distortions.

2.5.21 Data release delay

The maximum time for a SCSI initiator port to release the DATA BUS, DB(P_CRCA), and/or DB(P1) signals, following the transition of the I/O signal from false to true.

2.5.22 DIFFSENS voltage filter time

The minimum time DIFFSENS voltage shall be sensed continuously within the voltage range of a valid SCSI bus mode.

2.5.23 Offset induced time asymmetry

Time symmetry error created by the cumulative sum of all offset errors seen by the receiver. This includes non-symmetrical transmitter drive plus terminator current mismatch, receiver offset, and voltage drop due to resistance in the interconnect within the cable or backplane.

2.5.24 Physical disconnection delay

The minimum time that a SCSI target port shall wait after releasing BSY before participating in an ARBITRATION phase when honoring a DISCONNECT MESSAGE from the initiator.

2.5.25 Power on to selection

The recommended maximum time from power application until a SCSI target is able to respond with appropriate status and sense data to the TEST UNIT READY, INQUIRY, and REQUEST SENSE commands (see ANSI SCSI Primary Commands-4 standard).

2.5.26 QAS arbitration delay

The minimum time a SCSI device with QAS enabled shall wait from the detection of the MSG, C/D, and I/O signals being false to start QAS until the data bus is examined to see if QAS has been won (see Section 4.3.12).

2.5.27 QAS assertion delay

The maximum time allowed for a SCSI device to assert certain signals during QAS.

2.5.28 QAS release delay

The maximum time allowed for a SCSI device to release certain signals during QAS.

2.5.29 QAS non-data phase REQ(ACK) period

The minimum time a QAS-capable initiator shall ensure the REQ and ACK signals are asserted and that the data is valid during the COMMAND, MESSAGE, and STATUS phases.

2.5.30 Receive assertion period

The minimum time required at a SCSI device receiving a REQ signal for the signal to be asserted while using synchronous transfers or paced transfers, provided P_CRCA is not transitioning during data group transfers. Also, the minimum time required at a SCSI device receiving an ACK signal for the signal to be asserted while using synchronous transfers or paced transfers. For SE Fast-5 and Fast-10 operation, the time period is measured at the 0,8 V level. For SE Fast-20 operation, the period is measured at the 1,0 V level. For LVD, see SPI-5 Section 9, for signal measurement points.

2.5.31 Receive hold time

For ST data transfers, the minimum time required at the receiving SCSI device between the assertion of the REQ signal or the ACK signals and the changing of the Data Bus, DB(P_CRCA), and/or DB(P1) signals while using synchronous data transfers, provided P_CRCA is not transitioning during data group transfers. For DT data transfers, the minimum time required at the receiving SCSI device between the transition (i.e., assertion or negation) of the REQ signal or the ACK signals and the changing of the data bus, DB(P_CRCA), and/or

DB(P1) signals. For paced data transfers negative values as measured at the device connector are accommodated by skew compensation in the receiver. Receive hold time measured at the device connector shall not exceed the skew correction range.

2.5.32 Receive internal hold time

The minimum time provided for hold time in the receive data detector after allowance for timing errors and timing compensation from all sources measured from the worst-case bit (i.e., data or parity) to the compensated offset strobe.

Note. This time may not be observable to other than the SCSI device designer.

2.5.33 Receive internal setup time

The minimum time provided for setup time in the receive data detector after allowance for timing errors and timing compensation from all sources measured from the worst-case bit (i.e., data or parity) to the compensated offset strobe.

Note. This time may not be observable to other than the SCSI device designer.

2.5.34 Receive negation period

The minimum time required at a SCSI device receiving a REQ signal for the signal to be negated while using synchronous transfers or paced transfers. Also, the minimum time required at a SCSI device receiving an ACK signal for the signal to be asserted while using synchronous transfers or paced transfers. For SE Fast-5 and Fast-10 operation, the time period is measured at the 2,0 V level. For SE Fast-20 operation the period is measured at the 1,9 V level. For LVD, see SPI-5 Section 9, for signal measurement points.

2.5.35 Receive setup time

For ST data transfers, the minimum time required at the receiving SCSI device between the changing of Data Bus, DB(P_CRCA), and/or DB(P1) signals and the assertion of the REQ signal or the ACK signal while using synchronous data transfers. For DT data transfers, the minimum time required at the receiving SCSI device between the changing of Data Bus, DB(P_CRCA), and/or DB(P1) signals and the transition of the REQ signal or the ACK signal. For paced data transfers negative values as measured at the device connector are accommodated by skew compensation in the receiver. Receive setup time measured at the device connector shall not exceed the skew correction range.

2.5.36 Receive REQ(ACK) period tolerance

The minimum tolerance that a SCSI device shall allow to be subtracted from the REQ(ACK) period. The tolerance comprises the transmit REQ(ACK) tolerance plus a measurement error due to noise.

2.5.37 Receive REQ assertion period with P_CRCA transitioning

The minimum time required at a SCSI device receiving a REQ signal for the signal to be asserted while P_CRCA is transitioning during data group transfers. Specified to ensure that the assertion period is longer than the receive hold time plus the receive setup time.

2.5.38 Receive REQ negation period with P_CRCA transitioning

The minimum time required at a SCSI device receiving a REQ signal for the signal to be negated while P_CRCA is transitioning during data group transfers. Specified to ensure that the negation period is longer than the receive hold time plus the receive setup time.

2.5.39 Receive skew compensation

The effective reduction in worst-case timing skew of data, parity, and strobe signals provided by the receiving SCSI device but not directly observable at the receiving SCSI device connector.

2.5.40 Receiver amplitude time skew

The maximum time shift of SCSI bus signals caused by the difference in receiver switching delay of a minimum amplitude signal versus a maximum amplitude signal.

2.5.41 REQ(ACK) period

The REQ(ACK) period during synchronous transfers or paced transfers is the nominal time between adjacent assertion edges of the REQ or ACK signal for the fastest negotiated transfer rate. For the purpose of calculating the actual REQ(ACK) period tolerance the REQ(ACK) period should be measured without interruptions (e.g., offsets pauses). To minimize the impact of crosstalk and ISI the measurements should be made by averaging the time between edges during long (i.e., greater than 512 bytes) all zeros or all ones data transfers and by ignoring the first and last 10 transitions.

In DT DATA phases, the negotiated transfer period for data is half that of the REQ(ACK) period since data is qualified on both the assertion and negation edges of the REQ or ACK signal. In ST DATA phases, the negotiated transfer period for data is equal to the REQ(ACK) period during synchronous transfers since data is only qualified on the assertion edge of the REQ or ACK signal.

2.5.42 Reset delay

The minimum time that the RST signal shall be continuously true before the SCSI device shall initiate a hard reset.

2.5.43 Reset hold time

The minimum time that the RST signal is asserted. There is no maximum time.

2.5.44 Reset to selection

The recommended maximum time from after a reset condition until a SCSI target is able to respond with appropriate status and sense data to the TEST UNIT READY, INQUIRY, and REQUEST SENSE commands (see SCSI Primary Commands-4 Standard).

2.5.45 Residual skew error

The maximum timing error between the deskewed data and REQ or ACK internal to the receiving SCSI device after skew compensation.

2.5.46 Selection abort time

The maximum time that a SCSI device shall take from its most recent detection of being selected or reselected until asserting the BSY signal in response. This timeout is required to ensure that a SCSI target port or initiator does not assert the BSY signal after a SELECTION or RESELECTION phase has been aborted.

2.5.47 Selection timeout delay

The minimum time that a SCSI initiator port or target should wait for the assertion of the BSY signal during the SELECTION or RESELECTION phase before starting the timeout procedure. Note that this is only a recommended time period.

2.5.48 Signal timing skew

The maximum signal timing skew occurs when transferring random data and in combination with interruptions of the REQ or ACK signal transitions (e.g., pauses caused by offsets). The signal timing skew includes cable skew (measured with 0101...) patterns and signal distortion skew caused by random data patterns and transmission line reflections as shown in ANSI standard SPI-5, T10/1525D. The receiver detection range is the part of the signal between the “may detect” level and the “shall detect” level on either edge (see Section 8.3).

Note. For timing budget purposes the value stated in Table 6 is calculated without the benefit of skew compensation.

2.5.49 Skew correction range

The minimum skew correction capability of the receiver of a signal on the DATA BUS or DB(P1) relative to the ACK or REQ signal as measured at the receiver's connector. The skew correction range shall be equal to:

+/– [(transmitter chip skew) + (cable skew) + (two times trace skew)]

relative to the corresponding ACK or REQ clock signal for that transition. Receiver chip skew is not included, as it is internal to the receiver.

2.5.50 Strobe offset tolerance

The time tolerance of centering the compensated REQ or ACK strobe in the transfer period during the training pattern.

2.5.51 System deskew delay

The minimum time that a SCSI device should wait after receiving a SCSI signal to ensure that asynchronous transfers at the same time are valid. The system deskew delay shall not be applied to the synchronous transfers or paced transfers.

2.5.52 System noise at launch

The maximum time shift of SCSI bus signals caused by system noise at the transmitter (e.g., noise caused by current changes in the voice coil) measured at the transmitting SCSI device connector.

2.5.53 System noise at receiver

The maximum time shift of SCSI bus signals caused by system noise at the receiver (e.g., noise caused by current changes in the voice coil) measured at the receiving SCSI device connector not including the time shift from the system noise at launch.

2.5.54 Time asymmetry

The maximum time difference between the asserted and negated signal for data, REQ, or ACK transitions that are intended to be equidistant.

2.5.55 Transmit assertion period

The minimum time that a SCSI target port shall assert the REQ signal while using synchronous transfers or paced transfers, provided it is not transitioning P_CRCA during data group transfers. Also, the minimum time that a SCSI initiator port shall assert the ACK signal while using synchronous transfers or paced transfers.

2.5.56 Transmit hold time

For ST data transfers, the minimum time provided by the transmitting SCSI device between the assertion of the REQ signal or the ACK signal and the changing of the Data Bus, DB(P_CRCA), and/or DB(P1) while using synchronous data transfers. For DT data transfers, the minimum time provided by the transmitting SCSI device between the transition of the REQ signal or the ACK signal and the changing of the Data Bus, DB(P_CRCA), and/or DB(P1).

2.5.57 Transmit ISI compensation

The effective reduction in worst-case ISI timing shift provided by the transmitting SCSI device as seen at the receiving SCSI device connector.

2.5.58 Transmit negation period

The minimum time that a SCSI target port shall negate the REQ signal while using synchronous transfers or paced transfers, provided it is not transitioning P_CRCA during data group transfers. Also, the minimum time that a SCSI initiator port shall negate the ACK signal while using synchronous transfers or paced transfers.

2.5.59 Transmit setup time

For ST data transfers, the minimum time provided by the transmitting SCSI device between the changing of Data Bus, DB(P_CRCA), and/or DB(P1) and the assertion of the REQ signal or the ACK signal while using synchronous data transfers. For DT data transfers, the minimum time provided by the transmitting SCSI device between the changing of Data Bus, DB(P_CRCA), and/or DB(P1) and the transition of the REQ signal or the ACK signal.

2.5.60 Transmit REQ(ACK) period tolerance

The maximum tolerance that a SCSI device may subtract from the REQ(ACK) period.

2.5.61 Transmit REQ assertion period with P_CRCA transitioning

The minimum time that a SCSI target port shall assert the REQ signal while transitioning P_CRCA during data group transfers. Specified to provide the increased receive REQ assertion period, subject to loss on the interconnect.

2.5.62 Transmit REQ negation period with P_CRCA transitioning

The minimum time that a SCSI target port shall negate the REQ signal while transitioning P_CRCA during data group transfers. Specified to provide the increased receive REQ negation period, subject to loss on the interconnect.

2.5.63 Transmitter skew

The maximum difference in time allowed between the rising or falling edge of a “1010...” pattern on the DATA BUS or DB(P1) signal and its clocking signal on the ACK or REQ signal as measured at their zero-crossing points. The signals for the output waveforms shall be measured at the connector of the transmitting device.

2.5.64 Transmitter time asymmetry

The maximum time on DATA BUS, DB(P1), ACK, or REQ signal from any transition edge to the subsequent transition edge during a “1010...” pattern, as measured at their zero-crossing points, minus the data transfer period. The signals for the output waveforms shall be measured at the connector of the transmitting device.

2.6 Measurement points

The measurement points for low voltage differential (LVD) ACK, REQ, DATA, P_CRCA, and PARITY signals are defined in ANSI standard SPI-5 section 9.3. All measurements are at the SCSI connector.

When paced transfers are enabled, the timing shall be measured relative to the zero crossing of the differential signal. In paced transfers the timing budget and receiver masks account for the differences between the setup and hold detection thresholds that occur for synchronous transfers.

2.6.1 SE Fast-5 and Fast-10 measurement points

SE SCSI devices with data transfer rates up to and including Fast-10 shall use the measurement points defined in Section 9 of ANSI standard SPI-4, for the measurement of the timing values. The rise and fall times for the SE REQ/ACK signals shall be nominally the same as for the SE Data, DB(P_CRCA), and DB(P1) signals.

2.6.2 SE Fast-20 measurement points

SE SCSI devices with data transfer rates up to and including Fast-20 shall use the measurement points defined in Section 9 of ANSI standard SPI-4, for the measurement of the timing values. The rise and fall times for the SE REQ/ACK signals shall be nominally the same as for the SE Data, DB(P_CRCA), and DB(P1) signals.

SE Fast-20 timing measurement points shall apply even if a slower transfer rate is negotiated.

SE SCSI devices are not capable of Fast-40 and Fast-80 data transfer rates. LVD circuits are required for these faster transfer rates.

2.6.3 LVD measurement points

When transferring data using ST LVD phases, LVD SCSI devices shall use the measurement points defined in Section 9 Figure 60 of ANSI standard SPI-5, for the measurement of timing values. When transferring data using DT DATA phases, LVD SCSI devices shall use the measurement points defined in Section 9 Figure 61 of SPI-5, for the measurement of timing values. When transferring data using DT DATA phases LVD SCSI devices using paced transfers shall use the measurement points defined in Section 9 Figure 62 of SPI-5 for the measurement of the timing values. The rise and fall times for the LVD REQ/ACK signals shall be nominally the same as for the LVD Data, P_CRCA, and DP-1 signals.

2.7 Clocking methods for data transfers

This manual defines optional methods of latching data from the REQ and ACK signals depending on whether ST DATA phases or DT DATA phases are being used for information transfers as shown in Figure 5. Data shall only be latched on the asserting edge of the REQ or ACK signal except in DT DATA phases. When DT DATA phases are used, data shall be latched on both the asserting edge and the negating edge of the REQ or ACK signal.

Regardless of whether ST or DT transfers are enabled, the negotiated transfer period sets the maximum rate at which the data is clocked at in megatransfers per second. As a result, the time from rising edge to rising edge for REQ and ACK signals for the same transfer rate is twice as long for a DT transfer as it is for an ST transfer. An example of a negotiated transfer period of 25 ns with ST transfers is shown in figure 6. An example of a negotiated transfer period of 25 ns with DT transfers is shown in Figure 7. For ST and DT synchronous transfers the clocking signal (i.e., REQ or ACK) occurs when the DATA BUS is in a steady state.

Figure 8 shows an example of transfers with a negotiated transfer period of 6.25 ns at the receiving SCSI device's connector. There is no difference as to when data is latched on paced transfers, however, the relationship between the data and REQ or ACK is required to be adjusted in the SCSI devices receiver to match the synchronous transfers DT Data shown in Figure 5. For paced transfers, the clocking signal (i.e., REQ or ACK) may occur when the data bus is changing state.

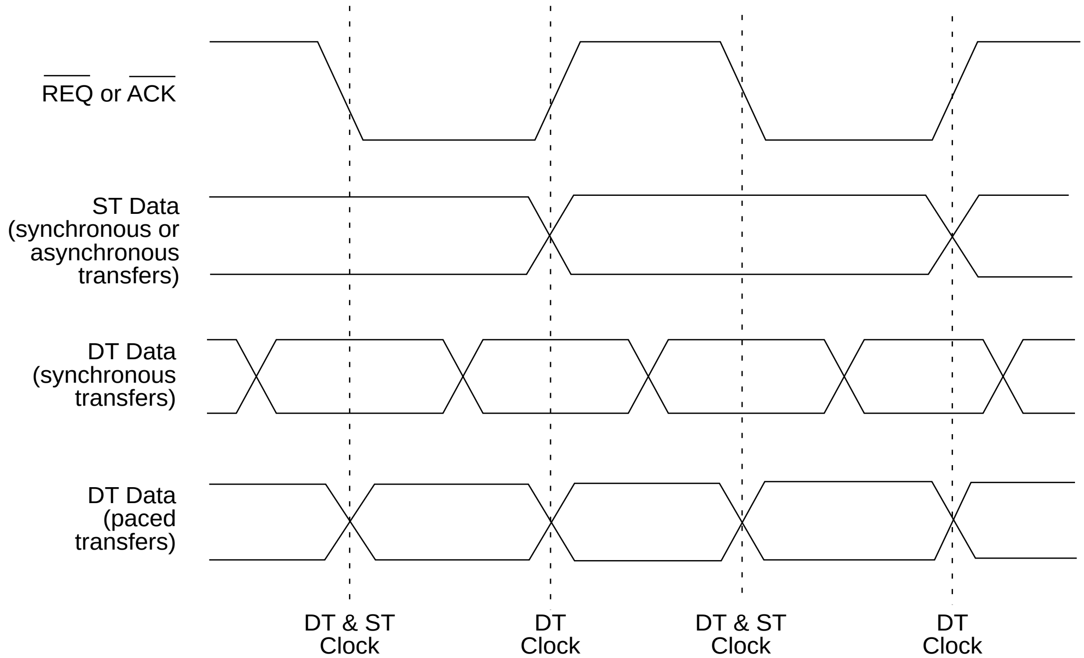


Figure 5. ST latching data vs. DT latching data

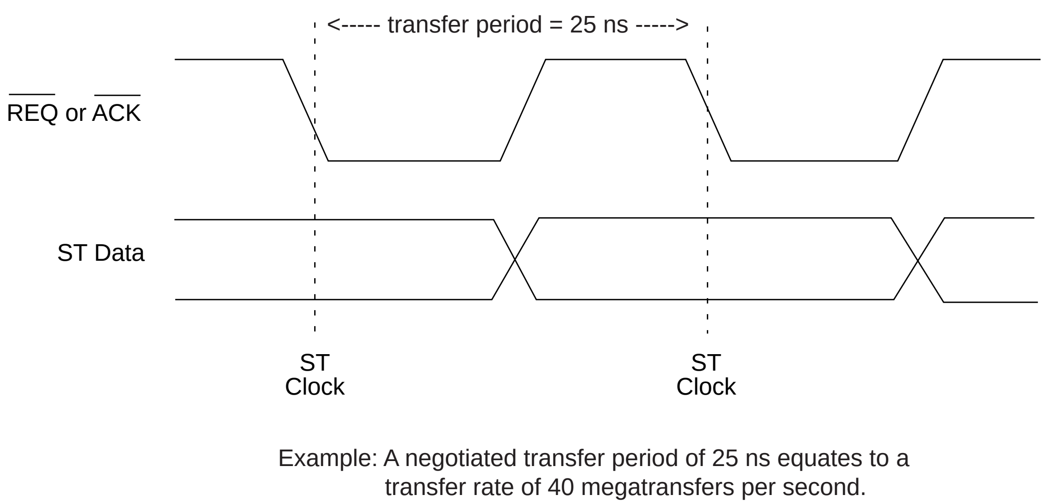
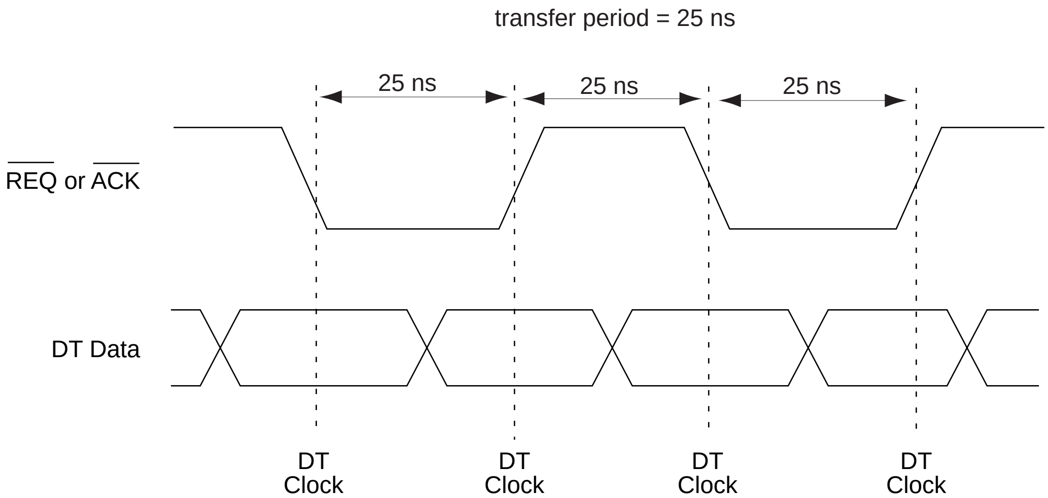
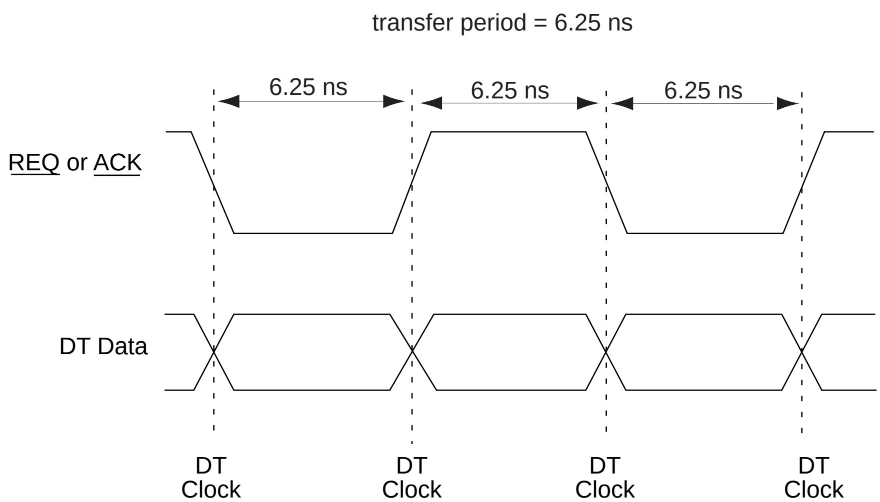


Figure 6. ST synchronous transfer example



Example: A negotiated transfer period of 25 ns equates to a transfer rate of 40 megatransfers per second.

Figure 7. DT synchronous transfer example



Example: A negotiated transfer period of 6.25 ns equates to a transfer rate of 160 megatransfers per second.

Figure 8. Paced transfer example

2.8 Paced transfer on a SCSI bus

A SCSI bus that supports paced transfers has additional driver and receiver functions required over those used with synchronous transfers or asynchronous transfers. These functions include driver precompensation, receiver skew compensation, receiver clock shifting, and an optional receiver signal adjustment. In addition, the drive precompensation may be switched out of the data path at the request of the receiving SCSI device.

The receiver skew compensation and clock shifting adjust the timing relationship between the clocking signal (i.e., REQ or ACK) and the signals being clocked (e.g., the data bus signals). That adjustment causes the clock signal to align with the middle of the signals being clocked when those signals enter the receiver. The receiver is then able to the clock signal to latch valid data.

During paced transfers, the clock signal (i.e., REQ or ACK) transitions at the negotiated transfer period. Data is qualified by the clock signal and the phase of the P1 signal.

Receiver skew compensation is not defined in this manual.

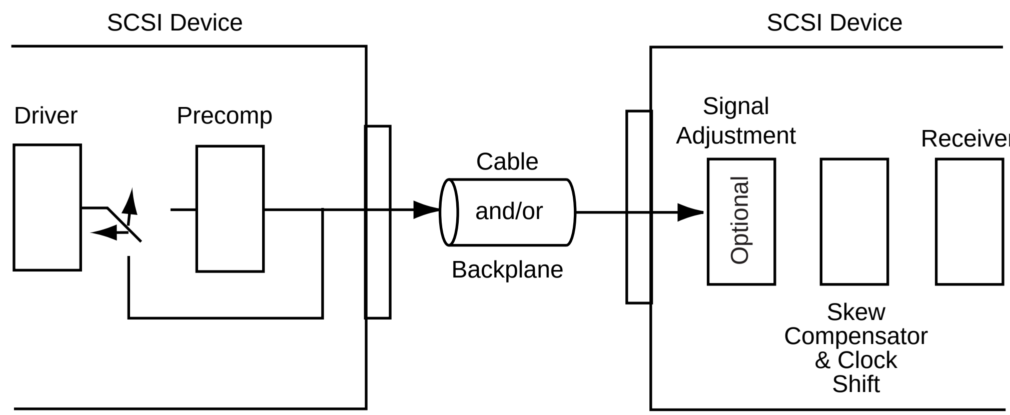


Figure 9. Example of a SCSI bus with paced transfers

2.9 Data transfer modes

There are three types of transfer modes:

- Asynchronous
- Synchronous
- Paced

This section provides a brief description of each of these types of transfer modes.

2.9.1 Asynchronous transfers

SCSI device ports default to 8-bit asynchronous transfers.

8-bit asynchronous transfers are used for all COMMAND, STATUS, and MESSAGE phases.

ST DATA phases may use 8-bit or 16-bit asynchronous transfers. Asynchronous transfers are not permitted in DT DATA phases.

2.9.2 Synchronous transfers

ST DATA phases shall use synchronous transfers when a synchronous transfer agreement is in effect. ST DATA phases may use 8-bit or 16-bit synchronous transfers.

DT DATA phases shall use synchronous transfers when a synchronous transfer agreement is in effect. DT DATA phases shall only use wide transfers.

2.9.3 Paced transfers

Paced transfers shall only be used in DT DATA phases when a paced transfer agreement is in effect. DT phases shall only use wide transfers.

2.10 ST DATA phase parallel transfers

The format of data transmitted during ST DATA phases consists of data and protection. Parity generation and checking give some error detection protection in the ST phase data.

2.11 DT DATA phase parallel transfers

During DT DATA phases, communicating SCSI devices format information according to one of two protocol options:

- Data group transfers. Data groups encapsulate all data and associated error protection.
- Information unit transfers. Information units encapsulate all nexus, task management, task attribute, command, data, and error protection. Usually these are called “SPI information units.”

Sections 2.11.1 and 2.11.2 contain a brief description of how packetized information transfer and CRC protection fit into the SCSI I/O system operation.

2.11.1 Data group transfers

Data group transfers are permitted when a synchronous transfer agreement is in effect. Data group transfers are not permitted when an asynchronous transfer agreement or a paced transfer agreement is in effect

When using data group transfers, each DT DATA IN phase and DT DATA OUT phase contains one or more data groups. A data group consists of a non-zero length data field containing an even number of bytes, followed by a pad field (when pad bytes are needed), and then followed by a pCRC field. The number of bytes transferred within a data group shall always be a multiple of four.

If the number of bytes in the data field is not a multiple of four, the transmitting SCSI device shall place two pad bytes into the pad field. If the number of bytes in the data field is a multiple of four, the transmitting SCSI device shall omit the pad field. Regardless of the number of bytes in the data field, the pCRC field shall be the last four bytes of the data group.

The value of the pad bytes within the pad field is vendor specific.

During DT DATA IN phase, if the number of bytes in a data field is not a multiple of two bytes, then after sending the pad and pCRC fields, the target shall change to MESSAGE IN phase and send an IGNORE WIDE RESIDUE message (see Section 4.3.4) with the NUMBER OF BYTES TO IGNORE FIELD set to 01h.

During DT DATA OUT phase, if a SCSI target port requests a pCRC field prior to the last data field of a task, the initiator shall transmit an even number of bytes in that data field.

The pCRC shall be used to protect all data group transfers. The SCSI device transmitting data sends the necessary pad field(s) and a pCRC field at a point determined by the target.

2.11.2 Information unit transfers

Information unit transfers are permitted when a synchronous transfer agreement is in effect. Information unit transfers are mandatory when a paced transfer agreement is in effect. Information unit transfers are not permitted when an asynchronous transfer agreement is in effect.

During information unit transfers, each DT DATA IN phase and DT DATA OUT phases contains one or more SPI information units. The number of bytes transferred within a SPI information unit shall always be a multiple of four.

If the number of bytes in the SPI information unit is not a multiple of four, the transmitting SCSI device shall transmit one, two, or three pad bytes as is necessary to make the transfer a multiple of four bytes before transmitting an iuCRC. If the number of bytes in the SPI information unit is a multiple of four, the transmitting SCSI device shall not transmit any pad bytes. Regardless of the number of bytes in the SPI information unit, the last four bytes of the SPI information unit shall be an iuCRC.

The value of the pad bytes is vendor-specific.

The iuCRC shall be used to protect all SPI information units. The SCSI device that originates the SPI information unit sends the necessary pad bytes and iuCRC fields.

An iuCRC interval may also be specified. The iuCRC interval specifies the number of bytes transferred before pad bytes (if any) and the iuCRC is transferred within SPI data information units and SPI data stream information units. A SPI data information unit or a SPI data stream information unit may contain zero or more iuCRC intervals depending on the values specified in the SPI L-Q information unit. At a minimum there shall be at least one iuCRC at the end of each SPI data information unit and SPI data stream information unit regardless of the size of the iuCRC interval. If specified, an iuCRC interval shall begin on the first transfer of each data information unit or data stream information unit.

The iuCRC interval is required to be a multiple of two, however, if it is not a multiple of four, then two pad bytes shall be transmitted before the iuCRC is transmitted.

SPI data stream information units may be used to transfer data to or from a SCSI device. Support of data streaming during DT DATA OUT phases, called write streaming, is mandatory. Support of data streaming during DT DATA IN phases, called read streaming, is optional. The use of read streaming is part of the negotiated transfer agreement between two SCSI devices (i.e., the RD_STRM bit set to one). A SCSI target port is not required to use read streaming even if streaming support is enabled.

A SCSI target port, while streaming data, may give an indication that the stream of SPI data stream information nits are about to end while still sending the current SPI data stream information unit. This early warning is called flow control. Support of flow control during DT DATA OUT phases, called write flow control, is optional. Support of flow control during DT DATA IN phases, called read flow control, is mandatory if read streaming is enabled. The use of write flow control is part of the negotiated transfer agreement between two SCSI devices (i.e., the WR_FLOW bit set to one).

2.12 Negotiation

PARALLEL PROTOCOL REQUEST (PPR) (see Section 4.3.12), SYNCHRONOUS DATA TRANSFER REQUEST (SDTR) (see Section 4.3.16), and WIDE DATA TRANSFER REQUEST (WDTR) (see Section 4.3.18) messages are used to alter the transfer agreement between two ports. The transfer agreement defines the protocol used during data phases (e.g., transfer period, REQ/ACK offset, transfer width) and agreement on features not affecting data phases (e.g., QAS). All other information transfer phases (i.e., COMMAND, MESSAGE, and STATUS) use eight-bit asynchronous data transfers.

PPR, SDTR, and WDTR messages are called negotiation messages. When a SCSI initiator port sends one of them, the message names are PPR OUT, SDTR OUT, and WDTR OUT. When a SCSI target port sends one of them, the message names are PPR IN, SDTR IN, and WDTR IN. A negotiation sequence consists of at least one matching set of negotiation messages (e.g., PPR OUT and PPR IN).

A transfer agreement is maintained by each port for each other port on the SCSI bus. Each port may be used as either a SCSI target port or a SCSI initiator port. The same transfer agreement applies whether the port is being used as a SCSI target port or as a SCSI initiator port.

2.12.1 Negotiation algorithm

A SCSI initiator port and SCSI target port exchange negotiation messages to perform negotiation. The originating port is the one that sends the first negotiation message and the responding port is the one that replies. Ports shall not set message fields to values they do not support. The originating port should set the fields in the originating negotiation message to the maximum values (e.g., fastest transfer period, largest REQ/ACK offset) it supports. If the responding port is able to support the requested values, it shall return the same values in the responding negotiation message. If the responding port requires different values (i.e., a subset of the originating port's request), it shall return those values in the responding negotiation message (e.g., if the originating port asks for a REQ/ACK offset of 32 and the responding port only supports a REQ/ACK offset of 16, then the responding port replies with an offset of 16).

If the responding negotiation message contains values the originating port does not support, the originating port shall respond with a MESSAGE REJECT message.

2.12.2 When to negotiate

Each port shall maintain a negotiation required flag for each other port. A port shall set its negotiation required flags to true for all other ports after a reset event. A port shall set its negotiation required flag to true for a given port after an error occurs while transmitting a responding negotiation message to that port.

A SCSI initiator port shall set its negotiation required flag to true for a SCSI target port after an unexpected COMMAND phase occurs when selecting without using attention condition (i.e., when selecting a SCSI target port with information units enabled).

A logical unit reset has no effect on negotiation required flags or on transfer agreements.

After a reset event a port shall set its transfer agreements for all other ports to the default transfer agreement (see Table 13).

A SCSI initiator port shall originate negotiation before sending a command to a SCSI target port whenever its negotiation required flag is true for that SCSI target port. A SCSI target port shall originate negotiation before accepting a command from a SCSI initiator port whenever its negotiation required flag is true for that SCSI initiator port. After successful negotiation or reaching the default transfer agreement, the negotiation required flag shall be set to false.

A port may originate negotiation even if its negotiation required flag is false (e.g., to change the settings, as part of integrity checking procedures, or, for a SCSI initiator port, after a SCSI target port has originated negotiation). Negotiation should not be originated after every selection and reselection as this may impact performance.

Note. SCSI target ports may have had their support for originating negotiation after power on disabled to support illegal SCSI initiator device software. If a SCSI initiator port sends a command to a SCSI target device that has been powered on (e.g., after a hot plug) that results in a unit attention condition, the SCSI initiator port determines that negotiation is required and originates negotiation before the next command. However, if the command is INQUIRY, REPORT LUNS, or REQUEST SENSE, a unit attention condition is not created. An invalid data phase may occur if the SCSI target port does not originate negotiation. If the SCSI initiator port always originates negotiation before sending those commands, the data phase runs correctly. When information units are disabled, a SCSI initiator port may originate negotiation with its currently negotiated settings before each INQUIRY, REPORT LUNS, or REQUEST SENSE command to avoid this problem. When information units are enabled, the selection without attention results in an unexpected COMMAND phase that notifies the SCSI initiator port that negotiation before each INQUIRY, REPORT LUNS, or REQUEST SENSE command is not needed.

2.12.3 Negotiable fields

Table 11 lists the fields that may be negotiated and the effects of successful negotiation on those fields by each of the different negotiation messages. Ports shall implement a given message if they implement fields that are negotiable with that message.

Table 11: Negotiable fields and effects of successful negotiation

Field name	Negotiation message pair		
	PPR	WDTR	SDTR
Transfer Period Factor	Negotiated (valid values: 08h-FFh)	No requirement	Negotiated (valid values: 0Ah-FFh)
REQ/ACK Offset	Negotiated	Sets to 00h	Negotiated
Transfer Width Exponent	Negotiated (valid values: 00h-01h)	Negotiated (valid values: 00h-01h)	Unchanged

Table 11: Negotiable fields and effects of successful negotiation

Field name		Negotiation message pair PPR	WDTR	SDTR
Protocol options	PCOMP_EN	Negotiated	Sets to 00h	Sets to 00h
	RTI	Negotiated	Sets to 00h	Sets to 00h
	RD_STRM	Negotiated	Sets to 00h	Sets to 00h
	WR_FLOW	Negotiated	Sets to 00h	Sets to 00h
	HOLD_MCS	Negotiated	Sets to 00h	Sets to 00h
	QAS_REQ	Negotiated	Sets to 00h	Sets to 00h
	DT_REQ	Negotiated	Sets to 00h	Sets to 00h
	IU_REQ	Negotiated	Sets to 00h	Sets to 00h

When negotiating, the responding port shall respond with values that are a subset of the values in the originating message as indicated in Table 12 (e.g., if the originating message requests a REQ/ACK offset of 10h, the responding message has a REQ/ACK offset field set to 10h or lower).

Table 12: Responding message requirements

Field Name		Message	Response shall be numerically
Transfer Period Factor		PPR, SDTR	Greater than or equal
REQ/ACK Offset		PPR, SDTR	Less than or equal
Transfer Width Exponent		PPR, WDTR	00h or 01h
Protocol Options	PCOMP_EN	PPR	Less than or equal
	RTI	PPR	Less than or equal
	RD_STRM	PPR	Less than or equal
	WR_FLOW	PPR	Less than or equal
	HOLD_MCS	PPR	Less than or equal
	QAS_REQ	PPR	Less than or equal
	DT_REQ	PPR	Less than or equal
	IU_REQ	PPR	Less than or equal

2.12.4 Transfer agreements

The transfer agreements that are in effect for various combinations of field values are described in Table 13.

Table 13: Transfer agreements

Transfer agreement	REQ/ACK offset	Transfer period factor	Transfer width exponent	DT_REQ	IU_REQ	QAS_REQ	All other protocol options
Default	00h	Any	00h	0	0	0	0
Asynchronous	00h	Any	Any	0	0	Any	0
Synchronous	GE 01h	GE 09h	Any	Any	Any	Any	Any
ST synchronous	GE 01h	GE 0Ah	Any	0	0	Any	Any
DT synchronous	GE 01h	GE 09h	01h	1	Any	Any	Any
Paced	GE 01h	08h	01h	1	1	Any	Any
Wide	Any	Any	01h	Any	Any	Any	Any
Narrow	Any	Any	00h	Any	Any	Any	Any
Data group	GE 01h	Any	01h	1	0	Any	Any
Information unit	GE 01h	Any	01h	1	1	Any	Any
ST data	GE 01h	GE 0Ah	Any	0	Any	Any	Any
DT data	GE 01h	Any	01h	1	Any	Any	Any

2.12.5 Transfer period factor

The Transfer Period Factor field selects the transfer period and determines which transfer rate's timing values in Tables 5, 6, 7, and 8 shall be honored, provided that REQ/ACK OFFSET is greater than 00h. The field values are defined in Table 14.

Table 14: Transfer period factor

Value	Description	Message	Transfer rate
00h - 06h	Reserved	N/A	N/A
07h	Transfer period equals 3.125 ns	PPR	Fast-320
08h	Transfer period equals 6.25 ns	PPR	Fast-160
09h	Transfer period equals 12.5 ns	PPR	Fast-80
0Ah	Transfer period equals 25 ns	PPR, SDTR	Fast-40
0Bh	Transfer period equals 30.3 ns	PPR, SDTR	Fast-40
0Ch	Transfer period equals 50 ns	PPR, SDTR	Fast-20
0Dh - 18h	Transfer period equals the TRANSFER PERIOD FACTOR x 4	PPR, SDTR	Fast-20
19h - 31h	Transfer period equals the TRANSFER PERIOD FACTOR x 4	PPR, SDTR	Fast-10
32h - FFh	Transfer period equals the TRANSFER PERIOD FACTOR x 4	PPR, SDTR	Fast-5

Table 15 shows which transfer period factors may be used with different types of transfer agreements, provided REQ/ACK OFFSET is greater than 00h.

Table 15: Transfer period factor relationships

Value	Transfer agreement					
	Synchronous	Paced	Data group	Information unit	ST data	DT data
00h - 06h	Reserved					
07h - 08h	NS	M	NS	M	NS	M
09h	M	NS	O	O	NS	M
0Ah	M	NS	O	O	O	O
0Bh	M	NS	O	O	O	O
0Ch	M	NS	O	O	O	O
0Dh - 18h	M	NS	O	O	O	O
19h - 31h	M	NS	O	O	O	O
32h - FFh	M	NS	O	O	O	O

Table abbreviations are defined as follows:

M = Mandatory: Support for the indicated transfer agreement shall be implemented if the indicated transfer period factor is implemented.

O = Optional: Support for the indicated transfer agreement may be implemented if the indicated transfer period factor is implemented.

NS = Not Supported: The indicated transfer agreement shall not be allowed if the indicated transfer factor is selected.

Table 20 defines valid combinations of TRANSFER PERIOD FACTOR and other fields.

2.12.6 REQ/ACK offset

The REQ/ACK OFFSET field determines the maximum number of REQs allowed to be outstanding before a corresponding ACK is received at the SCSI target port during synchronous or paced transfers. For ST synchronous transfers the REQ/ACK offset is the number of REQ assertions that may be sent by the SCSI target port in advance of the number of ACK assertions received from the SCSI initiator port.

For DT synchronous transfers the REQ/ACK offset is the number of REQ transitions that may be sent by the SCSI target port in advance of the number of ACK transitions received from the SCSI initiator port. For paced transfers in DT DATA IN phase the REQ/ACK offset is the number of data valid state REQ assertions that may be sent by the SCSI target port in advance of ACK assertions received from the SCSI initiator port.

For paced transfers in DT DATA OUT phase the REQ/ACK offset is the number of REQ assertions that may be sent by the SCSI target port in advance of the number of data valid state ACK assertions received from the SCSI initiator port.

See Section 2.9 for an explanation of the differences between ST and DT data transfers.

The REQ/ACK OFFSET value is chosen to prevent overflow conditions in the port's receive buffer and offset counter. The REQ/ACK OFFSET values and which timing values shall be selected are defined in Table 7.

Table 16. REQ/ACK Offset

Value	Description	Timing values
00h	Specifies asynchronous transfer agreement. [a]	Asynchronous. (see Table 5)
01h - FEh	Synchronous or paced transfers with specified offset.	Determined by transfer period factor (see Table 14).
FFh	Synchronous or paced transfers with unlimited offset.	Determined by transfer period factor (see Table 14).
[a] **Transfer period factor and protocol options other than QAS_REQ shall be ignored.**		

Table 20 defines valid combinations of REQ/ACK OFFSET and other fields.

2.12.7 Transfer width exponent

The TRANSFER WIDTH EXPONENT field defines the transfer width to be used during DATA IN and DATA OUT phases. The values are defined in Table 17.

If any of the protocol options bits other than QAS_REQ are set to one, then only wide transfer agreements are valid. If all the protocol options bits other than QAS_REQ are set to zero, wide transfer agreements and narrow transfer agreements are valid.

Table 17: Transfer width exponent

Value	Description
00h	Specifies 8-bit data bus (i.e., narrow transfer agreement)
01h	Specifies 16-bit data bus (i.e., wide transfer agreement)
02h	Obsolete
03h - FFh	Reserved

Table 20 defines valid combinations of TRANSFER WIDTH EXPONENT and other fields.

2.12.8 Protocol options

The protocol options fields affect the protocol used between the ports. The SCSI target port uses the protocol options bits to indicate to the SCSI initiator port if it agrees to enable the requested protocol options. Except for the PCOMP_EN bit, the SCSI target port shall not enable any protocol options that were not enabled in the negotiation message received from the SCSI initiator port.

Table 18 lists the protocol options bits.

Table 18: Protocol options bits

Name	Description
PCOMP_EN	Precompensation enable.
RTI	Retain training information.
RD_STRM	Read streaming and read flow control enable.
WR_FLOW	Write flow control enable.
HOLD_MCS	Hold margin control settings.
QAS_REQ	QAS enable request.
DT_REQ	DT clocking enable request.
IU_REQ	Information units enable request.

2.12.8.1 IU_REQ

The SCSI initiator port shall set IU_REQ to one in the PPR OUT message to request that information unit transfers be enabled. In response, the SCSI target port shall set its IU_REQ to one if it agrees to use information unit transfers or zero if it does not.

The SCSI initiator port shall set IU_REQ to zero in the PPR OUT message to request that information unit transfers be disabled. In response, the SCSI target port shall set IU_REQ to zero in the PPR IN message.

If IU_REQ is one, an information unit transfer agreement is in effect. If IU_REQ is zero, an asynchronous, ST synchronous, or data group transfer agreement is in effect.

Table 20 defines valid combinations of IU_REQ and other fields.

Each SCSI target port shall maintain a bus free required flag. Each time a negotiation is successful that results in the IU_REQ bit being changed from the previous agreement (i.e., zero to one or one to zero) the SCSI target port shall set its bus free required flag to true. Any intermediate changes (e.g., from multiple successful PPR negotiations) shall be treated as changing IU_REQ even if the final value equals the initial value.

At the conclusion of the message phases, if the bus free required flag is set to true, the target port shall:

1) abort all tasks for the SCSI initiator port;

2) set the bus free required flag to false; and

3) go to a BUS FREE phase.

At the conclusion of the message phases, if the IU_REQ bit was changed as the result of a negotiation, the SCSI initiator device shall abort all tasks for the logical unit.

If the IU_REQ bit was set to one during a previous PPR negotiation and not changed by a subsequent PPR negotiation sequence, the target port shall not request that the task manager abort any tasks for that SCSI initiator port and shall go to the BUS FREE phase after responding with a PPR IN message.

Table 19 describes the bus phases resulting from IU_REQ changes.

Table 19: Bus phases resulting from IU_REQ changes

Initial IU_REQ	Modified IU_REQ value	Causes	BUS phase following MESSAGE phases
0	**0**	a. PPR negotiation keeping IU_REQ set to zero; b. WDTR negotiation; or c. SDTR negotiation	COMMAND, DATA, STATUS, or BUS FREE phase
0	**1**	a. PPR negotiation setting IU_REQ to one	BUS FREE phase
1	**0**	a. PPR negotiation setting IU_REQ to zero; b. WDTR negotiation; or c. SDTR negotiation	BUS FREE phase
1	**1**	a. PPR negotiation keeping IU_REQ set to one	BUS FREE phase

2.12.8.2 DT_REQ

The SCSI initiator port shall set DT_REQ to one to request that DT DATA phases be enabled. In response, the SCSI target port shall set DT_REQ to one if it agrees to use DT DATA phases or zero if it does not.

The SCSI initiator port shall set DT_REQ to zero to request that information unit transfers be disabled. In response, the SCSI target port shall set DT_REQ to zero in the PPR IN message.

If DT_REQ is one, a DT data transfer agreement is in effect. If DT_REQ is zero, an asynchronous or ST data transfer agreement is in effect.

Table 20 defines valid combinations of DT_REQ and other fields.

2.12.8.3 QAS_REQ

The SCSI initiator port shall set QAS_REQ to one to request that QAS be enabled. In response, the SCSI target port shall set QAS_REQ to one if it supports QAS or zero if it does not.

The SCSI initiator port shall set QAS_REQ to zero to request that QAS be disabled. In response, the SCSI target port shall set QAS_REQ to zero in the PPR IN message.

Table 12 defines valid combinations of QAS_REQ and other fields.

When an initiator port and a target port have negotiated with each other to enable QAS, either of the two ports may participate in QAS arbitrations when attempting to connect to the other port. When an initiator port and target port have negotiated with each other to disable QAS, neither port shall participate in QAS arbitrations when attempting to connect to the other port.

When QAS and information unit transfers are both enabled for a connected SCSI target port, that SCSI target port may issue a QAS REQUEST message to release the bus after a DT DATA phase. When QAS is enabled for and information unit transfers are disabled for a connected SCSI target port, that SCSI target port shall not issue QAS REQUEST messages.

2.12.8.4 HOLD_MCS

The SCSI initiator port shall set HOLD_MCS to one to indicate that the SCSI target port should hold any margin control settings set with the margin control subpage of the port control mode page. In response, the SCSI target port shall set HOLD_MCS to one if it is capable of retaining the settings and zero if it is not.

The SCSI initiator port shall set HOLD_MCS to zero to indicate that the SCSI target port shall reset to their default values any margin control settings set with the margin control subpage of the port control mode page. In response, the SCSI target port shall set HOLD_MCS to zero.

Table 20 defines valid combinations of HOLD_MCS and other fields.

2.12.8.5 WR_FLOW

The SCSI initiator port shall set WR_FLOW to one to indicate that the SCSI target port should enable write flow control during write streaming. In response, the SCSI target port shall set WR_FLOW to one if it is capable of write flow control and zero if it is not.

The SCSI initiator port shall set WR_FLOW to zero to indicate that the SCSI target port shall disable write flow control during write streaming. In response, the SCSI target port shall set WR_FLOW to zero. Write streaming and write flow control only occurs during information unit transfers.

Table 20 defines valid combinations of WR_FLOW and other fields.

2.12.8.6 RD_STRM

The SCSI initiator port shall set RD_STRM to one to indicate that the SCSI target port should enable read streaming and read flow control. In response, the SCSI target port shall set RD_STRM to one if it is capable of read streaming and read flow control and zero if it is not.

The SCSI initiator port shall set RD_STRM to zero to indicate that the SCSI target port shall disable read streaming and read flow control. In response, the SCSI target port shall set RD_STRM to zero. Read streaming and read flow control only occur during information unit transfers.

Table 20 defines valid combinations of RD_STRM and other fields.

2.12.8.7 RTI (Retain Training Information)

The SCSI initiator port shall set RTI to one to indicate it is capable of saving paced data transfer training information and to indicate that the SCSI target port does not need to retrain on each connection. In response, the SCSI target port shall set RTI to one if it is capable of saving paced data transfer training information and zero if it is not.

The SCSI initiator port shall set RTI to zero to indicate it is not capable of saving paced data transfer training information and to indicate the SCSI target port shall retrain on each connection. In response, the SCSI target port shall set RTI to zero.

Table 20 defines valid combinations of RTI and other fields. For negotiated transfer periods slower than Fast-160 the RTI bit shall be set to zero.

2.12.8.8 PCOMP_EN

The SCSI initiator port shall set PCOMP_EN to one to indicate that the SCSI target port shall enable precompensation on all signals transmitted during DT DATA phases. The SCSI initiator port shall set PCOMP_EN to zero to indicate that the SCSI target port shall disable precompensation.

The SCSI target port shall set PCOMP_EN to one to indicate that the SCSI initiator port shall enable precompensation on all signals transmitted during DT DATA phases. The SCSI target port shall set PCOMP_EN to zero to indicate that the SCSI initiator port shall disable precompensation.

Table 20 defines valid combinations of PCOMP_EN and other fields. Ports that support Fast-160 shall support enabling and disabling precompensation of their drivers. For negotiated transfer periods slower than Fast-160 the PCOMP_EN bit shall be set to zero.

Note. Unlike other fields and bits in the PPR message the PCOMP_EN bit is not a negotiated value; instead, it instructs the receiving SCSI device as to whether or not precompensation is to be disabled or enabled. Because of this, precompensation may be enabled on one of the SCSI devices and disabled on the other SCSI device at the completion of a successful PPR negotiation.

2.12.9 Negotiable field combinations

Not all combinations of the negotiable fields are valid. Only the combinations defined in Table 20 shall be allowed. All other combinations of the listed fields are reserved.

Table 20: Valid negotiable field combinations

Transfer period factor	REQ/ ACK offset	Transfer width exponent	Protocol options								Description
			PCOMP_EN	RTI	RD_STRM	WR_FLOW	HOLD_MCS	QAS_REQ	DT_REQ	IU_REQ	
Ignore	00h	00h or 01h	0	0	0	0	0	0	0	0	Use ST DATA IN and ST DATA OUT phases to transfer data with asynchronous transfers.
Ignore	00h	00h or 01h	0	0	0	0	0	1	0	0	Use ST DATA IN and ST DATA OUT phases to transfer data with asynchronous transfers, and participate in QAS arbitrations.
0Ah - FFh	01h - FFh	00h or 01h	0	0	0	0	0	0	0	0	Use ST DATA IN and ST DATA OUT phases to transfer data with synchronous transfers.
09h - FFh	01h - FFh	01h	0	0	0	0	0	0	1	0	Use DT DATA IN and DT DATA OUT phases with data group transfers.
09h - FFh	01h - FFh	01h	0	0	0	0	0	1	1	0	Use DT DATA IN and DT DATA OUT phases with data group transfers, and participate in QAS arbitrations.
0Ah - FFh	01h - FFh	00h or 01h	0	0	0	0	0	1	0	0	Use ST DATA IN and ST DATA OUT phases to transfer data with synchronous transfers, and participate in QAS arbitrations
09h - FFh	01h - FFh	01h	0	0	0 or 1	0 or 1	0	0	1	1	Use DT DATA IN and DT DATA OUT phases with synchronous transfers and information unit transfers
07h	01h- FFh	01h	0	0 or 1	0 or 1	0 or 1	0 or 1	0	1	1	Use DT DATA IN and DT DATA OUT phases with paced transfers and information unit transfers
08h	01h - FFh	01h	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0	1	1	Use DT DATA IN and DT DATA OUT phases with synchronous transfers and information unit transfers.
09h - FFh	01h - FFh	01h	0	0	0 or 1	0 or 1	0	1	1	1	Use DT DATA IN and DT DATA OUT phases with paced transfers and information unit transfers, participate in QAS arbitrations, and issue QAS_REQUEST messages to initiate QAS arbitrations.
07h	01h- FFh	01h	0	0 or 1	0 or 1	0 or 1	0 or 1	1	1	1	Use DT DATA IN and DT DATA OUT phases with paced transfers and information unit transfers, participate in QAS arbitrations, and issue QAS_REQUEST messages to initiate QAS arbitrations
08h	01h - FFh	01h	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	1	1	1	Use DT DATA IN and DT DATA OUT phases with information unit transfers, participate in QAS arbitrations, and issue QAS_REQUEST messages to initiate QAS arbitrations.

2.12.10 Message restrictions

PPR may be originated by SCSI initiator ports but shall not be originated by SCSI target ports. If bus expanders are present, SCSI initiator ports should only use PPR when requesting values not attainable via WDTR and SDTR (e.g., setting any protocol option bits to one). If a SCSI target port responds to PPR only with values that are attainable via WDTR and SDTR (i.e., all protocol option bits set to zero), the SCSI initiator port should repeat negotiation with a WDTR and SDTR negotiation sequence. This ensures that bus expanders that do not support PPR are still able to handle data phases correctly.

WDTR and SDTR may be originated by either SCSI target ports or SCSI initiator ports. Since WDTR resets all the values that SDTR sets, it shall be sent first if both are needed. SCSI target ports capable of wide transfer agreements shall originate negotiation with WDTR followed with SDTR.

Note. If IU_REQ was set to one and a successful SCSI target port originated WDTR negotiation occurs, a BUS FREE phase generated because the SCSI target port detected that IU_REQ was changed is indistinguishable from a BUS FREE phase generated because the target port was detecting parity errors on the WDTR OUT. Following the WDTR negotiation with an SDTR negotiation before the BUS FREE occurs ensures that the SCSI initiator port and SCSI target port both know that IU_REQ has changed.

2.12.11 Negotiation message sequences

A SCSI initiator port originated negotiation sequence contains up to four steps:

1. SCSI initiator port’s originating message;
2. SCSI target port response;
3. SCSI initiator port response; and
4. SCSI target port second response.

A SCSI target port originated negotiation sequence contains up to four steps:

1. SCSI target port’s originating message;
2. SCSI initiator port response;
3. SCSI target port response; and
4. SCSI initiator port second response.

If the negotiation fails after a vendor-specific number of retries, the SCSI port originating the negotiation sequence may discontinue communication with the other SCSI port.

For illustrations showing the various negotiation responses, refer to the ANSI SPI-5 specification Section 4.12.7.

3.0 Logical characteristics

The operations of the SCSI bus as described in this section are supported by the drive as specified in each individual drive's Product Manual. The drive always functions as the target unless otherwise stated.

3.1 SCSI bus phases overview

The drive responds to the following phases:

BUS FREE phase
ARBITRATION phase
SELECTION phase
RESELECTION phase

COMMAND phase Data (IN and OUT) STATUS (IN only) MESSAGE (IN and OUT)	These phases are collectively termed the Information transfer phases

The COMMAND, DATA, STATUS, and MESSAGE phases are collectively called the information transfer phases.

The SCSI bus can never be in more than one phase at a time. Signals that are not mentioned in a particular context shall not be asserted.

3.1.1 BUS FREE phase

The BUS FREE phase indicates that there is no current task and that the SCSI bus is available for a physical connection or physical reconnection. SCSI devices shall detect the BUS FREE phase after the SEL and BSY signals are both false for at least one bus settle delay.

SCSI devices shall release all SCSI bus signals within one bus clear delay after BSY and SEL are continuously negated (false) for one bus settle delay. If a SCSI device requires more than one bus settle delay to detect the BUS FREE phase, it shall release all SCSI bus signals within one bus clear delay minus the excess time to detect the BUS FREE phase. The total time to clear the SCSI bus shall not exceed one bus settle delay plus one bus clear delay.

During normal operation a SCSI target port enters the BUS FREE phase when it releases the BSY signal.

3.1.1.1 Unexpected and expected bus free phases

In some cases a SCSI target port (connected to a SCSI initiator port) unexpectedly reverts to the BUS FREE phase to indicate an error condition that it has no other way to handle. This is called an unexpected disconnect.

SCSI target ports shall create a BUS FREE phases after any of the following:

a. after any bus reset event.
b. after a transceiver mode change reset event.
c. after an Abort Task management function is successfully received by a SCSI target port;
d. after an Abort Task Set management function is successfully received by a SCSI target port;
e. after a Clear Task Set management function is successfully received by a SCSI target port;
f. after a Logical Unit Reset management function is successfully received by a SCSI target port;
g. after a TARGERT RESET management function is successfully received by a SCSI target port;
h. after a Clear ACA Task management function is successfully received by a SCSI target port;
i. after a DISCONNECT message is successfully transmitted from a CSI target port (see Section 4.3.2);
j. after a TASK COMPLETE message is successfully transmitted from a SCSI target port (see Section 4.3.17);
k. after a DISCONNECT message is successfully received by a SCSI target port when information unit transfers are enabled;
l. after the release of the SEL signal after a SELECTION or RESELECTION phase timeout;
m. after a PPR (Parallel Protocol Request) negotiation in response to a selection using attention condition when information unit transfers are enabled (see Section 4.3.12); or
n. after any successful negotiation that causes information unit transfers to be enabled or disabled.

An unexpected bus free occurs when a SCSI initiator port detects a BUS FREE phase that it does not expect see 3.1.1.2.

The target uses an unexpected bus free to inform the initiator of a protocol error. The target may switch to a BUS FREE phase at any time, except during an ARBITRATION phase, independent of any attention condition.

The target shall terminate the task that was the current task before the BUS FREE phase by clearing all data and status for that task. The target may optionally prepare sense data that may be retrieved by a REQUEST SENSE command. However, an unexpected bus free shall not create an exception condition.

The initiator shall terminate the task that was the current task before the BUS FREE phase occurred and shall manage this condition as an exception condition.

3.1.1.2 Expected bus free phases

Initiators may expect a bus free to occur after one of the following:

a. after the last SPI command information unit is successfully received by a SCSI target port;
b. after a SPI data information unit is successful received by or transmitted from a SCSI target port;
c. after a SPI status information unit is successfully transmitted from a SCSI target port;
d. after a SPI L_Q information unit, if the SPI L_Q information unit Data Length field is zero; or
e. during a QAS phase.

3.1.2 Arbitration and QAS overview

Arbitration allows one SCSI device to gain control of the SCSI bus so that it can initiate or resume a task.

There are two methods that a SCSI device may use to arbitrate for the SCSI bus: normal arbitration and QAS (Quick Arbitration and Selection). Normal arbitration is mandatory and requires the detection of a BUS FREE phase on the SCSI bus before starting. QAS is optional and, when enabled, requires the detection of a QAS REQUEST message before starting.

SCSI devices with arbitration fairness enabled shall maintain a fairness register that records the SCSI IDs of devices that need a chance to arbitrate (see Section 3.4). Fairness in normal arbitration is enabled in targets by the Disconnect-Reconnect mode page (see Seagate SCSI Command Reference Manual, Part number 100293068). Fairness is always enabled when QAS is enabled.

3.1.2.1 Normal ARBITRATION phase

The procedure for a SCSI device to obtain control of the SCSI bus is as follows:

1. The SCSI device shall first wait for the BUS FREE phase to occur. The BUS FREE phase is detected whenever both the BSY and SEL signals are simultaneously and continuously false for a minimum of one bus settle delay.

Note. This bus settle delay is necessary because a transmission line phenomenon known as a wired-OR glitch may cause the BSY signal to briefly appear false, even though it is being driven true.

2. The SCSI device shall wait a minimum of one bus free delay after detection of the BUS FREE phase (i.e., after the BSY and SEL signals are both false for one bus settle delay) before driving any signal.
3. Following the bus free delay in step 2, the SCSI device may arbitrate for the SCSI bus by asserting both the BSY signal and its own SCSI ID. However the SCSI device shall not arbitrate (i.e., assert the BSY signal and its SCSI ID) during this NORMAL ARBITRATION phase if more than one bus set delay has passed since the BUS FREE phase was last observed. If arbitration fairness is enabled, the SCSI device shall not arbitrate until its fairness register is cleared (see ANSI specification SPI-5, Annex B).

Note. There is no maximum delay before asserting the BSY signal and the SCSI ID following the bus free delay in step 2 as long as the bus remains in the BUS FREE phase. However, SCSI devices that delay longer than one bus settle delay plus one bus set delay from the time when the BSY and SEL signals first become false may fail to participate in arbitration when competing with faster SCSI devices, and may not be ensured fair arbitration by the arbitration fairness algorithm.

4. After waiting at least one arbitration delay, measured from its assertion of BSY, the SCSI device shall examine the Data Bus.

   (a) If no higher priority SCSI ID bit is true on the Data Bus [DB(7) is the highest], the SCSI device has won the arbitration and the SCSI device shall assert the SEL signal.

   (b) If a higher priority SCSI ID bit is true on the Data Bus, the SCSI device has lost the arbitration and it shall release the BSY signal and the SCSI ID after the SEL signal becomes true (asserted), within one bus clear delay after the SEL signal becomes true. A SCSI device that loses arbitration may return to step (1). If the SCSI device implements a “fairness algorithm” for arbitration, see Section 3.4.

Note. Step 4 above requires any device that begins normal ARBITRATION phase to complete the normal ARBITRATION phase to the point of SEL being asserted if it begins the normal ARBITRATION phase as stated in step 3. This precludes the possibility of the bus being hung.

5. After the bus free delay in step 2, SCSI devices with arbitration fairness enabled that are not arbitrating shall wait one bus set delay and start sampling the Data Bus to determine the SCSI devices that attempted arbitration, the SCSI device that won, and the SCSI devices that lost. This sampling shall continue for an arbitration delay after the bus free delay in step 2. Each SCSI device shall update its fairness register with all lower-priority device IDs that lost arbitration.

Note. For ease of implementation, this sampling may begin when BSY is true following Bus Free and end when SEL is true.

6. The SCSI device that wins arbitration shall wait at least a bus clear delay plus a bus settle delay after asserting SEL before changing any signals.

The SCSI ID bit is a single bit on the Data Bus that corresponds to the SCSI device’s unique SCSI address. All other of the Data Bus bits shall be released by the SCSI device. During the normal ARBITRATION phase, DB(P_CRCA) and DB(P1) (if present) may be released or asserted, but shall not be actively driven false.

3.1.2.2 QAS protocol

Quick Arbitration and Selection (QAS) allows a SCSI target port with an information unit transfer agreement in effect and QAS enabled (see Section 4.3.12) that is currently connected to a SCSI initiator port that has information unit transfers enabled and QAS enabled to transfer control of the bus to another SCSI device that has information unit transfers enabled and QAS enabled without an intervening BUS FREE phase. SCSI devices that support QAS shall report that capability in the INQUIRY command.

Before a SCSI initiator may use QAS, that initiator shall negotiate, using the PPR message, the use of the QAS phase with each SCSI target port that has indicated support of QAS. Any time a SCSI initiator port's negotiation required flag is true, that SCSI initiator port shall renegotiate to enable QAS (see Section 4.3.12).

SCSI devices that support QAS shall implement the fairness algorithm (see Annex B of SPI-5) during all QAS arbitrations. SCSI devices shall negotiate the use of QAS with a particular SCSI device before using QAS to select or reselect that SCSI device. Also, targets shall have negotiated the use of QAS with a particular initiator before using QAS REQUEST message to do a physical disconnect from that initiator, and initiators shall have negotiated the use of QAS with a particular target before accepting a QAS REQUEST message from that target. If a SCSI initiator port receives a QAS REQUEST message from a SCSI target port that has not negotiated the use of QAS, then the initiator shall create an attention condition for the QAS REQUEST message, and shall report Message Reject on the following MESSAGE OUT phase.

In an environment where some SCSI devices have QAS enabled and other SCSI devices do not, it is possible for the SCSI devices that have QAS enabled to prevent SCSI devices that do not have QAS enabled from arbitrating for the bus. This occurs when SCSI devices that have QAS enabled never go to a BUS FREE phase.

A QAS initiator may interrupt a sequence of QAS cycles to force a normal arbitration with the following procedure:

1. perform a QAS arbitration;
2. on winning QAS arbitration, continue driving the initiator's ID on the Data Bus instead of asserting SEL to enter selection phase;
3. wait until the target transitions to Bus Free (this occurs after two QAS arbitration delays);
4. after detecting BSY false, release the Data Bus; and
5. after one bus settle delay from when the target drove BSY false, the bus is in BUS FREE phase. The initiator may then arbitrate using normal arbitration and perform a selection if it wins.

3.1.2.3 QAS phase overview

For targets with both information unit transfers and QAS enabled to indicate it wants to release the bus, the following procedure is used:

1. The target shall change to a MESSAGE IN phase, issue a single QAS Request (55h) message, and then wait for ACK to be true.

Note. The timing requirements are required to ensure that all the SCSI devices that have QAS enabled see the message bytes.

2. After detection of the ACK signal being false and if the SCSI initiator port did not create an attention condition, the SCSI target port shall release all SCSI signals except the BSY, MSG, C/D, I/O, and REQ signals. Then the SCSI target shall negate the MSG, C/D, and I/O signals within two system deskew delays. The SCSI target port shall wait two system deskew delays after negating the C/D, I/O, and MSG signals before releasing the REQ signal.
3. If the SCSI initiator port did not create an attention condition, the SCSI initiator port shall release all SCSI signals except ACK and ATN within two system deskew delays after detecting MSG, C/D, and I/O signals false. The ACK and ATN signals shall follow the timing specified in section 7 of the SPI-5 specification.

4. If the SCSI initiator creates an attention condition, the SCSI target port shall go to a MESSAGE OUT phase, receive all the message bytes, and cause an unexpected bus free by generating a BUS FREE phase (see Section 3.1.1.1 on page 51).
5. If the SCSI target port detects the SEL signal being true, the SCSI target port shall release the BSY, MSG, C/D, and I/O signals within one QAS release delay.
6. After waiting at least one QAS arbitration delay from negating the SCSI MSG, C/D, and I/O signals in step 2, if there are no SCSI ID bits true, the SCSI target port shall transition to the BUS FREE phase.
7. After waiting at least one QAS arbitration delay from negating the MSG, C/D, and I/O signals in step 2, if there are any SCSI ID bits true, the SCSI target port shall wait at least a second QAS arbitration delay. If the SEL signal is not true by the end of the second QAS arbitration delay, the SCSI target port shall transition to the BUS FREE phase.

Note. The release of MSG, C/D, and I/O may cause release glitches. Step 5 above ensures these glitches occur at a time when no connection is established on the bus so that they do not interfere with proper operation.

The procedure for a SCSI device with QAS enabled to obtain control of the SCSI bus via QAS is as follows:

1. The SCSI device shall first wait for MESSAGE IN phase to occur with a single QAS REQUEST message. When the SCSI device detects the ACK signal being false for the QAS REQUEST message and the attention condition is cleared, it shall begin the QAS phase.
2. The SCSI device shall wait a minimum of a two system deskew delays after detection of the MSG, C/D, and I/O signals being false before driving any signal.
3. Following the delay in step 2, the SCSI device may arbitrate for the SCSI bus by asserting its own SCSI ID within one QAS assertion delay from detection of the MSG, C/D, and I/O signals being false. If arbitration fairness is enabled, the SCSI device shall not arbitrate until its fairness register is cleared.
4. After waiting at least one QAS arbitration delay, measured from the detection of the MSG, C/D, and I/O signals being negated, the SCSI device shall examine the Data Bus.

a. If no higher priority SCSI ID bit is true on the Data Bus and the fairness algorithm allowed the SCSI device to participate, then the SCSI device has won the arbitration and it shall assert the SEL signal.

b. If a higher priority SCSI ID bit is true on the Data Bus (see Table 1 for the SCSI ID arbitration priorities) or the fairness algorithm (see Section 3.4) prevented the SCSI device from participating in QAS arbitration, then the SCSI device has lost the arbitration.

c. Any SCSI device other than the winner has lost the arbitration and shall release its SCSI ID bit after two system deskew delays and within one QAS release delay after detection of the SEL signal being asserted. A SCSI device that loses arbitration may return to step 1.

5. The SCSI device that wins arbitration shall wait at least a QAS arbitration delay after asserting the SEL signal before changing any signals.
6. After the QAS arbitration delay in step 4, SCSI devices with arbitration fairness enabled that are not arbitrating shall start sampling the Data Bus to determine the SCSI devices that are attempting arbitration, the SCSI device that won, and the SCSI devices that lost. This sampling shall continue for one bus settle delay plus two system deskew delays. The SCSI devices shall update their fairness register with all device IDs that lost arbitration.

The SCSI ID bit is a single bit on the Data Bus that corresponds to the SCSI device's unique SCSI address. All other Data Bus bits shall be released by the SCSI device. The DB(P_CRCA) and DB(P1) are not valid during the QAS phase. During the QAS phase, DB(P_CRCA), and DB(P1) may be released or asserted, but shall not be actively driven false.

3.2 SELECTION phase

The SELECTION phase allows a SCSI initiator port to select a SCSI target port for the purpose of initiating some target function (e.g., READ or WRITE command). During the SELECTION phase, the I/O signal is negated to distinguish this phase from the RESELECTION phase.

Refer to Section 3.4 for a description of the fairness algorithm which applies during SELECTION and RESELECTION phases.

3.2.1 Selection overview

The SCSI device that won a normal arbitration has both the BSY and SEL signals asserted and has delayed at least one bus clear delay plus a bus settle delay before ending the normal ARBITRATION phase.

The SCSI device that won QAS has the SEL signal asserted and has delayed at least one QAS arbitration delay before ending the QAS phase.

The SCSI device that won the arbitration identifies itself as a SCSI initiator port by not asserting the I/O signal.

3.2.1.1 Selection using attention condition

3.2.1.1.1 Starting the SELECTION phase when using attention condition

The initiator shall set the Data Bus to a value that is the OR of its SCSI ID bit, the target's SCSI ID bit, and the appropriate parity bit(s) [i.e., DB(P_CRCA) and/or DB(P1)]. The initiator shall create an attention condition (indicating that a MESSAGE OUT phase is to follow the SELECTION phase).

If the arbitration was a normal arbitration, then the initiator shall wait at least two system deskew delays and release the BSY signal. The initiator shall then wait at least one bus settle delay before attempting to detect an assertion of the BSY signal from the target.

If QAS was used for arbitration then the SCSI initiator port shall wait at least one bus settle delay before attempting to detect an assertion of the BSY signal from the SCSI target port.

The target shall detect that it is selected when the SEL signal and its SCSI ID bit are true and the BSY and I/O signals are false for at least one bus settle delay. The selected target may examine the Data Bus in order to determine the SCSI ID of the selecting initiator. The selected target shall then assert the BSY signal within one selection abort time of its most recent detection of being selected; this is required for correct operation of the selection timeout procedure.

The target shall not respond to a selection if bad parity is detected (see sections 3.9.2.1 and 3.9.3.1). Also, if more or less than two SCSI ID bits are on the Data Bus, the target shall not respond to selection.

No less than two system deskew delays after the initiator detects the BSY signal is true, it shall release the SEL signal and may change the Data Bus. The target shall wait until the SEL signal is false before asserting the REQ signal to enter an information transfer phase.

3.2.1.1.2 Information unit transfers disabled

If information unit transfer agreement is not in effect for the connecting SCSI initiator port device, the SCSI target port shall follow the phase sequences defined in Section 3.11.

3.2.1.1.3 Information unit transfers enabled

If information unit transfers are enabled (see Section 4.3.12) for the connecting initiator, the target shall follow the phase sequences defined in Section 3.5. On detecting the MESSAGE OUT phase, the initiator shall begin a PPR (Parallel Protocol Request) negotiation (see Section 4.3.12 in this manual). On completion of the PPR negotiation, the target shall proceed to a BUS FREE phase. If the first message received by the target during the MESSAGE OUT phase is not a task management message or a PPR message, the target shall change to a MESSAGE IN phase and issue a MESSAGE REJECT message followed by a WDTR message with TRANSFER WIDTH EXPONENT field set to 00h. If the target does not support the WDTR message, it shall follow the MESSAGE REJECT message with an SDTR message with the REQ/ACK Offset field set to 00h.

3.2.1.1.4 Selection using attention condition timeout procedure

Two optional selection timeout procedures are specified for clearing the SCSI bus if the initiator waits a minimum of one selection timeout delay and there has been no BSY signal response from the target:

- Optionally, the initiator shall assert the RST signal.
- Optionally, the initiator shall continue asserting the SEL signal and shall release the Data Bus, DB(P_CRCA), and/or DB(P1). If the initiator has not detected the BSY signal to be true after at least one selection abort time plus two system deskew delays, the initiator shall release the SEL signal allowing the SCSI bus to go to the BUS FREE phase. SCSI devices shall ensure that, when responding to selection, the selecting was still valid within one selection abort time of their assertion of the BSY signal. Failure to comply with this requirement may result in an improper selection (for example, two targets connected to the same initiator, wrong target connected to a SCSI initiator port or a SCSI target port connected to no initiator).

3.2.1.2 Selection without using attention condition

3.2.1.2.1 Information unit transfers disabled or enabled

The initiator shall set the Data Bus to a value that is the OR of its SCSI ID bit, the target's SCSI ID bit, and the appropriate parity bit(s) (i.e., DB(P_CRCA), and/or DB(P1)) and it shall clear the attention condition, indicating that an INFORMATION UNIT OUT phase is to follow the SELECTION phase.

If the arbitration was a normal arbitration, then the initiator shall wait at least two system deskew delays and release the BSY signal. The initiator shall then wait at least one bus settle delay before attempting to detect an assertion of the BSY signal from the target.

If QAS was used for arbitration, then the initiator shall wait at least one bus settle delay before attempting to detect an assertion of the BSY signal from the target.

The target shall detect it is selected when the SEL signal and its SCSI ID bit are true and the BSY and I/O signals are false for at least one bus settle delay. The selected target may examine the Data Bus in order to determine the SCSI ID of the selecting initiator. The selected target shall then assert the BSY signal within one selection abort time of its most recent detection of being selected; this is required for correct operation of the selection time-out procedure.

The target shall not respond to a selection if bad parity is detected (see sections 3.9.2.1 and 3.9.3.1). Also, if more or less than two SCSI ID bits are on the Data Bus, the target shall not respond to selection.

The SCSI initiator port shall wait at least two system deskew delays after detecting that the BSY signal is true. The SCSI initiator port shall then release the SEL signal and may change the Data Bus signals. The target shall wait until the SEL signal is false before asserting the REQ signal to enter an information transfer phase.

If information unit transfers are enabled (see Section 4.3.12) for the connecting initiator, the target shall follow the phase sequences defined in Section 3.11.

If information unit transfers are disabled (see Section 4.3.12) for the connecting initiator, the target shall follow the phase sequences defined in Section 3.12.

If a SCSI initiator port, when selecting without using an attention condition, detects an unexpected COMMAND phase, it shall set its transfer agreement to the default transfer agreement and set its negotiation required flag to true, create an attention condition, and on the corresponding MESSAGE OUT phase shall issue an ABORT TASK message. On the next selection of the SCSI target port that received the ABORT TASK message the SCSI initiator port shall do a selection using the attention condition and should negotiate to enable information unit transfers.

3.2.1.2.2 Selection without using attention condition time-out procedure

Two optional selection time-out procedures are specified for clearing the SCSI bus if the initiator waits a minimum of one selection time-out delay and there has been no BSY signal response from the target:

(a) Optionally, the initiator shall assert the RST signal (see Section 5.3);

(b) Optionally, the initiator shall continue asserting the SEL signal and shall release the Data Bus, DB(P_CRCA), or DB(P1). If the initiator has not detected the BSY signal to be true after at least one selection abort time plus two system deskew delays, the initiator shall release the SEL signal allowing the SCSI bus to go to the BUS FREE phase. SCSI devices shall ensure that, when responding to selection, the selection was still valid within one selection abort time of their assertion of the BSY signal. Failure to comply with this requirement may result in an improper selection (for example, two targets connected to the same initiator, wrong target connected to a SCSI initiator port, or a SCSI target port connected to no initiator).

3.3 RESELECTION phase

3.3.1 RESELECTION phase overview

The RESELECTION phase allows a SCSI target port to physically reconnect to a SCSI initiator port for the purpose of continuing some operation that was previously started by the initiator but was suspended by the target (i.e., the target physically disconnected by allowing a BUS FREE phase to occur or issued a QAS REQUEST message before the operation was complete). During the RESELECTION phase, the I/O signal is asserted to distinguish this phase from the SELECTION phase.

Refer to Section 3.4 for a description of the fairness algorithm which applies during SELECTION and RESELECTION phases.

3.3.2 Physical reconnection

The SCSI device that won a normal arbitration has both the BSY and SEL signals asserted and has delayed at least a bus clear delay plus one bus settle delay before ending the normal ARBITRATION phase.

The SCSI device that won a QAS has the SEL signal asserted and has delayed at least a QAS arbitration delay before ending the QAS phase.

The SCSI device that won the arbitration identifies itself as a SCSI target port by asserting the I/O signal.

The winning SCSI device shall also set the Data Bus to a value that is the logical OR of its SCSI ID bit and the initiator's SCSI ID bit and the appropriate parity bit(s) [i.e., DB(P_CRCA), and/or DB(P1)].

If the arbitration was a normal arbitration, then the target shall wait at least two system deskew delays and release the BSY signal. The target shall then wait at least one bus settle delay before attempting to detect an assertion of the BSY signal by the initiator.

If QAS was used for arbitration, then the target shall wait at least a bus settle delay before attempting to detect an assertion of the BSY signal from the initiator.

The initiator shall be physically reconnected when the SEL and I/O signals and its SCSI ID bit are true and the BSY signal is false for at least one bus settle delay. The physically reconnected initiator may examine the Data Bus in order to determine the SCSI ID of the physically reconnected target. The physical reconnected initiator shall then assert the BSY signal within one selection abort time of its most recent detection of being physically reconnected; this is required for correct operation of the timeout procedure.

The initiator shall not respond to a physical reconnection if bad parity is detected (see sections 3.9.2.1 and 3.9.3.1). Also, if more than or less than two SCSI ID bits are on the Data Bus, the initiator shall not respond to a physical reconnection.

After the target detects the assertion of the BSY signal, it shall also assert the BSY signal and wait at least two system deskew delays and then release the SEL signal. The target may then change the I/O signal and the Data Bus. After the physically reconnected initiator detects the SEL signal is false, it shall release the BSY signal. The target shall continue asserting the BSY signal until it relinquishes the SCSI bus.

Note. When the target is asserting the BSY signal, a transmission line phenomenon known as a wired-OR glitch may cause the BSY signal to appear false for up to a round-trip propagation delay following the release of the BSY signal by the initiator. This is the reason why the BUS FREE phase is recognized only after both the BSY and SEL signals are continuously false for a minimum of one bus settle delay. For more information on glitches, see ANSI SPI-5, T10/1525D.

3.3.3 Physical reconnection timeout procedure

Two optional physical reconnection timeout procedures are specified for clearing the SCSI bus during a RESELECTION phase if the target waits a minimum of one selection timeout delay and there has been no BSY signal response from the initiator:

- Optionally, the target shall assert the RST signal (see sections 5.3 and 5.4).
- Optionally, the target shall continue asserting the SEL and I/O signals and shall release all Data Bus, DB(P_CRCA), and/or DB(P1) signals. If the target has not detected the BSY signal to be true after at least a selection abort time plus two system deskew delays, the target shall release the SEL and I/O signals allowing the SCSI bus to go to the BUS FREE phase. SCSI devices shall ensure that the physical reconnection was still valid within one selection abort time of their assertion of the BSY signal. Failure to comply with this requirement may result in an improper physical reconnection (two initiators connected to the same target or the wrong initiator connected to a SCSI target port).

3.4 SCSI bus fairness

Implementation of the SCSI bus fairness is optional, however, if implemented, the SCSI bus fairness protocol shall conform to ANSI specification SPI-5, Annex B.

A SCSI device determines “fairness” by monitoring prior arbitration attempts by other SCSI devices. It shall postpone arbitration for itself until all lower priority SCSI devices that previously lost arbitration either win a subsequent arbitration or discontinue their arbitration attempts (as in the case where the initiator aborted an outstanding command thus removing the need to re-arbitrate).

When a SCSI device does not need to arbitrate for the SCSI bus, it shall monitor the arbitration attempts of the other SCSI devices and update a fairness register with the SCSI IDs of any lower priority SCSI devices that lost arbitration.

When a requirement for arbitration arises, the SCSI device shall first check to see if its fairness register is clear (see Section 3.1.2.3). If it is clear, then no lower priority SCSI devices had attempted and lost the previous arbitration and therefore, this SCSI device may now participate in arbitration. If the fairness register is not clear, the SCSI device shall postpone arbitration until all lower priority SCSI IDs have been cleared from the fairness register. Lower SCSI IDs are cleared as those lower level SCSI devices win arbitration. SCSI IDs shall also be cleared if a SCSI device discontinues arbitration (e.g., as a result of an ABORT TASK message, ABORT TASK SET message, CLEAR TASK SET message, or logical unit reset).

The fairness register may be refreshed, updated or cleared. The fairness register is refreshed by copying the SCSI IDs of any lower priority SCSI devices that lost arbitration into the fairness register. A refresh of the fairness register completely replaces the previous contents of the fairness register. The fairness register is updated by removing the SCSI IDs of any lower priority devices that win arbitration or discontinue arbitration. The fairness register is cleared by setting all of its bits to zero. SCSI IDs may only be added to the fairness register by a refresh but may be subtracted by a refresh, update, or clear.

Since the fairness register is only refreshed when the SCSI device is not arbitrating for itself, the fairness register is effectively frozen by the SCSI device prior to a requirement for its own arbitration arising. Other lower priority SCSI devices that were not latched shall not be added to the fairness register until this SCSI device has successfully arbitrated.

See ANSI specification SPI-5, Annex B, for details and timing for the SCSI bus fairness algorithm.

3.5 Information transfer phases

The COMMAND, DATA, STATUS, and MESSAGE phases are grouped together as information transfer phases because they are all used to transfer data or control information via the data bus. The actual contents of the information is beyond the scope of this section.

The C/D, I/O, and MSG signals are used to distinguish between the different information transfer phases (see Table 21). The target drives these three signals and therefore controls all changes from one phase to another. The initiator requests a MESSAGE OUT phase by creating an attention condition, The target causes the BUS FREE phase by releasing MSG, C/D, I/O, and BSY signals.

Table 21: Information transfer phases

<table>
<tr><th colspan="3">Signal</th><th rowspan="2">Phase</th><th rowspan="2">Direction of transfer</th><th rowspan="2" colspan="2">Comment</th></tr>
<tr><th>C/D</th><th>MSG</th><th>I/O</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>ST DATA OUT</td><td>Initiator to target</td><td rowspan="2">ST DATA Phase</td><td rowspan="4">DATA phase</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>ST DATA IN</td><td>Initiator from target</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>DT DATA OUT</td><td>Initiator to target</td><td rowspan="2">DT DATA phase</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>DT DATA IN</td><td>Initiator from target</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>COMMAND</td><td>Initiator to target</td><td></td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>STATUS</td><td>Initiator from target</td><td></td><td></td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>MESSAGE OUT</td><td>Initiator to target</td><td rowspan="2">MESSAGE Phase</td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>MESSAGE IN</td><td>Initiator from target</td><td></td></tr>
</table>

Key: 0 = False, 1 = True

The information transfer phases use one or more REQ/ACK handshakes to control the information transfer. Each REQ/ACK handshake allows the transfer of 8- or 16-bits of information depending on the negotiated transfer width (see Section 4.3.18). During the information transfer phases the BSY signal shall remain true and the SEL signal shall remain false. Additionally, during the information transfer phases, the target shall continuously envelope the REQ/ACK handshakes with the C/D, I/O, and MSG signals in such a manner that these control signals are valid for one bus settle delay before the assertion of the REQ signal of the first handshake and remain valid until after the negation of the ACK signal at the end of the handshake of the last transfer of the phase.

The SCSI target port shall not transition into an information transfer phase unless the REQ/ACK signals are negated. The target shall not transition from an information transfer phase into another information transfer phase unless the REQ and ACK signals are negated.

Note. After the negation of the ACK signal of the last transfer of the phase, the target may prepare for a new phase by asserting or negating the C/D, I/O, and MSG signals. These signals may be changed together or individually. They may be changed in any order and may be changed more than once. It is desirable that each line change only once. A new phase does not begin until the REQ signal is asserted for the first byte of the new phase.

Note. A phase is defined as ending when the C/D, I/O, or MSG signals change after the negation of the ACK signal. The time between the end of a phase and the assertion of the REQ signal beginning a new phase is undefined.

There are three methods of transferring data using information transfers:

- Asynchronous transfers
- Synchronous transfers
- Paced transfers

Synchronous transfers shall only be used for negotiated transfer rates less than or equal to Fast-80.

Paced transfers shall only be used for a negotiated transfer rate of Fast-160.

3.5.1 Asynchronous transfer

The target shall control the direction of information transfer by means of the I/O signal. When the I/O signal is true, information shall be transferred from the target to the initiator. When the I/O signal is false, information shall be transferred from the initiator to the target.

If the I/O signal is true (i.e., transfer to the SCSI initiator port), the target shall first drive the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals to their desired values, delay at least one system deskew delay plus one cable skew, then assert the REQ signal. The DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals shall remain valid until the ACK signal is true at the target. The initiator shall read the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals after the REQ signal is true, then indicate its acceptance of the data by asserting the ACK signal. When the ACK signal becomes true at the target, the target may change or release the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals and shall negate the REQ signal. After the REQ signal is false, the initiator shall then negate the ACK signal. After the ACK signal is false, the target may continue the transfer by driving the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals and asserting the REQ signal as described above.

If the I/O signal is false (i.e., transfer to the SCSI target port), the target shall request information by asserting the REQ signal. The SCSI initiator port shall drive the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals to their values, delay at least one system deskew delay plus one cable skew, and assert the ACK signal. The initiator shall continue to drive the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals until the REQ signal is false. When the ACK signal becomes true at the target, the target shall read the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals then negate the REQ signal. When the REQ signal becomes false at the initiator, the initiator may change or release the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals and shall negate the ACK signal. After the ACK signal is false, the target may continue the transfer by asserting the REQ signal as described above.

3.5.2 Synchronous transfer

Synchronous transfer is optional and is only used in DATA phases. It shall be used in a DATA phase if a synchronous transfer agreement has been established (see section 4.3.16 or 4.3.12). The transfer agreement specifies the REQ/ACK offset and the transfer period.

When synchronous data transfers are being used, data may be transferred using ST data transfers or, optionally, DT data transfers. DT data transfers shall only be used on 16-bit-wide buses that transmit and receive data using LVD transceivers.

Implementors shall not use this section for timing requirements. For timing requirements, see Section 2.5.

3.5.2.1 ST synchronous data transfer

When a ST data transfer agreement has been established, the SCSI target port shall only use the ST DATA IN phase and ST DATA OUT phase for data transfers.

The REQ/ACK offset specifies the maximum number of REQ assertions that shall be sent by the target in advance of the number of ACK assertions received from the initiator, establishing a pacing mechanism. If the number of REQ assertions exceeds the number of ACK assertions by the REQ/ACK offset, the target shall not assert the REQ signal until after the next ACK assertion is received. For successful completion of the ST DATA phase, the number of ACK and REQ assertions shall be equal.

For the timing requirements of the negotiated transfer period see Section 3.5.2.

If the I/O signal is true (i.e., transfer to the SCSI initiator port), the target shall first drive the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals to their values, wait at least one transmit setup time, then assert the REQ signal. The DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals shall be held valid for a minimum of a transmit hold time after the assertion of the REQ signal. The target shall assert the REQ signal for a minimum of one transmit assertion period. The target may then negate the REQ signal and change or release the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals. The initiator shall read the value on the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals within one receive hold time of the transition of the REQ signal to true. The SCSI initiator port shall then respond with an ACK assertion.

If the I/O signal is false (i.e., transfer to the SCSI target port), the SCSI initiator port, after detecting a REQ assertion, shall first drive the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals to their values, delay at least one transmit setup time, then assert the ACK signal. The initiator shall hold the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals valid for at least one transmit hold time after the assertion of the ACK signal. The initiator shall assert the ACK signal for a minimum of one transmit assertion period. The SCSI initiator port may then negate the ACK signal and may change or release the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals. The target shall read the value of the DB(7-0,P_CRCA) or DB(15-0,P_CRCA,P1) signals within one receive hold time of the transition of the ACK signal to true.

3.5.2.2 DT synchronous transfer

When a DT data transfer agreement has been established the target shall only use the DT DATA IN phase and DT DATA OUT phase for data transfers.

During DT data transfers, data shall be clocked on both the assertion and negation of the REQ and ACK signal lines. References to REQ/ACK transitions in this section refer to either an assertion or a negation of the REQ or ACK signal.

The REQ/ACK offset specifies the maximum number of REQ transitions that shall be sent by the target in advance of the number of ACK transitions received from the initiator, establishing a pacing mechanism. If the number of REQ transitions exceeds the number of ACK transitions by the REQ/ACK offset, the target shall not transition the REQ signal until after the next ACK transition is received. For successful completion of the DT DATA phase the number of ACK and REQ transitions shall be equal and both REQ and ACK shall be negated.

For the timing requirements of the negotiated transfer period see Section 3.5.2.

3.5.2.2.1 Information unit transfer

When information unit transfer agreement has been established (see Section 4.3.12):

a. information units shall be transferred on the DT DATA OUT phase and the DT DATA IN phase, and
b. the information units' embedded iuCRC shall be used to detect information unit data errors.

If the I/O signal is true (i.e., transfer to the initiator), to transfer SPI information units, the target:

1. shall drive the DB(15-0) signals to their values;
2. shall wait at least one transmit setup time from DB(15-0) being driven with valid data;
3. shall transition the REQ signal;
4. shall hold the DB(15-0) signals valid for a minimum of one transmit hold time;
5. may change or release the DB(15-0) signals; and
6. shall not change the REQ signal for a minimum of one transmit assertion period.

If the I/O signal is true (i.e., transfer to the initiator), to receive SPI information units, the initiator shall:

1. read the value on the DB(15-0) signals within one receive hold time of the transition of the REQ signal; and
2. respond with an ACK transition.

If the I/O signal is false (i.e., transfer to the target), to transfer SPI information units, the initiator:

1. shall wait until after detecting a REQ transition;
2. shall drive the DB(15-0) signals to their desired values;

3. shall delay at least one transmit setup time;
4. shall transition the ACK signal;
5. shall hold the DB(15-0) signals valid for at least one transmit hold time;
6. shall not change the ACK signal for a minimum of one transmit assertion period; and
7. may then change or release the DB(15-0) signals.

If the I/O signal is false (i.e., transfer to the target), to receive SPI information units, the target:

1. shall read the value of the DB(15-0) signals within one receive hold time of the transition of the ACK;
2. shall not transition the REQ signal for the current SPI information unit until the initiator has responded with all ACK transitions for the previous SPI information unit.

As a result of a SPI information unit always being an even number of transfers, the REQ and ACK signals are negated both before and after the transmission of the SPI information unit.

3.5.2.2.1.1 DT DATA IN phase information unit transfer exception condition handling

The initiator shall not negate the ACK for the last byte of the last iuCRC in an information unit until the entire information unit has been verified and any required attention condition has been established.

If the nexus has been fully identified (i.e., an I_T_L_Q nexus has been established) and the initiator detects an iuCRC error in any information unit (other than a SPI status information unit) it receives while in the DT DATA IN phase, the initiator shall create an attention condition on or before the last iuCRC within the failed information unit is acknowledged. When the target switches to a MESSAGE OUT phase, the initiator should send a SCSI initiator port Detected Error message (see Section 4.3.5) to the target. This message notifies the target that data in the information unit was invalid.

If a SCSI initiator port detects an iuCRC error in a SPI status information unit, the initiator shall create an attention condition on or before the last iuCRC of the information unit is acknowledged. If the target detects an attention condition, it shall switch to a MESSAGE OUT phase and the initiator shall send a SCSI initiator port Detected Error message (see Section 4.3.5) or an ABORT TASK message to the target. These messages notify the target that the SPI status information unit was invalid and the message received from the initiator was a SCSI initiator port Detected Error message.

If the information unit that failed was not a SPI status information unit, then the target shall send a SPI L_Q/SPI status information unit pair to the initiator with a CHECK CONDITION status and a sense key set to Aborted Command and an additional sense code set to Initiator Detected Error Message Received for the task associated with the received INITIATOR DETECTED ERROR message.

If the information unit that failed was a SPI status information unit and the message received was a SCSI initiator port Detected Error message, then the target shall retry transferring the SPI L_Q/SPI status information unit pair to the initiator with the original status information.

If the information unit that failed was a SPI status information unit and the message received was an ABORT TASK message, then the target shall cause a bus free by generating a BUS FREE phase.

If the initiator is receiving a SPI L_Q information unit and the initiator detects an iuCRC error (i.e., the nexus identification fails) while in the DT DATA IN phase, the initiator shall create an attention condition on or before the iuCRC is acknowledged. When the target switches to a MESSAGE OUT phase, the initiator should send a SCSI initiator port Detected Error message (see Section 4.3.5) to the target. This message notifies the target that the nexus identification failed. The target shall then cause a bus free by generating a BUS FREE phase, however, the target shall retry the task associated with the failed SPI L_Q information unit.

If the initiator receives a SPI L_Q information unit with a type code that is not defined in Table 50, that initiator shall create an attention condition after negating the ACK for the last byte of the iuCRC in the SPI L_Q information unit and before negating the ACK for the last byte of the last iuCRC in the information unit that follows the SPI L_Q information unit. When the target switches to a MESSAGE OUT phase, the initiator shall send an

ABORT TASK message (see Section 4.5.2) to the target. The target shall send a SPI L_Q/SPI status information unit pair to the initiator with a CHECK CONDITION status and a sense key set to Aborted Command for the task associated with the received ABORT TASK message.

3.5.2.2.1.2 DT DATA OUT phase information unit transfer exception condition handling

The target shall only respond to an iuCRC error after all the data in an information unit has been received.

If the nexus has been fully identified (i.e., an I_T_L_Q nexus has been established) and the target detects an iuCRC error in any SPI information unit it receives while in the DT DATA OUT phase, the target shall, before receiving another SPI L_Q information unit, switch to a DT DATA IN phase and send a SPI L_Q/SPI status information unit pair to the initiator with a CHECK CONDITION status and a sense key set to Aborted Command and the additional sense code set to iuCRC Error Detected for the task associated with the iuCRC error.

If the target detects an iuCRC error on an iuCRC interval that is not at the end of a SPI information unit, the target shall not respond to the error until all the bytes of the SPI information unit in which the error occurred have been transferred, however, the target may discard the transmitted information.

If the target is receiving a SPI L_Q information unit and the target detects an iuCRC error (i.e., the nexus identification fails), the target shall cause an unexpected bus free by generating a BUS FREE phase (see Section 3.1.1).

If a SCSI target port receives a SPI L_Q information unit with a type code that is not defined in Table 50, that target shall transfer all the bytes indicated by the data length and iuCRC interval and shall discard the transmitted information. After transferring all the bytes, the target shall change to a DT DATA IN phase and transmit a SPI status information unit with a RSPVALID bit of one and the packetized failure code set to Invalid Type Code Received in SPI L_Q Information Unit.

If a SCSI target port receives a SPI L_Q information unit with an illegal data length (see 14.3.2) the SCSI target port shall transfer all the bytes indicated by the data length and iuCRC interval and shall discard the transmitted information. After transferring all the bytes the SCSI target port shall change to a DT DATA IN phase and transmit a SPI status information unit with a RSPVALID bit of one and the packetized failure code set to ILLEGAL REQUEST RECEIVED IN SPI L_Q INFORMATION UNIT.

3.5.2.2.2 Data group data field transfer

When the target is transferring consecutive data groups, it shall not transition the REQ signal while the P_CRCA signal is asserted for the current data group until the initiator has acknowledged the entire previous data group.

Note. The requirement above ensures the initiator is not required to maintain more than one simultaneous pCRC calculation in different data groups.

If the I/O signal is true (i.e., transfer to the initiator), to transfer the data field, the target:

1. shall drive the DB(15-0) signals to their values and shall negate the P_CRCA signal;
2. shall wait at least the longer of one pCRC transmit setup time from the negation of P_CRCA or one transmit setup time from DB(15-0) being driven with valid data;
3. shall transition the REQ signal;
4. shall hold the DB(15-0) signals valid for a minimum of one transmit hold time and shall hold the P_CRCA signal for a minimum of a pCRC transmit hold time;
5. may change or release the DB(15-0) and P_CRCA signals; and
6. shall not change the REQ signal for at least one transmit assertion period if asserted or one transmit negation period if negated.

If the I/O signal is true (i.e., transfer to the initiator), to receive the data field, the initiator shall:

1. read the value on the DB(15-0) signals within one receive hold time of the transition of the REQ signal;

2. read the value of the P_CRCA signal within one pCRC receive hold time of the transition of the REQ signal; and
3. respond with an ACK transition.

If the I/O signal is false (i.e., transfer to the target), to transfer the data field, the initiator:

1. shall wait until after detecting a REQ transition with P_CRCA negated;
2. shall drive the DB(15-0) signals to their values;
3. shall delay at least one transmit setup time;
4. shall transition the ACK signal;
5. shall hold the DB(15-0) signals valid for at least one transmit hold time;
6. may then change or release the DB(15-0) signals; and
7. shall not change the ACK signal for at least one transmit assertion period if asserted or one transmit negation period if negated.

If the I/O signal is false (i.e., transfer to the target), to receive the data field, the target:

1. shall read the value of the DB(15-0) signals within one receive hold time of the transition of the ACK.

3.5.2.2.2.1 Data group pad field and pCRC field transfer to SCSI initiator port

The target detects a pad field is required if the I/O signal is true (i.e., transfer to the initiator), the target has completed the data field transfer of the current data group, and REQ signal is asserted. In this case, the target shall:

1. wait at least one pCRC transmit hold time since the last REQ assertion to assert P_CRCA;
2. wait at least one transmit hold time since the last REQ assertion to assert the DB(15-0) signals to their pad values;
3. wait at least the longer of one pCRC transmit setup time from the assertion of P_CRCA or one transmit setup time from DB(15-0) being driven with valid pad data;
4. wait until the initiator has responded with all ACK transitions for the previous data group;
5. wait at least one transmit REQ assertion period with P_CRCA transitioning since the last REQ assertion;
6. negate the REQ signal without waiting for the ACK transition corresponding to the previous REQ transition unless the negotiated offset would be exceeded;
7. hold the DB(15-0) signals valid for a minimum of one transmit hold time and hold the REQ signal negated for a minimum of one transmit negation period;
8. drive the DB(15-0) signals to their pCRC values;
9. wait at least one transmit setup time;
10. assert the REQ signal without waiting for the ACK transition corresponding to the previous REQ transition unless the negotiated offset would be exceeded;
11. hold the DB(15-0) signals for a minimum of one transmit hold time and hold the REQ signal asserted for a minimum of one transmit assertion period;
12. drive the DB(15-0) signals to their pCRC values;
13. wait at least one transmit setup time;
14. negate the REQ signal without waiting for the ACK transition corresponding to the previous REQ transition unless the negotiated offset would be exceeded;
15. hold the DB(15-0) signals for a minimum of one transmit hold time and hold the P_CRCA signal asserted for at least one pCRC transmit hold time; and
16. hold the REQ signal negated for at least one transmit REQ negation period with P_CRCA transitioning since the last REQ negation.

Note. The above requirements in steps 6), 10), and 14) to not wait for the ACK transition corresponding to the previous REQ transition were not present in the SPI-3 standard. For compatibility with old designs SCSI initiator ports should generate ACK transitions for all received REQ transitions.

If the SCSI target port determines that a pad field is not required, has completed the data field transfer of the current data group, the I/O signal is true (i.e., transfer to the SCSI initiator port), and the REQ signal is negated, the SCSI target port shall:

1. wait at least one pCRC transmit hold time since the last REQ negation to assert P_CRCA;
2. wait at least one transmit hold time since the last REQ negation to assert the DB(15-0) signals to their pCRC values;
3. wait at least the longer of one pCRC transmit setup time from the assertion of P_CRCA or a transmit setup time from DB(15-0) being driven with valid pCRC data;
4. wait until the initiator has responded with all ACK transitions for the previous data group;
5. wait at least one transmit REQ negation period with P_CRCA transitioning since the last REQ negation;
6. assert the REQ signal;
7. hold the DB(15-0) signals for a minimum of one transmit hold time and hold the REQ signal asserted for a minimum of one transmit assertion period;
8. drive the DB(15-0) signals to their pCRC values;
9. wait at least one transmit setup time;
10. negate the REQ signal;
11. hold the DB(15-0) signals for a minimum of one transmit hold time and hold the P_CRCA signal asserted for a minimum of one pCRC transmit hold time; and
12. hold the REQ signal negated for at least one transmit REQ negation period with P_CRCA transitioning since the last REQ negation.

After either of the above sequences is complete, the target has ended a data group transfer.

The initiator shall read the value on the DB(15-0) signals within one receive hold time of the transition of the REQ signal. The initiator shall then respond with an ACK transition.

The initiator shall continue to use the pad bytes, if any, for checking against the computed pCRC for the current data group. Upon receipt of the last byte of the pCRC field, the received pCRC and computed pCRC shall be compared. If they do match (i.e., no pCRC error), then the initiator shall negate the ACK signal.

If received pCRC and computed pCRC do not match (i.e., a pCRC error is detected) or if an improperly formatted data group is transferred, then the initiator shall create an attention condition on or before the last transfer of the data group. When the target switches to a MESSAGE OUT phase, the initiator should send a SCSI initiator port Detected Error message (see Section 4.3.5) to the target. This message notifies the target that data contained within the data group was invalid.

If the target does not retry transferring the information transfer or it exhausts its retry limit, the target shall go into a STATUS phase and send a CHECK CONDITION status with a sense key set to Aborted Command and an additional sense code set to Initiator Detected Error Message Received for the task associated with the received INITIATOR DETECTED ERROR message.

3.5.2.2.2.2 Data group pad field and pCRC field transfer to SCSI target port

If the I/O signal is false (i.e., transfer to the target), the initiator determines the data field. Transfer to the target is completed by detecting an assertion of the P_CRCA signal. If the REQ signal is asserted (i.e., pad field required), the initiator shall first transfer the two pad bytes, then the four pCRC bytes. If the REQ signal is negated (i.e., no pad field required), the initiator shall transfer the four pCRC bytes.

Pad field data and pCRC field data are transferred using the same negotiated transfer period as the data field data.

The target may continue to send REQs, up to the negotiated offset, for the next data group. The target shall not transition REQ with P_CRCA asserted until the initiator has responded with all ACK transitions for the previous data group.

When the initiator detects an assertion of the P_CRCA signal and the REQ signal is asserted (i.e., pad field required), it shall then:

1. transfer data bytes for all outstanding REQs received prior to the REQ that had the P_CRCA signal asserted;
2. drive the DB(15-0) signals to their pad values;
3. delay at least one transmit setup time;
4. negate the ACK signal;
5. hold the DB(15-0) signals valid for a minimum of one transmit hold time and hold the ACK signal negated for a minimum of one transmit assertion period;
6. drive the DB(15-0) signals to their pCRC values;
7. delay at least one transmit setup time;
8. assert the ACK signal;
9. hold the DB(15-0) signals valid for a minimum of one transmit hold time and hold the ACK signal asserted for a minimum of one transmit assertion period;
10. drive the DB(15-0) signals to their pCRC values;
11. delay at least one transmit setup time;
12. negate the ACK signal; and
13. hold the DB(15-0) signals valid for a minimum of one transmit hold time and hold the ACK signal negated for a minimum of one transmit assertion period.

When the initiator detects an assertion of the P_CRCA signal and the REQ signal is negated (i.e., no pad field required), it shall then:

1. transfer data bytes for all outstanding REQs received prior to the REQ that had the P_CRCA signal asserted;
2. drive the DB(15-0) signals to their pCRC values;
3. delay at least one transmit setup time;
4. assert the ACK signal;
5. hold the DB(15-0) signals valid for a minimum of one transmit hold time and hold the ACK signal asserted for a minimum of one transmit assertion period;
6. drive the DB(15-0) signals to their pCRC values;
7. delay at least one transmit setup time;
8. negate the ACK signal; and
9. hold the DB(15-0) signals valid for a minimum of one transmit hold time and hold the ACK signal negated for a minimum of one transmit assertion period.

After either of the above sequences is complete, the target has ended a data group transfer.

As a result of a data group always being an even number of transfers, the REQ and ACK signals are negated both before and after the transmission of the data group.

The target shall read the value of the DB(15-0) signals within one receive hold time of the transition of the ACK signal. The initiator shall use the pad bytes, if any, in the generation of the transmitted pCRC. The target shall then use those pad bytes, if any, for checking against the computed pCRC for the current data group. Upon receipt of the last byte of the pCRC field, the received pCRC and computed pCRC shall be compared.

If received pCRC and computed pCRC do not match (i.e., a pCRC error is detected), or if an improperly formatted data group is transferred, then the associated data group shall be considered invalid.

If the target does not retry transferring the information transfer or it exhausts its retry limit, the target shall go into a STATUS phase and send a CHECK CONDITION status with a sense key set to Aborted Command and an additional sense code set to SCSI Parity Error for the task associated with the pCRC error.

3.5.3 Paced transfer

If a paced transfer agreement has been established, it shall be used in DT DATA phase and information unit transfers shall be used. The transfer agreement also specifies the REQ/ACK offset and the transfer period.

When paced transfers are being used data shall be transferred using DT data transfers on 16-bit wide buses that transmit and receive data using LVD transceivers.

If driver precompensation is enabled at the SCSI device, that SCSI device shall apply driver precompensation to all the data, parity, REQ, and ACK signals.

During paced DT data transfers, if the phase of the P1 signal indicates data is valid on REQ or ACK assertions, data shall be clocked by the originating SCSI device by both the assertion and negation of the REQ or ACK signal lines. The receiving SCSI device shall clock DT data on both the assertion and negation of the REQ or ACK signal line after having been processed by the receiving SCSI device. If the phase of the P1 signal indicates data is invalid on REQ or ACK assertions, data shall not be clocked by the originating SCSI device and shall be ignored by the receiving SCSI device. If driver precompensation is enabled at the originating SCSI device, the originating SCSI device shall apply driver precompensation to all the data signals, the P_CRCA signal, the P1 signal, and the REQ, and or ACK signal.

For paced DT DATA IN phases the REQ/ACK offset specifies the maximum number of data valid state REQ assertions that shall be sent by the SCSI target port in advance of the number of ACK assertions received from the SCSI initiator port. If the number of data valid state REQ assertions exceeds the number of ACK assertions by the REQ/ACK offset, the SCSI target port shall change P1 to enable the data invalid state prior to the next assertion of REQ and shall not change P1 to enable a data valid state until after the next ACK assertion is received. For successful completion of a paced DT DATA IN phase, the number of data valid state REQ assertions and ACK assertions shall be equal. Each assertion indicates a single 32-bit data transfer.

For paced DT DATA OUT phases the REQ/ACK offset specifies the maximum number of REQ assertions that shall be sent by the SCSI target port in advance of the number of data valid state ACK assertions that shall be sent by the SCSI target port in advance of the number of data valid state ACK assertions received from the SCSI initiator port. If the number of REQ assertions exceeds the number of data valid state ACK assertions by the REQ/ACK offset, the SCSI target port shall not assert REQ until after the next data valid state ACK assertion is received. For successful completion of a paced DT DATA OUT phase, the number of REQ assertions and data valid state ACK assertions shall be equal. Each assertion indicates a single 32-bit data transfer.

Implementors shall not use the following subsections for timing requirements. For timing requirements, see Section 2.5.

3.5.3.1 Paced transfer training pattern

After any PPR negotiation occurs that enables paced transfers, a training pattern shall be transferred at the start of the first DT data phase for each data transfer direction regardless of the negotiated value of the RTI bit.

If retain training information is disabled, a training pattern shall be transferred at the start of the first DT DATA phase for each data transfer direction after each physical connect and physical reconnect. The training pattern shall not be transferred again until after a physical disconnection occurs.

If the retain training information is enabled, a training pattern shall be transferred at the start of the first DT DATA phase for each data transfer direction after the retain training information is enabled. The SCSI device shall save training configuration values for each I_T nexus that has negotiated to retain training information. The SCSI device shall use the saved training configuration values for all paced transfers. The SCSI target port may retrain an I_T nexus if it determines the training configuration values are invalid, without having to renegotiate the retain training information protocol option.

Note. The training configuration values are vendor specific.

If the retain training information is enabled and a port changes from a SCSI initiator port to a SCSI target port that SCSI target port shall retrain if the saved training configuration values were saved while the port was a SCSI initiator port.

The training pattern for a DATA IN phase shall conform to Section 3.5.3.1.1. The training pattern for a DATA OUT phase shall conform to Section 3.5.3.1.2. The receiving SCSI device shall use some or all elements of the training pattern to achieve deskewing. The transmitting SCSI device shall not make an intentional shift in relative timing between the data bus signals and the REQ or ACK signal during the DT data phase.

Note. The requirement to not intentionally change relative timing does not include the effects of ISI, noise, or jitter.

The training pattern consists of three sections: A, B, and C. Each section contains a different pattern that may be used to train circuits within a receiver.

3.5.3.1.1 DT DATA IN phase training pattern

The SCSI target port shall indicate a training pattern is going to occur on a DT DATA IN phase by:

1. releasing SEL for a minimum of two system deskew delays;
2. asserting the SEL signal a minimum of two system deskew delays; and
3. then asserting the REQ signal.

The SCSI target port shall begin the section A of its training pattern only after all the signal restrictions between information transfers phases listed in Section 3.10 or the signal restrictions between a RESELECTION phase and a DT DATA IN phase listed in Section 3.3 are met. The SCSI target port shall transmit the following training pattern:

Start of section A

1. if precompensation is enabled, then set the drivers to the strong driver state;
2. simultaneously assert REQ, P1, P_CRCA, and DB(15-0) signals;
3. wait the equivalent of 32 transfer periods (e.g., 200 ns at Fast-160);
4. simultaneously negate REQ, P1, P_CRCA, and DB(15-0) signals;
5. wait the equivalent of 32 transfer periods;
6. set precompensation to negotiated state;
7. negate SEL signal;
8. simultaneously assert and negate REQ, P1, P_CRCA, and DB(15-0) signals at the negotiated transfer period for 800 ns, (e.g., the equivalent of 128 transfer periods at Fast-160);

Start of section B

1. wait the equivalent of 192 transfer periods from the first assertion of REQ in step 2 of section A (e.g., 1200 ns at Fast-160);
2. keep the P1, P_CRCA, and DB(15-0) signals negated while continuing to assert and negate REQ at the negotiated transfer period for the equivalent of 8 transfer periods (e.g., 50 ns at Fast-160);
3. keep the P1, P_CRCA, DT(15-0), and REQ signals negated for the equivalent of 8 additional transfer periods;

4. simultaneously assert and negate P1, P_CRCA, and DB(15-0) signals at twice the negotiated transfer period (i.e., simultaneously repeat a 1100b bit pattern 12 times on each signal) while asserting and negating REQ at the negotiated transfer period 24 times [e.g., (2 x 6.25 ns) x 24 = 300 ns at Fast-160]; and

Start of section C

1. assert and negate REQ at the negotiated transfer period 64 times and at the same time assert and negate P1 at twice the negotiated transfer period while repeating a 0000010011111011b bit pattern on each of the P_CRCA and DB(15-0) signals [e.g., the equivalent of 128 transfer periods at Fast-160].

The SCSI initiator port shall begin its training pattern independent of the start of the SCSI target ports training pattern if it detects the SEL, MSG, and I/O true and C/D false on the first assertion of the REQ signal. The SCSI initiator port shall transmit the following training pattern:

1. assert ACK signal within 200 ns of the first REQ assertion;
2. if precompensation is enabled then set the drivers to the strong driver state;
3. wait the equivalent of 32 transfer periods (e.g., 200 ns at Fast-160);
4. negate ACK signal;
5. wait the equivalent of 32 transfer periods (e.g., 200 ns at Fast-160);
6. set precompensation to negotiated state; and
7. assert and negate ACK signal at the negotiated transfer period for 400 ns, (e.g., (2 x 6.25 ns) x 32 = 400 ns at Fast-160.

At the completion of its training pattern, the SCSI target port continues asserting and negating the REQ signal at the negotiated transfer period (e.g., 6.25 ns transfer period at Fast-160) and the P1 signal at twice the negotiated transfer period (e.g., 12.5 ns transfer period at Fast-160). When the SCSI target port is ready to transfer data it shall reverse the phase of P1 (see Section 3.5.3.2).

3.5.3.1.2 DT DATA OUT phase training pattern

The SCSI target port shall request a training pattern on a DT DATA OUT phase by asserting the SEL signal a minimum of two system deskew delays before asserting the REQ signal.

The SCSI target port shall begin its training pattern only after all the signal restrictions between a SELECTION phase and a DT DATA OUT phase listed in Section 3.2.1.2 are met. The SCSI target port shall transmit the following training pattern:

1. if precompensation is enabled, set the drivers to the strong driver state;
2. simultaneously assert REQ and P_CRCA signals;
3. wait the equivalent of 32 transfer periods (e.g., 200 ns at Fast-160);
4. simultaneously negate REQ and P_CRCA signals;
5. wait the equivalent of 32 transfer periods (e.g., 200 ns at Fast-160);
6. set precompensation to negotiated state;
7. negate SEL signal;
8. simultaneously assert and negate REQ and P_CRCA signals at the negotiated transfer period 32 times (e.g., (2 x 6.25) x 32 = 400 ns at Fast-160);
9. negate REQ and P_CRCA for at least the equivalent of 16 transfer periods (e.g., 100 ns at Fast-160); and
10. the SCSI target port shall begin asserting and negating REQ to indicate to the SCSI initiator port valid data may be sent. The number of REQ assertions shall not exceed the negotiated REQ/ACK offset.

The SCSI initiator port shall begin the section A of its training pattern independent of the start of the SCSI target ports training pattern if it detects the SEL and MSG true, and C/D and I/O false on the first assertion of the REQ signal. The SCSI initiator port shall transmit the following pattern:

For fast-320 the SCSI initiator port shall transmit the training pattern described by section A, section B, and section C in this subclause except that the polarity of DB(0, 1, 4, 5, 9, 10, 13, and 14) shall be inverted during transmission of section A, section B, and section C (i.e., where it is specified that these signals shall be asserted, they shall be negated, and where it is specified that these signals shall be negated, they shall be asserted). These signals shall return to their normal polarity after completion of training pattern

transmission.

Start of section A:

1. if precompensation is enabled, set the drivers to the strong driver state;
2. simultaneously assert ACK, P1, and DB(15-0) signals within the equivalent of 32 transfer periods of the first REQ assertion (e.g., 200 ns at Fast-160);
3. wait the equivalent of 32 transfer periods (e.g., 200 ns at Fast-160);
4. simultaneously negate ACK, P1, and DB(15-0) signals;
5. wait the equivalent of 32 transfer periods;
6. set precompensation to negotiated state;
7. simultaneously assert and negate ACK, P1, and DB(15-0) signals at the negotiated transfer period 64 times (e.g., (2 x 6.25) x 64 = 800 ns at Fast-160);

Start of section B:

1. wait the equivalent of 192 transfer periods from the first assertion of ACK in step 2 of section A (e.g., 1200 ns at Fast-160);
2. keep the P1, and DB(15-0) signals negated while continuing to assert and negate ACK at the negotiated transfer period for the equivalent of 8 transfer periods (e.g., 50 ns at Fast-160);
3. keep the P1, DB(15-0), and ACK signals negated for the equivalent of 8 additional transfer periods;
4. simultaneously assert and negate PA and DB(15-0) signals at twice the negotiated transfer period (i.e., simultaneously repeat a 1100b bit pattern 12 times on each signal) while asserting and negating ACK at the negotiated transfer period 24 times (e.g., (2 x 6.25) x 24 = 300 ns at Fast-160);

Start of section C:

1. assert and negate ACK at the negotiated transfer period 64 times and at the same time assert and negate P1 at twice the negotiated transfer period while repeating a 0000010011111011b bit pattern eight times on each of the DB(15-0) signals (e.g., 2 x 6.25 ns) x 64 = 800 ns at Fast-160).

At the completion of its training pattern, the SCSI initiator port continues asserting and negating the ACK signal at the negotiated transfer period (e.g., 6.25 ns transfer period at Fast-160) and the P1 signal at twice the negotiated transfer period (e.g., 12.5 ns transfer period at Fast-160). When the SCSI initiator port is ready to transfer data and the REQ/ACK offset value is not zero it shall reverse the phase of P1 (see Section 3.5.3.2 below).

3.5.3.2 P1 data valid/invalid state transitions

The transmitting SCSI device port shall indicate the start of a data valid state by reversing the phase of the P1 signal coincident with a REQ or ACK assertion. This is accomplished by withholding the next transition of P1 at the start of the first two transfer periods of valid data. Beginning with the third valid data word, P1 shall be toggled every two transfer periods, coincident with a REQ or ACK assertion. The minimum duration of the data valid state is two transfer periods, and the data valid state shall consist of an even number of transfer periods.

Anytime the sending SCSI device port pauses the sending of data, it shall reverse the phase of P1 by withholding the next transition of P1 at the start of the first two transfer periods that have invalid data. Beginning with the third transfer period with invalid data, P1 shall be toggled every two transfer periods until valid data is sent.

The data invalid state shall have at least one transition of P1 before changing states. The minimum data invalid time is four transfer periods. This ensures a maximum run length of three cycles for P1. The data invalid state shall last an even number of transfer periods.

From the data invalid state, the sending SCSI device port may resume sending data by reversing the phase of P1 again.

P1 has the same transmit setup and hold time requirements as data and shall always be detected by the receiving device on the assertion edge of the delayed clocked REQ or ACK signal.

See Figure 10 for examples of how the P1 signal is used to determine when the REQ or ACK transition clocks valid data.

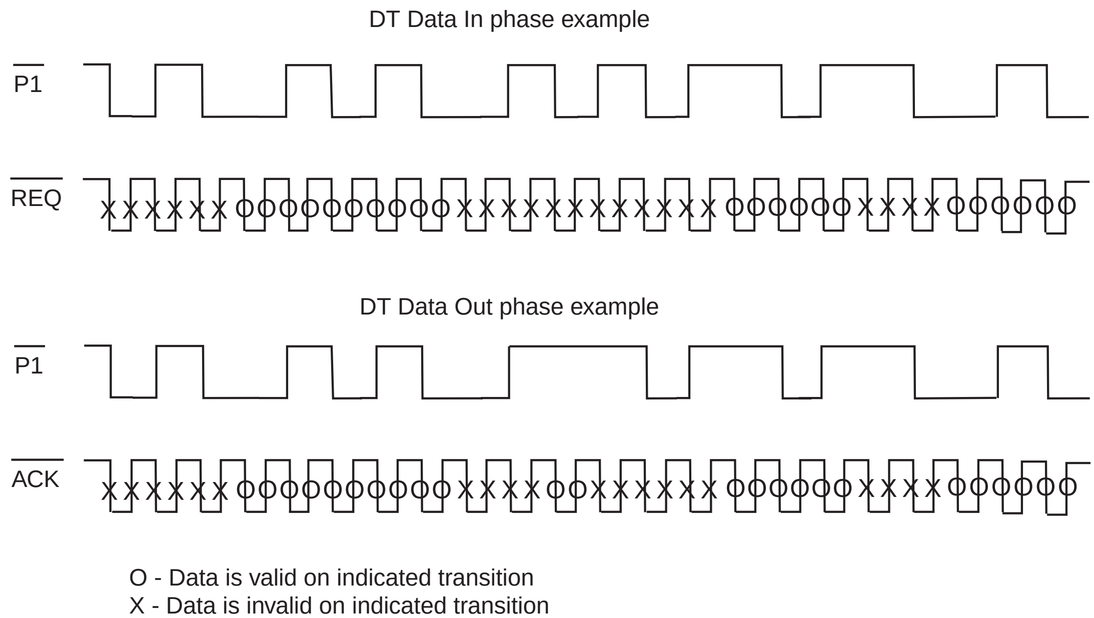


Figure 10. Use of P1 to establish data valid and data invalid states

3.5.3.2.1 Starting pacing transfers at end of training pattern

See Section 3.5.3.1 for the description of starting a data valid state after a training pattern.

3.5.3.2.2 Starting pacing transfers with no training pattern

Before starting the DT DATA IN phase, the SCSI target port shall wait at least two system deskew delays after the SEL signal is negated before the first assertion of the REQ signal.

The DT DATA IN phase without training starts on the first assertion of REQ if the SEL is not asserted.

The SCSI target port shall begin pacing transfers only after meeting all the following:

- signal restrictions between information transfer phases listed in Section 3.10;
- the signal restrictions between a RESELECTION phase and a DT DATA IN phase listed in Section 3.3.2; or
- the signal restrictions between a SELECTION phase and a DT DATA OUT phase listed in Section 3.2.1.2.

The SCSI target port shall begin pacing transfers by:

- simultaneously with the assertion of REQ, the SCSI target port shall begin asserting and negating P1 at twice the negotiated transfer period (e.g., 12.5 ns for Fast-160);
- SCSI target port shall assert and negate P1 at least 8 times [e.g., (2 x 6.25 ns) x 8 = 100 ns at Fast-160]; and
- the SCSI target port may establish a data valid state as described in Section 3.5.3.2.

The DT DATA OUT phase without training starts on the first assertion of REQ if the SEL is not asserted.

The SCSI target port shall begin pacing transfers only after meeting all the following:

- signal restrictions between information transfer phases listed in Section 3.10;
- the signal restrictions between a RESELECTION phase and a DT DATA IN phase listed in Section 3.3.2; or
- the signal restrictions between a SELECTION phase and a DT DATA OUT phase listed in Section 3.2.1.2.

The SCSI initiator port shall begin pacing transfers by:

- simultaneously with the assertion of ACK the SCSI initiator port shall begin asserting and negating P1 at twice the negotiated transfer period (e.g., 12.5 ns for Fast-160);
- SCSI initiator port shall assert and negate P1 at least 8 times (e.g., (2 x 6.25 ns) x 8 = =100 ns at Fast-160); and
- the SCSI initiator port may establish a data valid state as described in Section 3.5.3.2.

3.5.3.2.3 Ending pacing transfers

After transmitting the last data word of a DT DATA IN phase, the SCSI target port shall end pacing by waiting for all REQs to be responded to by ACKs then negate the REQ and P1 signals. After the SCSI target port stops asserting and negating REQ, it shall not assert REQ again until the requirements in Section 3.10 are met.

After transmitting the last data word of a DT DATA OUT phase, the SCSI initiator port shall:

- continue asserting and negating the ACK and P1 signals until it detects a change to the C/D, I/O, or MSG signals; and
- negate the ACK and P1 signals within 200 ns of detecting a change to the C/D, I/O, or MSG signals.

When the SCSI target port changes from a DT DATA OUT phase to any other phase it shall wait at least a bus settle delay plus a data release delay before asserting REQ and shall ignore any ACK transitions for at least a bus settle delay plus a data release delay after transitioning the C/D, I/O, or MSG signals.

3.5.3.3 Paced information unit transfer

Information units shall be transferred on the DT DATA OUT phase and the DT DATA IN phase, and the information units' embedded iuCRC shall be used to detect information unit data errors.

If the I/O signal is true (i.e., transfer to the SCSI initiator port) and the phase of the P1 signal indicates data is valid, to transfer SPI information units the SCSI target port:

(a) shall drive the DB(15-0) signals to their values simultaneous with the next REQ signal assertion;

(b) shall hold the DB(15-0) signals valid for a minimum of one transmit hold time;

(c) shall drive the DB(15-0) signals to their values simultaneous with the next REQ signal negation; and

(d) shall hold the DB(15-0) signals valid for a minimum of one transmit hold time.

If the I/O signal is true (i.e., transfer to the SCSI initiator port), to receive SPI information units the SCSI initiator port shall:

(a) read the value on the DB(15-0) signals within one receive hold time of the transition of the REQ signal; and

(b) respond with an ACK signal assertion after each REQ assertion/negation pair.

If the I/O signal is false (i.e., transfer to the SCSI target port) and the phase of the P1 signal indicates data is valid, to transfer SPI information units the SCSI initiator port:

(a) shall wait until detecting a REQ assertion;

(b) shall drive the DB(15-0) signals to their values simultaneous with the next ACk signal assertion;

(c) shall hold the DB(15-0) signals valid for a minimum of one transmit hold time;

(d) shall drive the DB(15-0) signals to their values simultaneous with the next ACK signal negation;

and

(e) shall hold the DB(15-0) signals valid for a minimum of one transmit hold time.

If the I/O signal is false (i.e., transfer to the SCSI target port), to receive SPI information units the SCSI target port:

(a) shall read the value of the DB(15-0 signals within one receive hold time of the transition of the ACK signal.

If write flow control is enabled and the current SPI data stream information unit is the last SPI data stream information unit of the stream:

(a) the SCSI target port shall assert the P_CRCA signal a minimum of a flow control transmit setup time before the end of the last information unit and shall keep the P_CRCA signal asserted for a flow control transmit hold time;

(b) the SCSI target port shall not assert the P_CRCA signal until a minimum of a flow control hold time after the end of the previous information unit; and

(c) the SCSI target port shall negate the P_CRCA signal a minimum of a flow control transmit setup time before the start of the next information unit.

Note. The earlier in a SPI data stream information unit that the SCSI target port asserts the P_CRCA signal, the better the SCSI initiator port may manage data pre-fetch.

As a result of a SPI information unit always being an even number of transfers, the REQ and ACK signals are negated both before and after the transmission of the SPI information unit.

Paced information unit transfers exception handling conditions are defined in Section 3.5.2.2.1.1 and Section 3.5.2.2.1.2.

3.5.3.4 Deskewing

The deskewing technique used in the receiving SCSI device is vendor specific. Any technique that works with the specified training pattern and complies with the specified receive skew compensation timing requirement is allowed. Deskewing shall only be enabled for paced transfers.

3.5.4 Wide transfer

Wide data transfers shall be used for DT DATA phases. Wide data transfer may be used in the ST DATA phase only if a wide transfer agreement is in effect (see section 4.3.18 or 4.3.12).

All SCSI devices shall support 8-bit narrow data transfers.

During narrow transfers, all information shall be transferred in bytes across the DB(7-0) and DB(P_CRCA) signals on the SCSI bus. At the receiving device the DB(15-8) (if present) and DB(P1) (if present) signals are undefined.

During wide transfers, the first and second information bytes for each DATA phase shall be transferred across the DB(7-0) and DB(15-8) signals, respectively, on the SCSI bus. Subsequent pairs of information bytes are likewise transferred in parallel across the SCSI bus (see Table 22).

The IGNORE WIDE RESIDUE message may be used to indicate that the last byte of a data field or the last data byte of information unit is undefined.

Table 22: Wide SCSI byte order

Transfer number	SCSI bus 15....8	SCSI bus 7....0	Data transfer width
1	N/A	W	8-bit
2	N/A	X	
3	N/A	Y	
4	N/A	Z	
1	X	W	16-bit
2	Z	Y	
When transferring consecutive bytes W, X, Y, and Z across the buses, they are transferred as shown above. This table does not necessarily represent how these bytes are stored in device memory.			

If the last information byte transferred does not fall on the DB(15-8) signals for a 16-bit wide transfer, then the values of the remaining higher-numbered bits are undefined. However, when using parity protection, the DB(P1) signal for this undefined byte shall be valid for whatever data is placed on the bus.

3.6 COMMAND phase

3.6.1 COMMAND phase description

The COMMAND phase allows the target to request command information from the initiator.

The SCSI target port shall assert the C/D signal and negate the I/O and MSG signals during the REQ/ACK handshakes of this phase.

A QAS-capable initiator shall wait a minimum of a QAS non-data phase REQ(ACK) period to assert ACK after detecting the assertion of REQ.

A QAS-capable initiator shall assert ACK for a minimum of a QAS non-data phase REQ(ACK) period and shall keep the command data valid until the negation of ACK.

3.6.2 COMMAND phase exception condition handling

If the target detects one or more parity errors on the command bytes received, it may retry the command by switching to the MESSAGE IN phase and sending a RESTORE POINTERS message. The target shall then switch to the COMMAND phase to receive the original command.

If the target does not retry the COMMAND phase or it exhausts its retry limit it shall return CHECK CONDITION status and set the sense key to Aborted Command and the additional sense code to SCSI Parity Error.

3.7 DATA phase

3.7.1 DATA phase overview

DATA phase is a term that encompasses both the ST DATA phases and the DT DATA phases. ST DATA phase is a term that encompasses both the ST DATA IN phase and ST DATA OUT phase. DT DATA phase is a term that encompasses both the DT DATA IN phase, and the DT DATA OUT phase.

3.7.2 DT DATA IN phase

The DT DATA IN phase allows the target to request that data be sent to the initiator from the target using DT data transfers. The target shall assert the I/O and MSG signals and negate the C/D signal during the REQ/ACK handshakes of this phase.

3.7.3 DT DATA OUT phase

The DT DATA OUT phase allows the SCSI target port to request that data be sent from the SCSI initiator device to the SCSI target device using DT data transfers. The target shall assert the MSG signal and negate the C/D and I/O signals during the REQ/ACK handshakes of this phase.

3.7.4 ST DATA IN phase

The ST DATA IN phase allows the SCSI target port to request that data be sent to the initiator from the target using ST data transfers. The target shall assert the I/O signal and negate the C/D and MSG signals during the REQ/ACK handshake(s) of this phase.

3.7.5 ST DATA OUT phase

The ST DATA OUT phase allows the SCSI target device to request that data be sent from the SCSI initiator device to the target device using ST data transfers. The SCSI target port shall negate the C/D, I/O, and MSG signals during the REQ/ACK handshakes of this phase.

3.8 STATUS phase

3.8.1 STATUS phase description

The STATUS phase allows the SCSI target device to request that a status byte be sent from the target to the SCSI initiator device.

The SCSI target port shall assert the C/D and I/O signals and negate the MSG signal during the REQ/ACK handshake of this phase.

A QAS-capable SCSI initiator port shall wait a minimum of one QAS non-data phase REQ(ACK) period to assert ACK after detecting the assertion of REQ.

A QAS-capable SCSI initiator port shall assert ACK for a minimum of one QAS non-data phase REQ(ACK) period.

3.8.2 STATUS phase exception condition handling

If the initiator detects a parity error on the status byte, the initiator shall create an attention condition. When the target switches to a MESSAGE OUT phase, the initiator should send a SCSI initiator port Detected Error message (see Section 4.3.5) to the target. This message notifies the target that the Status byte was invalid.

3.9 MESSAGE phase

3.9.1 MESSAGE phase overview

The MESSAGE phase is a term that references either a MESSAGE IN or a MESSAGE OUT phase. Multiple messages may be sent during either phase. The first byte transferred in either of these phases shall be either a single-byte message or the first byte of a multiple-byte message. Multiple-byte messages shall be wholly contained within a single MESSAGE phase.

3.9.2 MESSAGE IN phase

The MESSAGE IN phase allows the SCSI target port to request that messages be sent to the SCSI initiator port from the SCSI target port.

The SCSI target port shall assert the C/D, I/O, and MSG signals during the REQ/ACK handshakes of this phase.

A QAS-capable SCSI initiator port shall wait a minimum of one QAS non-data phase REQ(ACK) period to assert ACK after detecting the assertion of REQ.

A QAS-capable SCSI initiator port shall assert ACK for a minimum of one QAS non-data phase REQ(ACK) period.

3.9.2.1 MESSAGE IN phase exception condition handling

If the SCSI initiator port detects a parity error on any message byte it receives, the SCSI initiator port shall create an attention condition. When the SCSI target port switches to a MESSAGE OUT phase, the SCSI initiator port shall send a MESSAGE PARITY ERROR message (see Section 4.3.7) to the SCSI target port. This message notifies the SCSI target port that the MESSAGE IN byte was invalid.

3.9.3 MESSAGE OUT phase

The MESSAGE OUT phase allows the SCSI target port to request that message(s) be sent from the initiator to the target. The target invokes this phase in response to the attention condition created by the initiator (see Section 5.1).

The target shall assert the C/D and MSG signals and negate the I/O signal during the REQ/ACK handshakes of this phase. The target shall handshake bytes in this phase until the attention condition is cleared, except when rejecting a message.

A QAS-capable initiator shall wait a minimum of one QAS non-data phase REQ(ACK) period to assert ACK after detecting the assertion of REQ.

A QAS-capable initiator shall assert ACK for a minimum of one QAS non-data phase REQ(ACK) period and shall keep the message data valid until the negation of ACK.

If the target receives all of the message bytes successfully (i.e. no parity errors), it shall indicate that no retry is being attempted by changing to any information transfer phase other than the MESSAGE OUT phase and transferring at least one byte. The target may also indicate that it has successfully received the message bytes by changing to the BUS FREE phase (e.g., after receiving ABORT TASK SET or TARGET RESET messages).

3.9.3.1 MESSAGE OUT phase exception condition handling

If the target detects one or more parity errors on the message bytes received, it may request a retry of the messages by asserting the REQ signal after detecting the attention condition has been cleared and prior to changing to any other phase. The initiator, upon detecting this condition, shall resend all of the previous message bytes in the same order as previously sent during this phase. When resending more than one message byte, the initiator shall reestablish the attention condition as described in Section 5.1.

If the target does not retry the MESSAGE OUT phase or it exhausts its retry limit, it may:

a. return CHECK CONDITION status and set the sense key to Aborted Command and the additional sense code to Message Error; or
b. indicate a protocol error by performing an unexpected bus free.

The target may act on messages as received as long as no parity error is detected and may ignore all remaining messages sent under one attention condition after a parity error is detected. When a sequence of messages is resent by a SCSI initiator port because of a SCSI target port detected parity error, the target shall not act on any message that it acted on the first time received.

3.10 Signal restrictions between phases

When the SCSI bus is between two information transfer phases, the following restrictions shall apply to the SCSI bus signals:

a. The BSY, SEL, and ACK signals shall not change.
b. The REQ signal shall not change until it is asserted to qualify the start of a new phase.
c. The C/D, I/O, MSG, Data Bus, and DB(P_CRCA) signals may change.

d. When switching the Data Bus or DB(P_CRCA) signal direction from out (SCSI initiator port driving) to in (SCSI target port driving), the target shall delay driving the Data Bus, DB(P_CRCA), and/or DB(P1) by at least one data release delay plus one bus settle delay after asserting the I/O signal, and the initiator shall release the Data Bus, DB(P_CRCA), and/or DB(P1) no later than one data release delay after the transition of the I/O signal to true. When switching the Data Bus, DB(P_CRCA), and/or DB(P1) direction from in (target driving) to out (initiator driving), the target shall release the Data Bus, DB(P_CRCA), and/or DB(P1) no later than one system deskew delay after negating the I/O signal. The initiator shall assert the Data Bus, DB(P_CRCA), and/or DB(P1) no sooner than one system deskew delay after the detection of the negation of the I/O signal.

e. The DB(P_CRCA) signal direction may switch direction while the Data Bus and/or DB(P1) does not (e.g., changing from COMMAND phase to DT DATA OUT phase). When switching the DB(P_CRCA) signal direction from out (initiator driving) to in (target driving), the target shall delay driving the DB(P_CRCA) by at least one data release delay plus one bus settle delay after negating the C/D signal and the initiator shall release the DB(P_CRCA) signal no later than one data release delay after the transition of the C/D signal to false. When switching the DB(P_CRCA) signal direction from in (target driving) to out (initiator driving), the target shall release the DB(P_CRCA) signal no later than one system deskew delay after asserting the C/D signal. The initiator shall negate the DB(P_CRCA) signal no sooner than one system deskew delay after the detection of the assertion of the C/D signal.

f. The ATN and RST signals may change as defined under the descriptions for the attention condition (see Section 5.1) and hard reset (see Section 5.3).

3.11 SCSI bus phase sequences

3.11.1 SCSI bus phase sequences overview

The order in which phases are used on the SCSI bus follows a prescribed sequence.

During DT DATA phases, the target shall not change phases except at data group boundaries or SPI information unit boundaries. If a SCSI initiator port detects a phase change within a data group or information unit, it shall consider any data transferred for that data group or information unit to have been transferred incorrectly. The SCSI initiator port shall consider this condition a protocol error and respond accordingly.

A hard reset aborts any phase and is always followed by the BUS FREE phase. Also, any phase may be followed by the BUS FREE phase, but many such instances are exception conditions for SCSI initiator ports (see Section 3.1.1).

3.11.2 Phase sequences for physical reconnection and selection using attention condition with information unit transfers disabled

The allowable sequences for either physical reconnection or selection using attention condition while an information unit transfer agreement is not in effect shall be as shown in Figure 11. The normal progression for selection using attention condition (see Section 3.2.1.1) is:

1. from the BUS FREE phase to ARBITRATION;
2. from ARBITRATION to SELECTION or RESELECTION; and
3. from SELECTION or RESELECTION to one or more of the information transfer phases (i.e., COMMAND, DATA, STATUS, or MESSAGE).

The final information transfer phase is normally the MESSAGE IN phase where a DISCONNECT or TASK COMPLETE message is transferred, followed by the BUS FREE phase.

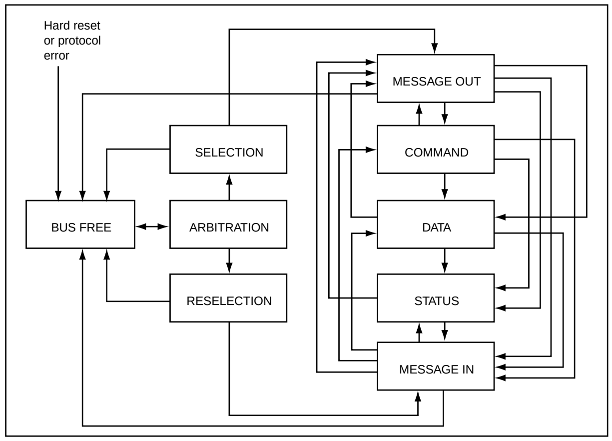


Figure 11. Phase sequences for physical reconnection and selection using attention condition with information unit transfers disabled

3.11.3 Phase sequences for selection without using attention condition with information unit transfers disabled

The allowable sequences for either physical reconnection or selection using attention condition while an information unit transfer agreement is not in effect shall be as shown in Figure 12.

The normal progression for selection without using attention condition (see Section 3.2.1.1.2) is:

1. from the BUS FREE phase to ARBITRATION;
2. from ARBITRATION to SELECTION;
3. from SELECTION to COMMAND phase;
4. from COMMAND phase to DATA phase;
5. from DATA phase to STATUS phase;
6. from STATUS phase to MESSAGE IN phase where a TASK COMPLETE message is transferred; and
7. from MESSAGE IN to BUS FREE phase.

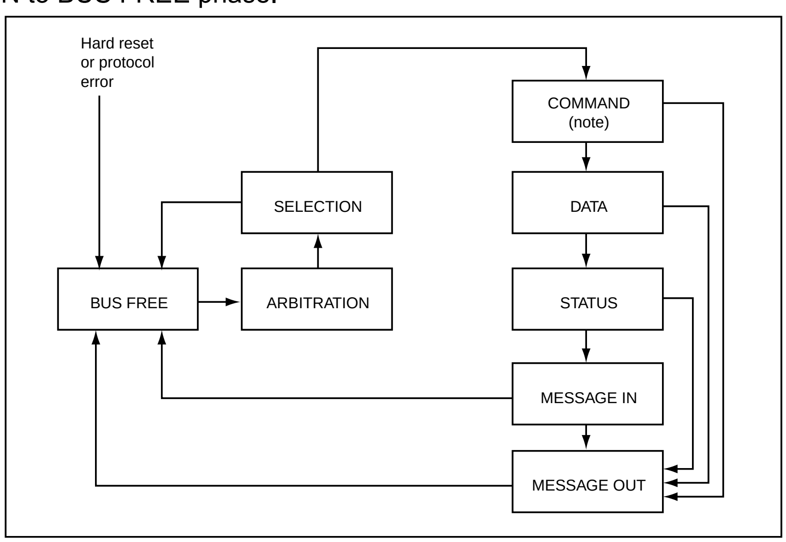


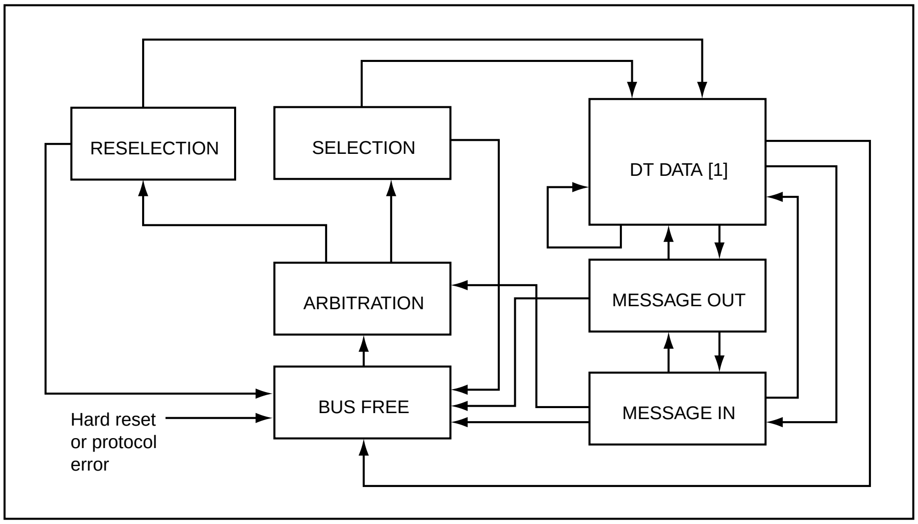
Figure 12. Phase sequences for selection without using attention condition with information unit transfers disabled

3.11.4 Phase sequences for physical reconnection or selection without using attention condition with information unit transfers enabled

The sequences for physical reconnection or selection without using attention condition while an information unit transfer agreement is in effect shall be as shown in Figure 13.

The normal progression for selection without using attention condition (see Section 3.2.1.1.3) if QAS is disabled is:

1. from the BUS FREE phase to ARBITRATION;
2. from ARBITRATION to SELECTION or RESELECTION;
3. from SELECTION or RESELECTION to one or more DT DATA phases; and
4. from the final DT DATA phase to BUS FREE.



[1] See figures 15, 16, 17, and 18 for the sequencing of SPI units within the DT DATA phases.

Figure 13. Phase sequences for physical reconnection or selection without using attention condition/ with information unit transfers enabled

3.11.5 Phase sequences for physical selection using attention condition with information unit transfers enabled

The sequences for a selection with attention condition while an information unit transfer agreement is in effect shall be as shown in Figure 14.

The normal progression for selection using attention condition (see Section 3.2.1.1.3) if QAS is disabled is:

1. from the BUS FREE phase to ARBITRATION;
2. from ARBITRATION to SELECTION;
3. from SELECTION to MESSAGE OUT;
4. from MESSAGE OUT to MESSAGE IN; and
5. from MESSAGE IN to BUS FREE.

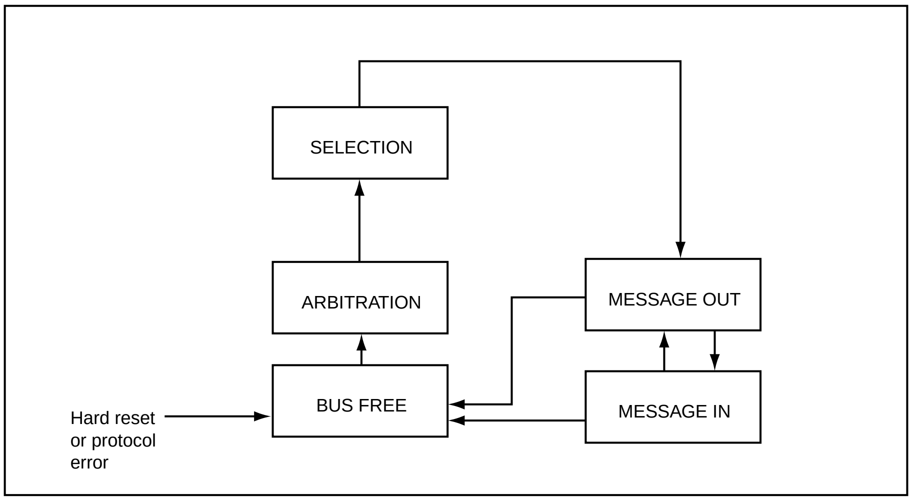


Figure 14. Phase sequences for selection with attention condition/physical reconnection and information unit transfers enabled

3.12 Data bus protection

3.12.1 Data bus protection overview

The data bus DB(P_CRCA) signal and the DB(P1) signals are used to generate parity or control the transfer of pCRC information on the Data Bus.

3.12.2 ST data bus protection using parity

For ARBITRATION phase the DB(P_CRCA) and DB(P1) signals shall not be checked for parity errors. For SELECTION and RESELECTION phases, valid parity is determined by the rules in Table 23.

Table 23: Parity checking rules for SELECTION and RESELECTION phases

Action	Condition
Check for odd parity on:	**If at least one bit is active on:**
DB(7-0,P_CRCA)	DB(15-0,P_CRCA,P1)
DB(15-8,P1)	DB(15-8,P1)

Note. These rules are necessary to permit interoperation of SCSI devices with different Data Bus widths. For example, if an 8-bit SCSI device selects a 16-bit SCSI device, the 16-bit SCSI device observes invalid parity on the upper 8 bits of the Data Bus.

For COMMAND, MESSAGE, and STATUS phases, the DB(P_CRCA) signal shall indicate odd parity for DB(7-0). The DB(P1) signal shall not be checked.

For ST DATA phases, the DB(P_CRCA) signal shall indicate odd parity for DB(7-0). If 8-bit transfers are enabled, the DB(P1) signal shall not be checked. If 16-bit data transfers are enabled, the DB(P1) signal shall indicate odd parity for DB(15-8). If 16-bit transfers are enabled and the last information byte transferred does not fall on the DB(15-8) signals, DB(P1) shall be valid for whatever data is placed on the bus.

Parity protection is not enabled during DT DATA phases.

3.12.3 DT data bus protection using CRC

3.12.3.1 DT data bus protection using CRC overview

When pCRC protection or iuCRC protection are enabled, the error detecting code is a 32-bit (four byte) Cyclic Redundancy Check (CRC), referred to as CRC-32. It is also used by several other device I/O standards. Four CRC bytes are transferred with data to increase the reliability of data transfers.

3.12.3.2 Error detection capabilities

The CRC detects all single bit errors, any two bits in error, or any combination of errors within a single 32-bit range.

3.12.3.3 Order of bytes in the CRC field

Figure 74 of ANSI SCSI Parallel Interface (SPI-5), T10/1525D, shows how transmitted data is used to calculate the CRC and how the CRC information is then transmitted.

4.0 Message system specification

SCSI protocol messages allow communication between a SCSI initiator port and a SCSI target port for the purpose of link management. The link management messages used for this purpose are defined within this standard and their use is confined to this standard. Other SCSI protocol messages allow communication between the application client and the task manager for the purpose of task management. The task management functions are defined in the SCSI Architecture Model-4 standard. Messages that convey the task management functions are defined by this manual.

4.1 General message protocols and formats

One or more messages may be sent during a single MESSAGE phase, but a message shall not be split between multiple message phases.

If an information unit agreement is not in effect, the first message sent by the SCSI initiator port after a successful SELECTION phase with an attention condition shall be an IDENTIFY, ABORT TASK SET (see Section 4.5.3), or TARGET RESET message. If a SCSI target port receives any other message, it shall cause an unexpected bus free by generating a BUS FREE phase (see Section 3.1.1).

If the first message is an IDENTIFY message, it may be followed by other messages, such as the first of a pair of SYNCHRONOUS DATA TRANSFER REQUEST messages. With tagged queuing, a task attribute shall follow the IDENTIFY message, then more messages may follow. The IDENTIFY message establishes a logical connection between the SCSI initiator port and the specified logical unit within the SCSI target device known as an I_T_L nexus.

If an information unit agreement is not in effect, after the RESELECTION phase the SCSI target port's first message shall be IDENTIFY. This allows the I_T_L nexus to be re-established. Only one logical unit shall be identified for any physical connection or physical reconnection; if a SCSI target port receives a second IDENTIFY message with a different logical unit number during a physical connection or physical reconnection, it shall cause an unexpected bus free by generating a BUS FREE phase (see Section 3.1.1).

If an information unit agreement is in effect, the SCSI target port enters a DT DATA phase after the RESELECTION phase as illustrated in Figure 13.

All SCSI initiator ports shall implement the mandatory messages tabulated in the "Initiator" column of tables 27, 40, and 45. All SCSI target ports shall implement the mandatory messages tabulated in the "Target" column of tables 27, 40, and 45.

Whenever an I_T_L nexus is established by a SCSI initiator port that is allowing physical disconnection, the initiator shall ensure that the active pointers are equal to the saved pointers for that particular logical unit. An implied restore pointers operation shall occur as a result of a RESELECTION phase or a successful receipt of a SPI L_Q information unit.

4.2 Message formats

One-byte, two-byte, and extended message formats are defined. The first byte of the message determines the format as defined in Table 24.

Table 24: Message format

Message code	Message format
00h	One-byte message (TASK COMPLETE)
01h	Extended messages
02h–0Ah	One-byte messages
0Bh	Obsolete one-byte messages
0Ch–0Eh	One-byte messages
0Fh–10h	Reserved one-byte messages
11h–13h	Obsolete one-byte messages
14h–15h	Reserved one-byte messages
16h–17h	One-byte messages
20h–24h	Two-byte messages
25h–2Fh	Reserved two-byte messages
30h–54h	Reserved
55h	One-byte message
56–7Fh	Reserved
80h–FFh	One-byte message (IDENTIFY)

4.2.1 One-byte messages

One-byte messages consist of a single byte transferred during a MESSAGE phase. The byte's message code determines the message to be performed as defined in Tables 27, 40, and 45. The IDENTIFY message is a one-byte code, but its format is different.

4.2.2 Two-byte messages

Two-byte messages consist of two consecutive bytes transferred during a MESSAGE IN phase or a MESSAGE OUT phase. The value of the first byte determines the message to be performed as defined in Tables 27, 40, and 45. The second byte is a parameter byte used as defined in the message descriptions.

4.2.3 Extended messages

A value of 01h in the first byte of a message indicates the beginning of a multiple-byte extended message. The minimum number of bytes sent for an extended message is three. All of the extended message bytes shall be transferred in consecutive MESSAGE IN phases or consecutive MESSAGE OUT phases. See sections referenced in Table 26 for details of the extended messages.

Table 25: Extended message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01h)							
1	EXTENDED MESSAGE LENGTH (n–1) [1]							
2	EXTENDED MESSAGE CODE [2]							
3-n	EXTENDED MESSAGE ARGUMENTS [3]							

1. The EXTENDED MESSAGE LENGTH field specifies the length in bytes of the Extended Message Code plus the extended message arguments to follow. The total length of the message is equal to the EXTENDED MESSAGE LENGTH plus two. A value of zero in the EXTENDED MESSAGE LENGTH field indicates 256 bytes follow.
2. The extended message codes are listed in Table 26.

Table 26: Extended Message Codes

Code	Extended message
00h	MODIFY DATA POINTERS
01h	SYNCHRONOUS DATA TRANSFER REQUEST
02h	Reserved
03h	WIDE DATA TRANSFER REQUEST
04h	PARALLEL PROTOCOL REQUEST
05h	MODIFY BIDIRECTIONAL DATA POINTER
06h - FFh	Reserved

1. The EXTENDED MESSAGE ARGUMENTS are specified within the Extended Message Descriptions in Paragraphs referenced in Table 27.

4.3 Message categories

The messages supported by the various drives are divided up into categories and listed in tables in the following subsections. The individual drive's Product Manual. Table 27 lists messages that are supported by the particular drive. In the tables, the message code values are given a direction specification (In-Out). Detailed descriptions follow the tables. Messages other than those supported by a drive are answered by the drive with a MESSAGE REJECT message.

4.3.1 LINK CONTROL MESSAGES

Table 27: Link Control message codes

Code	Support				Message name	Reference section	Direction		Clear attention condition
	IU transfers disabled		IU transfers enabled						
	Init	Targ	Init	Targ			In	Out	
04h	O	O	O	O	DISCONNECT	4.3.2	In	Out	Yes
80h+	M	O	NS	NS	IDENTIFY	4.3.3	In		N/A
80h+	M	M	NS	NS	IDENTIFY	4.3.3		Out	Not required
23h	O	O	NS	NS	IGNORE WIDE RESIDUE	4.3.4	In		N/A
05h	M	M	M	M	INITIATOR DETECTED ERROR	4.3.5		Out	Yes
0Ah	O	O	NS	NS	LINKED COMMAND COMPLETE		In		N/A
09h	M	M	M	M	MESSAGE PARITY ERROR	4.3.7		Out	Yes
07h	M	M	M	M	MESSAGE REJECT	4.3.8	In	Out	Yes
01h,05h,00h	O	O	O	O	MODIFY DATA POINTER	4.3.9	In		N/A
01h,09h,05h	O	O	O	O	MODIFY BIDIREC-TIONAL DATA POINTER	4.3.9	In		N/A
08h	M	M	M	M	NO OPERATION	4.3.11		Out	Yes
01h,06h,04h	M	M	M	M	PARALLEL PROTOCOL REQUEST	4.3.12	In	Out	Yes

Code	Support				Message name	Reference section	Direction		Clear attention condition
	IU transfers disabled		IU transfers enabled						
	Init	Targ	Init	Targ			In	Out	
55h	NS	O	O	O	QAS REQUEST	4.3.13	In		N/A
03h	O	O	O	O	RESTORE POINTERS	4.3.14	In		N/A
02h	O	O	NS	NS	SAVE DATA POINTER	4.3.15	In		N/A
01h,03h,01h	O	O	M	M	SYNCHRONOUS DATA TRANSFER REQUEST	4.3.16	In	Out	Yes
00h	M	M	NS	NS	TASK COMPLETE	4.3.17	In		N/A
01h,02h,03h	O	O	M	M	WIDE DATA TRANSFER REQUEST	4.3.18	In	Out	Yes

Key:

M = Mandatory support O = Optional support NS = Not supported

IN = SCSI target port to SCSI initiator port OUT = SCSI Initiator port to SCSI target port

Yes = SCSI Initiator port shall clear the attention condition before last ACK of the Message Out phase.

Not required = SCSI Initiator port may or may not clear the attention condition before last ACK of the MESSAGE OUT phase (see Section 5.1)

N/A = Not applicable

Init = SCSI initiator port Targ = SCSI target port

80h+ = Codes 80h through FFh are used for IDENTIFY messages

Note. The clear attention condition only applies during MESSAGE OUT phase.

4.3.2 DISCONNECT

The DISCONNECT message is sent from a SCSI target port to inform an initiator that the target plans to do a physical disconnect by releasing the BSY signal, and that a later physical reconnect is going to be required in order to complete the current task. This message shall not cause the initiator to save the data pointer.

The target shall consider the message transmission to be successful when there is no attention condition on the DISCONNECT message.

After successfully sending this message, the target shall go to the BUS FREE phase by releasing the BSY signal.

If an information unit transfer agreement is not in effect, any SCSI target port that breaks data transfers into one or more physical reconnections shall end each successful data transfer (except possibly the last) with a SAVE DATA POINTER—DISSCONNECT message sequence.

If an information unit transfer agreement is in effect, SCSI target ports shall not transmit a DISCONNECT message.

This message may also be sent from an initiator to a target to instruct the target to do a physical disconnect. If this option is enabled and a DISCONNECT message is received, the target shall either:

a. if an information unit transfer agreement is not in effect—switch to MESSAGE IN phase, send the DISCONNECT message to the initiator (possibly preceded by SAVE DATA POINTER message), and then do a physical disconnect by releasing BSY; or
b. if an information unit transfer agreement is in effect, regardless of the QAS mode—do a physical disconnect by releasing BSY.

After releasing the BSY signal, the target shall not participate in another ARBITRATION phase for at least a disconnection delay or the time limit specified in the Physical Disconnect Time Limit mode parameter (see Seagate SCSI Command Reference Manual, Part number 100293068.) whichever is greater. If this option is disabled or the target is not able to do a physical disconnect at the time when it receives the DISCONNECT message from the initiator, the target shall respond by sending a MESSAGE REJECT message to the initiator.

4.3.3 IDENTIFY

The IDENTIFY message (see Table 28) is sent by either the initiator or the target to establish an I_T_L nexus when information unit transfers are disabled.

Table 28: IDENTIFY message format

Bit / Byte	7	6	5	4	3	2	1	0
0	IDENTIFY	DISCPRIV	LUN					

The IDENTIFY bit shall be set to one to specify that this is an IDENTIFY message.

A DISCONNECT PRIVILEGE (DISCPRIV) bit of one specifies that the initiator has granted the target the privilege of doing physical disconnects. A DISCPRIV bit of zero specifies that the target shall not do physical disconnects. This bit is not defined and shall be set to zero when an IDENTIFY message is sent by a target.

The target shall generate a BUSY status (see Section 3.8) for a task not granting a physical disconnect privilege (DISCPRIV bit set to zero) in the IDENTIFY message if:

a. there are any pending tasks, and
b. the target determines that a physical reconnection of one or more pending tasks is required before the current task may be completed.

The LUN field specifies a logical unit number.

Only one logical unit number shall be identified per task. The initiator may send one or more IDENTIFY messages during a task. A second IDENTIFY message with a different value in the LUN field shall not be issued before a BUS FREE phase; if a target receives a second IDENTIFY message with a different value in this field, it shall cause an unexpected bus free (see Section 3.1.1.1) by generating a BUS FREE phase. Thus an initiator may change the DISCPRIV bit, but shall not attempt to switch to another task. (See the DTDC field of the physical disconnect/reconnect mode page in Seagate SCSI Command Reference Manual, Part number 100293068 for additional controls over physical disconnection.)

An implied RESTORE POINTERS message shall be performed by the initiator following successful identification of the nexus during the MESSAGE IN phase of a physical reconnection or a successful receipt of a SPI L_Q information unit.

Identification is considered successful during an initial connection or an initiator's physical reconnect when the target detects no error during the transfer of the IDENTIFY message and an optional task attribute message in the MESSAGE OUT phase following the SELECTION phase. See Section 4.4 for the ordering of the IDENTIFY and TASK ATTRIBUTE messages. See Section 3.9.3.1 for handling target detected errors during the MESSAGE OUT phase.

Identification is considered successful during a target's physical reconnect when there is no attention condition on either the IDENTIFY message or the simple message for an I_T_L_Q nexus in the MESSAGE IN phase following the RESELECTION phase. See Section 4.5 for the ordering of the IDENTIFY and TASK ATTRIBUTE messages. See Section 3.9.2.1, for handling target detected errors during the MESSAGE IN phase.

4.3.4 IGNORE WIDE RESIDUE

The IGNORE WIDE RESIDUE message (see Table 29) shall be sent from a target to indicate that the number of valid bytes sent in the last REQ/ACK handshake data of a DATA IN phase is less than the negotiated transfer width. When information unit transfers are disabled, the IGNORE WIDE RESIDUE message shall be sent following that DATA IN phase and prior to any other messages.

If the residual byte contains valid data, then the IGNORE WIDE RESIDUE message should not be sent.

Table 29: IGNORE WIDE RESIDUE message format

Bit Byte	7	6	5	4	3	2	1	0
0	MESSAGE CODE (23H)							
1	NUMBER OF BYTES TO IGNORE (01H)							

The NUMBER OF BYTES TO IGNORE field indicates the number of invalid data bytes transferred. See Table 30.

Note. More than one IGNORE WIDE RESIDUE message may occur during a task.

Table 30: Ignore field definition

Codes	Invalid data bits Wide transfers
00h	Reserved
01h	DB(15-8)
02h	Obsolete
03h	Obsolete
04h-FFh	Reserved

4.3.5 INITIATOR DETECTED ERROR

The INITIATOR DETECTED ERROR message is sent from an initiator to inform a target that an error has occurred that does not preclude the target from retrying the task. The source of the error may either be related to previous activities on the SCSI bus or may be internal to the initiator and unrelated to any previous SCSI bus activity. Although the integrity of the currently active pointers is not assured, a RESTORE POINTERS message or a physical disconnect followed by a reconnect shall cause the pointers to be restored to their defined prior state.

4.3.6 LINKED COMMAND COMPLETE

The LINKED COMMAND COMPLETE message is sent from a target to an initiator to indicate that a linkied command has completed and that status has been sent. The initiator shall then set the pointers to the initial state for the next linked command.

4.3.7 MESSAGE PARITY ERROR

The MESSAGE PARITY ERROR message is sent from the initiator to the target to indicate that it received a message byte with a parity error (see Section 3.9.3.1).

To indicate its intentions of sending this message, the initiator shall create an attention condition on the message byte that has the parity error. This provides an interlock so that the target is able to determine which message byte has the parity error. If the target receives this message under any other circumstance, it shall signal a catastrophic error condition by going to a BUS FREE phase without any further information transfer attempt (see Section 3.1.1).

If the target attempts a retry after receiving the MESSAGE PARITY ERROR message, the target shall return to the MESSAGE IN phase before switching to some other phase, the target shall resend the entire message that had the parity error.

4.3.8 MESSAGE REJECT

The MESSAGE REJECT message is sent from either the initiator or target to indicate that the last message or message byte it received was inappropriate or has not been implemented.

To indicate its intentions of sending this message, the initiator shall create an attention condition on the message byte that is to be rejected. If the target receives this message under any other circumstance, it shall reject this message.

When a target sends this message, it shall change to MESSAGE IN phase and send this message prior to requesting additional message bytes from the initiator. This provides an interlock so that the initiator is able to determine which message byte is rejected.

After a target sends a MESSAGE REJECT message and if the attention condition is still set, then it shall return to the MESSAGE OUT phase. The subsequent MESSAGE OUT phase shall begin with the first byte of a message.

4.3.9 MODIFY DATA POINTER

The MODIFY DATA POINTER message (see Table 31) is sent from the target to the initiator and requests that the signed argument be added to the value of the current data pointer using two's complement arithmetic. The data pointer is the data-out or data-in pointer being used by the command. The ENABLE MODIFY DATA POINTER (EMDP) bit in the Disconnect-reconnect mode page (see Seagate SCSI Command Reference Manual, Part number 100293068) indicates whether or not the target is permitted to issue the MODIFY DATA POINTER message. The target shall only issue the MODIFY DATA POINTER message during a unidirectional command.

It is recommended that the target not attempt to move the data pointer outside the range addressed by the command. Initiators may or may not place further restrictions on the acceptable values. Should the target send an Argument value that is not supported by the initiator, the initiator may reject the value by responding with the MESSAGE REJECT message. In this case, the data pointer is not changed from its value prior to the rejected MODIFY DATA POINTER message.

If an information unit transfer agreement is in effect the target shall not transmit a MODIFY DATA POINTER message.

Table 31: MODIFY DATA POINTER message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01H)							
1	EXTENDED MESSAGE LENGTH (05H)							
2	MODIFY DATA POINTER (00H)							
3	(MSB)							
4 5	ARGUMENT							
6								(LSB)

4.3.10 MODIFY BIDIRECTIONAL DATA POINTER

The MODIFY BIDIRECTIONAL DATA POINTER message (see Table 32) is sent from the target to the initiator and requests that the signed data-out argument be added using two's complement arithmetic to the value of the current data-out pointer and signed data-in argument be added using two's complement arithmetic to the value of the current data-in pointer. The ENABLE MODIFY DATA POINTER (EMDP) bit in the Disconnect-reconnect mode page indicates whether the target is permitted to issue the MODIFY BIDIRECTIONAL DATA POINTER message. The target shall only issue the MODIFY BIDIRECTIONAL DATA POINTER message during a bidirectional command.

It is recommended that the target not attempt to move the data-out pointer or the data-in pointer outside the range addressed by the command. Initiators may or may not place further restrictions on the acceptable values. Should the target send a data-out argument or a data-in argument value that is not supported by the initiator, the initiator may reject the value by responding with the MESSAGE REJECT message. In this case, both the data-out pointer and the data-in pointer are not changed from their values prior to the rejected MODIFY BIDIRECTIONAL DATA POINTER message.

If an information unit transfer agreement is in effect the target shall not transmit a MODIFY DATA POINTER message.

Table 32: MODIFY BIDIRECTIONAL DATA POINTER message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01H)							
1	EXTENDED MESSAGE LENGTH (09H)							
2	MODIFY BIDIRECTIONAL DATA POINTER (05H)							
3	(MSB)							
4	DATA--OUT ARGUMENT							
5								
6								(LSB)
7	(MSB)							
8	DATA--IN ARGUMENT							
9								
10								(LSB)

4.3.11 NO OPERATION

The NO OPERATION message is sent from an initiator in response to a target's request for a message when the initiator does not currently have any other valid message to send.

For example, if the target does not respond to the attention condition until a later phase and at that time the original message is no longer valid the initiator may send the NO OPERATION message when the target switches to a MESSAGE OUT phase.

4.3.12 PARALLEL PROTOCOL REQUEST

PARALLEL PROTOCOL REQUEST messages (see Table 33) are used to negotiate a synchronous data transfer agreement, a wide data transfer agreement, and set the protocol options between two SCSI devices.

Table 33: PARALLEL PROTOCOL REQUEST message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01H)							
1	EXTENDED MESSAGE LENGTH (06H)							
2	PARALLEL PROTOCOL REQUEST (04H)							
3	TRANSFER PERIOD FACTOR							
4	RESERVED							
5	REQ/ACK OFFEST							
6	TRANSFER WIDTH EXPONENT (M)							
7[1]	PCOMP	RTI	RD_STRM	WR_FLOW	HOLD_MCS	QAS_REQ	DT_REQ	IU_REQ

[1] For details on protocol option bits See Table 18, Section 2.12.8.

TRANSFER PERIOD FACTOR

00h-007h Reserved. Faster timings may be allowed by future SCSI parallel interface standards.

08h Transfer period equals 6.25 ns. Fast-160 data is latched every 6.25 ns.

09h Transfer period equals 12.5 ns. Fast-80 data is latched every 12.5 ns.

0Ah Transfer period equals 25 ns. Fast-40 data is latched every 25 ns or 30.3 ns.

0Bh Transfer period equals 30.3 ns. Fast-40 data is latched every 25 ns or 30.3 ns.

0Ch Transfer period equals 50 ns. Fast-20 data is latched using a transfer period of less than or equal to 96 ns and greater than or equal to 50 ns.

ODh - 18h Transfer period equals the period factor x 4. Fast-20 data is latched using a transfer period of less than or equal to 96 ns and greater than orequal to 50 ns.

19h - 31h Transfer period equals the period factor x 4. Fast-10 data is latched using a transfer period of less than or equal to 196 ns and greater than or equal to 100 ns.

32h - FFh Transfer period equals the period factor x 4. Fast-5 data is latched using a transfer period of less than or equal 1,020 ns and greater than or equal to 200 ns.

REQ/ACK OFFSET

This field determines the maximum number of REQs allowed to be outstanding before a correesponding ACK is received at the target during synchronous or paced transfers.

For ST synchronous transfers, the REQ/ACK OFFSET is the number of REQ transitions that may be sent by the target in advance of the number of ACK transitions received from the initiator.

For paced transfers in DT DATA IN phase, the REQ/ACK OFFSET is the number of data valid state REQ assertions that may be sent by the target in advance of ACK assertions received from the initiator.

For paced transfers in DT DATA OUT phase, the REQ/ACK OFFSET is the number of REQ assertions that may be sent by the target in advance of the number of data valid state ACK assertions received from the initiator.

See Section 3.5 for an explanation of the differences between ST and DT data transfers.

The REQ/ACK OFFSET value is chosen to prevent overflow conditions in the port's receive buffer and offset counter. The REQ/ACK OFFSET values and which timing values shall be selected are defined below.

Value	Description	Timing values
00h	Specifies asynchronouse transfer agreement. The Transfer period factor and protocol options other than QAS_REQ shall be ignored.	Asynch
01h-FEh	Synchronous or paced transfers with specified offset.	Determined by transfer period factor.
FFh	Synchronous or paced transfers with unlimited offset.	Determined by transfer period factor.

TRANSFER WIDTH EXPONENT

This field defines the transfer width to be used during DATA IN phases, and DATA OUT phases. The transfer width that is established applies to both SCSI devices.

Valid transfer widths are:

a. 8 bits (m=00h)
b. and 16 bits (m=01h) if all the protocol options bits are zero.
c. The only valid transfer width is 16 bits (m=01h) if any of the protocol options bits are one.

TRANSFER PERIOD EXPONENT field values greater than 01h are reserved.

If any of the protocol options bits (other than QAS_REQ) are set to one, then only wide transfer agreements are valid. If all the protocol options bits other than QAS_REQ are set to zero, wide transfer agreements and narrow transfer agreements are valid.

The protocol options bits (IU_REQ, DT_REQ, and QAS_REQ) are used by the initiator to indicate the protocol options to be enabled. The target uses the protocol options bits to indicate to the initiator if the requested protocol options are enabled. The target shall not enable any options that were not enabled in the PPR message received from the initiator.

An information units enable request bit (IU_REQ) of zero indicates that information unit transfers shall not be used (i.e., data group transfers shall be enabled) when received from the initiator and that information unit transfers are disabled when received from the target. An IU_REQ bit of one indicates that information unit transfers shall be used when received from the initiator and that information unit transfers are enabled when received from the target. If the IU_REQ bit is changed from the previous agreement (i.e., zero to one or one to zero) as a result of a negotiation the target shall go to a BUS FREE phase on completion of the negotiation.

A DT enable request bit (DT_REQ) of zero indicates that DT DATA phases are to be disabled when received from the initiator and that DT DATA phases are disabled when received from the target. A DT_REQ bit of one indicates that DT DATA phases are to be enabled when received from the initiator and that DT DATA phases are enabled when received from the target.

A QAS enable request bit (QAS_REQ) of zero indicates that QAS is to be disabled when received from the initiator and that QAS is disabled when received from the target. A QAS_REQ bit of one indicates that QAS is to be enabled when received from the initiator and that QAS is enabled when received from the target.

Not all combinations of the protocol options bits are valid. Only the bit combinations defined in Table 34 shall be allowed. All other combinations are reserved.

Table 34: Valid protocol options bit combinations

QAS_REQ	DT_REQ	IU_REQ	Description
0	0	0	Use ST DATA IN and ST DATA OUT phases to transfer data
0	1	0	Use DT DATA IN and DT DATA OUT phases with data group transfers
0	1	1	Use DT DATA IN and DT DATA OUT phases with information unit transfers
1	1	1	Use DT DATA IN and DT DATA OUT phases with information unit transfers and use QAS for arbitration

A Parallel Protocol Request agreement applies to all logical units of the two SCSI devices that negotiated agreement. That is, if SCSI device A, acting as an initiator, negotiates a data transfer agreement with SCSI device B (a target), then the same data transfer agreement applies to SCSI devices A and B even if SCSI device B changes to an initiator.

A data transfer agreement only applies to the two SCSI devices that negotiate the agreement. Separate data transfer agreements are negotiated for each pair of SCSI devices. The data transfer agreement only applies to DATA phases and information unit transfers.

A PARALLEL PROTOCOL REQUEST message exchange shall be initiated by an initiator whenever a previously arranged parallel protocol agreement may have become invalid. The agreement becomes invalid after any condition that may leave the parallel protocol agreement in an indeterminate state such as:

a. after a hard reset;
b. after a TARGET RESET message;
c. after a power cycle; or
d. after a change in the transceiver mode (e.g., LVD mode to SE mode).

If a target determines that the agreement is in an indeterminate state, it shall initiate a negotiation to enter an asynchronous eight-bit wide data transfer mode with all the protocol options bits set to zero, using a WDTR message with the transfer bit exponent set to 00h.

Any condition that leaves the data transfer agreement in an indeterminate state shall cause the SCSI device to enter an asynchronous, eight-bit wide data transfer mode with all the protocol options bits set to set to zero.

An initiator may initiate a PARALLEL PROTOCOL REQUEST message exchange whenever it is appropriate to negotiate a data transfer agreement. SCSI devices that are currently capable of supporting any of the Parallel Protocol Request options shall not respond to a PARALLEL PROTOCOL REQUEST message with a MESSAGE REJECT message.

Renegotiation after every selection is not recommended, since a significant performance impact is likely.

The Parallel Protocol Request message exchange establishes an agreement between the two SCSI devices;

a. on the permissible periods and the REQ/ACK offsets for all logical units on the two SCSI devices. This agreement only applies to ST DATA IN phases, ST DATA OUT phases, DT DATA IN phases, and DT DATA OUT phases. All other phases shall use asynchronous transfers;
b. on the width of the data path to be used for Data phase transfers between two SCSI devices. This agreement only applies to ST DATA IN phases, ST DATA OUT phases, DT DATA IN phases, and DT DATA OUT phases. All other information transfer phases shall use an eight-bit data path; and
c. on the protocol option is to be used.

The initiator sets its values according to the rules above to permit it to receive data successfully. If the target is able to receive data successfully with these values (or smaller periods or larger REQ/ACK offsets or both), it returns the same values in its PARALLEL PROTOCOL REQUEST message. If it requires a larger period, a smaller REQ/ACK offset, or a smaller transfer width in order to receive data successfully, it substitutes values in its PARALLEL PROTOCOL REQUEST message as required, returning unchanged any value not required to be changed. Each SCSI device when transmitting data shall respect the negotiated limits set by the other's PARALLEL PROTOCOL REQUEST message, but it is permitted to transfer data with larger periods, smaller synchronous REQ/ACK offsets, or both. The completion of an exchange of PARALLEL PROTOCOL REQUEST messages implies an agreement as shown in Table 35.

If the target does not support the selected protocol option it shall clear as many bits as required to set the protocol option field to a legal value that it does support.

Table 35: PARALLEL PROTOCOL REQUEST messages implied agreement

Target's Parallel Protocol Request response	Implied agreement
Non-zero REQ/ACK OFFSET	Synchronous transfer (i.e., each SCSI device transmits data with a period equal to or greater than and a REQ/ACK OFFSET equal to or less than the negotiated values received in the target's PPR message).
REQ/ACK OFFSET equal to zero	Asynchronous transfer
Transfer width exponent equal to one	16-bit data
Transfer width equal to zero	Eight-bit data
Protocol options equal to 0h and transfer period factor equal to 9h	Eight-bit/asynchronous data transfer with protocol options field set to 0h
IU_REQ, DT_REQ, and QAS_REQ equal to zero	ST DATA IN and ST DATA OUT phases to transfer data
DT_REQ equal to one	DT DATA IN and DT DATA OUT phases to transfer data
IU_REQ and DT_REQ equal to one	DT DATA IN and DT DATA OUT phases with information units
IU_REQ, DT_REQ, and QAS_REQ equal to one	DT DATA IN and DT DATA OUT phases with information units and use QAS for arbitration
MESSAGE REJECT message	The initiator shall set eight-bit/asynchronous data transfer with protocol options field set to 0h.
Parity error (on responding message)	Eight-bit/asynchronous data transfer with protocol options field set to 0h
Unexpected bus free (as a result of the responding message)	Eight-bit/asynchronous data transfer with protocol options field set to 0h
No response	Eight-bit/asynchronous data transfer with protocol options field set to 0h

If there is an unrecoverable parity error on the initial PARALLEL PROTOCOL REQUEST message (see Section 4.3.12), the initiator shall retain its previous data transfer mode and protocol options. If there is an unexpected bus free on the initial PARALLEL PROTOCOL REQUEST message, the initiator shall retain its previous data transfer mode and protocol options.

4.3.12.1 PARALLEL PROTOCOL REQUEST

If the initiator recognizes that Parallel Protocol Request negotiation is required, it creates an attention condition and sends a PARALLEL PROTOCOL REQUEST message to begin the negotiating process. After successfully completing the MESSAGE OUT phase, the target shall respond with a PARALLEL PROTOCOL REQUEST message or a MESSAGE REJECT message.

If an abnormal condition prevents the target from responding with a PARALLEL PROTOCOL REQUEST message or with a MESSAGE REJECT message, then both SCSI devices shall use the eight-bit/asynchronous data transfer mode with all the protocol options bits set zero to indicate ST DATA IN and ST DATA OUT phases between the two SCSI devices.

Following a target's responding PARALLEL PROTOCOL REQUEST message, an implied agreement for data transfers shall not be considered to exist until:

a. the initiator receives the last byte of the PARALLEL PROTOCOL REQUEST message and parity is valid; and
b. the target does not detect an attention condition on the last byte of the PARALLEL PROTOCOL REQUEST message.

If the initiator does not support the target’s responding PARALLEL PROTOCOL REQUEST message’s values, the initiator shall create an attention condition and the first message shall be a MESSAGE REJECT message.

If during the PARALLEL PROTOCOL REQUEST message the initiator creates an attention condition and the first message of the MESSAGE OUT phase is either a MESSAGE PARITY ERROR or MESSAGE REJECT message, the data transfers shall be considered to be negated by both SCSI devices. In this case, both SCSI devices shall use the eight-bit asynchronous data transfer mode with all the protocol options bits set zero to indicate ST DATA IN and ST DATA OUT phases for data transfers between the two SCSI devices.

4.3.13 QAS REQUEST

The QAS REQUEST message is sent from a target that has both information unit transfers and QAS enabled to begin a QAS phase (see Section 3.1.2.3) after a DT DATA phase.

4.3.14 RESTORE POINTERS

The RESTORE POINTERS message is sent from a target to direct the initiator to copy the most recently saved command, data, and status pointers for the task to the corresponding active pointers. The command and status pointers shall be restored to the beginning of the present command and status areas. The data pointer shall be restored to the value at the beginning of the data area in the absence of a SAVE DATA POINTERS message or to the value at the point at which the last SAVE DATA POINTERS message occurred for that task.

When information unit transfers are enabled, there are implied restore pointers, and the target shall not transmit a RESTORE POINTERS message. For more information on this see Section 6 and particularly Section 6.2.3.

4.3.15 SAVE DATA POINTERS

The SAVE DATA POINTERS message is sent from a target to direct the initiator to copy the current data pointer to the saved data pointer for the current task.

4.3.16 SYNCHRONOUS DATA TRANSFER REQUEST

Synchronous Data Transfer Request (SDTR) messages (see Table 36) are used to negotiate a synchronous data transfer agreement between two SCSI devices.

Table 36: Synchronous Data Transfer Request message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01h)							
1	EXTENDED MESSAGE LENGTH (03h)							
2	SYNCHRONOUS DATA TRANSFER REQUEST (01h)							
3	TRANSFER PERIOD FACTOR [1]							
4	REQ/ACK OFFSET							

The TRANSFER PERIOD FACTOR field values are defined below.

Code	Description
00h-07h	Reserved. Faster timings may be allowed by future SCSI parallel interface standards
08h	Transfer period equals 6.25 ns. Fast-160. DT transfers are required for this transfer period factor.
09h	Transfer period equals 12.5 ns. Fast-80. DT transfers are required for this transfer period factor.
0Ah	Transfer period equals 25 ns. Fast-40 data is latched every 25 ns or 30.3 ns.
0Bh	Transfer period equals 30.3 ns. Fast-40 data is latched every 25 ns or 30.3 ns.
0Ch	Transfer period equals 50 ns. Fast-20 data is latched using a transfer period of less than or equal to 96 ns and greater than or equal to 50 ns.
0D-18h	Transfer period equals the period factor x 4. Fast-20 data is latched using a transfer period of less than or equal to 96 ns and greater than or equal to 50 ns.
19h-31h	Transfer period equals the period factor x 4. Fast-10 data is latched using a transfer period of less than or equal to 196 ns and greater than or equal to 100 ns.
32h-FFh	Transfer period equals the period factor x 4. Fast-5 data is latched using a transfer period of less than or equal to 1,020 ns and greater than or equal to 200 ns.

The REQ/ACK OFFSET field contains the maximum number of REQ assertions allowed to be outstanding before a corresponding ACK assertion is received at the target. The size of a data transfer may be 1 or 2 bytes depending on what values, if any, have been previously negotiated through an exchange of WIDE DATA TRANSFER REQUEST messages or PPR messages. The REQ/ACK OFFSET value is chosen to prevent overflow conditions in the SCSI device's reception buffer and offset counter. A REQ/ACK OFFSET value of zero shall indicate asynchronous data transfer mode and that the TRANSFER PERIOD FACTOR field shall be ignored; a value of FFh shall indicate unlimited REQ/ACK OFFSET.

An SDTR agreement applies to all logical units of the two SCSI devices that negotiated agreement. That is, if SCSI device A, acting as an initiator, negotiates a synchronous data transfer agreement with SCSI device B (a target), then the same data transfer agreement applies to SCSI devices A and B even if SCSI device B changes to an initiator.

An SDTR agreement only applies to the two SCSI devices that negotiate the agreement. Separate synchronous data transfer agreements are negotiated for each pair of SCSI devices. The synchronous data transfer agreement only applies to DATA phases.

An SDTR message exchange shall be initiated by a SCSI device whenever a previously arranged synchronous data transfer agreement may have become invalid. The agreement becomes invalid after any condition that may leave the data transfer agreement in an indeterminate state such as:

a. after a HARD RESET;
b. after a TARGET RESET message;
c. after a power cycle; and
d. after a change in the transceiver mode (e.g., LVD mode to MSE mode).

Any condition that leaves the data transfer agreement in an indeterminate state shall cause the SCSI device to enter an asynchronous data transfer mode.

A SCSI device may initiate an SDTR message exchange whenever it is appropriate to negotiate a new data transfer agreement (either synchronous or asynchronous). SCSI devices that are capable of synchronous data transfers shall not respond to an SDTR message with a MESSAGE REJECT message.

Renegotiation after every selection is not recommended since a significant performance impact is likely.

The SDTR message exchange establishes the permissible transfer periods and the REQ/ACK offsets for all logical units on the two SCSI devices. This agreement only applies to ST DATA IN phases and ST DATA OUT phases. COMMAND, MESSAGE, and STATUS phases shall use asynchronous transfers.

The originating SCSI device (the SCSI device that sends the first of the pair of SDTR messages) sets its values according to the rules above to permit it to receive data successfully. If the responding SCSI device is able to also receive data successfully with these values (or smaller transfer periods or larger REQ/ACK offsets or both), it returns the same values in its SDTR message. If it requires a larger transfer period, a smaller REQ/ACK offset, or both in order to receive data successfully, it substitutes values in its SDTR message as required, returning unchanged any value not required to be changed. Each SCSI device, when transmitting data, shall respect the negotiated limits set by the other's SDTR message, but it is permitted to transfer data with larger transfer periods, smaller synchronous REQ/ACK offsets, or both. The completion of an exchange of SDTR messages implies an agreement as shown in Table 37.

Table 37: SDTR messages implied agreements

Responding SCSI device Parallel Protocol Request response	Implied agreement
Non-zero REQ/ACK offset	Synchronous transfer (i.e., each SCSI device transmits data with a transfer period equal to or greater than and a REQ/ACK offset equal to or less than the values received in the other device's SDTR message) with ST DATA IN and ST DATA OUT phases. This is equivalent to protocol options of 0h in the PARALLEL PROTOCOL REQUEST message.
REQ/ACK offset equal to zero	Asynchronous transfer.
MESSAGE REJECT message	The originating SCSI device shall set synchronous transfer.
Parity error (on responding message)	Asynchronous transfer
Unexpected bus free (as a result of the responding message)	Asynchronous transfer
No response	Asynchronous transfer

If there is an unrecoverable parity error on the initial SDTR message (see sections 3.12.2 and 3.12.3), the initiating SCSI device shall retain its previous data transfer mode. If there is an unexpected bus free on the initial SDTR message, the initiating SCSI device shall retain its previous data transfer mode.

4.3.16.1 Target initiated SDTR negotiation

If the target recognizes that SDTR negotiation is required, it sends an SDTR message to the initiator. The initiator shall create an attention condition on the last byte of the SDTR message from the target, and the initiator shall respond with its SDTR message, MESSAGE PARITY ERROR message, or with a MESSAGE REJECT message.

If an abnormal condition prevents the initiator from responding with a SDTR message or with a MESSAGE REJECT message, then both SCSI devices shall return to asynchronous data transfer mode for data transfers between the two SCSI devices.

Following an initiator's responding SDTR message, an implied agreement for synchronous operation shall not be considered to exist until the target leaves MESSAGE OUT phase, indicating that the target has accepted the SDTR negotiation.

If the target does not support any of the initiator’s responding SDTR message’s values, the target shall switch to a Message In phase and the first message shall be a MESSAGE REJECT message. In this case the implied agreement shall be considered to be negated and both SCSI devices shall use the asynchronous data transfer mode for data transfers between the two SCSI devices.

If a parity error occurs, the implied agreement shall be reinstated if a retransmission of a subsequent pair of messages is successfully accomplished. After a vendor-specific number of retry attempts (greater than zero), if the target continues to receive parity errors, it shall terminate the retry activity. This is done by the target causing an unexpected bus free. The initiator shall accept such action as aborting the SDTR negotiation and both SCSI devices shall go to asynchronous data transfer mode for data transfers between the two SCSI devices.

4.3.16.2 Initiator initiated SDTR negotiation

If the initiator recognizes that SDTR negotiation is required, it creates an attention condition and sends a SDTR message to begin the negotiating process. After successfully completing the MESSAGE OUT phase, the target shall respond with the SDTR message or a MESSAGE REJECT message.

If an abnormal condition prevents the target from responding with a SDTR message or with a MESSAGE REJECT message, then both SCSI devices shall go to asynchronous data transfer mode for data transfers between the two SCSI devices.

Following a target's responding SDTR message, an implied agreement for synchronous data transfers shall not be considered to exist until:

a. the initiator receives the last byte of the SDTR message and parity is valid; and
b. the target does not detect an attention condition on the last byte of the SDTR message. If the initiator does not support the target’s responding SDTR message’s values, the initiator shall create an attention condition and the first message shall be a MESSAGE REJECT message.

If during the SDTR message the initiator creates an attention condition and the first message out is either Message Parity Error or Message Reject, the synchronous operation shall be considered to be negated by both the initiator and the target. In this case, both SCSI devices shall go to asynchronous data transfer mode for data transfers between the two SCSI devices.

4.3.17 TASK COMPLETE

The TASK COMPLETE message is sent from a target to an initiator to indicate that a task has completed and that valid status has been sent to the initiator when information unit transfers are disabled.

After successfully sending this message, the target shall go to the Bus Free phase by releasing the BSY signal. The target shall consider the message transmission to be successful when there is no attention condition on the TASK COMPLETE message.

The task may have completed successfully or unsuccessfully as indicated in the status.

4.3.18 WIDE DATA TRANSFER REQUEST

WIDE DATA TRANSFER REQUEST (WDTR) messages (see Table 38) are used to negotiate a wide data transfer agreement between two SCSI devices.

Table 38: WIDE DATA TRANSFER REQUEST message format

Bit / Byte	7	6	5	4	3	2	1	0
0	EXTENDED MESSAGE (01h)							
1	EXTENDED MESSAGE LENGTH (02h)							
2	WIDE DATA TRANSFER REQUEST (03h)							
3	TRANSFER WIDTH EXPONENT							

The TRANSFER WIDTH EXPONENT field defines the transfer width to be used during ST DATA IN phases and ST Data Out phases. The transfer width that is established applies to both SCSI devices. Valid transfer widths are 8 bits (m=00h) and 16 bits (m=01h). A Transfer Width Exponent Field value of 02h is obsolete and values greater than 02h are reserved.

A WDTR agreement applies to all logical units of the two SCSI devices that negotiated agreement. That is, if SCSI device A, acting as an initiator, negotiates a wide data transfer agreement with SCSI device B (a target), then the same transfer width agreement applies to SCSI devices A and B even if SCSI device B changes to an initiator.

A WDTR only applies to the two SCSI devices that negotiate the agreement. Separate wide transfer agreements are negotiated for each pair of SCSI devices. The wide data transfer agreement only applies to Data phases.

A WDTR message exchange shall be initiated by a SCSI device whenever a previously arranged wide transfer agreement may have become invalid. The agreement becomes invalid after any condition that may leave the wide transfer agreement in an indeterminate state such as:

a. after a hard reset;
b. after a TARGET RESET message;
c. after a power cycle; and
d. after a change in the transceiver mode (e.g., LVD mode to MSE mode).

Any condition that leaves the data transfer agreement in an indeterminate state shall cause the SCSI device to enter an eight-bit wide data transfer mode.

A SCSI device may initiate a WDTR message exchange whenever it is appropriate to negotiate a new wide transfer agreement. SCSI devices that are capable of wide data transfers (greater than 8 bits) shall not respond to a WDTR message with a MESSAGE REJECT message.

Renegotiation after every selection is not recommended, since a significant performance impact is likely.

The WDTR message exchange establishes an agreement between the two SCSI devices on the width of the data path to be used for Data phase transfers between two SCSI devices. This agreement only applies to ST DATA IN phases and ST DATA OUT phases. All other information transfer phases, except DT Data phases, shall use an eight-bit data path.

If a SCSI device implements both wide data transfer option and synchronous data transfer option and uses the SDTR and WDTR messages, then it shall negotiate the wide data transfer agreement prior to negotiating the synchronous data transfer agreement. If a synchronous data transfer agreement is in effect, then:

a. If a WDTR message is rejected with a MESSAGE REJECT message, the prior synchronous data transfer agreement shall remain intact;

b. If a WDTR message fails for any other reason, the prior synchronous data transfer agreement shall remain intact; or

c. If a WDTR message is not rejected with a MESSAGE REJECT message, a WDTR message shall reset the synchronous data transfer agreement to asynchronous mode and any protocol option bits (see Section 4.3.16) shall be set to zero.

The originating SCSI device (the SCSI device that sends the first of the pair of WDTR messages) sets its transfer width value to the maximum data path width it elects to accommodate. If the responding SCSI device is able to also accommodate this transfer width, it returns the same value in its WDTR message. If it requires a smaller transfer width, it substitutes the smaller value in its WDTR message. The successful completion of an exchange of WDTR messages implies an agreement as shown in Table 39.

Table 39: WDTR message implied agreement

Responding SCSI device WDTR response	Implied agreement
Transfer Width Exponent equal to one	16-bit data
Transfer Width equal to zero	Eight-bit data transfer
MESSAGE REJECT message	The originating SCSI device shall set eight-bit data transfer
Parity error (on responding message)	Eight-bit data transfer
Unexpected bus free (as a result of the responding message)	Eight-bit data transfer
No response	Eight-bit data transfer

If there is an unrecoverable parity error on the initial WDTR message (see sections 3.9.2 and 3.9.3) the initiating SCSI device shall retain its previous data transfer mode. If there is an unexpected bus free on the initial WDTR message the initiating SCSI device shall retain its previous data transfer mode.

4.3.18.1 Target initiated WDTR negotiation

If the target recognizes that WDTR negotiation is required, it sends a WDTR message to the initiator.

The initiator shall create an attention condition on the last byte of the WDTR message from the target, and the initiator shall respond with its WDTR message, MESSAGE PARITY ERROR message, or with a MESSAGE REJECT message.

If an abnormal condition prevents the initiator from responding with a WDTR message or with a MESSAGE REJECT message, then both SCSI devices shall go to eight-bit data transfer mode for data transfers between the two SCSI devices.

Following an initiator's responding WDTR message, an implied agreement for wide data transfers operation shall not be considered to exist until the target leaves the MESSAGE OUT phase, indicating that the target has accepted the negotiation.

If the target does not support the initiator’s responding Transfer Width Exponent, the target shall switch to a Message In phase and the first message shall be a MESSAGE REJECT message. In this case, the implied agreement shall be considered to be negated and both SCSI devices shall use the eight-bit data transfer mode for data transfers between the two SCSI devices. Any prior synchronous data transfer agreement shall remain intact.

If a parity error occurs, the implied agreement shall be reinstated if a retransmission of a subsequent pair of messages is successfully accomplished. After a device-dependent number of retry attempts (greater than zero), if the target continues to receive parity errors, it shall terminate the retry activity. This is done by the target causing an unexpected bus free. The initiator shall accept such action as aborting the WDTR negotiation, and both SCSI devices shall go to eight-bit data transfer mode for data transfers between the two SCSI devices. Any prior synchronous data transfer agreement shall remain intact.

4.3.18.2 Initiator initiated Wide Data Transfer Request (WDTR) negotiation

If the initiator recognizes that WDTR negotiation is required, it creates an attention condition and sends a WDTR message to begin the negotiating process. After successfully completing the MESSAGE OUT phase, the target shall respond with a WDTR message or a MESSAGE REJECT message.

If an abnormal condition prevents the target from responding with a WDTR message or with a MESSAGE REJECT message, then both SCSI devices shall go to eight-bit transfer mode for data transfers between the two SCSI devices.

Following a target's responding WDTR message, an implied agreement for wide data transfers shall not be considered to exist until:

a. the initiator receives the last byte of the WDTR message and parity is valid; and
b. the target does not detect an attention condition before the ACK signal is released on the last byte of the WDTR message.

If the initiator does not support the target’s responding Transfer Width Exponent, the initiator shall create an attention condition and the first message shall be a MESSAGE REJECT message. If during the WDTR message the initiator creates an attention condition and the first message of the MESSAGE OUT phase is either a Message Parity Error or MESSAGE REJECT message, the wide data transfers shall be considered to be negated by both SCSI devices. In this case, both SCSI devices shall use the eight-bit data transfer mode for data transfers between the two devices.

4.4 Task attribute messages

Two-byte task attribute messages are used to specify an identifier, called a tag, for a task that establishes the I_T_L_Q nexus. The Task Attribute messages are listed in Table 40. The Tag field is an 8-bit unsigned integer assigned by the application client and sent to the initiator in the send SCSI command request (see ANSI SPI-4, T10/1365D Rev. 8, Section 19.3.2).

4.4.1 Task attribute message overview and codes

The tag for every task for each I_T_L nexus shall be uniquely assigned by the application client. There is no requirement for the task manager to check whether a tag is currently in use for another I_T_L nexus. If the task manager checks the tag value and receives a tag that is currently in use for the I_T_L nexus, then it shall abort all tasks for the initiator and the associated logical unit and shall return Check Condition status for the task that caused the overlapped tag. The sense key shall be set to Aborted Command and the additional sense code shall be set to Overlapped Commands Attempted with the additional sense code qualifier set to the value of the duplicate tag (see Section 7.3). A tag becomes available for reassignment when the task ends. The numeric value of a tag is arbitrary, providing there are no outstanding duplicates, and shall not affect the order of execution.

For each logical unit on each target, each application client has up to 256 tags to assign to tasks. Thus a target with eight logical units may have up to 14,336 tasks concurrently in existence if there were seven initiators on the bus.

Whenever an initiator does a physical connection to a target, the appropriate task attribute message shall be sent following the Identify message to establish the I_T_L_Q nexus for the task. Only one I_T_L_Q nexus may be established during an initial connection or physical reconnection. If a task attribute message is not sent, then only an I_T_L nexus is established for the task (i.e., an untagged command).

When a target does a physical reconnection to an initiator to continue a tagged task, the Simple Queue message shall be sent following the Identify message to resume the I_T_L_Q nexus for the task. Only one I_T_L_Q nexus may occur during a physical reconnection. If the Simple Tag message is not sent, then only an I_T_L nexus occurs for the task (i.e., an untagged command).

If a target attempts to do a physical reconnection using an invalid tag, then the initiator should create an attention condition. After the corresponding MESSAGE OUT phase, the initiator shall respond with an Abort Task message.

If a target does not implement tagged queuing and a queue tag message is received, the target shall switch to a Message In phase with a MESSAGE REJECT message and accept the task as if it were untagged, provided there are no outstanding untagged tasks from that initiator.

See Section 7.0 of this manual and the ANSI SCSI Architecture Model-2 standard for the task set management rules.

Table 40: Task attribute message codes

Code	Support				Message name	Direction		Clear attention condition
	IU Transfers Disabled		IU Transfers Enabled					
	Iniator	Target	Initiator	Target				
24h	O	O	N/A	N/A	ACA		Out	Not required
21h	Q	Q	N/A	N/A	Head of Queue		Out	Not required
22h	Q	Q	N/A	N/A	Ordered		Out	Not required
20h	Q	Q	N/A	N/A	Simple	In	Out	Not required

Key:

M = Mandatory support O = Optional support Q = Mandatory if tagged queuing is implemented

In = Target to initiator Out = Initiator to target

Yes = Initiator shall clear the attention condition before last ACK of the MESSAGE OUT phase.

Not required = Initiator may or may not clear the attention condition before last ACK of the MESSAGE OUT phase (see Section 5.1).

N/A = Not applicable

4.4.2 ACA (AUTO CONTINGENT ALLEGIANCE)

See Table 41 for the format of the ACA message.

Table 41: ACA message format

Bit / Byte	7	6	5	4	3	2	1	0
0	MESSAGE CODE (24h)							
1	TAG (00h-FFh)							

The ACA message specifies that the task shall be placed in the task set as an ACA task. The rules used by the task manager to handle ACA tasks within a task set are defined in Section 7.0 of this manual and the SCSI Architecture Model-2 standard.

4.4.3 HEAD OF QUEUE

See Table 42 for the format of the Head of Queue message.

Table 42: Head of Queue message format

Bit Byte	7	6	5	4	3	2	1	0
0	**MESSAGE CODE (21h)**							
1	**TAG (00h-FFh)**							

The Head of Queue message specifies that the task shall be placed in the task set as a Head of Queue task. The rules used by the device server to handle Head of Queue tasks within a task set are defined in Section 7.0 of this manual and in the SCSI Architecture Model-2 standard.

4.4.4 ORDERED

See Table 43 for the format of the Ordered message.

Table 43: Ordered message format

Bit Byte	7	6	5	4	3	2	1	0
0	MESSAGE CODE (22h)							
1	TAG (00h-FFh)							

The Ordered message specifies that the task shall be placed in the task set as an Ordered task. The rules used by the task manager to handle Ordered tasks within a task set are defined in Section 7.0 of this manual and in the SCSI Architecture Model-2 standard.

4.4.5 SIMPLE

See Table 44 for the format of the Simple message.

Table 44: Simple message format

Bit Byte	7	6	5	4	3	2	1	0
0	MESSAGE CODE (20h)							
1	TAG (00h-FFh)							

The Simple message specifies that the task shall be placed in the task set as a Simple task. The rules used by the task manager to handle Simple tasks within a task set are defined in Section 7.0 of this manual and in the SCSI Architecture Model-2 standard.

4.5 Task management messages

Table 45 lists the codes used for messages that manage tasks. Details about these task management messages is given in subsections following the table.

4.5.1 Task management message codes

Table 45: Task management message codes

Code	Support				Message name	Direction	Clear attention condition
	IU Transfers Disabled		IU Transfers Enabled				
	Initiator	Target	Initiator	Target			
0Dh	Q	Q	M	M	ABORT TASK	Out	Yes
06h	O	M	N/A	N/A	ABORT TASK SET	Out	Yes
16h	O	O	N/A	N/A	CLEAR ACA	Out	Not required
0Eh	Q	Q	N/A	N/A	CLEAR TASK SET	Out	Yes
17h	M	O	N/A	N/A	LOGICAL UNIT RESET	Out	Yes
0Ch	O	M	O	M	TARGET RESET	Out	Yes

Key:

M = Mandatory support O = Optional support Q = Mandatory if tagged queuing is implemented

In = Target to initiator Out = Initiator to target

Yes = Initiator shall clear the attention condition before last ACK of the MESSAGE OUT phase.

Not required = Initiator may or may not clear the attention condition before last ACK of the MESSAGE OUT phase (see Section 5.1).

N/A = Not applicable. The receiving SCSI device shall reject this message.

4.5.2 ABORT TASK

The ABORT TASK message is defined in the SCSI Architecture Model-4 standard.

In addition to the requirements in the SCSI Architecture Model-4 standard, the target shall go to the Bus Free phase following the successful receipt of the ABORT TASK message.

If only an I_T nexus has been established, the target shall go to the Bus Free phase. No status or message shall be sent for the current task and no pending data, status, or tasks are affected.

Note. The ABORT TASK message in the case of only an I_T nexus is useful to an initiator that is not able to get an Identify message through to the target due to parity errors and just needs to end the current task. Any pending data, status, or tasks for the I_T nexus are not affected. It is not possible to abort an I_T nexus on a physical reconnection because of item (f) in Section 5.1.

On a physical reconnection, the ABORT TASK message aborts the current task if it is fully identified. If the current task is not fully identified (i.e., an I_T_L nexus exists, but the target is doing a physical reconnecting for an I_T_L_Q nexus), then the current task is not aborted and the target goes to the BUS FREE phase.

Note. A nexus may not be fully identified on a physical reconnection if an attention condition is created during the Identify message and the target has any tagged tasks for that initiator on that logical unit.

It is not an error to issue this message to an I_T_L or I_T_L_Q nexus that does not have any pending tasks.

4.5.3 ABORT TASK SET

The ABORT TASK SET message is defined in the SCSI Architecture Model-4 standard. In addition to the requirements in the SCSI Architecture Model-4 standard the target shall go to the BUS FREE phase following the successful receipt of the ABORT TASK SET message.

If only an I_T nexus has been established, the target shall switch to a BUS FREE phase. No status or message shall be sent for the current task and no pending data, status, or tasks are affected.

Note. The ABORT TASK SET message in the case of only an I_T nexus is useful to an initiator that is not able to send an Identify message through to the target due to parity errors and just needs to end the current task or task management function. It is not an error to issue this message to an I_T_L nexus that does not have any pending or current tasks.

4.5.4 CLEAR ACA

The CLEAR ACA message is defined in the SCSI Architecture Model-4 standard.

The CLEAR ACA message shall only be sent by an initiator during an initial connection. If the target receives the CLEAR ACA message at any other time the target shall switch to a Message In phase and issue a MESSAGE REJECT message. The target shall then continue processing the task that was in process when the CLEAR ACA message was received.

On receipt of a CLEAR ACA message, the task manager, in addition to clearing the ACA condition, shall go to the BUS FREE phase following the successful receipt of the CLEAR ACA message.

It is not an error to issue a CLEAR ACA message when no ACA condition is in effect.

4.5.5 CLEAR TASK SET

The CLEAR TASK SET message is defined in the SCSI Architecture Model-4 standard.

In addition to the requirements in the SCSI Architecture Model-4 standard, the target shall go to the BUS FREE phase following the successful receipt of the CLEAR TASK SET message.

4.5.6 LOGICAL UNIT RESET

The LOGICAL UNIT RESET message is defined in the SCSI Architecture Model-4 standard.

Receipt of the LOGICAL UNIT RESET message after an I_T_L nexus has been astablished is a logical unit reset event as defined in the SCSI Architecture Model-4 standard.

In addition to the requirements in the SCSI Architecture Model-4 standard, the target shall go to the BUS FREE phase following the successful receipt of the LOGICAL UNIT RESET message.

4.5.7 TARGET RESET

The TARGET RESET message is defined in the SCSI Architecture Model-4 standard.

In addition to the requirements in the SCSI Architecture Model-4 standard, the target shall go to the BUS FREE phase following the successful receipt of the TARGET RESET message.

5.0 Miscellaneous SCSI bus characteristics

Asynchronous conditions can occur on the SCSI bus that cause a SCSI device to perform certain actions that may alter the phase sequence of an I/O process. Furthermore, SCSI devices may not all be powered on at the same time. This publication does not address power sequencing issues. However, each SCSI device, as it is powered on, should perform appropriate internal reset operations and internal test operations. Following a power on to selection time after powering on, SCSI targets should be able to respond with appropriate status and sense data to the TEST UNIT READY, INQUIRY, and REQUEST SENSE commands. Sections 5.1, 5.3, and 5.4 describe some asynchronous conditions that could arise during operations on a SCSI bus. Section 5.5 describes SCSI pointers, a subject related to the first three. Section 4 describes the SCSI message system, which also enters into the overall SCSI operating picture.

5.1 Attention condition

The attention condition allows a SCSI initiator port to inform a SCSI target port that the initiator has a message ready. The target shall honor all valid attention conditions by performing a MESSAGE OUT phase.

The initiator may create an attention condition during the SELECTION phase and during all information transfer phases.

To create an attention condition during the SELECTION phase following normal arbitration, the initiator shall assert the ATN signal at least two system deskew delays before releasing the BSY signal.

To create an attention condition during the SELECTION phase following a QAS, the initiator shall assert the ATN signal at least two system deskew delays before asserting the target's ID in the bus.

To create an attention condition during an information transfer phase, the initiator shall assert the ATN signal at least an attention setup time before negating the ACK signal. To reestablish an attention condition during a multi-byte Message Out retry, the initiator shall assert the ATN signal two system deskew delays before asserting the ACK signal on the first message byte. To clear an attention condition during an information transfer phase, the initiator shall negate the ATN signal at least two system deskew delays before asserting the ACK signal. The initiator shall not negate the ATN signal while the ACK signal is asserted during a MESSAGE OUT phase.

The initiator shall create the attention condition on or before the last information transfer in a bus phase or information unit, for the attention condition to be honored before transition to a new bus phase or information unit. If the initiator does not meet the attention condition setup time, the target may not honor the attention condition until a later bus phase or information unit, possibly resulting in an unexpected action. The initiator shall keep the ATN signal asserted until the target responds to the attention condition.

Once the target has responded to the attention condition by going to MESSAGE OUT phase, the initiator shall keep the attention condition set if more than one message byte is to be transferred. The initiator shall clear the attention condition on the last message byte to be sent. The initiator shall clear the attention condition while transferring the last byte of the messages indicated with a Yes in Tables 27, 40, and 45. If the target detects that the initiator failed to meet this requirement, then the target shall go to BUS FREE phase (see Section 3.1).

A SCSI target port shall respond to an attention condition with MESSAGE OUT phase as follows:

a. If an attention condition is created during a COMMAND phase, the target shall enter MESSAGE OUT phase after transferring part or all of the command descriptor block.
b. If an attention condition is created during a DATA phase, the target shall enter MESSAGE OUT phase at the target's earliest convenience (for example, on a logical block boundary). The initiator shall continue REQ/ACK handshakes until it detects the phase change.
c. If an attention condition is created during a STATUS phase, the target shall enter MESSAGE OUT phase after the status byte has been acknowledged by the initiator.
d. If an attention condition is created during a MESSAGE IN phase, the target shall enter MESSAGE OUT phase before it sends another message. This permits a MESSAGE PARITY ERROR message from the initiator to be associated with the appropriate message.
e. If an attention condition is created during a SELECTION phase, the target shall enter MESSAGE OUT phase after that SELECTION phase.
f. If SPI (SCSI Parallel Interface) information unit transfers are disabled and an attention condition is detected during a RESELECTION phase, the SCSI target port shall enter MESSAGE OUT phase after the SCSI target has sent its IDENTIFY message for that RESELECTION phase.
g. If an attention condition is created during a RESELECTION phase the target shall enter MESSAGE OUT phase after the target has sent the first Information Transfer phase.
h. If the attention condition is detected during the transfer of a SPI data stream information unit, the target shall terminate the current stream by entering into MESSAGE OUT phase at the end of any SPI data stream information unit in the current stream.
i. If the attention condition is detected between SPI information units, the target shall enter MESSAGE OUT phase at the completion of the next SPI information unit.

During a RESELECTION phase, the initiator should only create an attention condition to transmit an ABORT TASK SET, Abort Task, Target Reset, Clear Task Set, Disconnect, Logical Unit Reset, or NO OPERATION message. Other uses may result in ambiguities concerning the nexus.

The initiator shall keep the ATN signal asserted throughout the MESSAGE OUT phase if more than one byte is to be transferred. Unless otherwise specified, the initiator may negate the ATN signal at any time that does not violate the specified setup and hold times, except it shall not negate the ATN signal while the ACK signal is asserted during a MESSAGE OUT phase. Normally, the initiator negates the ATN signal while the REQ signal is true and the ACK signal is false during the last REQ/ACK handshake of the MESSAGE OUT phase.

5.2 Bus reset condition

The bus reset condition is used to create a hard reset (see Section 5.3) for all SCSI devices on the bus and change the bus to a BUS FREE phase. This condition shall take precedence over all other phases and conditions. Any SCSI device may create the bus reset condition by asserting the RST signal for a minimum of a reset hold time.

Environmental conditions (for example, static discharge) may generate brief glitches on the RST signal. SCSI devices shall not react to glitches on the RST signal that are less than a reset delay. The manner of rejecting glitches is vendor-specific. The bus clear delay following a RST signal transition to true is measured from the original transition of the RST signal. This limits the time to confirm the RST signal to a maximum of a bus clear delay.

5.3 Hard reset

A SCSI device detecting a reset event shall release all SCSI bus signals within a bus clear delay of the transition of the RST signal to true, except the RST signal if it is asserting RST. The BUS FREE phase always follows the hard reset condition. The SCSI device shall not assert the RST signal in response to a reset event on the same bus segment.

The effect of the hard reset on tasks that have not completed, SCSI device reservations, and SCSI device operating modes is defined in the SCSI Architecture Model-4 standard.

Any SCSI device that detects a hard reset shall also set its transfer agreement to the default transfer agreement.

5.4 Reset events

When a SCSI device detects a reset event it shall initiate hard reset.

5.4.1 Bus reset event

When a SCSI device detects a bus reset condition by detecting RST true for a reset delay, it shall cause a reset event. In response to a bus reset event, a SCSI target port shall create a unit attention condition for all initiators. The sense key shall be set to Unit Attention with the additional sense code set to either SCSI Bus Reset Occurred or Power On, Reset, or Bus Device Reset Occurred.

5.4.2 Power on reset event

When a SCSI device is powered on, it shall cause a reset event. In response to a power on reset event, the target shall create a unit attention condition for all initiators. The sense key shall be set to Unit Attention with the additional sense code set to either Power On Occurred or Power On, Reset, or Bus Device Reset Occurred.

5.4.3 Target reset event

When a SCSI device successfully receives a SCSI target port Reset message, it shall cause a reset event. In response to a SCSI target port reset event, the SCSI target shall create a unit attention condition for all SCSI initiator ports. The sense key shall be set to Unit Attention with the additional sense code set to either Bus Device Reset Function Occurred or Power On, Reset, or Bus Device Reset Occurred.

5.4.4 Transceiver mode change reset event

When a SCSI device that contains multimode transceivers detects a transceiver mode change from LVD mode to MSE mode it shall cause a reset event. In response to the transceiver mode change reset event, a SCSI target port shall create a unit attention condition for all initiators. The sense key shall be set to Unit Attention, and the additional sense code set to Transceiver Mode Changed to Single-ended.

When a SCSI device that contains multimode transceivers detects a transceiver mode change from MSE mode to LVD mode it shall cause a reset event. In response to the transceiver mode change reset event, a SCSI target port shall create a unit attention condition for all initiators. The sense key shall be set to Unit Attention, and the additional sense code set to Transceiver Mode Changed to LVD.

5.5 Asynchronous condition recovery

Part of the asynchronous condition recovery stem are sets of pointers to storage areas where information is kept that, in addition to being necessary to the start of a task, allows tasks to be interrupted, restarted or repeated. The following subsections describe these pointers.

5.5.1 SCSI pointers

The SCSI architecture provides for sets of pointers to be kept within each initiator task control area. These pointers are in sets of three pointers per set. The pointers in each set point to three storage area sections in the initiator. The three sections contain the following information:

a. A command from initiator to target.
b. Status (from target) associated with the command.
c. Data (to/from target) associated with the command.

Of these three-pointer sets there are two types:

- Active pointers (one set per initiator only).
- Saved pointers (one or more sets per initiator, up to fifteen sets total).

The use of these two types of pointers is described in the following paragraphs.

5.5.2 Active pointers

Active pointers represent the current state of the interface between the initiator and the target that the initiator is currently connected to and servicing. The pointers for the current task remain in the initiator Active Pointer registers from the time they are put there after the completion of all activities associated with the previous task, until the logic of the Initiator dictates a new task shall be executed. Normally, successful receipt by the initiator of good status associated with the current task triggers the initiator to insert a new set of the three Active pointers for the next task. The initiator does not wait for the TASK COMPLETE message before deciding whether to retry the Active task or transfer in new task pointers. If the current task was never satisfactorily completed, the initiator logic may dictate that some special response action be taken, such as restoring the values in the Active pointer registers to their beginning values so the current task can be restarted (see note following). The initiator may choose to send a command such as a REQUEST SENSE command to the target, or the initiator could ignore the unsatisfactorily completed task and start the next originally scheduled task.

Note. For example, if the drive detects a parity error in the data out from the current command, it sends the RESTORE POINTERS message to the initiator. In this case, the Restore Pointers request causes the initiator to restore as the active pointers the values existing at the beginning of the current task so the current task can be resent. The “beginning” pointer values point to the first byte of the current task (very likely a command descriptor block), the first byte of the area set aside for status to be returned and the first byte of the area set aside for data associated with the current task. (See Section 4.3.14 for a detailed description of operations resulting from the RESTORE POINTERS message).

5.5.3 Saved pointers

Saved pointers point to initiator storage locations where command, status and data information are stored for a task that was saved at some point in the past. There is one set of saved pointers for the current task for each target on the interface bus that is currently active (whether or not it is currently connected to the initiator). The saved command pointer always points to the first byte of the Command Descriptor Block (see Section 7.2) for the “current” task for each target, the saved status pointer always points to the first byte of the area used for the status associated with the command, and the saved data pointer points to some location (not necessarily the beginning) in the area used for data associated with the command. When a SCSI target port disconnects the initiator saves the current pointers. Before a SCSI target port disconnects it may send a SAVE DATA POINTERS message to the initiator, which copies the data pointer that is for the current command for that target into the location set aside for the target’s saved pointers. When a SCSI target port reconnects, the initiator performs a restore pointers operation that copies the saved pointers for the reconnected target into the initiator current pointer registers so that the current command for that target may continue its operation from where it left off before disconnecting. If the target had sent a Save Data Pointer message previously, the current data pointer points to the place in the data store area where operations left off. The data pointer otherwise points to the beginning of the data area, unless the data pointers were modified by a MODIFY DATA POINTERS message from the target prior to disconnecting. The MODIFY DATA POINTERS message adds a value to the data pointer that allows data to be taken, upon reconnection, from a location before or after the last byte transferred location.

Since the data pointer value may be modified by the target before the task ends, it should not be used to test for actual transfer length, because the value may no longer be valid for that purpose.

5.6 Command processing considerations and exception conditions

5.6.1 Command processing considerations and exception conditions overview

The following subclauses describe some aspects of command processing, including exception conditions and error handling that are specific to this standard.

5.6.2 Asynchronous event notification

Notification of an asynchronous event is performed using the SEND command with the AER bit set to one. The information identifying the condition being reported shall be returned during the data out delivery phase of the SEND command (see SCSI Primary Commands-4 standard).

An error condition or unit attention condition shall be reported once per occurrence of the event causing it. The target may choose to use an asynchronous event notification or to return CHECK CONDITION status on a subsequent command, but not both. Notification of command-related error conditions shall be sent only to the initiator that requested the task.

The asynchronous event notification protocol may be used to notify processor devices that a system resource has become available. If a SCSI target port chooses to use this method, the sense key in the sense data sent to the processor device shall be set to Unit Attention.

The asynchronous event notification protocol shall be used only with SCSI devices that return processor device type with an AERC bit of one in response to an INQUIRY command. The INQUIRY command should be sent to logical unit zero of each SCSI device responding to selection. This procedure shall be conducted prior to the first asynchronous event notification and shall be repeated whenever the SCSI device deems it appropriate or when an event occurs that may invalidate the current information. (See SYNCHRONOUS DATA TRANSFER REQUEST message, Section 4.3.16, for examples of these events.)

Each SCSI device that returns processor device type with an AERC bit of one shall be issued a TEST UNIT READY command to determine that the SCSI device is ready to receive an asynchronous event notification. A SCSI device returning CHECK CONDITION status is issued a REQUEST SENSE command. This clears any pending unit attention condition. A SCSI device that returns processor device type with an AERC bit of one and returns Good status when issued a TEST UNIT READY command shall accept a SEND command with an AER bit of one.

Note. A SCSI device that uses asynchronous event notification at initialization time should provide means to defeat these notifications. This may be done with a switch or jumper wire. SCSI devices that implement saved parameters may alternatively save the asynchronous event notification permissions either on a per SCSI device basis or as a system wide option. In any case, a SCSI device conducts a survey with INQUIRY commands to be sure that the SCSI devices on the SCSI bus are appropriate destinations for SEND commands with an AER bit of one. (The SCSI devices on the bus or the SCSI ID assignments may have changed.)

See asynchronous event reporting in the SCSI Architecture Model-4 standard for more information on asynchronous event notification.

This feature is not likely supported at this time. Check individual drive's Product Manual, in features supported table to see if it is supported by a particular drive model.

5.6.3 Incorrect initiator connection

An incorrect initiator connection occurs during an initial connection if a SCSI initiator port creates a nexus that already exists and does not send an ABORT TASK SET, ABORT TASK, TARGET RESET, CLEAR TASK SET, DISCONNECT, or LOGICAL UNIT RESET message as one of the messages of the MESSAGE OUT phase or as one of the task management flags in the SPI command information unit.

A task manager that detects an incorrect initiator connection shall abort all tasks for the initiator and the associated logical unit and shall return CHECK CONDITION status for the task that caused the incorrect initiator connection. The sense key shall be set to Aborted Command and the additional sense code shall be set to Overlapped Commands Attempted with the additional sense code qualifier set to the value of the duplicate tag (see Section 4.4).

Note. An incorrect initiator connection may be indicative of a serious error and, if not detected, could result in a task operating with a wrong set of pointers. This is considered a catastrophic failure on the part of the initiator. Therefore, vendor-specific error recovery procedures may be required to guarantee the data integrity on the medium. The target may return additional sense data to aid in this error recovery procedure (e.g., sequential-access devices may return the residue of blocks remaining to be written or read at the time the second command was received).

5.6.4 Unexpected RESELECTION phase

An unexpected RESELECTION phase occurs if a SCSI target port attempts to do a physical reconnect to a task for which a nexus does not exist. a SCSI initiator port should respond to an unexpected RESELECTION phase by sending an ABORT TASK message.

6.0 SPI information units

An information unit transfer transfers data in SPI information units. The order in which SPI information units are transferred within an information unit transfer follows a prescribed sequence. When information unit transfers are enabled, only SPI information units shall be transferred within the DT DATA OUT phase and DT DATA IN phase.

The SPI information unit sequences shall be as shown in figures 15, 16, 17, and 18. See figures 12 and 13 for the sequencing rules between the DT Data In or DT DATA OUT phases and the other phases.

For information on information unit exception handling, see sections 3.5.2.2.1.1 and 3.5.2.2.1.2.

The normal progression is from SPI L_Q information unit/SPI command information unit pairs, to SPI L_Q information unit/SPI data information unit pairs, to SPI L_Q information unit/SPI status information unit pairs.

Note. a SCSI initiator port may request a BUS FREE phase by creating an attention condition and sending a DISCONNECT message on the corresponding MESSAGE OUT phase. This allows a SCSI initiator port to request the target break up a long sequence of SPI L_Q information unit/SPI data information unit pairs into smaller sequences.

After message phases complete that contain any negotiation (such as, PPR, WDTR, or SDTR) that results in UI_REQ being changed, the SCSI target device shall abort all tasks, except the current task, for the initiator participating in the negotiation and the SCSI initiator shall abort all tasks, except the current task, for the target device.

When an information unit transfer agreement is in effect, there is no option equivalent to the "physical disconnect without sending a save Data Pointer message." The initiator shall save the data pointers as soon as the last byte of the last iuCRC for a SPI information unit is transferred. The save shall occur even if the initiator detects an error in the SPI data information unit.

The target shall not start a new information unit transfer until all previous REQ(s) have been responded to by an equal number of ACK(s) except during a sequence of SPI data stream information units (see Section 6.2.4).

6.1 Information unit transfer logical operations

SCSI devices using information unit transfers may transfer SPI information units for any number of I/O processes by using logical connects, logical disconnects, and logical reconnects.

If there are no phase changes to a MESSAGE OUT phase or a MESSAGE IN phase, then logical disconnects shall occur at the completion of:

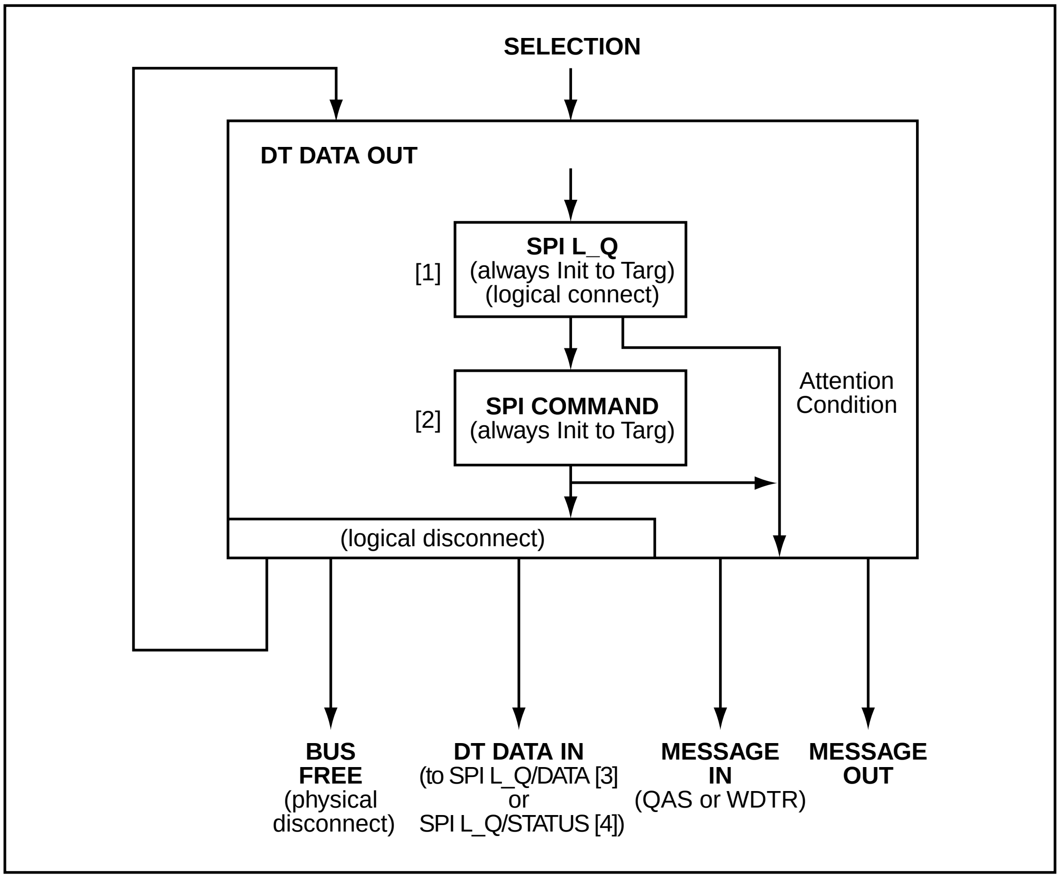
a. each SPI command information unit;
b. each SPI status information unit;
c. each SPI data information unit;
d. any SPI L_Q information unit if SPI L_Q information unit Data Length field is zero; and
e. the last SPI data stream information unit.

At completion of those SPI information units the I_T_L_Q nexus becomes an I_T nexus. The I_T nexus remains in place until the target does a physical disconnect or an I_T_L_Q nexus is reestablished by the target transmitting a SPI L_Q information unit.

Logical reconnections occur on the successful target transmission and initiator receipt of a SPI L_Q information unit for an existing I/O process. The logical reconnection reestablishes the I_T_L_Q nexus for that I/O process.

SCSI devices using information unit transfers may receive several commands during an initial connection. This occurs when a SCSI initiator port uses the multiple command option in the SPI L_Q information unit. For each SPI L_Q received with a multiple command type or a last command type, a logical connection occurs and an I_T_L_Q nexus is formed.

If there is a phase change to a MESSAGE OUT phase or a MESSAGE IN phase, then there is no logical disconnect and the I_T_L_Q nexus remains in place. If a DT DATA phase follows the message phase, then the L_Q portion of the current I_T_L_Q nexus shall be replaced with the L_Q in the next SPI L_Q information unit.



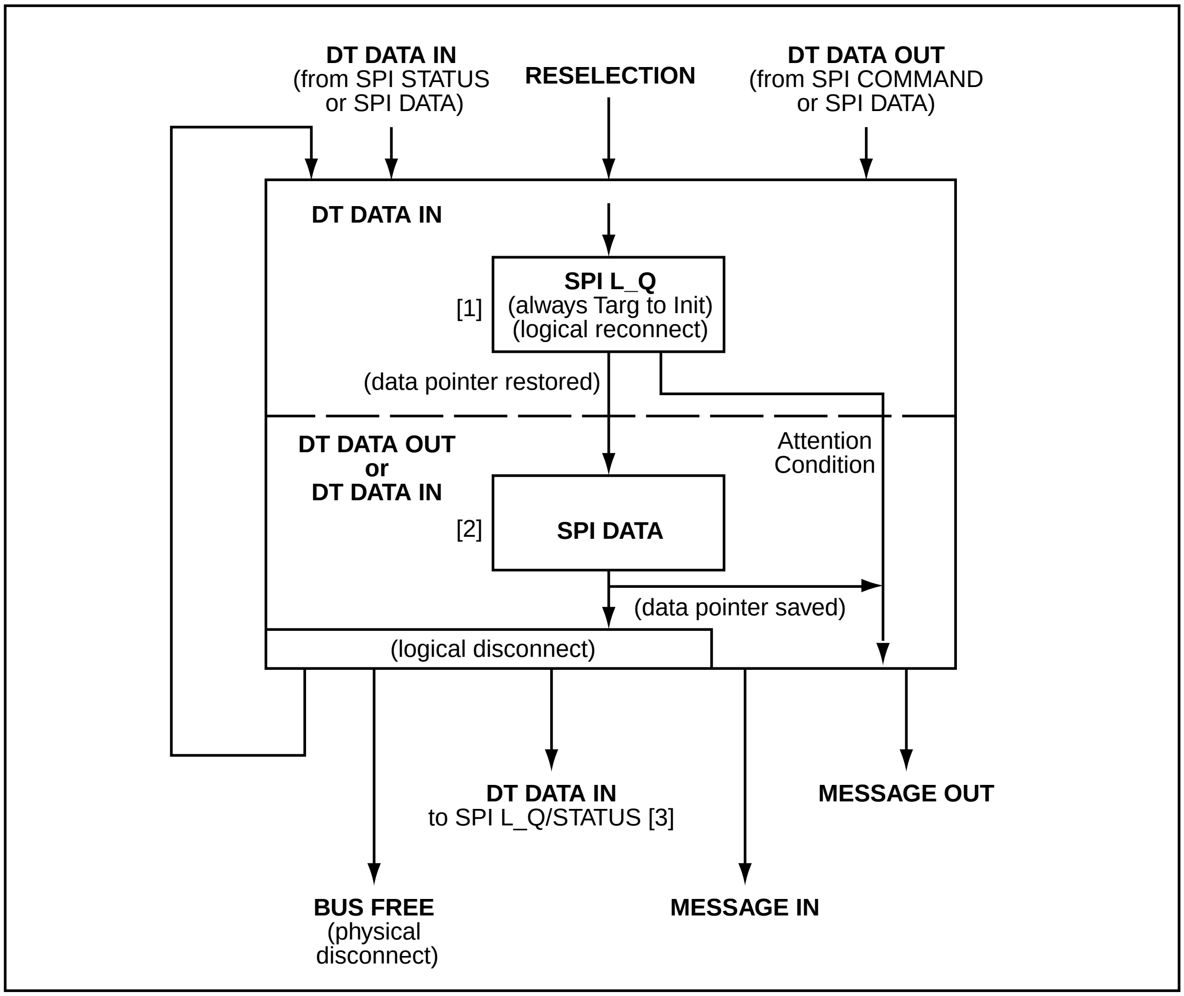
[1] See Table 49 for the format of the SPI L_Q information unit.

[2] See Table 46 for the format of the SPI Command information unit.

[3] See Figure 16.

[4] See Figure 18.

Figure 15. SPI information unit sequence during initial connection

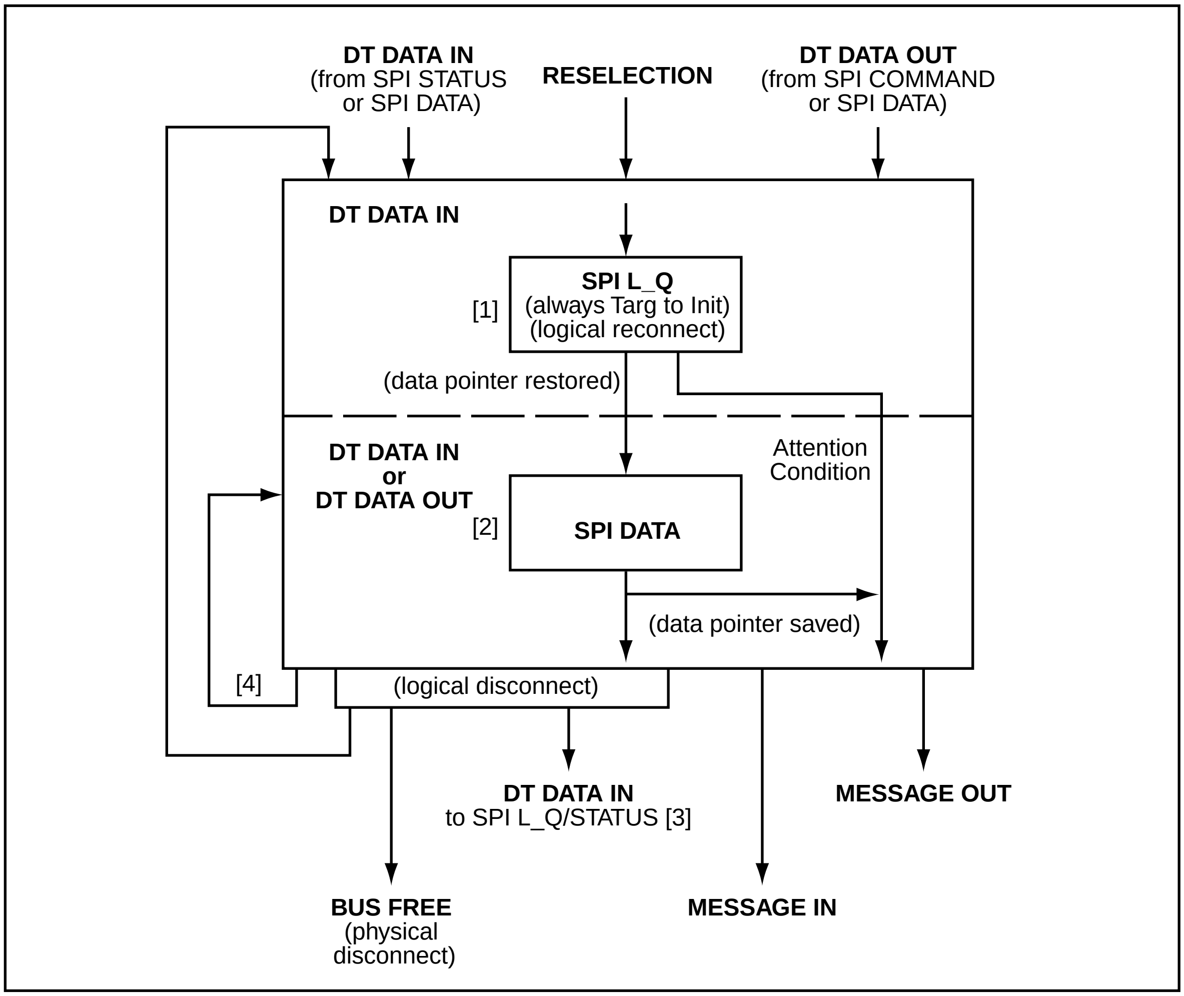


[1] See Table 49 for the format of the SPI L_Q information unit.

[2] See Table 52 for the format of the SPI Data information unit.

[3] See Figure 18.

Figure 16. SPI information unit sequence during data type transfers

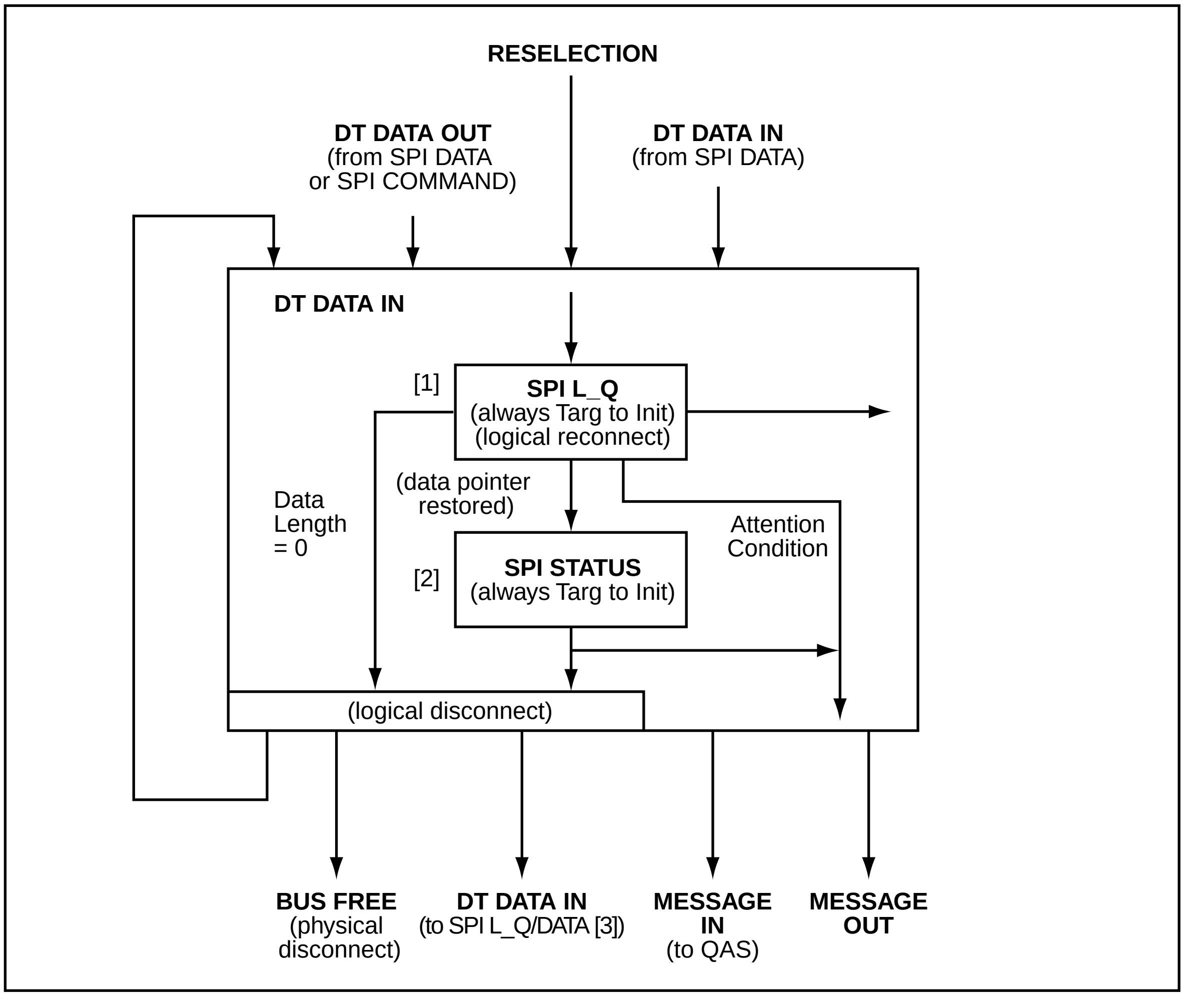


[1] See Table 49 for the format of the SPI L_Q information unit.

[2] See Table 53 for the format of the SPI Data information unit.

[3] See Figure 18.

Figure 17. SPI information unit sequence during data stream type transfers



[1] See Table 49 for the format of the SPI L_Q information unit.

[2] See Table 54 for the format of the SPI Status information unit.

[3] See Figure 16.

Figure 18. SPI information unit sequence during status transfers

6.2 SPI information units

6.2.1 SPI command information unit

The SPI command information unit (see Table 46) transfers CDBs, task attributes, and task management requests to be performed by a device server.

a SCSI initiator port shall consider a BUS FREE phase after the transfer of a SPI command information unit to be equivalent to receiving a DISCONNECT message.

If a SCSI target port does not have the resources to accept a SPI command information unit and the Task Management Flags field equals 00h the target shall transfer all the bytes of the current SPI command information unit but need not hold the transmitted information. After transferring all the SPI command information unit bytes the target shall change to a DT DATA IN phase and transmit a SPI status information unit with the status defined in the SCSI Architecture Model-4 standard (SAM-4) for this condition. If the initiator has more commands to send the target, the initiator shall wait at least until the next selection before those remaining commands may be sent.

If the Task Management Flags field is a supported value not equal to 00h, the target shall perform the selected task management function before processing any further SPI information units regardless of the command type. On completion of a support task management function, the target shall go to a BUS FREE phase. No SPI status information unit shall be reported for the task management function. If the Task Management Flags field is not a supported value, then the task manager shall terminate the task with a Good status and the packetized failure code shall be set to task management function not supported. If a task management function fails, the task manager shall terminate the task with a Good status. The packetized failure code shall be set to task management function failed.

If the target terminates a SPI L_Q/SPI command information unit pair for one of the following reasons:

a. Task Set Full status,
b. Busy status,
c. Check Condition due to a SPI command information unit iuCRC error, or
d. a bus free due to a SPI L_Q information unit iuCRC error,

it shall have no effect on any other SPI L_Q/SPI command information unit pair beyond those caused by any task management functions contained within the last SPI L_Q/SPI command information unit pair.

Table 46: SPI command information unit

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved					Task Attribute [1] [7]		
2	Task Management Flags [2] [7]							
3	ADDITIONAL CDB LENGTH = (number of additional CDB bytes)/4 [3]						RDDATA [4]	WRDATA [4]
4	(MSB)	CDB [5] [7] [8]						
19								(LSB)
20	(MSB)	Additional CDB [6] [7] [8]						
n								(LSB)
n+1	(MSB)	iuCRC [9]						
n+2								
n+3								
n+4								(LSB)

[1] The Task Attribute field is defined in Table 47.

Table 47: Task Attribute

Codes	Description
000b	Requests that the task be managed according to the rules for a simple task attribute. See the SCSI Architecture Model-4 standard.
001b	Requests that the task be managed according to the rules for head of queue task attribute. See the SCSI Architecture Model-4 standard.
010b	Requests that the task be managed according to the rules for an ordered attribute. See the SCSI Architecture Model-4 standard.
011b	Reserved.
100b	Requests that the task be managed according to the rules for a simple task attribute. See the SCSI Architecture Model-4 standard.
101b-111b	Reserved.

[2] The Task Management Flags field is defined in Table 48. If a Task Management function fails, the Task Manager shall terminate the task with Good status. The packetized failure code shall be set to Task Management Function Failed.

Table 48: Task Management flags

Codes	Description
00h	Indicates no task management requests for the current task.
01h	The task manager shall abort the task as defined in the ABORT TASK message (see Section 4.5.2).
02h	The task manager shall abort the task set as defined in the ABORT TASK SET message (see Section 4.5.3).
04h	The task manager shall clear the task set as defined in the CLEAR TASK SET message (see Section 4.5.5).
08h	The task manager shall perform a hard reset to the selected logical unit as defined in the LOGICAL UNIT RESET message (see Section 4.5.6).
20h	The task manager shall perform a hard reset as defined in the TARGET RESET message (see Section 4.5.7).
40h	The task manager shall perform a clear ACA as defined in the CLEAR ACA message (see Section 4.5.4).
All other values reserved	The task manager shall terminate the task with a Good status. The packetized failure code shall be set to Task Management Function Not Supported.

[3] The Additional CDB Length field contains the length in 4-byte words of the Additional CDB field.

[4] The Write Data (WRDATA) bit and READ Data (RDDATA) bit are defined in SCSI protocol standards T10/1144D (FCP-2), sections 9.1, 9.1.16, and 9.1.17. They are also discussed in the Seagate *Fibre Channel Interface Manual*, part number 77767496. These two bits are used by drives in a Fibre Channel I/O system and shall be ignored by this section.

[5] The CDB field contains the actual CDB to be interpreted by the addressed logical unit. The maximum CDB length is 16 bytes. The CDB field and the task attribute field is not valid and is ignored if the Task

Management Flags field is not zero. Any bytes between the end of a 6 byte CDB, 10 byte CDB, or 12 byte CDB and the end of the CDB field shall be reserved.

[6] The Additional CDB field contains any CDB bytes beyond those contained within the standard 16 byte CDB field.

[7] The CDB, Additional CDB, and Task Attribute fields are not valid and are ignored if the Task Management Flags field is not zero.

[8] The contents of the CDB and Additional CDB fields shall be as defined in the SCSI command standards.

[9] The iuCRC field shall use the algorithm defined in Subsection 11.3 of ANSI SCSI Parallel Interface (SPI-5).

6.2.2 SPI L_Q information unit

The SPI L_Q information unit (see Table 49) contains L_Q nexus (Logical unit—Q tag relationship) information for the information unit that follows, the type of information unit that follows, and the length of information unit that follows. A SPI L_Q information unit shall precede all SPI command information units, SPI multiple command information units, SPI data information units, SPI status information units, and the first of an uninterrupted sequence of SPI data stream information units.

The receipt of an error free (i.e., no iuCRC error) SPI L_Q information unit by a SCSI initiator port shall cause the initiator to restore the data pointer.

Table 49: SPI L_Q information unit

Bit Byte	7	6	5	4	3	2	1	0
0	Type [1]							
1	Reserved							
2	MSB	Tag [2]						
3								LSB
4	MSB	Logical Unit Number [3]						
5								
6								
7								
8								
9								
10								
11								LSB
12	Reserved							

Bit / Byte	7	6	5	4	3	2	1	0
13	MSB							
14	Data Length [4]							
15								LSB
16	BIDI Direction [5]		Reserved					
17	Reserved							
18	MSB							
19	iuCRC Interval [6]							LSB
20	MSB							
21	iuCRC [7]							
22								
23								LSB

[1] The Type field is defined in Table 50. If a SCSI initiator port receives a type code that is not defined in Table 50, that initiator shall follow the procedures defined in Section 3.5.2.2.1.1. If a SCSI target port receives a type code that is not defined in Table 50, that target shall follow the procedures defined in Section 3.5.2.2.1.2.

Table 50: SPI L_Q information unit type

Codes	Type	Description
01h	Last Command	Sent by a SCSI initiator port device to indicate a SPI command information unit shall follow this SPI L_Q information unit. Indicates the initiator device shall not send any more SPI command information units during the current connection. The value of the Data Length field shall be greater than or equal to 14h and less than or equal to 90h. The iuCRC Interval field shall be set to zero and ignored by the target device. The Bidi Direction field shall be set to zero and ignored by the receiving SCSI target device.
02h	Multiple Command	Sent by a SCSI initiator port device to indicate a SPI command information unit shall follow this SPI L_Q information unit. Indicates the initiator device has another SPI L_Q information unit and SPI command information unit during the current connection. The value of the Data Length field shall be greater than or equal to 14h and less than or equal to 90h. The iuCRC Interval field shall be set to zero and ignored by the SCSI target device. The IUCRC Interval field shall be set to zero and ignored by the SCSI target device. The BIDI Direction field shall be set to zero and ignored by the receiving target device.
04h	Data	Sent by a SCSI target port device to indicate a SPI data information unit shall follow this SPI L_Q information unit. The Data Length field shall not be set to zero. For a bidirectional command, the direction of the SPI data information unit shall be indicated in the BIDI Direction field of the SPI L_Q information unit as defined in Table 51.
05h	Data stream	Sent by a SCSI target port device to indicate an unspecified number of SPI data stream information units shall follow this SPI L_Q information unit. The Data Length field shall not be set to zero. For a bidirectional command, the direction of the SPI data stream information units shall be indicated in the BIDI Direction field of the SPI L_Q information unit as defined in Table 51.
08h	Status	Sent by a SCSI target port device to indicate a SPI status information unit may follow this SPI L_Q information unit. A length of zero in the Data Length field shall indicate no SPI status information unit shall follow the SPI L_Q information unit (see Section 6.2.5). The iuCRC Interval field shall be set to zero and ignored by the SCSI target device. The BIDI Direction field shall be set to zero and ignored by the receiving SCSI initiator device.
F0H - FFh		vendor-specific.
All others		Reserved.

[2] Tag field is a 16-bit unsigned integer assigned by the application client and sent to the initiator in the Send SCSI Command request. See Section 4.4.1, Task attribute message overview and codes.

[3] The Logical Unit Number field specifies the address of the logical unit of the I_T_L_Q nexus for the current task. The structure of the Logical Unit Number field shall be as defined in the SCSI Architecture Model-4 standard, Section 4.11. If the addressed logical unit does not exist, the task manager shall follow the SCSI rules for selection of invalid logical units as defined in the SCSI Primary Commands - 4 (SPC-4) standard. The eight bytes in this field provide 4 levels of addressing. Bytes 4-5 address the first level, etc. to bytes 10-11, which address the fourth level. If LUN in SPI LQ information unit packet is X which is greater than 0x3F (>63), then the LUN in ID message sent back will be set to 0x3F (63). If LUN in SPI LQ information unit packet is X which is not greater than 0x3F (>63), then the LUN in ID message sent back will be set to X.

[4] The Data Length field contains the length in bytes of the following information units. For SPI data stream information units, the data length field contains the length in bytes of each SPI data stream information unit that follows (i.e., the total number of bytes transferred would equal the data length times the number of SPI data stream information units transferred). The data length shall not include any of the 4-byte iuCRC nor any transmitted pad bytes (e.g., a data length of 509 with a iuCRC interval of zero or greater

than 509 would transfer 509 bytes of data plus 3 bytes of pad plus 4 bytes of iuCRC for a total transfer of 516 bytes). The target shall not set the data length to a value that exceeds the maximum burst size as defined in the disconnect-reconnect page (see Seagate SCSI Command Reference Manual, Part number 100293068.)

[5] The BIDI Direction field determines the data direction if the command is a bidirectional command and the type code is data or data stream. The code values for the BIDI Direction field are defined below.

Table 51: BIDI Direction

Codes	Description
00b	A unidirectional command or a type code other than data or data stream (see Table 50).
01b	A bidirectional command transferring data from the SCSI initiator device to the SCSI target device.
10b	A bidirectional command transferring data from the SCSI target device to the SCSI initiator device.
11b	Reserved.

[6] The iuCRC Interval field contains the length in bytes of the data to be sent before a iuCRC is transferred. The iuCRC interval length shall not include the 4 byte iuCRC nor any transmitted pad bytes (e.g., an iuCRC interval length of 510 transfer 510 bytes of data plus 2 bytes of pad plus 4 bytes of iuCRC for a total transfer of 516 bytes). The iuCRC interval shall be a multiple of two (i.e., odd numbers are not allowed). If the iuCRC interval is equal to zero or is greater than or equal to the data length only one iuCRC shall occur at the end of the SPI information unit.

[7] The iuCRC field shall use the algorithm defined in Subsection 11.3 of the ANSI SCSI Parallel Interface (SPI-5).

6.2.3 SPI data information unit

The SPI data information unit (see Table 52) contains data.

The detection of a BUS FREE phase following a SPI data information unit by a SCSI initiator port shall be equivalent to the initiator receiving a DISCONNECT message.

The detection of a QAS REQUEST message following a SPI data information unit by a SCSI initiator port shall be equivalent to the initiator receiving a DISCONNECT message.

Table 52: SPI data information unit

Bit Byte	7	6	5	4	3	2	1	0
0	MSB			DATA [1]				
n								LSB
n+1	MSB							
n+2				iuCRC [2]				
n+3								
n+4								LSB

[1] The Data field may contain any type of information (e.g., parameter lists, mode pages, user data).

[2] The iuCRC field shall use the algorithm defined in Subsection 11.3 of ANSI SCSI Parallel Interface (SPI-5), T10/1525D. If the iuCRC Interval field of the SPI L_Q information unit contains a value greater than zero and less than the data length, then there is an iuCRC field at each iuCRC interval in addition to the iuCRC shown in this table. These additional iuCRC fields are not shown in this table.

6.2.4 SPI data stream information unit

The SPI data stream information unit (see Table 53) contains data.

All the SPI data stream information units transferred after a SPI L_Q information unit with a type of data stream shall be the size indicated in the Data Length field of the SPI L_Q information unit.

If the data transfer size is not a multiple of the data length, the SCSI target shall end the stream at a data length boundary and shall send a new SPI L_Q with a smaller data length to finish the data transfer. The new SPI L_Q may or may not be sent during the current physical connection.

During write streaming, the sequence of SPI data stream information units shall end with any change to the C/D, I/O, or MSG signals on a SPI data stream information unit boundary. If during write streaming SPI data stream information nits a SCSI initiator detects a REQ transition after transmitting the last iuCRC for a SPI data stream information unit that initiator shall transmit the next SPI data stream information unit.

During read streaming, the SCSI target shall end a sequence of SPI data stream information units by performing one of the following:

a. should assert the P_CRCA signal before the end of the current SPI data stream information unit boundary; or
b. may change the C/D, I/O, or MSG signals on a SPI data stream information unit boundary.

If, during the last SPI data stream information unit of a read stream, the P_CRCA signal was not asserted and a SCSI initiator detects a REQ transition after receiving the last iuCRC for a SPI data stream information unit, that initiator shall receive the next SPI data stream information unit. If, during the last SPI data stream information unit the P_CRCA signal was asserted and a SCSI initiator port detects a REQ transition after receiving the last iuCRC for a SPI data stream information unit, that SCSI initiator shall logically disconnect from the current I_T_L_Q nexus.

If, during a sequence of SPI data stream information units a SCSI initiator detects any change to the C/D, I/O, or MSG signals after transmitting or receiving the last iuCRC for a SPI data stream information unit, that initiator shall consider the current I/O process to be logically disconnected or in the case of detecting a BUS FREE phase or a MESSAGE IN phase to be physically disconnected.

The detection of a BUS FREE phase following a SPI data stream information unit by a SCSI initiator shall be equivalent to the initiator port receiving a DISCONNECT message.

The detection of a QAS REQUEST message following a SPI data stream information nit by a SCSI initiator shall be equivalent to the initiator receiving a DISCONNECT message.

To end a sequence of SPI data stream information units, a SCSI initiator may request a disconnect by establishing an attention condition. The initiator shall continue to transfer or receive data, pad bytes (if any), and iuCRCs until the target changes to the MESSAGE OUT phase.

During a sequence of SPI data stream information units the offset count is not required to go to zero at the boundary of any SPI data stream information unit if the next SPI information unit is a SPI data stream information unit.

Table 53: SPI status data stream information unit

Bit / Byte	7	6	5	4	3	2	1	0
0	MSB			DATA [1]				
n								LSB
n+1	MSB							
n+2				iuCRC [2]				
n+3								
n+4								LSB

[1] The Data field may contain any type of information (e.g., parameter lists, mode pages, user data).

[2] The iuCRC field shall use the algorithm defined in Subsection 11.3 of ANSI SCSI Parallel Interface (SPI-5). If the iuCRC Interval field of the SPI L_Q information unit contains a value greater than zero and less than the data length, then there is an iuCRC field at each iuCRC interval in addition to the iuCRC shown in this table. These additional iuCRC fields are not shown in this table.

6.2.5 SPI status information unit

The SPI status information unit (see Table 54) contains the completion status of the task indicated by the preceding SPI L_Q information unit. The target shall consider the SPI status information unit transmission to be successful when there is no attention condition on the transfer of the information unit.

If a task completes with a Good status, a SNSVALID bit of zero, and a RSPVALID bit of zero, then the target shall set the Data Length field in the SPI L_Q information unit (see Section 6.2.2) to zero.

Table 54: SPI status information unit

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved				Reserved for FCP [1]		SNSVALID [2]	RSPVALID [4]
3	STATUS [5]							
4	MSB	SENSE DATA LIST LENGTH (n-m) [3]						
7								LSB
8	MSB	PACKETIZED FAILURES LIST LENGTH [4]						
11								LSB
12	MSB	PACKETIZED FAILURES [4]						
m								LSB
1+m	MSB	SENSE DATA LIST [3]						
n								LSB
n+1	MSB	iuCRC [6]						
n+2								
n+3								
n+4								LSB

[1] Reserved for Fiber Channel Protocol.

[2] A Sense Data Valid (SNSVALID) bit of zero indicates the sense data list length shall be ignored and no sense data is provided. A SNSVALID bit of one indicates the Sense Data List Length field specifies the number of bytes in the Sense Data field.

[3] If sense data is provided, the Sense Data Valid (SNSVALID) bit shall be set to one and the Sense Data List Length field shall specify the number of bytes in the Sense Data field. The Sense Data List Length field shall only contain even lengths greater than zero and shall not be set to a value greater than 252. If no sense data is provided, the sense data valid bit shall be set to zero. The initiator shall ignore the Sense

Data List Length field and shall assume a length of zero.

[4] If packetized failure data is provided, the Packetized Failures Valid (RSPVALID) bit shall be set to one and the Packetized Failures List Length field shall specify the number of bytes in the Packetized Failures field. The Packetized Failures List Length field shall contain a length of 4. Other lengths are reserved for future standardization. If no packetized failure data is provided, the packetized failures valid bit shall be set to zero. The initiator shall ignore the Packetized Failures List Length field and shall assume a length of zero. The Packetized Failures field (see Table 55) contains information describing the packetized failures detected during the execution of a task. The Packetized Failures field shall contain valid information if the target detects any of the conditions described by the packetized failure code (see Table 56).

[5] The Status field contains the status of a task that completes. See Section 7.3, Table 64, for a list of status codes.

[6] The iuCRC field shall use the algorithm defined in Subsection 11.3, ANSI SCSI Parallel Interface (SPI-5), T10/1525D.

The Packetized Failures field (see Table 55) contains information describing the packetized failures detected during the execution of a task. The Packetized Failures field shall contain valid information if the target detects any of the conditions described by the packetized failure code (see Table 56).

Table 55: Packetized Failures field

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved							
3	PACKETIZED FAILURE CODE [1]							

[1] The Packetized Failures Code is defined in Table 56.

Table 56: Packetized Failures code

Codes	Description
00h	Indicates no failure or task management function complete.
01h	Reserved.
02h	SPI command information unit fields invalid.
03h	Reserved.
04h	The task management function not supported.
05h	The task management function failed.
06h-FFh	Reserved.

The Sense Data field contains the information specified by the SCSI Primary Commands-4 (SPC-4), T10/1731-D, for presentation by the REQUEST SENSE command (see Seagate SCSI Command Reference Manual, Part number 100293068). The proper sense data shall be presented when a SCSI status byte of Check Condition is presented as specified by the SCSI Primary Commands-4 (SPC-4), T10/1731-D and this manual.

The iuCRC field shall use the algorithm defined in Subsection 11.3, ANSI SCSI Parallel Interface (SPI-5), T10/1525D.

7.0 SCSI commands

This section defines the SCSI command structure and describes a typical SCSI bus procedure involving a command, status return, and message interchange.

The command structure defined herein provides for a contiguous set of logical blocks of data to be transferred across the interface. The number of logical data blocks to be transferred is defined in the command. Initiator commands to the drive are structured in accordance with the requirements imposed by the drive physical characteristics. These physical characteristics are reported to the initiator in response to an INQUIRY command.

A single command may transfer one or more logical blocks of data. The drive may disconnect, if allowed by the initiators, from the SCSI bus to allow activity by other SCSI devices while the drive performs operations within itself.

Upon command completion (which may be executed either successfully or unsuccessfully), the drive returns a status byte to the initiator. Since most error and exception conditions cannot be adequately described with a single status byte, one status code that can be sent as the status byte is called Check Condition. It indicates that additional information is available. The initiator may issue a REQUEST SENSE command to request the return of the additional information as part of the DATA IN phase of the command.

Future implementations of the SCSI protocol will contain an autosense feature (see Section 7.6.4.2) allowing the application client to request the automatic return of sense data. Fibre Channel protocol already has this feature in its packetized command structure.

7.1 Command implementation requirements

The first byte of any SCSI command contains an operation code as defined in this document. Three bits (bits 7-5) of the second byte of each SCSI command have historically been used to specify the logical unit if it is not specified using the Identify Message (see Section 4). These three bits are now shown as “Reserved” in the new Command data block format. The last byte of most SCSI commands contains a control field. The exception to this is the new variable length CDB which has the control field in the second byte, rather than the last.

7.1.1 Reserved

Reserved bits, bytes, fields, and code values are set aside for future standardization. Their use and interpretation may be specified by future revisions to this specification. A reserved bit, field, or byte shall be set to zero, or in accordance with a future revision to this specification. A drive that receives a reserved code value shall terminate the command with a CHECK CONDITION status and the Sense Key shall be set to ILLEGAL REQUEST. It shall also be acceptable for the drive to interpret the bit, field, byte, or code value in accordance with a future revision to this specification.

7.2 Command Descriptor Block (CDB)

A request by a SCSI initiator port to a drive is performed by sending a Command Descriptor Block (CDB) to the drive. For several commands, the request is accompanied by a list of parameters sent during the DATA OUT phase. The field uses shown in tables 57, 58, 59, 60, and most of those in 61 are used consistently by most commands. See the specific commands for detailed information.

The Command Descriptor Block always has an operation code as the first byte of the command. This is followed by command parameters (if any), and a control field.

For all commands, if there is an invalid parameter in the Command Descriptor Block, the drive shall terminate the command without altering the medium.

The format description for the Command Descriptor Block as supported by Seagate drives is shown in tables 57, 58, 59, 60, and 61.

7.2.1 Fixed and variable length Command Descriptor Block formats

For all commands, if there is an invalid parameter in the command descriptor block, the device server terminates the command without altering the medium.

Table 57 shows the typical format of a 6-byte CDB. Table 58 shows a typical format of a 10-byte CDB. Table 59 shows the typical format of a 12-byte CDB. Table 60 shows the typical format of a 16-byte CDB. Table 61 shows the typical format for a variable length CDB.

Table 57: Typical CDB for 6-byte commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE [1]							
1	Reserved			(MSB)				
2	LOGICAL BLOCK ADDRESS (if required) [3]							
3								(LSB)
4	TRANSFER LENGTH (if required) [4] PARAMETER LIST LENGTH (if required) [5] ALLOCATION LENGTH (if required) [6]							
5	CONTROL [7]							

See notes following Table 61.

Table 58: Typical CDB for 10-byte commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE [1]							
1	Reserved			SERVICE ACTION (if required) [2]				
2	(MSB)							
3	LOGICAL BLOCK ADDRESS (if required) [3]							
4								
5								(LSB)
6	Reserved							
7	(MSB)	TRANSFER LENGTH (if required) [4] PARAMETER LIST LENGTH (if required) [5] ALLOCATION LENGTH (if required [6]						
8								(LSB)
9	CONTROL [7]							

See notes following Table 61.

Table 59: Typical CDB for 12-byte commands

Byte \ Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE [1]							
1	Reserved			SERVICE ACTION (if required) [2]				
2	(MSB)	LOGICAL BLOCK ADDRESS (if required) [3]						
3								
4								
5								(LSB)
6	(MSB)	TRANSFER LENGTH (If required) [4] PARAMETER LIST LENGTH (if required) [5] ALLOCATION LENGTH (if required) [6]						
7								
8								
9								(LSB)
10	Reserved							
11	CONTROL [7]							

See notes following Table 61.

Table 60: Typical CDB for 16-byte commands

Byte \ Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE [1]							
1	Reserved			SERVICE ACTION (if required) [2]				
2	(MSB)	Logical Block Address (if required) [3]						
3								
4								
5								(LSB)
6	(MSB)	ADDITIONAL CDB DATA (if required) [9]						
7								
8								
9								(LSB)

Bit / Byte	7	6	5	4	3	2	1	0
10	(MSB)	TRANSFER LENGTH (If required) [4] PARAMETER LIST LENGTH (if required) [5] ALLOCATION LENGTH (if required) [6]						
11								
12								
13								(LSB)
14	Reserved							
15	CONTROL [7]							

See notes following Table 61.

Table 61: Typical variable length CDB

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh) [1]							
1	CONTROL [7]							
2	Reserved							
3	Reserved							
4	Reserved							
5	ENCRYPTION IDENTIFICATION [8]							
6	Reserved							
7	ADDITIONAL CDB LENGTH (n–7) [9]							
8	(MSB)	SERVICE ACTION [2]						
9								(LSB)
10 : n	SERVICE ACTION SPECIFIC FIELDS [10]							

Notes for Tables 57, 58, 59, 60, and 61.

[1] The Operation Code field of the Command Descriptor Block contains the code value indentifying the operation being requested by the CDB. The Operation Code provides for a possible 256 command operation codes. Details of the various commands with their operation codes are defined in Section 8.0 of this manual.

[2] Service Action. All typical CDB formats except the typical 6-byte format provide for a Service Action field containing a coded value identifying a function to be performed under the more general command function specified in the Operation Code field. While the Service Action field is defined for typical CDB formats, it is used as described in this clause only in those CDB formats that explicitly contain a Service Action field. When the specific field Service Action is not defined in a CDB format, the bits identified as the Service Action field in a typical CDB may be used for other purposes. More details appear on this in Section

8.0 where the details of each command are described.

[3] The Logical Block Address on logical units or within a partition on device volumes shall begin with block zero and be contiguous up to the last logical block on that logical unit or within that partition.

A six-byte command descriptor block contains a 21-bit Parameter List Length field. The 10-byte, the 12-byte and the 16-byte command descriptor blocks contain 32-bit Logical Block Address fields. Logical Block Address fields in additional parameter data have their length specified for each occurrence. See the specific command descriptions.

[4] The Transfer Length specifies the amount of data to be transferred, usually the number of blocks. For several commands the transfer length indicates the requested number of bytes to be sent as defined in the command description. For these commands the transfer length field may be identified by a different name. See the following descriptions and the individual command descriptions for further information.

Commands that use one byte for Transfer Length allow up to 256 blocks of data to be transferred by one command. A Transfer Length value of 1 to 255 indicates the number of blocks that shall be transferred. A value of zero indicates 256 blocks.

Commands that use two bytes for Transfer Length allow up to 65,535 blocks of data to be transferred by one command. In this case, a Transfer Length of zero indicates that no data transfer shall take place. A value of 1 to 65,535 indicates the number of blocks that shall be transferred.

For several commands more than two bytes are allocated for Transfer Length. Refer to the specific command description for further information.

The Transfer Length of the commands that are used to send a list of parameters to a drive is called the Parameter List Length. The Parameter List Length specifies the number of bytes sent during the DATA OUT phase.

The Transfer Length of the commands used to return sense data (e.g., REQUEST SENSE, INQUIRY, MODE SENSE, etc.) to a SCSI initiator port is called the Allocation Length. The Allocation Length specifies the number of bytes that the initiator has allocated for returned data. The drive shall terminate the DATA IN phase when Allocation Length bytes have been transferred or when all available data have been transferred to the initiator, whichever is less.

[5] The Parameter List Length field is used to specify the number of bytes sent from the Data-Out Buffer. This field is typically used in command descriptor blocks for parameters that are sent to a device server (e.g., mode parameters, diagnostic parameters, log parameters, etc.). A parameter length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

[6] The Allocation Length field specifies the maximum number of bytes that an application client has allocated for returned data. An allocation length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. The device server shall terminate transfers to the Data-In Buffer when allocation length bytes have been transferred or when all available data have been transferred, whichever is less. The allocation length is used to limit the maximum amount of data (e.g., sense data, mode data, log data, diagnostic data, etc.) returned to an application client. If the information being transferred to the Data-In Buffer includes fields containing counts of the number of bytes in some or all of the data, the contents of these fields shall not be altered to reflect the truncation, if any, that results from an insufficient allocation length value, unless the standard that describes the Data-In Buffer format specifically states otherwise. If the amount of information to be transferred exceeds the maximum value that may be specified in the Allocation Length field, the device server shall transfer no data and return a CHECK CONDITION status; the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to Invalid Field in CDB.

[7] Only the Control fields have consistently defined meanings across all commands. The Control field is the last byte of every command descriptor block, except for the variable length CDB format. The Control field is defined in Table 62.

Table 62: Control field

Bit Byte	7	6	5	4	3	2	1	0
Last	VENDOR-SPECIFIC		Reserved			NACA [a]	Obsolete [c]	LINK [b] 0 or 1

[a] The Normal Auto Contingent Allegiance (NACA) bit is used to control the rules for handling an Auto Contingent Allegiance (ACA) condition caused by the command. Section 7.6.1.1 specifies the actions to be taken by the logical unit in response to an ACA condition for NACA bit values of one or zero. All drives implement support for the NACA value of zero and may support the NACA value of one. The ability to support an NACA value of one is indicated in standard INQUIRY data. See Seagate SCSI Command Reference Manual, Part number 100293068.

If the NACA bit is set to a value that is not supported, the drive completes the command with a status of Check Condition and a sense key of ILLEGAL REQUEST. The procedure for handling the resulting ACA condition operates in accordance with the supported bit value.

[b] The Link bit allows the drive to continue the task across multiple commands. Support for the Link bit is a logical unit option. A Link bit of one indicates that the initiator requests continuation of the task across two or more SCSI commands. If the Link bit is one and if the command completes successfully, a drive that supports the Link bit continues the task and returns a status of Intermediate or Intermediate–Condition Met and a service response of LINKED COMMAND COMPLETE message (see Section 4.3.6). The drive completes the command with a status of Check Condition and a sense key of ILLEGAL REQUEST if the Link bit is set to one and the drive does not support linked commands.

[c] Obsolete bit. Bit 0 provides an obsolete way to request interrupts between linked commands. If bit 0 is equal to one, device servers not implementing the obsolete capability (Link bit was formerly in this bit position and Flag bit was in bit 1 position) terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

[8] The Encryption Identification field indicates whether CDB bytes 8 through n and/or the data bytes are encrypted. The value also indicates which encryption key to use for decryption. A value of zero indicates no encryption. All other values are reserved.

[9] The Additional CDB Length field indicates the number of additional CDB bytes. This value in the Additional CDB Length field shall be a multiple of 4.

[10] Service Action specific fields. The Service Action field indicates the action being requested by the application client. The Service Action field is required in the variable length CDB format and is described in Note [2]. Each service action code description defines a number of service action specific fields that are needed for that service action. If the device server detects an error during decryption of encrypted CDB bytes, it shall return CHECK CONDITION status with an additional sense code of CDB Decryption Error. If the device server detects an error during decryption of encrypted data bytes, it shall return CHECK CONDITION status with an additional sense code of Data Decryption Error.

7.3 Status

A Status byte shall be sent from the target to the initiator during the STATUS phase at the termination of each command as specified in Tables 63 and 64 unless the command is cleared by one of the following conditions:

1. an Abort message
2. a Bus Device Reset message
3. a hard reset condition
4. an unexpected Bus Free condition (see Section 3.1.1)
5. an ABORT TASK message
6. a CLEAR TASK SET message

Table 63: Status byte

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved		STATUS BYTE CODE					Reserved

Table 64: Status byte code bit values

Status byte	Status represented	Task Ended
00h	Good	Yes
02h	Check Condition	Yes
04h	Condition Met/Good	Yes
08h	Busy	Yes
10h	Intermediate/Good	No
14h	Intermediate/Condition Met	No
18h	Reservation Conflict	Yes
22h	Obsolete [2]	Yes
28h	Queue Full/Task Set Full [1]	Yes
30h	ACA Active	Yes
40h	Task Aborted	Yes
All other codes	Reserved	

[1] What was formerly called a “Command Queue” is now called a “Task Set.”

[2] Formerly “Command Terminated.”

A description of the status byte codes is given below.

Good. This status indicates that the Device Server has successfully completed the task.

Check Condition. This status indicates that an auto contingent allegiance or contingent allegiance condition has occurred (see Section 7.6.1). Optionally, autosense data may be delivered (see Section 7.6.4.2).

Condition Met. This status shall be returned whenever the requested operation specified by an unlinked command is satisfied (see the PREFETCH commands in ANSI SCSI Block Commands-2, T10/1417D).

Busy. This status indicates that the logical unit is busy. This status shall be returned whenever a logical unit is unable to accept a command from an otherwise acceptable initiator (i.e., no reservation conflicts). The recommended initiator recovery action is to issue the command again at a later time.

Intermediate. This status or Intermediate-Condition Met shall be returned for each successfully completed command in a series of linked commands (except the last command), unless the command is terminated with Check Condition, Reservation Conflict, Task Set Full, Busy status. If Intermediate or Intermediate-Condition Met status is not returned, the series of linked commands is terminated and the task is ended.

Intermediate–Condition Met. This status is returned whenever the operation requested by a linked command is satisfied (see the PREFETCH commands in ANSI SCSI Block Commands-2, T10/1417D), unless the command is terminated with Check Condition, Reservation Conflict, Task Set Full, Busy status. If Intermediate or Intermediate-Condition Met status is not returned, the series of linked commands is terminated and the task is ended.

Reservation Conflict. This status shall be returned whenever a SCSI initiator port attempts to access a logical unit or an element of a logical unit that is reserved with a conflicting reservation type for another SCSI initiator. (See the RESERVE, RELEASE, PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands in ANSI SCSI Primary Commands - 4, T10/1731-D). The recommended initiator recovery action is to issue the command again at a later time. Removing a persistent reservation belonging to a failing initiator may require the execution of a PERSISTENT RESERVE OUT command with the Preempt or Preempt and Clear actions (see the SPC-4 standard, T10/1731-D).

Task Set Full. This status shall be implemented if the logical unit supports the creation of tagged tasks (see Section 7.7). This status shall not be implemented if the logical unit does not support the creation of tagged tasks.

When the logical unit has at least one task in the task set for a SCSI initiator port and a lack of task set resources prevents entering a newly received tagged task from that initiator in the task set, Task Set Full shall be returned. When the logical unit has no task in the task set for a SCSI initiator port and a lack of task set resources prevents entering a newly received tagged task from that initiator in the task set, Busy should be returned.

When the logical unit has at least one task in the task set and a lack of task set resources prevents entering a newly received untagged task in the task, Busy should be returned.

The logical unit should allow at least one queued command for each supported initiator that has identified itself to the target by a protocol specific procedure or by the successful transmission of a command.

ACA Active. This status shall be returned when an auto contingent allegiance exists within a task set and a SCSI initiator port issues a command for that task set when at least one of the following is true:

a. There is a task with the ACA attribute in the task set;
b. The initiator issuing the command did not cause the ACA condition; or
c. The task created to execute the command did not have the ACA attribute and the NACA bit was set to one in the CDB Control byte of the faulting command (see Section 7.6.1).

The initiator may reissue the command after the ACA condition has been cleared.

Task Aborted. This status shall be returned when a task is aborted by another SCSI Initiator and the Control mode page TAS bit is set to one.

7.3.1 Status precedence

If more than one condition applies to a completed task, the report of a Busy, Reservation Conflict, ACA Active or Task Set Full status shall take precedence over the return of any other status for that task.

7.4 Command examples

7.4.1 Single command example

A typical operation on the SCSI bus is likely to include a single READ command to a peripheral device such as the drive. This operation is described in detail starting with a request from the initiator. This example assumes that no linked commands and no malfunctions or errors occur and is illustrated in Figure 19.

The initiator has active pointers and a set of stored pointers representing active disconnected SCSI devices (a SCSI initiator port without disconnect capability does not require stored pointers). The initiator sets up the active pointers for the operation requested, arbitrates for the SCSI bus, and selects the drive. Once this process is completed, the drive assumes control of the operation.

The drive obtains the command from the initiator (in this case a READ command). The drive interprets the command and executes it. For this command, the drive reads the requested data from the Disc Media and sends this data to the initiator. After sending the read data to the initiator, the drive sends a status byte to the initiator. To end the operation, the drive sends a Command Complete message to the initiator and then goes to the Bus Free state.

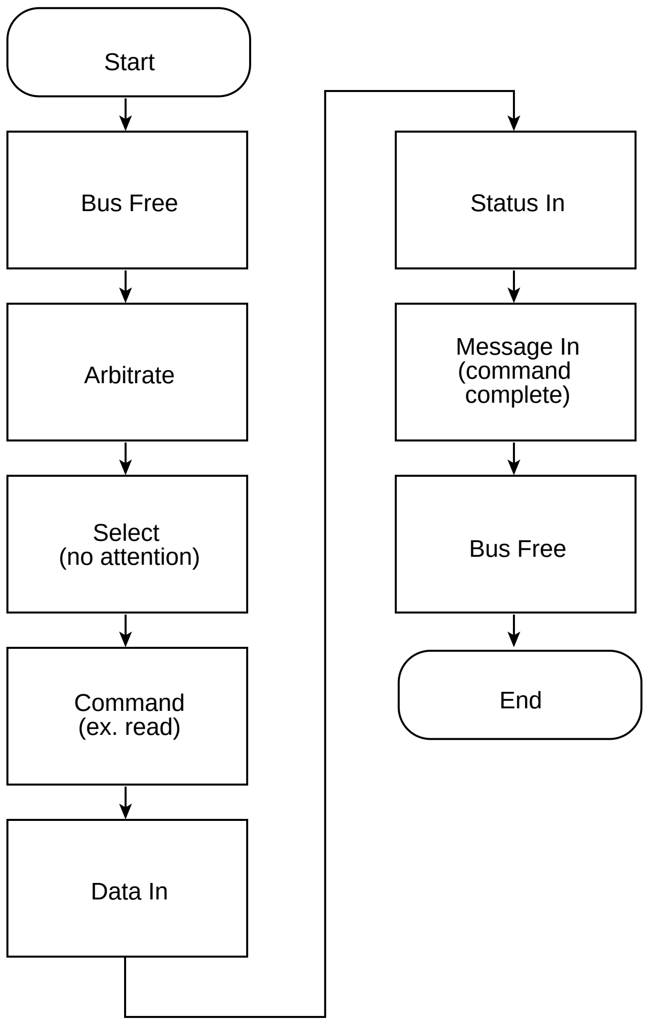


Figure 19. Single command example

7.4.2 Disconnect example

In the single command example, the length of time necessary to obtain the data may require a time consuming physical seek. In order to improve system throughput, the drive may disconnect from the initiator, freeing the SCSI bus to allow other requests to be sent to other SCSI devices. To do this, the initiator must be reselectable and capable of restoring the pointers upon reconnection. The drive must be capable of arbitrating for the SCSI bus and reselecting the initiator. See Figure 20.

After the drive has received the READ command (and has determined that there will be a delay), it disconnects by sending a DISCONNECT message and releasing BSY (goes to Bus Free state).

When the data is ready to be transferred, the drive reconnects to the initiator, the initiator restores the pointers to their most recently saved values (which in this case are the initial values), and the drive continues (as in the single command example) to finish the operation. The initiator recognizes that the operation is complete when a Command Complete message is received.

If the drive elects to disconnect after transferring part of the data (e.g., while crossing a cylinder boundary), it sends a Save Data Pointer message and a DISCONNECT message to the initiator and then disconnects. When reconnection is completed, the initiator restores the current data pointer to the value it was immediately before the Save Data Pointer message.

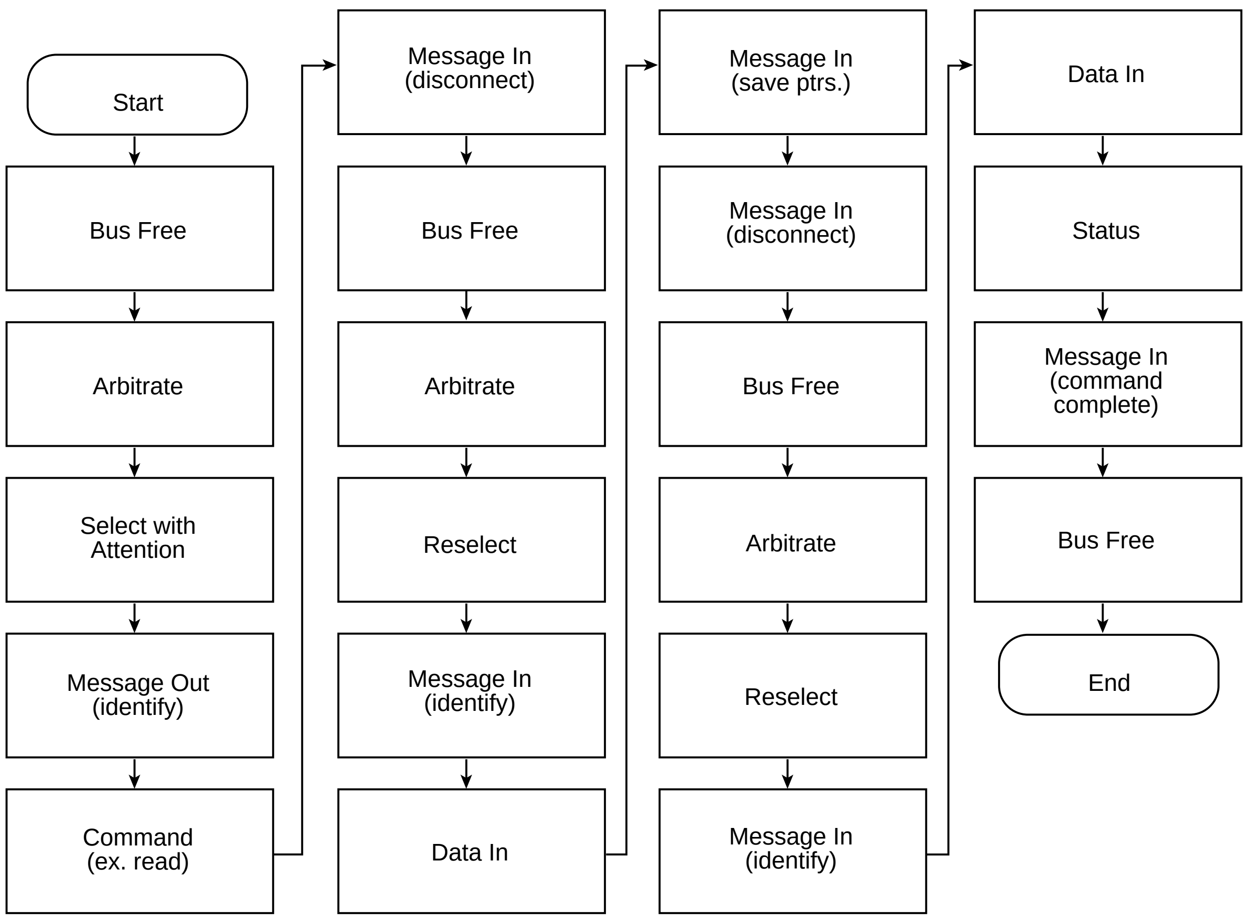


Figure 20. Disconnect example

7.5 Timing examples

Times (T00 through T35) necessary to define performance are listed in the individual drive's Product Manual, in the "Disc drive SCSI timing" section. For timing waveforms to define these times, refer to the SCSI Parallel Interface-5 specification (SPI-5).

7.6 Command processing considerations and exception conditions

The following clauses describe some exception conditions and errors associated with command processing and the sequencing of commands.

7.6.1 Auto Contingent Allegiance or Contingent Allegiance

The auto contingent allegiance (NACA=1, see Section 7.2.1, note [7], Table 62) or contingent allegiance (NACA=0) condition shall exist within the task set when the logical unit completes a command by returning a CHECK CONDITION status (see Section 7.3).

7.6.1.1 Logical Unit response to Auto Contingent Allegiance or Contingent Allegiance

The auto contingent allegiance (NACA=1, see Section 7.2.1, note [7], Table 62) or contingent allegiance (NACA=0) condition shall not cross task set boundaries and shall be preserved until it is cleared as described in Section 7.6.1.2. If requested by the application client and supported by the protocol and logical unit, sense data shall be returned as described in Section 7.6.4.2.

Note. The SCSI-2 contingent allegiance condition has had an alternate added and the extended contingent allegiance condition has been replaced in SCSI-3 (now called merely "SCSI") by auto contingent allegiance in conjunction with the NACA bit.

Note. If the SCSI-3 protocol does not enforce synchronization of client/server states, there may be a time delay between the occurrence of the auto contingent allegiance or contingent allegiance condition and the time at which the initiator becomes aware of the condition.

After sending status and a service response of Task Complete, the logical unit shall modify the state of all tasks in the faulted task set as described in SCSI Architecture Model-4 (SAM-4), T10/1683-D, "Task Set Management."

A task created by the faulted initiator while the auto contingent allegiance condition is in effect may be entered into the faulted task set under the conditions described below.

As described in Section 7.2.1, note [7], Table 62, the setting of the NACA bit in the Control byte of the faulting command CDB determines the rules that apply to an ACA or CA condition caused by that command. If the NACA bit was set to zero the SCSI-2 contingent allegiance rules shall apply.

If the NACA bit was set to one in the Control byte of the faulting command, then a new task created by the faulted initiator while the ACA condition is in effect shall not be entered into the faulted task set unless all of the following conditions are true:

a. The task has the ACA attribute; and
b. No other task from the faulted initiator having the ACA attribute is in the task set.

If the task is from the faulted initiator and any of the conditions listed above are not met, the newly created task shall not be entered into the task set and shall be completed with a status of ACA Active.

If a task having the ACA attribute is received and no auto contingent allegiance condition is in effect for the task set or if the NACA bit was set to zero in the CDB for the faulting command (i.e., a contingent allegiance condition is in effect), then the ACA task shall be completed with a status of Check Condition. The sense key shall be set to ILLEGAL REQUEST with an additional sense code of Invalid Message Error. As noted in Section 7.2.1, note [7], Table 62, any existing contingent allegiance condition shall be cleared and a new auto contingent allegiance (NACA=1) or contingent allegiance (NACA=0) condition shall be established.

The handling of tasks created by initiators other than the faulted initiator depends on the value in the TST field in the Control mode page (see ANSI SCSI Primary Commands-4, T10/1731-D).

If TST=000b, tasks created by other initiators while the ACA or CA condition is in effect shall not be entered into the faulted task set (except for a PERSISTENT RESERVE command with a Preempt and Clear action as described in Section 7.2.1, note [7], Table 62). Tasks rejected from the task set due to the presence of an ACA or CA condition shall be completed with a status of ACA Active (if NACA=1 in the new command's CDB Control byte, see Section 7.2.1, note [7], Table 62) or Busy (if NACA=0).

If TST=001b, tasks created by one initiator shall not be rejected based on an ACA or CA condition in effect for another initiator. Only ACA or CA condition for the sending initiator (as well as other task set management considerations described in SCSI Architecture Model-4, T10/1157D, clause 7, "Task Set Management.") shall affect acceptance into the task set or rejection for a task from that initiator.

7.6.1.2 Clearing an Auto Contingent Allegiance condition

If the NACA bit is set to zero in the Control byte of the faulting command, then the SCSI-2 rules for clearing contingent allegiance shall apply. In addition, the logical unit shall clear the associated contingent allegiance condition upon sending sense data by means of the autosense mechanism described in Section 7.6.4.2.

While the SCSI rules for clearing the ACA condition are in effect, a logical unit that supports the Clear ACA task management function shall ignore all Clear ACA requests and shall return a service response of Function Complete (see SAM-4).

If the logical unit accepts a value of one for the NACA bit and this bit was set to one in the Control byte of the faulting command, then the SCSI-2 rules for clearing a contingent allegiance condition shall not apply. In this case, the ACA condition shall only be cleared:

a. As the result of a power on or a logical unit reset (see ANSI SAM-4 document T10/1683-D);
b. Through a Clear ACA task management function issued by the faulting initiator as described in ANSI SAM-4 document T10/1683-D;
c. Through a Preempt and Clear action of a PERSISTENT RESERVE OUT command that clears the tasks of the faulting initiator (see the ANSI SCSI Primary Commands-4);
d. A command with the ACA attribute terminates with a CHECK CONDITION status. The state of all tasks in the task set when an auto contingent allegiance condition is cleared shall be modified as described in ANSI document T10/1157D, clause 7, "Task Set Management."

7.6.2 Overlapped commands

An overlapped command occurs when an application client reuses a Task Address (see Glossary) in a new command before a previous task to which that address was assigned completes its task lifetime as described in SCSI Architecture Model-4 (SAM-4), "Task and Command Lifetimes."

Each system that implements a SCSI protocol standard shall specify whether or not a logical unit is required to detect overlapped commands. A logical unit that detects an overlapped command shall abort all tasks for the initiator in the task set and shall return CHECK CONDITION status for that command. If the overlapped command condition was caused by an untagged task or a tagged task with a tag value exceeding FFh, then the sense key shall be set to Aborted Command and the additional sense code shall be set to Overlapped Commands Attempted. Otherwise, an additional sense code of Tagged Overlapped Tasks shall be returned with the additional sense code qualifier byte set to the value of the duplicate tag.

Note. An overlapped command may be indicative of a serious error and, if not detected, could result in corrupted data. This is considered a catastrophic failure on the part of the initiator. Therefore, vendor-specific error recovery procedures may be required to guarantee the data integrity on the medium. The target logical unit may return additional sense data to aid in this error recovery procedure (e.g., sequential-access devices may return the residue of blocks remaining to be written or read at the time the second command was received).

Note. Some logical units may not detect an overlapped command until after the command descriptor block has been received.

7.6.3 Incorrect logical unit selection

The target's response to an incorrect logical unit identifier is described in the following paragraphs.

The logical unit identifier may be incorrect because:

a. The target does not support the logical unit (e.g., some targets support only one peripheral device). In response to any other command except REQUEST SENSE and INQUIRY, the target shall terminate the command with CHECK CONDITION status. Sense data shall be set to the values specified for the REQUEST SENSE command in item b below;
b. The target supports the logical unit, but the peripheral device is not currently attached to the target. In response to an INQUIRY command the target shall return the INQUIRY data with the peripheral qualifier set to the value required in the SPC-4 standard. In response to a REQUEST SENSE command, the target shall return sense data. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to Logical Unit Not Supported.

   In response to any other command except REQUEST SENSE and INQUIRY, the target shall terminate the command with CHECK CONDITION status. Sense data shall be set to the values specified for the REQUEST SENSE command above;
c. The target supports the logical unit and the peripheral device is attached, but not operational. In response to an INQUIRY command the target shall return the INQUIRY data with the peripheral qualifier set to the value required in Seagate SCSI Command Reference Manual, Part number 100293068. In response to REQUEST SENSE, the target shall return sense data.

   The target's response to any command other than INQUIRY and REQUEST SENSE is vendor-specific; or
d. The target supports the logical unit but is incapable of determining if the peripheral device is attached or is not operational when it is not ready.

   In response to an INQUIRY command the target shall return the INQUIRY data with the peripheral qualifier set to the value specified in Seagate SCSI Command Reference Manual, Part number 100293068. In response to a REQUEST SENSE command the target shall return the REQUEST SENSE data with a sense key of No Sense unless an auto contingent allegiance exists. The target's response to any other command is vendor-specific.

7.6.4 Sense data

Sense data shall be made available by the logical unit in the event a command completes with a CHECK CONDITION status or other conditions. The format, content and conditions under which sense data shall be prepared by the logical unit are specified in this manual, the SPC-4 standard, and other applicable SCSI protocol standards.

Sense data shall be preserved by the logical unit for the initiator until it is transferred by one of the methods listed below or until another task from that initiator is entered into the task set.

The sense data may be transferred to the initiator through any of the following methods:

a. The REQUEST SENSE command specified in Seagate SCSI Command Reference Manual, Part number 100293068;
b. An asynchronous event report; or
c. Autosense delivery.

The following clauses describe the last two transfer methods.

7.6.4.1 Asynchronous Event Reporting

Asynchronous Event Reporting is used by a logical unit to signal another device that an asynchronous event has occurred. The mechanism automatically returns sense data associated with the event. Support for asynchronous event reporting is a logical unit option.

The control mode page contains parameters affecting the use of asynchronous event reporting (see Seagate SCSI Command Reference Manual, Part number 100293068).

Asynchronous Event Reporting is used to signal a device that one of the four events listed below has occurred:

a. an error condition was encountered after command completion;
b. a newly initialized device is available;
c. some other type of unit attention condition has occurred; or
d. an asynchronous event has occurred.

An example of the first case above occurs in a device that implements a write cache. If the target is unable to write cache data to the medium, it may use an asynchronous event report to inform the initiator of the failure.

An example of the second case above is a logical unit that generates an asynchronous event report, following a power-on cycle, to notify other SCSI devices that it is ready to accept I/O commands.

Sense data accompanying the report identifies the condition (see Section 7.6.4).

An error condition or unit attention condition shall be reported to a specific initiator once per occurrence of the event causing it. The logical unit may choose to use an asynchronous event report or to return CHECK CONDITION status on a subsequent command, but not both. Notification of an error condition encountered after command completion shall be returned only to the initiator that sent the affected task or tasks.

Asynchronous event reports may be used to notify devices that a system resource has become available. If a logical unit uses this method of reporting, the sense key in the AER sense data shall be set to Unit Attention.

7.6.4.2 Autosense

Autosense is the automatic return of sense data to the application client coincident with the completion of an SCSI command under the conditions described below. The return of sense data in this way is equivalent to an explicit command from the application client requesting sense data immediately after being notified that an ACA condition has occurred. Inclusion of autosense support in an SCSI protocol standard is optional.

An application client may request autosense service for any SCSI command. If supported by the protocol and logical unit and requested by the application client, the device server shall only return sense data in this manner coincident with the completion of a command with a status of Check Condition. After autosense data is sent, the sense data and the CA (NACA=0), if any, shall then be cleared. Autosense shall not affect ACA (NACA=1), see Section 7.6.1.

Protocol standards that support autosense shall require an autosense implementation to:

a. Notify the logical unit when autosense data has been requested for a command; and
b. Inform the application client when autosense data has been returned upon command completion (see Section 7.0).

It is not an error for the application client to request the automatic return of sense data when autosense is not supported by the SCSI protocol or logical unit implementation. If the application client requested the return of sense data through the autosense facility and the protocol service layer does not support this feature, then the confirmation returned by the initiator's service delivery port should indicate that no sense data was returned. If the protocol service layer supports autosense but the logical unit does not, then the target should indicate that no sense data was returned. In either case, sense information shall be preserved and the application client may issue a command to retrieve it.

7.6.5 Unexpected RESELECTION phase

An unexpected RESELECTION phase occurs if a SCSI target port attempts to do a physical reconnect to a task for which a nexus does not exist. a SCSI initiator port should respond to an unexpected RESELECTION phase by sending an ABORT TASK message.

7.6.6 Unit Attention condition

The drive sets up the Unit Attention condition when it stores (within itself) a Unit Attention condition flag for each device on the SCSI bus having a SCSI initiator port relationship with the drive, and this Unit Attention condition persists for each initiator until the condition is cleared (flag negated) by each initiator individually. The Unit Attention condition results when one of the following events occur:

a. A power-on sequence occurs.
b. A reset is generated internally by the drive (caused by a power glitch).
c. A Bus Device Reset message causes the drive to reset itself.
d. The Reset I/O line resets the drive.
e. a SCSI initiator port changes one or more of the mode select parameters in the drive (these changes could affect one or more of the other initiators).
f. The inquiry data has been changed.
g. The mode parameters in effect for a SCSI initiator port have been restored from non-volatile memory.
h. An event occurs that requires the attention of the initiator.
i. A Clear Queue message is received.
j. The Log parameters are changed. Unit Attention condition is posted for all initiators in the system other than the one that changed the Log Parameters.
k. Tasks for this initiator were changed by another initiator.
l. The version or level of microcode has changed.
m. The logical unit inventory has been changed (not generally applicable to Seagate disc drives); or
n. A change in the condition of a synchronized spindle occurred (not generally applicable to Seagate disc drives).

The Unit Attention Parameters page (page 00h, bit 4 of byte 2) of the MODE SELECT command controls whether or not a CHECK CONDITION status is to be reported to affected initiators when a Unit Attention condition exists (see Seagate SCSI Command Reference Manual, Part number 100293068).

Logical units may queue unit attention conditions. After the first unit attention condition is cleared, another unit attention condition may exist (for example, a power on condition followed by a microcode change condition).

The Unit Attention condition for a particular initiator is cleared when that initiator does one of the following:

a. It sends a REQUEST SENSE command; or
b. It sends any other legitimate command, with the exception of the INQUIRY command. The INQUIRY command does not clear the Unit Attention condition.

When a Unit Attention condition flag is stored in the drive for a SCSI initiator port, the commands that initiator issues to the drive operate as described in the following paragraphs.

If a SCSI initiator port sends an INQUIRY command to the drive when the drive has stored a Unit Attention condition flag for that initiator (before the drive generates the auto contingent allegiance condition), the drive shall perform the INQUIRY command and shall not clear the Unit Attention condition.

If a SCSI initiator port sends a REQUEST SENSE command to the drive when a Unit Attention condition flag is stored for that initiator (before the drive generates the auto contingent allegiance condition), the drive shall discard any pending sense data, report the Unit Attention sense key, and clear the Unit Attention condition (negate the flag) for that initiator.

If a SCSI initiator port issues a command other than INQUIRY or REQUEST SENSE while a Unit Attention condition flag is stored for that initiator, the drive may or may not perform the command and report CHECK CONDITION status, depending on whether or not the Unit Attention bit is zero or one in the Unit Attention Mode Parameters page (Page 00h, bit 4 of byte 2). See Seagate SCSI Command Reference Manual, Part number 100293068. If a REQUEST SENSE is issued next, the Unit Attention condition is reported and cleared (flag negated) as noted in the preceding paragraph. If another command other than REQUEST SENSE or INQUIRY is issued instead, the drive shall perform the command and return the appropriate status. The Unit Attention condition for the subject initiator is cleared (flag negated) and the sense data and flag indicating there has been a Unit Attention condition are lost.

7.6.7 Target hard reset

a SCSI target port hard reset is a SCSI target port response to a SCSI target port Reset task management request (see SCSI Architecture Model-4, T10/1157D), or a reset event within the service delivery subsystem. The definition of target reset events is protocol and interconnect specific. Each SCSI product standard shall specify the response to a SCSI target port reset event including the conditions under which a SCSI target port hard reset shall be executed.

To execute a hard reset, a SCSI target port shall initiate a logical unit reset for all attached logical units as described in Section 7.6.8.

7.6.8 Logical unit reset

A logical unit reset is a response to a Logical Unit Reset task management request (see SCSI Architecture Model-4, T10/1157D), or some other logical unit reset event, such as a SCSI target port hard reset (see Section 7.6.7). The definition of such events may be device-specific or dependent on the protocol and interconnect. Each appropriate SCSI standard shall specify the conditions under which a logical unit reset shall be executed.

To execute a logical unit reset the logical unit shall:

a. Abort all tasks in its task set;
b. Clear an auto contingent allegiance condition, if one is present;
c. Release all SCSI device reservations;
d. Return the device's operating mode to the appropriate initial conditions, similar to those conditions that would be found following device power-on. The mode select conditions (see SCSI Primary Commands-4 (SPC-4), T10/1731-D) shall be restored to their saved values if saved values have been established. Mode select conditions for which no saved values have been established shall be returned to their default values;
e. Set a Unit Attention condition (see Section 7.6.6); and
f. Initiate a logical unit reset for all nested logical units.

In addition to the above, the logical unit shall execute any additional functions required by the applicable standards.

7.7 Queued tasks (formerly "queued I/O processes")

Queuing of tasks allows a drive to accept multiple commands for execution at a later time.

There are two methods for implementation of queuing: tagged and untagged. Tagged task queuing allows the drive to accept multiple commands from each initiator. Untagged task queuing allows the drive to accept one command from each initiator. Drives that have SCSI-2 implementation support tagged queuing while in SCSI-1 or SCSI-2 mode. They can use untagged task queuing mode if the initiator does not send task queue tag messages.

Initiators may add or delete commands to the queue for the drive within the limitations specified in this specification. When adding a command, the initiator may specify fixed order of execution, allow the drive to define the order of execution, or specify that the command is to be executed next. See glossary in Section 1.2.1 for terminology definitions when reading the following explanations.

7.7.1 Untagged task queuing

Untagged task queuing allows the drive to accept a command from a SCSI initiator port while a command from another initiator is being executed. Only one command for each I T L nexus may be accepted at a time.

A new task may be initiated any time the BUS FREE phase exists even if another task from a different initiator is being executed. If the disconnect privilege is not granted, the drive returns Busy status to the new task.

The I T L nexus specifies the relationship so that the drive can always reconnect to the initiator to restore the pointers for task as long as only one command per I T L nexus is issued. It is the responsibility of the initiator to assure that only one command is issued at any time.

7.7.2 Tagged task queuing

Tagged task queuing allows a drive to accept multiple commands from the same or different initiators until the drive's task queue is full. A new task may be initiated any time the BUS FREE phase exists, if the disconnect privilege is granted. If the disconnect privilege is not granted for a tagged command, the drive returns Busy status to the new task.

The Task Queue Tag messages (see Section 4.4) allow the initiator to establish a unique I T L Q nexus to identify each task. Each task may be a command or a set of linked commands with a unique queue tag.

The I T L Q nexus allows the target to reconnect to the desired task and the initiator to restore the correct set of pointers. a SCSI initiator port may have several task ongoing to the same or different logical unit as long as each has a unique nexus.

If only Simple Task Queue Tag messages are used, the drive may execute the commands in any order that is deemed desirable within the constraints of the queue management algorithm specified in the Control Mode page (see Seagate SCSI Command Reference Manual, Part number 100293068). The command ordering is done by the drive to meet its performance and functional goals. The algorithm used by the drive attempts to achieve certain drive or system performance goals established in the drive firmware for the queued commands and guarantee that all commands will be executed. One possible goal would be to minimize seek times, but there could be others, possibly designed to meet some special system need. Commands from other initiators are also executed in an order selected in the same manner. The drive uses the Simple Task Queue Tag when reconnecting to the initiator.

If Ordered Task Queue Tag messages are used, the drive executes the commands in the order received with respect to other commands received with Ordered Task Queue Tag messages. All commands received with a Simple Task Queue Tag message prior to a command received with an Ordered Task Queue Tag message, regardless of initiator, are executed before that command with the Ordered Task Queue Tag message. All commands received with a Simple Task Queue Tag message after a command received with an Ordered Task Queue Tag message, regardless of initiator, are executed after that command with the Ordered Task Queue Tag message.

A command received with a Head of Task Queue Tag message is placed first in the queue, to be executed next. A command received with a Head of Task Queue Tag message does not suspend a task for which the drive has begun execution. Consecutive commands received with Head of Task Queue Tag messages are executed in a last-in-first-out order.

The Control Mode page specifies alternative queue management algorithms with additional rules on the order of execution of commands (see Seagate SCSI Command Reference Manual, Part number 100293068).

A task received from a SCSI initiator port without a task queue tag message while there are any tagged I/O commands in the command queue from that initiator is an incorrect initiator connection, unless there is a contingent allegiance condition. A task received from a SCSI initiator port with a task queue tag message while there is an untagged command in the command queue from that initiator is also an incorrect initiator connection. In either of these cases, the drive removes all commands in the queue from that initiator, aborts the command in process if it is from that initiator, and sets the Sense Key to Aborted Command and the Sense Code to Overlapped Commands Attempted.

The RESERVE and RELEASE commands should be sent with an Ordered Task Queue Tag message. Use of the Head of Task Queue Tag message with these commands could result in reservation conflicts with previously issued commands.

The TEST UNIT READY and INQUIRY commands are often sent with a Head of Task Queue Tag message, since the information returned is either available or has no effect on the condition of the drive.

The drive recovery option, is to continue execution of commands in the queue after the contingent allegiance condition has cleared. The drive returns Busy status to all other initiators while the contingent allegiance condition exists. During this time all commands in the queue are suspended. All commands used for recovery operations are untagged commands.

Deferred errors are normally related to a command that has already completed. As such, there is no attempt to return the queue tag value assigned to the original command.

7.8 Parameter rounding

Certain parameters sent to a SCSI target port with various commands contain a range of values. Targets may choose to implement only selected values from this range. When the target receives a value that it does not support, it either rejects the command (CHECK CONDITION status with Illegal Request Sense key) or it rounds the value received to a supported value. The target shall reject unsupported values unless rounding is permitted in the description of the parameter.

Rounding of parameter values, when permitted (Rounding is enabled by MODE SELECT command, page code 00h, byte 2, bit 2) shall be performed as follows:

a SCSI target port that receives a parameter value that is not an exact supported value shall adjust the value to one that it supports and shall return CHECK CONDITION status with a sense key of Recovered Error. The additional sense code shall be set to Rounded Parameter. The initiator is responsible to issue an appropriate command to learn what value the target has selected.

Implementors Note: Generally, the target should adjust maximum-value fields down to the next lower supported value than the one specified by the initiator. Minimum-value fields should be rounded up to the next higher supported value than the one specified by the initiator. In some cases, the type of rounding (up or down) is explicitly specified in the description of the parameter.

7.9 Programmable operating definition

Some applications require that the operating definition of a logical unit be modified to meet the special requirements of a particular initiator. The program-controlled modification of the operating definition is provided to allow operating systems to change the operating definition of a more recently developed target to one which is more compatible with the operating system. This ability requires that the system comply with the low-level hardware definitions of SCSI-2.

The parameters that can be changed by modifying the operating definition of a logical unit include the vendor identification, the device type, the device model, the SCSI compliance level, the SCSI specification level, the command set, and other parameters. The low-level hardware parameters including signal timing and parity definitions cannot be changed by modifying the operating definition. The present operating definition of a logical unit with respect to a SCSI initiator port can be determined at any time by execution of an INQUIRY command. In some vendor-specific cases, it may also be necessary to perform other commands including MODE SENSE and READ CAPACITY.

The more recent Seagate drives do not support the Change Operating Definition command (see individual drive’s Product Manual).

Each logical unit begins at a particular operating definition. If the logical unit supports the CHANGE DEFINITION command, the present operating definition can be changed to any other operating definition supported by the logical unit. The actual details of the operating definition of a logical unit are vendor-specific. If the operating definition is changed to one that does not include the CHANGE DEFINITION command, the target continues to accept the CHANGE DEFINITION command.

If an error occurs during execution of a CHANGE DEFINITION command, the original operating definition remains in effect after the command is executed. The new operating definition becomes active only after successful execution of the CHANGE DEFINITION command.

Since new operating definitions may preclude the execution of tasks that are already in progress, the target may disconnect to allow completion of any tasks that are in progress. Operating definition changes that may cause conflicts with the normal operation from other initiators shall be indicated to those initiators by generating a Unit Attention condition for each other initiator. The additional sense code shall be set to Changed Operating Definition.

A SCSI initiator port may request a list of the operating definitions that the target supports and descriptive text for each operating definition using the INQUIRY command.

7.10 Incorrect initiator connection

An incorrect initiator connection occurs on a reconnection if:

a. a SCSI initiator port attempts to reconnect to a task, and
b. a soft reset condition has not occurred, and
c. the initiator does not send an Abort, Abort Tag, Bus Device Reset, Clear Task Set, Continue Task, or Terminate Task message during the same MESSAGE OUT phase as the IDENTIFY message.

An incorrect initiator connection also occurs on an initial connection when a SCSI initiator port:

a. attempts to establish an I T L Q nexus when an I T L nexus already exists from a previous connection, or
b. attempts to establish an I T L nexus when an I T L Q nexus already exists unless there is a contingent allegiance or extended contingent allegiance condition present for the logical unit or target routine.

A SCSI target port that detects an incorrect initiator connection shall abort all tasks for the initiator on the logical unit or target routine and shall return CHECK CONDITION status. The sense key shall be set to Aborted Command and the additional sense code shall be set to Overlapped Commands Attempted.

An incorrect initiator connection may be indicative of a serious error and if not detected could result in a task operating with a wrong set of pointers. This is considered a catastrophic failure on the part of the initiator. Therefore, host-specific error recovery procedures may be required to guarantee the data integrity on the medium. The target may return additional sense data to aid in this error recovery procedure. Also, some targets may not detect an incorrect initiator connection until after the command descriptor block has been received.

8.0 Drive features

8.1 S.M.A.R.T. system

Some drive families mentioned in Section 1.1 implement what is called in the industry the S.M.A.R.T. system. S.M.A.R.T. is an acronym for Self-Monitoring Analysis and Reporting Technology. The intent of the S.M.A.R.T. system is to recognize conditions that indicate imminent drive failure and provide sufficient warning to the host system of impending failure. The host system may use the information provided to trigger it to perform diagnostic, preventative, and/or protective functions (e.g., data backup).

The initiator sets up the parameters for S.M.A.R.T. operation using MODE SELECT Informational Exceptions Control page 1Ch. The drive reports information about S.M.A.R.T. operation using REQUEST SENSE Additional Sense Code 5D 00 and MODE SENSE data page 1Ch. Refer to Seagate SCSI Command Reference Manual, Part number 100293068 for the description of the MODE SELECT/MODE SENSE commands and for more details on the Informational Exceptions Control page. Refer to the individual drive's Product Manual, to determine the extent of its implementation of the S.M.A.R.T. system.

8.2 Self-test operations

8.2.1 Default self-test

The default self-test is mandatory for all device types that support the SEND DIAGNOSTICs command. The response is simply a GOOD status if the test is successful or a CHECK CONDITION status if the test fails.

8.2.2 The short and extended self-tests

There are two optional types of self-test that may be invoked using the Self-Test Code field in the SEND DIAGNOSTICs command: a short self-test and an extended self-test. The goal of the short self-test is to quickly identify if the logical unit is faulty. A goal of the extended self-test routine is to simplify factory testing during integration by having logical units perform more comprehensive testing without application client intervention. A second goal of the extended self-test is to provide a more comprehensive test to validate the results of a short self-test, if its results are judged by the application client to be inconclusive.

The criteria for the short self-test are that it has one or more segments and completes in two minutes or less. The criteria for the extended self-test are that it is has one or more segments and that the completion time is vendor-specific. Any tests performed in the segments are vendor-specific.

The following are examples of segments:

a. An electrical segment wherein the logical unit tests its own electronics. The tests in this segment are vendor-specific, but some examples of tests that might be included are: a read/write circuitry test and/or a test of the read/write head elements;

b. A seek/servo segment wherein a device tests it capability to find and servo on data tracks; and

c. A read/verify scan segment wherein a device performs read scanning of some or all of the medium surface.

The tests performed in the segments may be the same for the short and extended self-tests. The time required by a logical unit to complete its extended self-test is reported in the Extended Self-Test Completion Time field in the Control mode page (see Seagate SCSI Command Reference Manual, Part number 100293068).

8.2.3 Self-test modes

There are two modes for short and extended self-tests: a foreground mode and a background mode. These modes are described in the following sections.

8.2.3.1 Foreground mode

When a device server receives a SEND DIAGNOSTICs command specifying a self-test to be performed in the foreground mode, the device server shall return status for that command after the self-test has been completed. Not all Seagate drives support this mode.

While performing a self-test in the foreground mode, the device server shall respond to all commands except INQUIRY, REPORT LUNS, and REQUEST SENSE with a CHECK CONDITION status, a sense key of Not Ready and an additional sense code of Logical Unit Not Ready, Self-Test In Progress.

If a device server is performing a self-test in the foreground mode and a test segment error occurs during the test, the device server shall update the Self-test results log page (see Seagate SCSI Command Reference Manual, Part number 100293068) and report CHECK CONDITION status with a sense key of Hardware Error and an additional sense code of Logical Unit Failed Self-Test. The application client may obtain additional information about the failure by reading the Self-test results log page. If the device server is unable to update the Self-test results log page it shall return a CHECK CONDITION status with a sense key of Hardware Error and an additional sense code of Logical Unit Unable To Update Self-Test Log.

An application client should reserve the logical unit before initiating a self-test in the foreground mode. An application client may terminate a self-test that is being performed in the foreground mode using an Abort Task, ABORT TASK SET, or Clear Task Set task management function. If a task manager receives an Abort Task, ABORT TASK SET, or Clear Task Set task management function while performing a self-test in the foreground mode, the it shall abort the self-test and update the Self-test results log page (see Seagate SCSI Command Reference Manual, Part number 100293068).

8.2.3.2 Background mode

When a device server receives a SEND DIAGNOSTICs command specifying a self-test to be performed in the background mode, the device server shall return status for that command as soon as the command descriptor block has been validated.

After returning status for the SEND DIAGNOSTICs command specifying a self-test to be performed in the background mode, the device server shall initialize the Self-test results log page (see Seagate SCSI Command Reference Manual, Part number 100293068) as follows. The self-test code from the SEND DIAGNOSTICs command shall be placed in the Self-Test Code field in the log page. The Self-Test Results field shall be set to Fh. After the Self-test results log page is initialized, the device server shall begin the first self-test segment.

While the device server is performing a self-test in the background mode, it shall terminate with a CHECK CONDITION status any SEND DIAGNOSTICs command it receives that meets one of the following criteria:

a. The Self-Test bit is one; or
b. The Self-Test Code field contains a value other than 000b or 100b.

When terminating the SEND DIAGNOSTICs command, the sense key shall be set to Not Ready and the additional sense code shall be set to Logical Unit Not Ready, Self-Test In Progress.

While performing a self-test in the background mode, the device server shall suspend the self-test to service any other commands received with the exceptions listed in Table 65. Suspension of the self-test to service the command shall occur as soon as possible, but shall never take longer than two seconds.

Table 65: Exception commands for background self-tests

Device type [a]	Command	Reference
All device types	SEND DIAGNOSTICs (with Self-Test Code field set to 100b)	[2]
	WRITE BUFFER (with the mode set to any download microcode option	[2]
Direct access	FORMAT UNIT	SBC
	START/STOP UNIT (stop only)	

a. Device types not listed in this table do not have commands that are exceptions for background self-tests, other than those listed above for all device types.

b. See Seagate SCSI Command Reference Manual, Part number 100293068.

If one of the exception commands listed in Table 65 is received, the device server shall abort the self-test, update the self-test log, and service the command as soon as possible but not longer than two seconds after the command descriptor block has been validated.

Note. An application client may terminate a self-test that is being performed in the background mode by issuing a SEND DIAGNOSTICs command with the Self-Test Code field set to 100b (Abort background self-test function).

8.2.3.3 Elements common to foreground and background self-test modes

The Progress Indication field returned in response to a REQUEST SENSE command (see Seagate SCSI Command Reference Manual, Part number 100293068) may be used by the application client at any time during execution of a self-test to poll the logical unit’s progress. While executing a self-test unless an error has occurred, a device server shall respond to a REQUEST SENSE command by returning a sense key of Not Ready and an additional sense code of Logical Unit Not Ready, Self-Test In Progress with the sense key specific bytes set for progress indication.

The application client may obtain information about the twenty most recently completed self-tests by reading the Self-test results log page (see Seagate SCSI Command Reference Manual, Part number 100293068). This is the only method for an application client to obtain information about self-tests performed in the background mode.

Table 66 summarizes when a logical unit returns status after receipt of a self-test command, how an application client may abort a self-test, how a logical unit handles new commands that are received while a self-test is in progress, and how a logical unit reports a self-test failure.

Table 66: Self-test mode summary

Mode	When status is returned	How to abort the self-test	Processing of subsequent commands while self-test is executing	Self-test failure reporting
Foreground	After the self-test is complete	Abort Task task management function	If the command is INQUIRY, REPORT LUNS, or REQUEST SENSE, process normally. Otherwise, terminate with CHECK CONDITION status, Not Ready sense key, and Logical Unit Not Ready, Self-Test In Progress additional sense code.	Terminate with CHECK CONDITION status, Hardware Error sense key, and Logical Unit Failed Self-Test or Logical Unit Unable To Update Self-Test Log additional sense code.
Background	After the CDB is validated	SEND DIAGNOSTICs command with Self-Test Code field set to 100b	Process the command, except as described in Seagate SCSI Command Reference Manual, Part number 100293068.	Application client checks Self-test results log page after the Progress Indication field returned from REQUEST SENSE indicates the self-test is complete.

8.3 Alternate error detection for the asynchronous information phases (AIP)—Command, Message, and Status

8.3.1 Error detection for asynchronous information phases

This section describes an enhanced error detection method for the Command, Message, and Status asynchronous information transfer phases. In systems not implementing this scheme, these phases only transfer information on the lower eight data bits of a SCSI bus with only normal parity protection on those transfers. Therefore, additional check information can be transferred on the upper eight data bits in order to improve error detection capabilities. Since the upper eight data bits of the bus are used for this scheme, this error detection method is only available on wide SCSI devices that are on wide SCSI busses.

8.3.2 Protection code

The following are the covered signals to be encoded and details of the protection code to be used on the asynchronous information phases.

8.3.2.1 Covered signals

Table 67 defines the signals to be covered by the protection code and their bit locations in the 21-bit code word. When a SCSI device receives an information byte, it also latches the state of the other SCSI signals and values noted in Table 67.

Table 67: Signals to be covered by the protection code and their bit locations

Code word location	SCSI signal	Meaning
0	DB(0)	Data bit 0 of the information byte
1	DB(1)	Data bit 1 of the information byte
2	DB(2)	Data bit 2 of the information byte
3	DB(3)	Data bit 3 of the information byte
4	DB(4)	Data bit 4 of the information byte
5	DB(5)	Data bit 5 of the information byte
6	DB(6)	Data bit 6 of the information byte
7	DB(7)	Data bit 7 of the information byte
8	DB(8)	Reserved [a]
9	DB(9)	Reserved [a]
10	RSVD	Reserved [b]
11	RSVD	Reserved [b]
12	RSVD	Reserved [b]
13	Seq ID 0	Sequence ID bit 0
14	Seq ID 1	Sequence ID bit 1
15	DB(10)	Redundant bit 0 of the code word
16	DB(11)	Redundant bit 1 of the code word
17	DB(12	Redundant bit 2 of the code word
18	DB(13)	Redundant bit 3 of the code word
19	DB(14)	Redundant bit 4 of the code word
20	DB(15)	Redundant bit 5 of the code word

a. DB(8) and DB(9) are reserved for future use. These signals are negated by the transmitting SCSI device and are ignored by the receiving SCSI device. Both the transmitter and receiver encode these signals in the protection code.

b. For calculation purposes these signals are zero. However, these virtual signals could be used for other functions in a future standard.

The Sequence IDs are encoded in the protection code. A sequence of consecutive information transfers during a Message, Command, or STATUS phase is a run. The Sequence ID increments during a run. A new run begins on every phase change or on each Message Out retry.

For each new run, the Sequence ID is set to zero for the first word transferred, set to one for the second word transferred, set to two for the third word transferred, and set to three for the fourth word transferred. The Sequence ID then cycles back to being set to zero for the fifth word transferred, and so forth until the run is complete. At the beginning of the next run, the Sequence ID is set to zero again.

The Sequence ID provides detection of errors that occur when an information transfer is missed or double clocked. A Sequence ID error causes a protection code error. If a protection code error is detected, then the information transfer is invalid. The method for recovery from these errors is the same as the method for parity error recovery.

8.3.2.2 Code description

The protection code (see Table 68) is a cyclic binary BCH code.

Table 68: Protection code

Code	Maximum data bits allowed	Number of redundant bits	Minimum distance of the code
(21,15,4)	15	6	4

The BCH protection code is a cyclic code with a generator polynomial of x 6 + x 5 + x 2 + 1.

The canonical form of the code generator is shown in Figure 21. This is a serial implementation: the register is initialized to zero, then the data is fed in one bit at a time, codeword bit 14 (as defined above) first, followed by codeword bits 13, 12, 11, and so on until bit 0. As each data bit is input, the shift register is clocked. When all 15 bits have been clocked into the generator, the check bits are available in the registers, check bit 0 (codeword bit 15) on the right in the diagram and check bit 5 (codeword bit 20) on the left. The + signs represent an XOR operation.

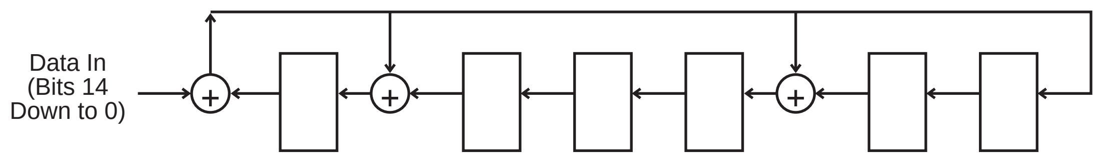


Figure 21. Protection code generator

Using this representation as a baseline, it is possible to construct logic to generate the six check bits from an input data stream of n-bit width, including all 15 bits simultaneously, which is the expected implementation, wherein:

Redundant bit 0 is the XOR of code word bits 0, 1, 2, 3, 5, 6, 7, 10, 11, and 13.
Redundant bit 1 is the XOR of code word bits 1, 2, 3, 4, 6, 7, 8, 11, 12, and 14.
Redundant bit 2 is the XOR of code word bits 0, 1, 4, 6, 8, 9, 10, 11, and 12.
Redundant bit 3 is the XOR of code word bits 1, 2, 5, 7, 9, 10, 11, 12, and 13.
Redundant bit 4 is the XOR of code word bits 2, 3, 6, 8, 10, 11, 12, 13, and 14.
Redundant bit 5 is the XOR of code word bits 0, 1, 2, 4, 5, 6, 9, 10, 12, and 14.

8.3.2.3 Error detection properties

This protection code was selected to have adequate detection properties for asynchronous information transfer phases, given that these transfers are inherently less prone to errors and that these transfers have short code words (approximately 20 bits as compared to thousands of bits during a DT data phase). The BCH protection code Hamming distance is a minimum of four, the same as achieved by the data CRC for transfers of less than eight kilobytes. The protection code will detect all errors of three bits or fewer, all errors of an odd number of bits, and 98.4% of all possible errors.

8.3.3 Protection code usage

Protection code checking is enabled or disabled on an I_T nexus basis. All Command, Message, and STATUS phase information is checked for an I_T nexus while checking is enabled. Protection code checking is disabled after a power cycle, after a hard reset, after a SCSI target port Reset message, and after a change in the transceiver mode (e.g., LVD mode to MSE mode). Protection code checking is always disabled for information unit transfers.

8.3.3.1 Protection code transmission

SCSI devices supporting this protection code transmit the protection code check data during all Command, Message, and STATUS phases. The protection code byte is transferred on the upper eight bits of a wide bus simultaneously with the information day byte on the lower eight bits of the bus using the same clock for the transfer. Thus the transfer of the information byte and the protection code byte is performed exactly like a normal wide transfer. The check data is transmitted even if detection is not enabled.

8.3.3.2 Enabling protection code checking

A SCSI device enables protection code checking for an I_T nexus when it detects that valid protection code data is being transmitted on the upper byte of the SCSI bus. The frequency that a SCSI device will try to enable protection code checking and the number of valid protection code bytes required is vendor-specific. The following are some possible times when a SCSI device could try to enable protection code checking:

a. During the first Command, Message, or STATUS phase after a power cycle, after a hard reset, after a SCSI target port Reset message, or after a change in the transceiver mode.

b. Any time that removal and insertion of a SCSI device is possible, i.e. after a Unit Attention condition.

c. During the MESSAGE phases of a negotiation.

8.3.3.3 Disabling protection code checking

The removal and insertion of a SCSI device could require that protection code checking be disabled for a previously enabled I_T nexus. A SCSI device disables protection code checking when it detects that no protection code data is being transmitted on the upper byte. The determination that no protection code data is being transmitted is vendor-specific. The following are some possible ways that a SCSI device could determine that no protection code data is being transmitted:

a. The DB(15-8) and DB(P1) signals are continuously deasserted while there is good parity on DB(7-0) and DB(P_CRCA).

b. The protection code has a consistent error while there is good parity on DB(7-0) and DB(P_CRCA).

8.3.4 Parity

When protection code checking is enabled normal wide parity is used during a protected transfer of Command, Message, or Status information. DB(P_CRCA) contains the parity for DB(7-0), and DB(P1) contains the parity for DB(15-8).

8.3.5 Error handling

Protection code errors are handled exactly like parity errors during Command, Message, or STATUS phases as defined in the relevant subclauses on exception condition handling in clause 11.1 of SPI-5.

Some examples of how API works are given in SPI-5 Annex M, subsection M.6.

8.4 Removal and insertion of SCSI devices (popularly known as “hot plugging”)

8.4.1 Removal and insertion of SCSI devices overview

This section defines the physical requirements for removal and insertion of SCSI devices on the SCSI bus. The issues related to the logical configuration of the SCSI bus and characteristics of the SCSI devices when a replacement occurs are beyond the scope of this standard. It should also be noted that the cases listed are distinguished for compatibility reasons and in most cases describe a system environment independent of this standard.

Four cases are addressed. The cases are differentiated by the state of the SCSI bus when the removal or insertion occurs.

8.4.2 Case 1—Power off during removal or insertion

a. All SCSI devices are powered off during physical reconfiguration.

8.4.3 Case 2—RST signal asserted continuously during removal or insertion

a. RST signal shall be asserted continuously by the initiator during removal or insertion.

b. The system shall be designed such that the SCSI device being inserted shall make its power ground and logic ground connections at least 1 ms prior to the connection of any device connector contact to the SCSI bus. The ground connections shall be maintained during and after the connection of the SCSI device to the SCSI bus.

c. The system shall be designed such that the SCSI device being removed shall maintain its power ground and logic ground prior to, during, and for at least 1 ms after the disconnection of any device connector contact from the SCSI bus.

Note. The translation of the 1 ms time to mechanical provisions is vendor-specific.

8.4.4 Case 3—Current I/O processes not allowed during insertion or removal

a. All I/O processes for all SCSI devices shall be quiesced.

b. The system shall be designed such that the SCSI device being inserted shall make its power ground and logic ground connections at least 1 ms prior to the connection of any device connector contact to the SCSI bus. The ground connections shall be maintained during and after the connection of the SCSI device to the SCSI bus.

c. The system shall be designed such that the SCSI device being removed shall maintain its power ground and logic ground prior to, during, and for at least 1 ms after the disconnection of any device connector contact from the SCSI bus.

d. The SCSI device being removed or inserted shall employ transceivers that conform to the applicable requirements in ANSI SPI-5 specification, T10/1525D, sections 7.2.2 and 7.3.5.3, for glitch-free powering on and off. The SCSI device shall maintain the high-impedance state at the device connector contacts during a power cycle until the transceivers are enabled. Power cycling includes on-board TERMPWR cycling caused by plugging, and SCSI device power cycling caused by plugging and switching.

Note. Any on board switchable terminators as well as SCSI device transceivers may affect the impedance state at the device connector contacts.

e. The SCSI device power may be simultaneously switched with the SCSI bus contacts if the power distribution system is able to maintain adequate power stability to other SCSI devices during the transition and the grounding requirements in items (b) and (c) above are met.

f. The SCSI bus termination shall be external to the SCSI device being inserted or removed.

g. Resumption of I/O processes is vendor-specific but shall not occur sooner than 200 milliseconds after the completion of the insertion or removal event.

h. Bypassing capacitors connecting to the TERMPWR line on the SCSI device being inserted or removed shall not exceed 10 mF. For single-ended applications, SCSI bus terminations shall use voltage regulation.

Note. In a multimode environment any insertion or removal that changes the SCSI bus mode causes a transceiver mode change reset event (see Section 5.4.4).

8.4.5 Case 4—Current I/O process allowed during insertion or removal

a. All I/O processes for the SCSI device being inserted or removed shall be quiesced prior to removal.

b. A SCSI device being inserted shall make its power ground and logic ground connection at least 1 ms prior to the connection of any device connector contact to the SCSI bus. The ground connections shall be maintained during and after the connection of the SCSI device to the SCSI bus.

c. A SCSI device being removed shall maintain its power ground and logic ground prior to, during, and for at least 1 ms after the disconnection of any device connector contact from the SCSI bus.

d. The SCSI device being removed or inserted shall employ transceivers that conform to the applicable requirements in ANSI SPI-5 specification, T10/1525D, sections 7.2.2 and 7.3.5.3 for glitch-free powering on and off. The SCSI device shall maintain the high-impedance state at the device connector contacts during a power cycle until the transceivers are enabled. Power cycling includes on board TERMPWR cycling caused by plugging, and SCSI device power cycling caused by plugging and switching.

Note. Any on-board switchable terminators as well as SCSI device transceivers may affect the impedance state at the device connector contacts.

e. The SCSI device power may be simultaneously switched with the SCSI bus contacts if the power distribution system is able to maintain adequate power stability to other SCSI devices during the transition and the grounding requirements in items (b) and (c) above are met.

f. The SCSI bus termination shall be external to the SCSI device being inserted or removed.

g. Initiation or resumption of I/O processes for a newly inserted or removed SCSI device is vendor-specific but shall not occur sooner than 200 milliseconds after the completion of the insertion or removal event.

h. Bypassing capacitors connecting to the TERMPWR line on the SCSI device being inserted or removed shall not exceed 10 mF. For single-ended applications, SCSI bus terminations shall use voltage regulation.

Note. In a multimode environment, any insertion or removal that changes the bus mode causes a transceiver mode change reset event (see Section 5.4.4).

Note. LVD SCSI devices may require more stringent system design to tolerate transients that occur during Case 4 insertion or removal.

8.5 SPI-3 to SCSI-2 terminology mapping

This section contains a mapping of terminology used in SCSI-2 to the terminology used in this manual (see Table 69).

Table 69: SPI-3 to SCSI-2 terminology mapping

SPI-3 equivalent term	SCSI-2 term
abort task	abort tag
abort task set	abort
cable skew	cable skew delay
clear task set	clear queue
head of queue	head of queue tag
ordered	ordered queue tag
simple	simple queue tag
target reset	bus device reset
task	I/O process
task complete	command complete
task set	queue

Index

D

E

F

G

H

I

K

L

M

N

O

P

S

T

U

V

W

Seagate

Seagate Technology LLC
920 Disc Drive, Scotts Valley, California 95066-4544, USA
Publication Number: 100293069, Rev. A, Printed in USA