NOTE: Copies of this document may be purchased from:
Global Engineering Documents, 2805 McGaw, Irvine, CA 92714
(800) 854-7179 or (714) 261-1455.
Please refer to document X3.131-199X.

X3T9.2/91-11R1

Working draft proposed
American National Standard
for information systems -

SCSI-3 Interlocked Protocol
(SIP)

March 14, 1992

Secretariat

Computer and Business Equipment Manufacturers Association

Abstract: This standard defines interlocked protocol requirements for the SCSI-3 Parallel Bus. The SCSI-3 Interlocked Protocol combined with the SCSI-3 Parallel Bus define an interface which facilitates the interconnection of computers and intelligent peripherals and thus provides a common interface specification for both systems integrators and suppliers of intelligent peripherals.

This is an internal working document of X3T9.2, a Task Group of Accredited Standards Committee X3. As such, this is not a completed standard. The contents are actively being modified by the X3T9.2 Task Group. This document is made available for review and comment only.

POINTS OF CONTACT:

X3T9.2 Chair	X3T9.2 Vice-Chair	Technical Editor
John B. Lohmeyer	I. Dal Allan	Kurt Chan
NCR Corporation	ENDL	Hewlett-Packard
3718 N. Rock Road	14426 Black Walnut Court	8000 Foothills Blvd.
Wichita, KS 67226	Saratoga, CA 95070	Roseville, CA 95678
Tel: (316) 636-8703	Tel: (408) 867-6630	Tel: (916) 785-5621
Fax: (316) 636-8889	Fax: (408) 867-2115	Fax: (916) 786-9185
Internet: john.lohmeyer@wichitaks.ncr.com	Internet: 2501752@mcimail.com	Internet: kc@hprnd.rose.hp.com

STATUS OF EDITING:

The SCSI-3 Interlocked Protocol contains several changes to the SCSI-2 standard:

1. Target Transfer Disable message
2. Continue I/O Process message
3. 5-bit LUN field (expanded from 3 bits)
4. Dual porting definition and BUD DEVICE RESET OTHER PORT message

1 Forward

The SCSI protocol is designed to provide an efficient peer-to-peer I/O bus with up to 8 devices, including one or more hosts. Data may be transferred asynchronously at rates that depend primarily on device implementation and cable length. Synchronous data transfers are supported at rates up to 10 mega-transfers per second. With the 32-bit wide data transfer option, data rates of up to 40 megabytes per second are possible.

With any technical document there may arise questions of interpretation as new products are implemented. The X3 Committee has established procedures to issue technical opinions concerning the standards developed by the X3 organization. These procedures may result in SCSI Technical Information Bulletins being published by X3.

These Bulletins, while reflecting the opinion of the Technical Committee which developed the standard, are intended solely as supplementary information to other users of the standard. This standard, ANS X3.xxx-199x, as approved through the publication and voting procedures of the American National Standards Institute, is not altered by these bulletins. Any subsequent revision to this standard may or may not reflect the contents of these Technical Information Bulletins.

Current X3 practice is to make Technical Information Bulletins available through:

Global Engineering Documents
2805 McGaw
Irvine, CA 92714
(800) 854-7179
(714) 261-1455

2 Scope

This American National Standard defines the physical attributes of an input/output bus for interconnecting computers and peripheral devices. Figure 1 shows the relationship of this document to other SCSI-3 standards.

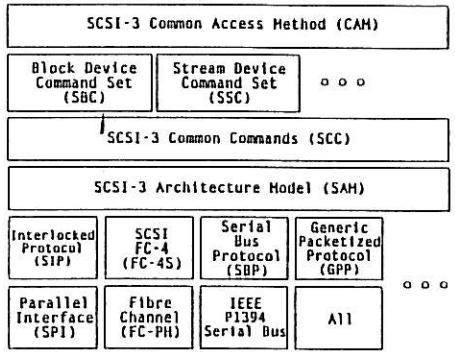


Figure 1 - SCSI-3 Document Roadmap

This roadmap is intended to show the general applicability of documents to one another, not a hierarchy, protocol stack, or system architecture. For example:

1. SIP, SBP, and FC-4S are link-specific protocols designed to be applied to the physical interfaces directly below them.
2. CAM, SBC, SSC, SCC, and SAM are applicable to all link protocols.
3. GPP is intended to be used with any physical interface.

Of the above standards, only the following fall under the jurisdiction of X3T9.2:

1. The Common Access Method, which defines Operating System-independent methods of interfacing host software (e.g., SCSI device drivers) to host hardware (e.g., SCSI host adapters).
2. The Block Device Command Set (e.g., direct access devices)
3. The Stream Device Command Set (e.g., sequential access devices). Other device command sets (e.g., medium changers, printers, scanners) will also be documented here.
4. The Common Command Set, which defines the commands used by all device types.
5. The Architecture Model, which describes I/O process management and the data structures presented to all link protocols.
6. The Interlocked Protocol, which describes the link protocol used on the Parallel Bus (e.g., messages, bus phase sequences).
7. The Parallel Interface, which defines the physical attributes of the Parallel Bus (e.g., cable, connector, and electrical specifications).

The original Small Computer System Interface Standard, X3.131-1986, is referred to herein as SCSI-1. SCSI-1 was revised resulting in the Small Computer System Interface - 2 (X3.131-199x), referred to herein as SCSI-2. This standard, the SCSI-3 Parallel Interface, the SCSI Packetized Protocol, and the SCSI-3 Command Set are referred to herein as SCSI-3. The term SCSI is used wherever it is not necessary to distinguish between the versions of SCSI.

This document defines the interlocked protocol requirements of the interface to allow interoperatability of conforming devices at the data transport level.

SCSI is a local I/O bus that can be operated over a wide range of data rates. The objectives of the Parallel Interface and Interlocked Protocol on SCSI-3 are:

1. To provide host computers with device independence within a class of devices. Thus, different disk drives, tape drives, printers, optical media drives, and other devices can be added to the host computers without requiring modifications to generic system hardware. Provision is made for the addition of special features and functions through the use of vendor unique indications. Reserved areas are provided for future standardization.

2. To provide compatibility with SCSI-2 devices. Devices meeting SCSI-2 and the SCSI-3 Parallel Interface standards can co-exist on the same bus. Properly conforming SCSI-2 devices, both initiators and targets, should respond in an acceptable manner to reject SCSI-3 protocol extensions. All SCSI-3 protocol extensions are designed to be permissive of such rejections and to allow the SCSI-2 device to continue operation without requiring the use of the extension.

3. To move device-dependent intelligence out to the SCSI-3 devices. Refer to the SCSI-3 Command Set standard.

The interface protocol includes provision for the connection of multiple initiators (SCSI devices capable of initiating an I/O process) and multiple targets (SCSI devices capable of responding to a request to perform an I/O process). Distributed arbitration (i.e., bus-contention logic) is built into the architecture of SCSI. A priority system awards interface control to the highest priority SCSI device that is contending for use of the bus.

The SCSI-3 Interlocked Protocol standard is divided into four major sections:

Section 5:	Interlocked protocol attributes and bus conditions resulting from parallel bus events.
Section 6:	Message system used by the interlocked protocol.
Section 7:	Status and Command structure, command processing exception conditions.
Annexes A-D:	Examples of a SCSI command, a disconnect sequence, a linked command, queuing.

3 Referenced Standards and Organizations

American National Standard Small Computer System Interface, X3.131-1991, may also be useful to achieve compatibility with devices that conform to version 2 of SCSI.

American National Standard Small Computer System Interface - 2, X3.131-1991, may also be useful to achieve compatibility with devices that conform to version 2 of SCSI.

The SCSI-3 Parallel Interface (X3T9.2/91-010) defines the bus conditions, phase sequences, messaging, and interlocked-specific commands and status needed to support the Parallel Interface.

The SCSI-3 Command Set (X3T9.2/91-yyy) defines the device-specific command sets used by both the SCSI-3 Packetized Protocol and the SCSI-3 Interlocked Protocol.

4 Glossary and Conventions

4.1 Glossary

4.1.1 accept (a command): A received command is accepted by a Logical Unit or Target Routine when it passes validation.

4.1.2 Active Initiator I/O Process: An I/O Process which begins with an Initial Connection.

4.1.3 Active Target I/O Process: An I/O process that is not in the Queued I/O Process Queue.

4.1.4 Active I/O Process Queue: A queue in a Target which contains zero or more Active Target I/O Processes, or a queue in an Initiator which contains zero or more Active Initiator I/O Processes.

4.1.5 ASCII BYTE: A byte whose value is interpreted for graphic presentation according to the ASCII collating sequence.

4.1.6 Asynchronous Event Notification: A procedure used by targets to notify initiators of events occurring in the target which do not occur while the target is executing an Active Target I/O Process (e.g., Unit Attention or medium removal). The procedure involves the target becoming a temporary initiator for the purpose of issuing a SEND command to an initiator which behaves as a temporary target (processor device). Detection of the event results in either Contingent Allegiance or Extended Contingent Allegiance.

4.1.7 assigned: A Target Controller Function is assigned when it is permitted to accept I/O Processes only from certain Path Groups.

4.1.8 autosense data: The sense data provided by a target immediately following the status byte.

4.1.9 block: A logical or physical byte string.

4.1.10 block peripheral device: A peripheral device shich is in one of the following device classes: direct access, write once, CD-ROM, or optical memory.

4.1.11 byte: In this standard, this term indicates an 8-bit (octet) construct.

4.1.12 byte string: A contiguous set of ASCII bytes.

4.1.13 command: A Command Descriptor Block and associated command parameter data sent from an initiator to a target which causes a logical unit or target routine to perform an action.

4.1.14 command descriptor block: A collection of bytes used to specify the fixed length portion of commands. The operation code of the command, the logical unit number of the target, the logical block address (if required), and the length of the variable length portion of the command (if required) are among the fields specified.

4.1.15 command parameter data: A collection of zero or more fields provided by an initiator as an addnedum to some command descriptor blocks.

4.1.16 command response data: A collection of zero or more fields provided by a target in response to some commands (e.g., Sense Data).

4.1.17 connect: An initiator function that selects a target for the purpose of establishing a nexus and starting an I/O process. The connection that results is an initial connection.

4.1.18 connection: An initial connection or reconnection. A connection can only occur between one initiator and one target on the same bus. A connection begins when conditions exist between an initiator and a target for information transfer. A connection ends with the next disconnect.

4.1.19 Contingent Allegiance: A condition in a target established when an error is detected. During this condition the target preserves sense data. This condition is typically generated by a CHECK CONDITION status.

4.1.20 current I/O process: The I/O process connected on a bus. The I/O process may be an active target I/O process, or it may be a queued I/O process. Only one I/O process may be current per connect.

4.1.21 device class: One ore more targets all having the same logical model and specified independently in the SCSI Command Set standard. All device classes conform to the mandatory requirements of the logical system.

4.1.22 device type: A specific target function which implements some or all of the functions for a device class. Each device type implements all mandatory commands and functions for its device class.

4.1.23 disconnect: The action that occurs when an SCSI device releases control of the SCSI bus, allowing it to go to the BUS FREE phase.

4.1.24 dual port: SCSI devices may provide two SCSI connectors in a dual port configuration that allows any port to connect to the attached logical unit(s). If the option is implemented, the SCSI device shall provide separate transmitters and receivers for each port. If a port is active and a command is sent to the inactive port, the inactive port shall either:

(1) accept the command bytes and disconnect, or
(2) will not accept the command bytes and respond with appropriate status (e.g. BUSY, CHECK CONDITION, QUEUE FULL, or RESERVATION CONFLICT).

The SCSI device may allow the SCSI ID assigned to each port to be the same value or different values.

4.1.25 execute (a command): A command is executed after it has been accepted and its I/O Process has become active.

4.1.26 Extended Contingent Allegiance: A condition in a target established when an error is detected. This condition assists in extended error recovery procedures typically used in multi-initiator systems, and is generated by an INITIATE RECOVERY message. See RELEASE RECOVERY.

4.1.27 field: A set of one or more contiguous bits.

4.1.28 host adapter: A device which connects between a host system and the SCSI bus. The device usually performs the lower layers of the SCSI protocol and normally operates in the initiator role. This function may be integrated into the host system.

4.1.29 initial connection: An initial connection is the result of a connect. It exists from the assertion of the BSY signal in a SELECTION phase until the next BUS FREE phase occurs.

4.1.30 initiator: An SCSI device (usually a host system) that requests an I/O process to be performed by another SCSI device (a target).

4.1.31 initiator mode: The mode of operation of a port in which the port performs initiator functions.

4.1.32 invalid: An illegal, reserved, or unsupported bit, field, code value, bus phase, or protocol sequence.

4.1.33 I/O process: An I/O process consists of one initial connection and zero or more reconnections, all pertaining to a single command or a group of linked commands. More specifically, the connection(s) pertain to a nexus as defined below in which zero or more command descriptor blocks are transferred. An I/O process begins with the establishment of a nexus. An I/O process normally ends with the BUS FREE phase following successful transfer of a COMMAND COMPLETE or a RELEASE RECOVERY message. An I/O process also ends with the BUS FREE phase following an ABORT, ABORT TAG, BUS DEVICE RESET, or CLEAR QUEUE message or when a hard RESET condition or an unexpected disconnect occurs.

4.1.34 I/O process queue: An active target I/O process queue, active I/O initiator process queue, or a queued I/O process queue. The contents of the I/O process queues in an initiator and a target may be different.

4.1.35 I_T nexus: A nexus which exists between an initiator and a target.

4.1.36 I_T_L nexus: A nexus which exists between an initiator, a target, and a logical unit. This relationship replaces the prior I_T nexus.

4.1.37 I_T_L_Q nexus: A nexus between an initiator, a target, 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.

4.1.38 I_T_R nexus: A nexus which exists between an initiator, a target, and a target routine. This relationship replaces the prior I_T nexus.

4.1.39 I_T_x nexus: A nexus which is either an I_T_L or I_T_R nexus.

4.1.40 I_T_x_y nexus: A nexus which is either an I_T_x or I_T_L_Q.

4.1.41 logical block: A unit of data supplied or requested by an initiator usually for a peripheral device. A logical block may be fixed or variable in length.

4.1.42 logical block data: A sequence of bytes from or to a logical block.

4.1.43 logical unit: An addressable function within a target which executes active target I/O processes. A logical unit is addressed by the logical unit number, and it has a command set.

4.1.44 logical unit number: The address of a logical unit.

4.1.45 LSB: Least significant bit.

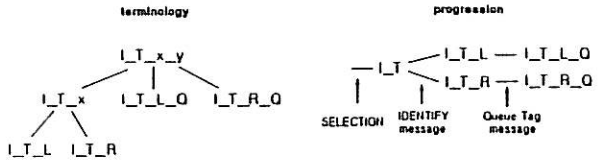
4.1.46 LUN: Logical unit number.

4.1.47 mn:** A mathematical expression representing the number m raised to the n'th power. In SCSI, both m and n are integer decimal numbers.

4.1.48 message: One or more bytes transferred between an initiator and a target for the purpose of controlling the nexus.

4.1.49 MSB: Most significant bit.

4.1.50 nexus: A relationship between an initiator and a target that begins with an initial connection and ends with the completion of the associated I/O process. The relationship may be restricted to a single logical unit or target routine using the Identify function. Each initiator and target may have zero or more nexuses established at a time. The relationship may be further restricted by the successful transfer of a queue tag message. For dual port implementations, the relationship is restricted to the port on which the initial connection was established.



4.1.51 one: A true signal value or a true condition of a variable. A field value numerically equal to 1b.

4.1.52 other peripheral device: A peripheral device which is in one of the following device classes: scanner, medium changer, graphic arts (2 classes).

4.1.53 page: A self-describing set of fields used with some command parameter data and command response data. Each page is identified with a page code.

4.1.54 page code: A field within a page whose value is unique within a command. The page code provides the means to interpret the remaining fields in a page.

4.1.55 parameter: The value of a field.

4.1.56 pending sense data: Sense data within a target for a contingent allegiance or extended contingent allegiance that has not been sent to the initiator.

4.1.57 peripheral device: A physical peripheral device attached to an SCSI device. A peripheral device and an SCSI device may be one unit. Examples of peripheral devices are: magnetic disks, printers, optical disks, and magnetic tapes.

4.1.58 physical block: The minimum recording unit, in bytes, for certain device classes. The physical block may be fixed or variable in length and is device class dependent.

4.1.59 port: The portion of an SCSI device that attaches to the SCSI bus. An SCSI device may have more than one port. Each port may attach to a different bus.

4.1.60 queue: See I/O process queue.

4.1.61 queue tag: The parameter associated with an I/O process that uniquely identifies it from other tagged I/O processes for a logical unit from the same initiator.

4.1.62 queued I/O process: An I/O process that is in a queued I/O process queue and is not in an active target I/O process queue or an active initiator I/O process queue. A queued I/O process in an initiator is an I/O process for which a connect has not been made. A queued I/O process in a target is an I/O Process that is not in the active target I/O process queue.

4.1.63 queued I/O process queue: A queue in either an initiator or a target that holds I/O processes prior to acceptance and execution. The I/O processes in the queued I/O process queue may be tagged or untagged.

4.1.64 receive (a command): A command is received after it has been transferred from an initiator to a Target.

4.1.65 reconnect: The act of reviving a nexus to continue an I/O process. A target completes when conditions are appropriate for the physical bus to transfer data associated with a nexus between an initiator and a target.

4.1.66 reconnection: A reconnection is the result of a reconnect. After a connect, a reconnection exists from the end of a reconnect until the next disconnect.

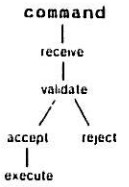
4.1.67 reject (a command): A received command is rejected by a Logical Unit or Target Routine when it fails validation.

4.1.68 reserved: The term used for fields, code values, and bus phases set aside for future standardization.

4.1.69 SCSI: SCSI-3.

4.1.70 SCSI-3: The Small Computer System Interface - 3 (X3.xxx-199X).

4.1.71 SCSI Command Processing: SCSI commands are processed in the following order:



4.1.72 SPI: SCSI-3 Parallel Interface.

4.1.73 SPI Address: The unique address for one port on an SPI device. Multiple ports on the same SPI device connected to the same SPI bus have different SPI addresses.

4.1.74 SPI Device: A device capable of attaching to one or more busses via one or more ports.

4.1.75 SPI ID: The bit-significant representation of the SPI address referring to one of the data bus signal lines available to the SPI device, excluding parity signals.

4.1.76 status: A field sent from a target to an initiator upon completion of each command. Status is also produced to report some conditions which occur in an I/O process when a command is not executing.

4.1.77 storage device: A peripheral device which is capable of reading and optionally writing logical blocks to a medium.

4.1.78 stream peripheral device: A peripheral device which is in one of the following device classes: sequential access, printer, processor, or communications.

4.1.79 target: An SPI device that performs an I/O process requested by an initiator.

4.1.80 target mode: The mode of operation of a port in which the port performs target functions.

4.1.81 target routine: An addressable function within a target which executes active target I/O processes. A target routine is addressed by a target routine number, and it has a command set to execute.

4.1.82 target routine number: The address of a target routine.

4.1.83 unexpected disconnect: A disconnection that occurs because of a protocol error.

4.1.84 validate (a command): . A command is validated when a Logical Unit or Target Routine verifies that all fields contain legal values.

4.1.85 vendor specific (VS): Something (e.g., a bit, field, code value, etc.) that is not defined by this standard and may be used differently in various implementations.

4.1.86 xx: Digits 0-9, except those used as section numbers, in the text of this standard that are not immediately followed by lower-case "b" or "h" are decimal values. Large Numbers are not separated by commas or spaces (e.g., 12345; not 12,345 or 12 345).

4.1.87 xxb: Digits 0 and 1 immediately followed by lower-case "b" are binary values.

4.1.88 xxh: Digits 0-9 and the upper-case letters "A"-"F" immediately followed by lower-case "h" are hexadecimal values.

4.1.89 zero: A field with a value of 0b in each bit position. A false signal value or a false condition of a variable.

4.2 Editorial Conventions

Certain words and terms used in this standard have specific meanings beyond the normal English meaning. Either the glossary defines these words and terms, or the definition appears at first use. Names of signals, phases, messages, commands, statuses, sense keys, additional sense codes, and additional sense code qualifiers are in all uppercase (e.g., REQUEST SENSE).

Normal English capitalization (or lack thereof) is used for other words. Words have the normal technical English definition unless the word or phrase is defined in context or in a glossary. Some words may have definitions unique to the sections where they are used.

5 SCSI Interlocked Protocol Attributes

5.1 SCSI Bus Conditions

The SCSI bus has two asynchronous conditions; the attention condition and the reset condition. These conditions cause the SCSI device to perform certain actions and can alter the phase sequence. The attention condition is caused by the attention event and the reset condition is caused by the reset event. These events are described in the SCSI Parallel Interface document.

Furthermore, an SCSI device may not all be powered on at the same time. This standard 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. It is recommended that following a power-on to selection time after power is applied, SCSI targets be able to respond with appropriate status and sense data to the TEST UNIT READY, INQUIRY, and REQUEST SENSE commands.

5.1.1 Attention Condition

The attention condition allows an initiator to inform a target that the initiator has a message ready. The target may get this message by performing a MESSAGE OUT phase. The attention condition is created by an attention event (see SCSI-3 Parallel Bus).

A target shall respond to an attention event with MESSAGE OUT phase as follows:

(1) If the ATN signal becomes true during a COMMAND phase, the target shall enter MESSAGE OUT phase after transferring part or all of the command descriptor block bytes.

(2) If the ATN signal becomes true during a DATA phase, the target shall enter MESSAGE OUT phase at the target's earliest convenience (often, but not necessarily on a logical block boundary). The initiator shall continue REQ/ACK handshakes until it detects the phase change.
(3) If the ATN signal becomes true during a STATUS phase, the target shall enter MESSAGE OUT phase after the status byte has been acknowledged by the initiator.
(4) If the ATN signal becomes true 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.
(5) If the ATN signal becomes true during a SELECTION phase and before the initiator releases the BSY signal, the target shall enter MESSAGE OUT phase immediately after that SELECTION phase.
(6) If the ATN signal becomes true during a RESELECTION phase, the target shall enter MESSAGE OUT phase after the target has sent its IDENTIFY message for that RESELECTION phase.

5.1.2 Reset Condition

The reset condition is used to immediately clear all SCSI devices from the bus. This condition shall take precedence over all other phases and conditions. Any SCSI device may create the reset condition by generating a reset event. During the reset condition, the state of all SCSI bus signals other than the RST signal is not defined.

The effect of the reset condition on I/O processes which have not completed, SCSI device reservations, and SCSI device operating modes is determined by whether the SCSI device has implemented the hard reset alternative or the soft reset alternative (one of which shall be implemented) as defined in 5.1.2.1 and 5.1.2.2. The hard and soft reset alternatives are mutually exclusive within a system. A facility for targets to report which reset alternative is implemented is provided in the SftRe bit of the INQUIRY data (see SCSI-3 Command Set).

5.1.2.1 Hard Reset Alternative

SCSI devices that implement the hard reset alternative, upon detection of the reset condition, shall:

(1) Clear all I/O processes including queued I/O processes.
(2) Release all SCSI device reservations.
(3) Return any SCSI device operating modes to their appropriate initial conditions, similar to those conditions that would be found after a normal power-on reset. MODE SELECT conditions shall be restored to their last saved values if saved values have been established. MODE SELECT conditions for which no saved values have been saved shall be returned to their default values.
(4) Unit attention condition shall be set.

It is recommended that, following a reset to selection time after a hard reset condition ends, SCSI targets be able to respond with appropriate status and sense data to the TEST UNIT READY, INQUIRY, and REQUEST SENSE commands.

For dual port implementations, the SCSI device shall only apply the hard reset to the port on which the reset was received. In this case,

(1) All I/O processes for the other port are unaffected,
(2) Reservations granted to initiators on that port are not released,
(3) Operating modes for the other port are not changed, and
(4) Unit attention is not set for any initiators on the other port.

If MODE SELECT parameters affecting the other port are changed, the unit attention condition may be set.

5.1.2.2 Soft Reset Alternative

SCSI devices that implement the soft reset alternative, upon detection of the reset condition, shall:

1) Attempt to complete any I/O processes which have not completed and that were fully identified
2) Preserve all SCSI device reservations
3) Preserve any SCSI device operating modes (MODE SELECT, PREVENT/ALLOW MEDIUM REMOVAL commands, etc.)
4) Preserve all the information required to continue normal dispatching of I/O processes queued prior to the reset condition.

The soft reset alternative allows an initiator to reset the SCSI bus with minimum disruption to the operation of other initiators in a multiple initiator system. To ensure proper operation the following conditions shall be met:

1) An initiator shall not consider an I/O process to be fully identified until the IDENTIFY message (and queue tag message, if any) is sent to the target and the target responds by changing to any other information transfer phase and requests that at least one byte be transferred.
2) A target shall consider an I/O process to be fully identified when it successfully receives the IDENTIFY message and any queue tag message and the initiator negates the ATN signal.
3) If an initiator selects a logical unit for which there already is an active I/O process with the same queue tag (if any) for the same initiator, the target shall clear the original I/O process and perform the new I/O process.
4) If a target reselects an initiator to continue an I/O process for which the initiator has no record, the initiator shall abort that I/O process by sending the ABORT or ABORT TAG message, depending on whether the reselecting I/O process is a tagged I/O process.
5) An initiator shall consider an I/O process to be completed when it negates ACK for a successfully received COMMAND COMPLETE message.
6) A target shall consider an I/O process to be completed when it detects the transition of ACK to false for the COMMAND COMPLETE message with the ATN signal false.
7) An initiator shall not negate the ACK signal for the SAVE DATA POINTER message until it has actually saved the data pointer for the I/O process.
8) A target shall consider the data pointer to be saved when it detects the transition of the ACK signal to false for the SAVE DATA POINTER message with the ATN signal false.
9) If the reset condition occurs between the time that the target asserts the REQ signal for the SAVE DATA POINTER message and it detects the transition of the ACK signal to false, the target shall terminate the I/O process with CHECK CONDITION status. The target shall set the sense key to ABORTED COMMAND. This is necessary because the target cannot determine whether the data pointer has actually been saved.

If the ATN signal is true in conditions (6) or (8), the target would normally switch to MESSAGE OUT phase and attempt to transfer a message byte. If the reset condition occurs before the target successfully receives the message byte, it may assume that the initiator has not successfully received the COMMAND COMPLETE message or the SAVE DATA POINTER message. In the case of COMMAND COMPLETE message, the target may reselect the initiator and attempt to send the COMMAND COMPLETE message again. In the case of the SAVE DATA POINTER message, the target may reselect the initiator and terminate the I/O process as described in condition (9).

5.1.3 Unexpected Disconnect Condition

The Unexpected Disconnect condition is created by a disconnect event which is unexpected by the initiator. Initiators normally do not expect a disconnect event except after one of the following occurrences:

a) after a Reset Event has been detected;
b) after an ABORT message is successfully received by a target;
c) after a BUS DEVICE RESET message is successfully received by a target;
d) after a DISCONNECT message is successfully received by a target;
e) after a COMMAND COMPLETE message is successfully transmitted from a target
f) after a RELEASE RECOVERY message is successfully received by a target;
g) after an ABORT TAG message is successfully received by a target;
h) after a CLEAR QUEUE message is successfully received by a target.
i) after an unsuccessful selection or reselection.

If an initiator detects a disconnect event at any other time an Unexpected Disconnect condition is said to exist. A target uses this condition to indicate a protocol error to the initiator. The target may create the Unexpected Disconnect condition independent of the existence of the Attention Condition.

The initiator shall manage this condition as an unsuccessful I/O process termination. The target terminates the I/O Process by clearing all pending data and status information for the affected nexus. The target may optionally prepare sense data that may be retrieved by a REQUEST SENSE command. When an initiator detects an Unexpected Disconnect Condition, it is recommended that a REQUEST SENSE command be attempted to obtain any valid sense data that may be available.

5.1.4 Response to SELECTION

Targets in interlocked mode shall respond to SELECTION as described in the table below.

Table 1: Target Response to Selection

	Number of SCSI ID bits (n) true during SELECTION phase		
SELECTION with:	1>n>2	n=1	n=2
ATN True	Unexpected Disconnect Condition	MESSAGE OUT, no disconnect allowed	MESSAGE OUT
ATN False	Unexpected Disconnect Condition		

5.2 SCSI Pointers

Consider the system shown in Figure 5-3 in which an initiator and target communicate on the SCSI bus in order to execute an I/O process.

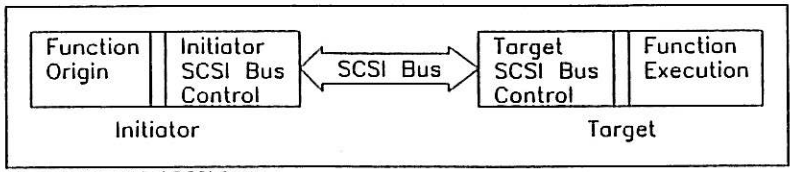


Figure 2: Simplified SCSI System

The SCSI architecture provides for a set of three pointers for each I/O process, called the saved pointers. The set of three pointers consist of one for the command, one for the data, and one for the status. When an I/O process becomes active, its three saved pointers are copied into the initiator's set of three active pointers. There is only one set of active pointers in each initiator. The active pointers point to the next command, data, or status byte to be transferred between the initiator's memory and the target. The saved and active pointers reside in the initiator.

The saved command pointer always points to the start of the command descriptor block for the I/O process. The saved status pointer always points to the start of the status area for the I/O process. The saved data pointer points to the start of the data area until the target sends a SAVE DATA POINTER message for the I/O process.

In response to the SAVE DATA POINTER message, the initiator stores the value of the active data pointer into the saved data pointer for that I/O process. The target may restore the active pointers to the saved pointer values for the active I/O process by sending a RESTORE POINTERS message to the initiator. The initiator then copies the set of saved pointers into the set of active pointers. Whenever a target disconnects from the bus, only the set of saved pointers are retained. The set of active pointers is restored from the set of saved pointers upon reconnection of the I/O process.

Since the data pointer value may be modified by the target before the I/O process ends, it should not be used to test for actual transfer length because it is not reliable.

6 SCSI Interlocked Message System Description

The message system allows communication between an initiator and target for the purpose of interface management. A message may be one, two, or multiple bytes in length. One or more messages may be sent during a single MESSAGE phase, but a message may not be split over MESSAGE phases. The initiator is required to end the MESSAGE OUT phase (by negating ATN) when it sends certain messages identified in Table 2.

One-byte, Two-byte, and extended message formats are defined. The first byte of the message determines the format as defined in Table 2.

Table 2: Message Format

Value	Message Format
00h	One-Byte Message (COMMAND COMPLETE)
01h	Extended Messages
02h - 1Fh	One-Byte Messages
20h - 2Fh	Two-Byte Messages
30h - 7Fh	Reserved
80h - FFh	One-Byte Message (IDENTIFY)

One-byte messages consist of a single byte transferred during a MESSAGE phase. The value of the byte determines which message is to be performed as defined in Table 3.

Table 3: Message Codes

Code	Support Init	Support Targ	Message Name	Direction	Negate ATN Before last ACK
06h	O	M	ABORT	Out	Yes
0Dh	O	O	ABORT TAG (Note 1)	Out	Yes
0Ch	O	M	BUS DEVICE RESET	Out	Yes
14h	O	M	BUS DEVICE RESET OTHER PORT (Note 3)	Out	No
0Eh	O	O	CLEAR QUEUE (Note 1)	Out	Yes
00h	M	M	COMMAND COMPLETE	In	—
12h	O	O	CONTINUE I/O PROCESS	Out	Yes
04h	O	O	DISCONNECT	In	—
04h	O	O	DISCONNECT	Out	Yes
80h+	M	O	IDENTIFY	In	—
80h+	M	M	IDENTIFY	Out	No
23h	O	O	IGNORE WIDE RESIDUE (Two Bytes)	In	—
0Fh	O	O	INITIATE RECOVERY	In	—
0Fh	O	O	INITIATE RECOVERY (Note 2)	Out	Yes
05h	M	M	INITIATOR DETECTED ERROR	Out	Yes
0Ah	O	O	LINKED COMMAND COMPLETE	In	—
0Bh	O	O	LINKED COMMAND COMPLETE (WITH FLAG)	In	—
09h	M	M	MESSAGE PARITY ERROR	Out	Yes
07h	M	M	MESSAGE REJECT	In Out	Yes
***	O	O	MODIFY DATA POINTER	In	—
08h	M	M	NO OPERATION	Out	Yes
			Queue Tag Messages (Two Bytes)		
21h	O	O	HEAD OF QUEUE TAG	Out	No
22h	O	O	ORDERED QUEUE TAG	Out	No
20h	O	O	SIMPLE QUEUE TAG	In Out	No
10h	O	O	RELEASE RECOVERY	Out	Yes
03h	O	O	RESTORE POINTERS	In	—
02h	O	O	SAVE DATA POINTER	In	—
***	O	O	SYNCHRONOUS DATA TRANSFER REQUEST	In Out	Yes
13h	O	O	TARGET TRANSFER DISABLE	Out	Yes
***	O	O	WIDE DATA TRANSFER REQUEST	In Out	Yes
11h	O	O	TERMINATE I/O PROCESS	Out	Yes
15h - 1Fh			Reserved		
24h - 2Fh			Reserved for two-byte messages		
30h - 7Fh			Reserved		

Key: M = Mandatory support, O = Optional support.
In = Target to initiator, Out = Initiator to target.
Yes = Initiator shall negate ATN before last ACK of message.
No = Initiator may or may not negate ACK before last ACK of message.
— = Not Applicable
*** = Extended message
80h+ = Codes 80h through FFh are used for IDENTIFY messages

NOTES:
(1) The ABORT TAG and CLEAR QUEUE messages are required if tagged queuing is implemented.
(2) Outbound INITIATE RECOVERY messages are only valid during the asynchronous event notification protocol.
(3) The BUS DEVICE RESET OTHER PORT message is required if the dual port option is implemented.

Two-byte messages consist of two consecutive bytes transferred during a MESSAGE phase. The value of the first byte determines which message is to be performed as defined in Table 3. The second byte is a parameter byte which is used as defined in the message description.

A value of one 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. The extended message format and the extended message codes are shown in Tables 4 and 5, respectively.

Table 4: Extended Message Format

Bit / Byte	7	6	5	4	3	2	1	0
0	Extended message (01h)							
1	Extended message length (n)							
2	Extended message code (y)							
3	Extended message arguments							
n+1								

The extended message length specifies the length in bytes of the extended message code plus the extended message arguments to follow. Therefore, the total length of the message is equal to the extended message length plus two. A value of zero for the extended message length indicates 256 bytes follow.

The extended message codes are listed in Table 5. The extended message arguments are specified within the extended message descriptions.

Table 5: Extended Message Codes

Code (y)	Description
02h	Reserved
00h	MODIFY DATA POINTER
01h	SYNCHRONOUS DATA TRANSFER REQUEST
03h	WIDE DATA TRANSFER REQUEST
04h - 7Fh	Reserved
80h - FFh	Vendor Specific

NOTE: Extended message code 02h was used for the EXTENDED IDENTIFY message in SCSI-1.

The first message sent by the initiator after the SELECTION phase shall be an IDENTIFY, ABORT, or BUS DEVICE RESET message. If a target receives any other message it shall go to BUS FREE phase (see Unexpected Disconnect condition, 5.1.3)

If the first message is an IDENTIFY message, then it may be immediately followed by other messages, such as the first of a pair of SYNCHRONOUS DATA TRANSFER REQUEST messages. If tagged queuing is used the queue tag message immediately follows the IDENTIFY message. The IDENTIFY message establishes a logical connection between the initiator and the specified logical unit or target routine within the target known as an I_T_L nexus or I_T_R nexus. After the RESELECTION phase, the target's first message shall be IDENTIFY. This allows the I_T_L nexus or I_T_R nexus to be re-established. Only one logical unit or target routine shall be identified for any connection; if a target receives a second IDENTIFY message with a different logical unit number or target routine number during a connection, it shall go to BUS FREE phase (see Unexpected Disconnect condition, 5.1.3). The treatment of other logical unit addressing errors is described in 6.5.

All initiators shall implement the mandatory messages tabulated in the "Init" column of Table 3. All targets shall implement the mandatory messages tabulated in the "Targ" column of Table 3.

Whenever an I_T_L nexus or I_T_R nexus is established by an initiator that is allowing disconnection, the initiator shall ensure that the active pointers are equal to the saved pointers for that particular logical unit or target routine. An implied restore pointers operation shall occur as a result of a reconnection.

6.1 ABORT

The ABORT message is sent from the initiator to the target to clear any I/O process for the I_T_x nexus. The target shall go to the BUS FREE phase following successful receipt of this message. The pending data, status, and I/O processes for any other nexus shall not be cleared.

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 I/O process and no other I/O process shall be affected.

The ABORT message in the case of only an I_T nexus is useful to an initiator that cannot get an IDENTIFY message through to the target due to parity errors and just needs to end the current connection. No pending data, status, or queued I/O processes are affected.

It is not possible to abort an I_T nexus on a reconnection because of item (6) in 5.1.3.

It is not an error to issue this message to an I_T_x nexus that does not have an active or queued I/O process.

Previously established conditions, including MODE SELECT parameters, reservations, and extended contingent allegiance shall not be changed by the ABORT message. For dual port implementations, no I/O processes or previously established conditions for the other port are affected.

Table 6 summarizes the effect of the ABORT, ABORT TAG, BUD DEVICE RESET, and CLEAR QUEUE messages on I/O Processes.

Table 6: I/O Process Initialization Messages

		BUS DEVICE RESET	CLEAR QUEUE		ABORT		ABORT TAG	
Current nexus		Any	I_T	I_T_x_y	I_T	I_T_x_y	I_T	I_T_x_y
Which I/O Processes Cleared	Current	Yes	Illegal	Yes	Yes	Yes	Illegal	Yes
	Active	Yes	Illegal	Yes	No	Yes	Illegal	No
	Queued	Yes	Illegal	Yes	No	Yes	Illegal	No
For Which Initiators		All	Illegal	All	Selecting	Selecting	Illegal	Selecting
Which Other Conditions Cleared	Mode Select	Yes	Illegal	No	No	No	Illegal	No
	ECA	Yes	Illegal	No	No	No	Illegal	No
	Reservations	Yes	Illegal	No	No	No	Illegal	No
	CA	Yes	Illegal	Yes[1]	Yes	Yes	Illegal	Yes[1]
Unit Attention Established for Which Initiators		All	Illegal	All but Selecting	None	None	Illegal	None

[1] Except I_T_R nexuses

6.2 ABORT TAG

The ABORT TAG message shall be implemented if tagged queuing is implemented. The target shall go to the BUS FREE phase following the successful receipt of this message. The target shall clear the current I/O process. If the target has already started execution of the I/O process, the execution shall be halted. The medium contents may have been modified before the execution was halted. In either case, any pending status or data for the I/O process shall be cleared and no status or ending message shall be sent to the initiator. Pending status, data, and commands for other active or queued I/O processes shall not be affected. Execution of other I/O processes queued for the I_T_x nexus shall not be aborted.

Previously established conditions, including MODE SELECT parameters, reservations, and extended contingent allegiance shall not be changed by the ABORT TAG message. For dual port implementations, no I/O processes or previously established conditions for the other port are affected.

On a reconnection, the ABORT TAG message aborts the current I/O process if it is fully identified. If the I/O process is not fully identified (i.e., an I_T_L nexus exists, but the target is reconnecting for an I_T_L_Q nexus), then the I/O process is not aborted and the target goes to the BUS FREE phase.

A nexus is not fully identified on a reconnection if the ATN signal is asserted during or prior to the IDENTIFY message and the target only has tagged I/O processes for that initiator on that logical unit.

6.3 BUS DEVICE RESET

The BUS DEVICE RESET message is sent from an initiator to direct a target to clear all I/O processes on that SCSI device. This message has the same effect as a hard reset condition to the selected SCSI device. The target shall go to the BUS FREE phase following successful receipt of this message. The target shall create a unit attention condition for all initiators.

For dual port implementations, the hard reset condition shall apply only to the port from which the message was received. Any active I/O processes, queued I/O processes, device reservations, and operating modes for the other port are unaffected and the unit attention condition is not set for initiators on the other port.

6.4 BUS DEVICE RESET OTHER PORT

The BUS DEVICE RESET OTHER PORT message shall be implemented if the device implements the dual port option. The BUS DEVICE RESET OTHER PORT message is sent from an initiator to direct a target to clear the I/O processes associated with the other port on that SCSI device. This message has the same effect as a hard reset condition on the other port of the selected SCSI device but has no effect on I/O processes, reservations, and operating modes of the port from which the message was received. The target shall create a unit attention condition for all initiators on the other port.

125

6.5 CLEAR QUEUE

The CLEAR QUEUE message shall be implemented if tagged queuing is implemented and may be implemented if untagged queuing is implemented. The target shall go to the BUS FREE phase following successful receipt of this message. The target shall perform an action equivalent to receiving a series of ABORT messages from each initiator. All I/O processes, from all initiators, in the queue for the specified logical unit or target routine shall be cleared from the queue. All active I/O processes shall be terminated. The medium may have been altered by partially executed commands. All pending status and data for that logical unit or target routine for all initiators shall be cleared. No status or message shall be sent for any of the I/O processes. A unit attention condition shall be generated for all other initiators with I/O processes that either were active or were queued for that logical unit or target routine. When reporting the unit attention condition the additional sense code shall be set to COMMANDS CLEARED BY ANOTHER INITIATOR.

For dual port implementations, only I/O processes for the port from which the message was received are affected. No I/O processes are cleared for the other port and no unit attention conditions are generated for initiators on the other port.

Previously established conditions, including MODE SELECT parameters, reservations, and extended contingent allegiance shall not be changed by the CLEAR QUEUE message.

6.6 COMMAND COMPLETE

The COMMAND COMPLETE message is sent from a target to an initiator to indicate that the execution of an I/O process has completed and that valid status has been sent to the initiator. 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 it detects the negation of ACK for the COMMAND COMPLETE message with the ATN signal false.

The I/O process may have completed successfully or unsuccessfully as indicated in the status.

6.7 CONTINUE I/O PROCESS

The CONTINUE I/O PROCESS message is sent from the initiator to the target to reconnect to an I/O process. This message shall be sent in the same MESSAGE OUT phase as the IDENTIFY message.

IMPLEMENTORS NOTE: Thus the MESSAGE OUT phase following SELECTION phase consists of the IDENTIFY, queue tag (if any), and CONTINUE I/O PROCESS messages.

The purpose of the CONTINUE I/O PROCESS message is to distinguish a valid initiator reconnection from an incorrect initiator reconnection (see 7.4.1).

If the target expects a significant delay before it will be ready to continue processing the reconnected I/O PROCESS, it may attempt to free the SCSI bus by sending a DISCONNECT message to the initiator. The initiator may reject the disconnection attempt by responding with MESSAGE REJECT message.

If the CONTINUE I/O PROCESS message occurs on an initial connection then the target should go to the BUS FREE phase.

If the CONTINUE I/O PROCESS message occurs on a subsequent connection then the target may either treat this as a dynamic head-of-queue request or it may reject the message with a MESSAGE REJECT message.

An initiator that gets rejected should assert the ATN signal and send an ABORT TAG message on the resulting MESSAGE OUT phase. Otherwise, the target may treat the connection as an incorrect initiator connection.

Initiators should avoid sending this message to targets which have not implemented this message. Such targets may not respond as described in this section. An initiator can determine whether a target implements this message by examining the TranDis bit in the standard INQUIRY data (see SCSI-3 Command Set).

6.8 DISCONNECT

The DISCONNECT message is sent from a target to inform an initiator that the present connection is going to be broken (the target plans to disconnect by releasing the BSY signal), but that a later reconnect will be required in order to complete the current I/O process. This message shall not cause the initiator to save the data pointer. 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 it detects the negation of the ACK signal for the DISCONNECT message with the ATN signal false.

Targets which break data transfers into multiple connections shall end each successful connection (except possibly the last) with a SAVE DATA POINTER - DISCONNECT message sequence.

This message may also be sent from an initiator to a target to instruct the target to disconnect from the SCSI bus. If this option is supported, and after the DISCONNECT message is received, the target shall switch to MESSAGE IN phase, send the DISCONNECT message to the initiator (possibly preceded by SAVE DATA POINTER message), and then disconnect by releasing BSY. After releasing the BSY signal, the target shall not participate in another ARBITRATION phase for at least a disconnection delay of 200 microseconds. A disconnection delay is defined as the minimum time that a target shall wait after releasing BSY before participating in an ARBITRATION phase when honoring a DISCONNECT message from the initiator. If this option is not supported or the target cannot 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.

6.9 IDENTIFY

The IDENTIFY message is sent by either the initiator or the target to establish an I_T_L or an I_T_R nexus. For dual port implementations: if the target disconnects from the bus during an I/O process, it shall reconnect through the same port when the I/O process is continued.

Use of the IDENTIFY message to establish an I_T_R nexus allows connection to one of up to eight target routines or functions in the target itself. These target routines are expected to be used for maintenance and diagnostic purposes.

Table 7: IDENTIFY Message Format

Bit	7	6	5	4	3	2	1	0
	Identify	DiscPriv	LUNTAR	LUNTRN				

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 disconnecting. A DiscPriv bit of zero specifies that the target shall not disconnect. This bit is not defined and shall be set to zero when an IDENTIFY message is sent by a target.

A logical unit target (LUNTAR) bit of zero specifies that the I/O process is directed to or from a logical unit. A LUNTAR bit of one specifies that the I/O process is directed to or from a target routine.

The logical unit number target routine number (LUNTRN) field specifies a logical unit number if the LUNTAR bit is zero. The LUNTRN field specifies a target routine number if the LUNTAR bit is one. Only the INQUIRY and REQUEST SENSE commands are valid for target routines. If a target receives any other command for a target routine, it shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST.

An IDENTIFY message is invalid if a reserved bit is set to one or if the LUNTAR bit is set to one and the target does not implement target routines. A device may respond to an invalid IDENTIFY message by immediately sending a MESSAGE REJECT message or by returning CHECK CONDITION status. If a CHECK CONDITION status is returned, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID BITS IN IDENTIFY MESSAGE FIELD.

Only one logical unit number or target routine number shall be identified per I/O process. The initiator may send one or more IDENTIFY messages during a connection. A second IDENTIFY message with a different value in either the LUNTAR bit or LUNTRN field shall not be issued before a BUS FREE phase has occurred; if a target receives a second IDENTIFY message with a different value in either of these fields, it shall go to BUS FREE phase (see Unexpected Disconnect condition, 5.1.3). Thus an initiator may change the DiscPriv bit, but may not attempt to switch to another I/O process. (See the DTDC field of the disconnect-reconnect mode page in the SCSI-3 Command Set for additional controls over disconnection.)

An implied RESTORE POINTERS message shall be performed by the initiator prior to the assertion of the ACK signal on the next phase for an inbound IDENTIFY message sent during reconnection.

6.10 IGNORE WIDE RESIDUE

Table 8: IGNORE WIDE RESIDUE Message Format

Bit Byte	7	6	5	4	3	2	1	0
0	Message code (23h)							
1	Ignore (01h, 02h, 03h)							

The IGNORE WIDE RESIDUE message shall be sent from a target to indicate that the number of valid bytes sent during the last REQ/ACK handshake and REQB/ACKB handshake of a DATA IN phase is less than the negotiated transfer width. The ignore field indicates the number of invalid data bytes transferred. This message shall be sent immediately following that DATA IN phase and prior to any other messages. The ignore field is defined in Table 9.

Table 9: Ignore Field Definition

Ignore	Invalid Data Bits	
	32-bit Transfers	16-bit Transfers
00h	Reserved	Reserved
01h	DB(31-24)	DB(15-8)
02h	DB(31-16)	Reserved
03h	DB(31-8)	Reserved
04h - FFh	Reserved	Reserved

Even though a byte is invalid its corresponding parity bit shall be valid for the value transferred. For 16-bit transfers, DB(31-16) are always invalid and the corresponding parity bits are also invalid.

6.11 INITIATE RECOVERY

A target that supports extended contingent allegiance shall inform the initiator it is entering this condition by sending an INITIATE RECOVERY message immediately following a CHECK CONDITION or COMMAND TERMINATED status. The extended contingent allegiance condition remains in effect until terminated as described in 6.7.

If an asynchronous event occurs, the target may enter an extended contingent allegiance condition by becoming a temporary initiator and sending the INITIATE RECOVERY message following the IDENTIFY message and any queue tag message and before the COMMAND phase of the SEND command that is used to perform the asynchronous event notification (see 7.4.2). The successful transmission of this message establishes the extended contingent allegiance condition which remains in effect until terminated as described in 6.7.

If the target notifies multiple initiators of the asynchronous event, it should include the INITIATE RECOVERY message in only one of the notifications.

A MESSAGE REJECT response to an INITIATE RECOVERY message indicates that an extended contingent allegiance condition shall not be established. The enabled or disabled state of an extended contingent allegiance (see the DQue bit of the control mode page, SCSI-3 Command Set) is not changed by the rejection of an INITIATE RECOVERY message.

6.12 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 I/O process. 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 present pointer integrity is not assured, a RESTORE POINTERS message or a disconnect followed by a reconnect, shall cause the pointers to be restored to their defined prior state.

6.13 LINKED COMMAND COMPLETE

The LINKED COMMAND COMPLETE message is sent from a target to an initiator to indicate that the execution of a linked 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.

6.14 LINKED COMMAND COMPLETE (WITH FLAG)

The LINKED COMMAND COMPLETE (WITH FLAG) message is sent from a target to an initiator to indicate that the execution of a linked command (with the flag bit set to one) has completed and that status has been sent. The initiator shall then set the pointers to the initial state of the next linked command. Typically this message would be used to cause an interrupt in the initiator between two linked commands.

6.15 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.

In order to indicate its intentions of sending this message, the initiator shall assert the ATN signal prior to its release of the ACK signal for the REQ/ACK handshake of the message byte that has the parity error. This provides an interlock so that the target can 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 releasing the BSY signal without any further information transfer attempt.

If after receiving the MESSAGE PARITY ERROR message the target returns to the MESSAGE IN phase before switching to some other phase (other than SETUP phase), the target shall re-send the entire message that had the parity error.

6.16 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.

In order to indicate its intentions of sending this message, the initiator shall assert the ATN signal prior to its release of the ACK signal for the REQ/ACK handshake of 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 can determine which message byte is rejected.

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

6.17 MODIFY DATA POINTER Message

Table 10: MODIFY DATA POINTER

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) Argument							
6								(LSB)

The MODIFY DATA POINTER message is sent from the target to the initiator and requests that the signed argument be added (two's complement) to the value of the current data pointer. The Enable Modify Data Pointer (EMDP) bit in the Disconnect-Reconnect mode page (see SCSI-3 Command Set) indicates whether or not the target is permitted to issue the MODIFY DATA POINTER message.

6.18 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 enters the MESSAGE OUT phase.

6.19 Queue Tag Messages

Table 11: Queue Tag Message Format

Bit / Byte	7	6	5	4	3	2	1	0
0	Message code (20h, 21h, 22h)							
1	Queue Tag (00h - FFh)							

Table 11 defines the format for the queue tag messages. If the target implements tagged queuing, all of the queue tag messages are mandatory: HEAD OF QUEUE TAG, ORDERED QUEUE TAG, and SIMPLE QUEUE TAG.

If a target does not implement tagged queuing and a queue tag message is received or if a queue tag message is received for a target routine, it shall respond with a MESSAGE REJECT message and accept the I/O process as if it were untagged.

The queue tag messages are used to specify an identifier, called a queue tag, for an I/O process which establishes the I_T_L_Q nexus. The queue tag field is an 8-bit unsigned integer assigned by the initiator during an initial connection. The queue tag for every I/O process for each I_T_L nexus should be unique. If the target receives a queue tag that is currently in use for the I_T_L nexus, then it shall respond as defined in 6.5.2. A queue tag becomes available for re-assignment when the I/O process ends. The numeric value of a queue tag has no effect on the order of execution.

For each logical unit on each target, each initiator has up to 256 queue tags to assign to I/O processes. Thus a target with eight logical units could have up to 14336 I/O processes concurrently in existence if there were seven initiators on the bus.

Whenever an initiator connects to a target, the appropriate queue tag message shall be sent immediately following the IDENTIFY message and within the same MESSAGE OUT phase to establish the I_T_L_Q nexus for the I/O process. Only one I_T_L_Q nexus may be established during a connection. If a queue tag message is not sent, then only an I_T_x nexus is established for the I/O process (untagged command).

Whenever a target reconnects to an initiator to continue a tagged I/O process, the SIMPLE QUEUE TAG message shall be sent immediately following the IDENTIFY message and within the same MESSAGE IN phase to revive the I_T_L_Q nexus for the I/O process. Only one I_T_L_Q nexus may be revived during a reconnection. If the SIMPLE QUEUE TAG message is not sent, then only an I_T_x nexus is revived for the I/O process (untagged command).

If a target attempts to reconnect using an invalid queue tag, then the initiator should respond with an ABORT TAG message.

6.19.1 HEAD OF QUEUE TAG

The HEAD OF QUEUE TAG message specifies that the I/O process be placed first in that logical unit's command queue. An I/O process already being executed by the target shall not be pre-empted. A subsequent I/O process received with a HEAD OF QUEUE TAG message shall be placed at the head of the command queue for execution in last-in, first-out order.

6.19.2 ORDERED QUEUE TAG

The ORDERED QUEUE TAG message specifies that the I/O process be placed in that logical unit's command queue for execution in the order received. All queued I/O processes for the logical unit received prior to this I/O process shall be executed before this I/O process is executed. All queued I/O processes received after this I/O process shall be executed after this I/O process, except for I/O processes received with a HEAD OF QUEUE TAG message.

6.19.3 SIMPLE QUEUE TAG

The SIMPLE QUEUE TAG message specifies that the I/O process be placed in that logical unit's command queue. The order of execution is described in 6.8.

6.20 RELEASE RECOVERY

The RELEASE RECOVERY message is sent from an initiator to a target to terminate an extended contingent allegiance condition previously established by an INITIATE RECOVERY message. This message shall be sent immediately following the IDENTIFY message in the same MESSAGE OUT phase. The extended contingent allegiance condition ends upon successful receipt of the RELEASE RECOVERY message. The target shall go to the BUS FREE phase following successful receipt of this message.

If a RELEASE RECOVERY message is received by a target that implements extended contingent allegiance when no extended contingent allegiance condition is active, the message shall not be rejected and the target shall go to the BUS FREE phase.

6.21 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 I/O process 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 POINTER message or to the value at the point at which the last SAVE DATA POINTER message occurred for that nexus.

6.22 SAVE DATA POINTER

The SAVE DATA POINTER message is sent from a target to direct the initiator to copy the active data pointer to the saved data pointer for the current I/O process (see 5.2 for a description of pointers).

6.23 SYNCHRONOUS DATA TRANSFER REQUEST Message

Table 12: SYNCHRONOUS DATA TRANSFER REQUEST

Bit / Byte	7	6	5	4	3	2	1	0
0	Extended message (01h)							
1	Extended message length (03h)							
2	SYNCHRONOUS DATA TRANSFER REQUEST code (01h)							
3	Transfer Period Factor							
4	REQ/ACK Offset							

SYNCHRONOUS DATA TRANSFER REQUEST (SDTR) messages are used to negotiate a synchronous data transfer agreement between two SCSI devices. The agreement applies to all logical units and target

routines of both SCSI devices, regardless of the target or initiator role. That is, if SCSI device A, acting as an initiator negotiates a synchronous data transfer agreement with SCSI device B (in the target role), then the same data transfer agreement applies to SCSI devices A and B even if SCSI device B changes to the initiator role.

A synchronous data transfer 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.

An SDTR message exchange shall be initiated by an SCSI device whenever a previously-arranged data transfer agreement may have become invalid. The agreement becomes invalid after any condition which may leave the data transfer agreement in an indeterminate state such as:

(1) after a hard reset condition;
(2) after a BUS DEVICE RESET message and;
(3) after a power cycle.

In addition, an 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.

Re-negotiation at every selection is not recommended, since a significant performance impact is likely.

Due to historical problems with early host adapters that could not accept an SDTR message, some targets may not initiate synchronous negotiation after a power cycle as required by this standard. Host adapters that support synchronous mode may avoid the ensuing failure modes when the target is independently power cycled by initiating a synchronous negotiation on each REQUEST SENSE and INQUIRY command.

The SDTR message exchange establishes the permissible transfer periods and the REQ/ACK offsets for all logical units and target routines on the two devices. This agreement only applies to data phases.

The transfer period factor times four is the value of the transfer period. The transfer period is the minimum time allowed between leading edges of successive REQ pulses and of successive ACK pulses to meet the device requirements for successful reception of data.

The REQ/ACK offset is the maximum number of REQ pulses allowed to be outstanding before the leading edge of its corresponding ACK pulse is received at the target. This value is chosen to prevent overflow conditions in the device's reception buffer and offset counter. A REQ/ACK offset value of zero shall indicate asynchronous data transfer mode; a value of FFh shall indicate unlimited REQ/ACK offset.

The originating device (the 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 device can 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 device when transmitting data shall respect the limits set by the other's SDTR message, but it is permitted to transfer data with larger transfer periods, smaller REQ/ACK offsets, or both than specified in the other's SDTR message. The successful completion of an exchange of SDTR messages implies an agreement as follows:

Responding Device SDTR response	Implied Agreement
1) Non-zero REQ/ACK offset	Each 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.
2) REQ/ACK offset equal to zero	Asynchronous transfer
3) MESSAGE REJECT message	Asynchronous transfer

If the initiator recognizes that negotiation is required, it asserts the ATN signal and sends a SDTR message to begin the negotiating process. After successfully completing the MESSAGE OUT phase, the target shall respond with the proper SDTR message. If an abnormal condition prevents the target from returning an appropriate response, both devices shall go to asynchronous data transfer mode for data transfers between the two devices.

Following target response (1) above, the implied agreement for synchronous operation shall be considered to be negated by both the initiator and the target if the initiator asserts the ATN signal and the first message out is either MESSAGE PARITY ERROR or MESSAGE REJECT. In this case, both devices shall go to asynchronous data transfer mode for data transfers between the two devices. For the MESSAGE PARITY ERROR case, the implied agreement shall be reinstated if a re-transmittal of the second of the pair of messages is successfully accomplished. After a vendor-specific number of retry attempts (greater than zero), if the target receives a MESSAGE PARITY ERROR message, it shall terminate the retry activity. This may be done either by changing to any other information transfer phase and transferring at least one byte of information or by going to the BUS FREE phase. The initiator shall accept such action as aborting the negotiation, and both devices shall go to asynchronous data transfer mode for data transfers between the two devices.

If the target recognizes that negotiation is required, it sends an SDTR message to the initiator. Prior to releasing the ACK signal on the last byte of the SDTR message from the target, the initiator shall assert the ATN signal and respond with its SDTR message or with a MESSAGE REJECT message. If an abnormal condition prevents the initiator from returning an appropriate response, both devices shall go to asynchronous data transfer mode for data transfers between the two devices.

Following an initiator's responding SDTR message, an implied agreement for synchronous operation shall not be considered to exist until the target leaves the MESSAGE OUT phase, indicating that the target has accepted the negotiation. After a vendor-specific number of retry attempts (greater than zero), if the target has not received the initiator's responding SDTR message, it shall go to the BUS FREE phase without any further information transfer attempt (see 5.1.3). This indicates that a catastrophic error condition has occurred. Both devices shall go to asynchronous data transfer mode for data transfers between the two devices.

If, following an initiator's responding SDTR message, the target shifts to MESSAGE IN phase and the first message in is MESSAGE REJECT, the implied agreement shall be considered to be negated and both devices shall go to asynchronous data transfer mode for data transfers between the two devices.

The implied synchronous agreement shall remain in effect until a BUS DEVICE RESET message is received, until a hard reset condition occurs, or until one of the two SCSI devices elects to modify the agreement. The default data transfer mode is asynchronous data transfer mode. The default data transfer mode is entered at power on, after a BUS DEVICE RESET message, or after a hard reset condition.

6.24 TARGET TRANSFER DISABLE

The TARGET TRANSFER DISABLE message is sent from an initiator to a target to request that subsequent reconnections for data transfer on the I/O process be done by the initiator instead of the target. The target may reconnect for other purposes, but shall not enter a data phase on a target reconnection. SCSI devices that implement this message shall also implement the CONTINUE I/O PROCESS message.

This message shall be sent as the last message of the first MESSAGE OUT phase of an initial connection. The target may continue the I/O process, including any DATA OUT phases on the initial connection, until the target would normally disconnect, but the target shall not reconnect to transfer data. That is, the target shall not enter a DATA IN phase on the initial connection and the target shall not enter any data phase on any subsequent target reconnection for the I/O process.

When the target is ready to transfer data for a disconnected I/O process for which a TARGET TRANSFER DISABLE message has been sent, the target shall reconnect to the initiator for the I/O process (via a RESELECTION phase, an IDENTIFY message, and an optional SIMPLE QUEUE TAG message), send a DISCONNECT message, and, if the initiator does not respond with a MESSAGE REJECT message, go to the BUS FREE phase. This connection serves to notify the initiator that the I/O process is ready for data transfer. If the initiator rejects the DISCONNECT message, the target may enter a data phase; otherwise, the initiator may reconnect to the I/O process as described in the CONTINUE I/O PROCESS message to perform the data transfer.

Initiators should avoid sending the TARGET TRANSFER DISABLE message to targets which have not implemented this message. Such targets may not respond as described in this section. An initiator can determine whether a target implements this message by examining the TranDis bit in the standard INQUIRY data (see SCSI-3 Command Set).

6.25 TERMINATE I/O PROCESS

The TERMINATE I/O PROCESS message is sent from the initiator to the target to terminate the current I/O process without corrupting the medium.

With the following exceptions, the target shall terminate the current I/O process and return COMMAND TERMINATED status. The sense key shall be set to NO SENSE. The additional sense code and qualifier are set to I/O PROCESS TERMINATED.

If the associated I/O process involves a data phase, the target shall set the valid bit in the sense data to one and set the information field as follows:

1) If the command descriptor block specifies an allocation length or parameter list length, the information field shall be set to the difference (residue) between the number of bytes successfully transferred and the requested length.

2) If the command descriptor block specifies a transfer length field, the information field shall be set as defined in the REQUEST SENSE command (see SCSI-3 Command Set).

If an error is detected for the associated I/O process the target shall ignore the TERMINATE I/O PROCESS message.

If the operation requested for the associated I/O process has been completed but status has not been returned, the target shall ignore the TERMINATE I/O PROCESS message.

If the target does not support this message or is unable to stop the current I/O process, it shall send a MESSAGE REJEECT message to the initiator and continue the I/O process in a normal manner.

The effect of a TERMINATED I/O PROCESS message on the command queue depends on the queue error recovery option specified in the control mode page (see SCSI-3 Command Set) and on whether or not a contingent allegiance condition is generated.

The TERMINATE I/O PROCESS message provides a means for the initiator to request the target to reduce the transfer length of the current command to the amount that has already been transferred. The initiator can use the sense data to determine the actual number of bytes or blocks that have been transferred. This message is normally used by the initiator to stop a lengthy read, write, or verify operation when a higher-priority command is available to be executed. It is up to the initiator to complete the terminated command at a later time, if required.

6.26 WIDE DATA TRANSFER REQUEST Message

Table 13: WIDE DATA TRANSFER MESSAGE

Bit / Byte	7	6	5	4	3	2	1	0
0	Extended message (01h)							
1	Extended message length (02h)							
2	WIDE DATA TRANSFER REQUEST code (03h)							
3	Transfer Width Exponent							

A WIDE DATA TRANSFER REQUEST (WDTR) message exchange shall be initiated by an SCSI device whenever a previously-arranged transfer width agreement may have become invalid. The agreement becomes invalid after any condition which may leave the data transfer agreement in an indeterminate state such as:

1) after a hard reset condition;
2) after a BUS DEVICE RESET message and;
3) after a power cycle.

In addition, an SCSI device may initiate an WDTR message exchange whenever it is appropriate to negotiate a new transfer width agreement. SCSI devices that are capable of wide data transfers (greater than eight bits) shall not respond to an WDTR message with a MESSAGE REJECT message.

Re-negotiation at every selection is not recommended, since a significant performance impact is likely.

The WDTR message exchange establishes an agreement between two SCSI devices on the width of the data path to be used for DATA phase transfers between the two devices. This agreement applies to

DATA IN and DATA OUT phases only. All other information transfer phases shall use an eight-bit data path.

If an SCSI device implements both wide data transfer option and synchronous data transfer option, 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 an SCSI device that accepts a WDTR message shall reset the synchronous agreement to asynchronous mode.

The transfer width is two to the transfer width exponent bytes wide. The transfer width that is established applies to all logical units on both SCSI devices. Valid transfer widths are 8 bits (m = 00h), 16 bits (m = 01h), and 32 bits (m = 02h). Values of m greater than 02h are reserved.

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 can 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 follows:

Responding Device WDTR Response	Implied Agreement
(1) Non-zero transfer width	Each device transmits and receives data with a transfer width equal to the responding SCSI device's transfer width.
(2) Transfer width equal to zero	Eight-bit Data Transfer
(3) MESSAGE REJECT message	Eight-bit Data Transfer

If the initiator recognizes that negotiation is required, it asserts the ATN signal and sends a WDTR message to begin the negotiating process. After successfully completing the MESSAGE OUT phase, the target shall respond with the proper WDTR message. If an abnormal condition prevents the target from returning an appropriate response, both devices shall go to eight-bit data transfer mode for data transfers between the two devices.

Following target response (1) above, the implied agreement for wide data transfers shall be considered to be negated by both the initiator and the target if the initiator asserts ATN and the first message out is either MESSAGE PARITY ERROR or MESSAGE REJECT. In this case, both devices shall go to eight-bit data transfer mode for data transfers between the two devices. For the MESSAGE PARITY ERROR case, the implied agreement shall be reinstated if a re-transmittal of the second of the pair of messages is successfully accomplished. After a vendor-specific number of retry attempts (greater than zero), if the target receives a MESSAGE PARITY ERROR message, it shall terminate the retry activity. This may be done either by changing to any other information transfer phase and transferring at least one byte of information or by going to the BUS FREE phase (see 5.1.3). The initiator shall accept such action as aborting the negotiation, and both devices shall go to eight-bit data transfer mode for data transfers between the two devices.

If the target recognizes that negotiation is required, it sends a WDTR message to the initiator. Prior to releasing the ACK signal on the last byte of the WDTR message from the target, the initiator shall assert the ATN signal and respond with its WDTR message or with a MESSAGE REJECT message. If an abnormal condition prevents the initiator from returning an appropriate response, both devices shall go to eight-bit data transfer mode for data transfers between the two devices.

Following an initiator's responding WDTR message, an implied agreement for wide data transfer operation shall not be considered to exist until the target leaves the MESSAGE OUT phase, indicating that the target has accepted the negotiation. After a vendor-specific number of retry attempts (greater than zero), if the target has not received the initiator's responding WDTR message, it shall go to the BUS FREE phase without any further information transfer attempt (see 5.1.3). This indicates that a catastrophic error condition has occurred. Both devices shall go to eight-bit data transfer mode for data transfers between the two devices.

If, following an initiator's responding WDTR message, the target shifts to MESSAGE IN phase and the first message in is MESSAGE REJECT, the implied agreement shall be considered to be negated and both devices shall go to eight-bit data transfer mode for data transfers between the two devices.

The implied transfer width agreement shall remain in effect until a BUS DEVICE RESET message is received, until a hard reset condition occurs, or until one of the two SCSI devices elects to modify the agreement. The default data transfer width is eight-bit data transfer mode. The default data transfer mode is entered at power on, after a BUS DEVICE RESET message, or after a hard reset condition.

132

7 SCSI Interlocked Commands and Status

This section defines the SCSI command and status structures and gives several examples.

By keeping to a minimum the functions essential to communicate via this protocol, a wide range of peripheral devices of varying capability can operate in the same environment. Because subsets of the full architecture may be implemented, optional functions are noted.

7.1 Command Implementation Requirements

The first byte of all SCSI commands shall contain an operation code as defined in this standard. Targets shall implement all commands with a mandatory operation code (see 7.1.2) both in section 7 and in the appropriate section for their device type.

7.1.1 Reserved

Reserved bits, fields, bytes, and code values are set aside for future standardization. Their use and interpretation may be specified by future extensions to this standard. A reserved bit, field, or byte shall be set to zero, or in accordance with a future extension to this standard. A target that receives a reserved bit, field, or byte that is not zero or receives a reserved code value shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. It shall also be acceptable for a target to interpret a bit, field, byte, or code value in accordance with a future extension to this standard.

7.1.2 Operation Code Types

The operation code types are defined in Table 14.

Table 14: Operation Code Type

Operation Code Type	Description
M	Mandatory - Commands so designated shall be implemented in order to meet the minimum requirement of this standard.
O	Optional - Commands so designated, if implemented, shall be implemented as defined in this standard.
V	Vendor specific - Operation codes so designated are available for vendor defined commands. See the vendor specifications where compatibility is desired.
R	Reserved - Operation codes so designated shall not be used. They are reserved for future extensions to this standard.

7.2 Command Descriptor Block

A command is communicated by sending a command descriptor block to the target. For several commands, the command descriptor block is accompanied by a list of parameters sent during the DATA OUT phase. See the specific commands in the SCSI-3 Command Set for detailed information.

The command descriptor block always has an operation code as its first byte and a control byte as its last byte.

For all commands, if there is an invalid parameter in the command descriptor block, then the target shall terminate the command without altering the medium.

Table 15: Typical Command Descriptor Block for Six-byte Commands

Bit Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			(MSB)				
2	Logical Block Address (if required)							
3								(LSB)
4	Transfer Length (if required) Parameter List Length (if required) Allocation Length (if required)							
5	Control							

Table 16: Typical Command Descriptor Block for Ten-byte Commands

Bit Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	(MSB)							
3	Logical Block Address (if required)							
4								
5								(LSB)
6	Reserved							
7	(MSB)			Transfer Length (if required) Parameter List Length (if required)				
8				Allocation Length (if required)				(LSB)
9	Control							

Table 17: Typical Command Descriptor Block for Twelve-byte Commands

Bit Byte	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	(MSB)							
3	Logical Block Address (if required)							
4								
5								(LSB)
6	(MSB)							
7	Transfer Length (if required) Parameter List Length (if required) Allocation Length (if required)							
8								
9								(LSB)
10	Reserved							
11	Control							

7.2.1 Operation Code

The operation code (Table 18) of the command descriptor block has a group code field and a command code field. The three-bit group code field provides for eight groups of command codes. The five-bit command code field provides for thirty-two command codes in each group. Thus, a total of 256 possible operation codes exist. Operation codes are defined in the subsequent Sections of this document.

The group code specifies one of the following groups:
Group 0 - six-byte commands
Group 1 - ten-byte commands
Group 2 - ten-byte commands
Group 3 - reserved
Group 4 - reserved
Group 5 - twelve-byte commands
Group 6 - vendor specific
Group 7 - vendor specific

133

Table 18: Operation Code

Bit	7	6	5	4	3	2	1	0
	Group Code			Command Code				

7.2.2 Logical Unit Number

The logical unit number is defined in the IDENTIFY message (5.6.7). The target shall ignore the logical unit number specified within the command descriptor block if an IDENTIFY message was received. It is recommended that the logical unit number in the command descriptor block be set to zero.

7.2.3 Logical Block Address

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 logical block address. The ten-byte and the twelve-byte command descriptor blocks contain 32-bit logical block addresses. Logical block addresses in additional parameter data have their length specified for each occurrence. See the specific command descriptions in the SCSI-3 Command Set.

7.2.4 Transfer Length

The transfer length field 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 in the SCSI-3 Command Set for further information.

Commands that use one byte for the 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.

In commands that use multiple bytes for the transfer length, a transfer length of zero indicates that no data transfer shall take place. A value of one or greater indicates the number of blocks that shall be transferred.

Refer to the specific command description for further information.

7.2.5 Parameter List Length

The parameter list length is used to specify the number of bytes sent during the DATA OUT phase. This field is typically used in command descriptor blocks for parameters that are sent to a target (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.

7.2.6 Allocation Length

The allocation length field specifies the maximum number of bytes that an initiator 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 target 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.

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 initiator.

7.2.7 Control Field

The control field is the last byte of every command descriptor block. The control field is defined in Table 19.

Table 19: Control Field

Bit	7	6	5	4	3	2	1	0
	Vendor specific		Reserved				Flag	Link

The flag bit specifies which message the target shall return to the initiator if the link bit is one and the command completes without error. The flag bit is typically used to cause an interrupt in the initiator between linked commands. Implementation of the flag bit is optional.

The flag bit should be set to zero if the link bit is zero. If link bit is zero and the flag bit is one, the target shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST.

If the flag bit is zero and the link bit is one, and if the command completes successfully, the target shall send the LINKED COMMAND COMPLETE message. If the flag bit is one and the link bit is one, and if the command completes successfully, the target shall send the LINKED COMMAND COMPLETE (WITH FLAG) message.

The link bit is used to continue the I/O process across multiple commands. Implementation of the link bit is optional.

A link bit of one indicates that the initiator requests a continuation of the I/O process and that the target should enter the command phase upon successful completion of the current command.

If the link bit is one, and if the command completes successfully, the target shall return INTERMEDIATE or INTERMEDIATE-CONDITION MET status and shall then send one of the two messages defined by the flag bit.

If either of the link and flag bits are set to one, and the target does not implement linked commands, it shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST.

7.3 Status

The status byte and status byte code are specified in Tables 20 and 21. A status byte shall be sent from the target to the initiator during the STATUS phase at the completion of each command unless the command is terminated by one of the following events:

a) an ABORT message;
b) an ABORT TAG message;
c) a BUS DEVICE RESET message;
d) a CLEAR QUEUE message;
e) a hard reset condition;
f) an unexpected disconnect (see 5.1.3).

The STATUS phase normally occurs at the end of a command but in some case may occur prior to transferring the command descriptor block.

Table 20: Status Byte

Bit	7	6	5	4	3	2	1	0
	Reserved		Status Byte Code					Reserved

Table 21: Status Byte Code

Bits of Status Byte 7	6	5	4	3	2	1	0	Status
R	R	0	0	0	0	0	R	GOOD
R	R	0	0	0	0	1	R	CHECK CONDITION
R	R	0	0	0	1	0	R	CONDITION MET
R	R	0	0	1	0	0	R	BUSY
R	R	0	1	0	0	0	R	INTERMEDIATE
R	R	0	1	0	1	0	R	INTERMEDIATE-CONDITION MET
R	R	0	1	1	0	0	R	RESERVATION CONFLICT
R	R	1	0	0	0	1	R	COMMAND TERMINATED
R	R	1	0	1	0	0	R	QUEUE FULL
All Other Codes								Reserved
Key: R = Reserved bit								

A definition of the status byte codes is given below.

GOOD. This status indicates that the target has successfully completed the command.

CHECK CONDITION. This status indicates that a contingent allegiance condition has occurred (see SCSI-3 Command Set).

CONDITION MET. This status or INTERMEDIATE-CONDITION MET is returned whenever the requested operation is satisfied.

BUSY. This status indicates that the target is busy. This status shall be returned whenever a target 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 every successfully completed command in a series of linked commands (except the last command), unless the command is terminated with CHECK CONDITION, RESERVATION CONFLICT, or COMMAND TERMINATED status. If INTERMEDIATE or INTERMEDIATE-CONDITION MET status is not returned, the series of linked commands is terminated and the I/O process is ended.

INTERMEDIATE-CONDITION MET. This status is the combination of the CONDITION MET and INTERMEDIATE statuses.

RESERVATION CONFLICT. This status shall be returned whenever an initiator attempts to access a logical unit or an extent within a logical unit that is reserved with a conflicting reservation type for another SCSI device (see the RESERVE and RESERVE UNIT commands in the SCSI-3 Command Set). The recommended initiator recovery action is to issue the command again at a later time.

COMMAND TERMINATED. This status shall be returned whenever the target terminates the current I/O process after receiving a TERMINATE I/O PROCESS message. This status also indicates that a contingent allegiance condition has occurred (see SCSI-3 Command Set).

QUEUE FULL. This status shall be implemented if tagged queuing is implemented. This status is returned when a SIMPLE QUEUE TAG, ORDERED QUEUE TAG, or HEAD OF QUEUE TAG message is received and the command queue is full. The I/O process is not placed in the command queue.

7.4 Command Processing Considerations and Exception Conditions

The following sections describe some exception conditions and errors associated with command processing and the sequencing of commands.

7.4.1 Incorrect Initiator Connection

An incorrect initiator connection occurs on a reconnection if:

(1) an initiator attempts to reconnect to an I/O process, and
(2) a soft reset condition has not occurred, and
(3) the initiator does not send an ABORT, ABORT TAG, BUS DEVICE RESET, CLEAR QUEUE, CONTINUE I/O PROCESS, or TERMINATE I/O PROCESS message during the same MESSAGE OUT phase as the IDENTIFY message.

An incorrect initiator connection also occurs on an initial connection when an initiator:

(1) attempts to establish an I_T_L_Q nexus when an I_T_L nexus already exists from a previous connection, or
(2) 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 target that detects an incorrect initiator connection shall abort all I/O processes 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.

If an initiator reconnects to an I/O process and a soft reset condition has occurred, the target shall meet the requirements of the soft reset alternative (see 5.1.2.2).

An incorrect initiator connection may be indicative of a serious error and, if not detected, could result in an I/O process 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).

Some targets may not detect an incorrect initiator connection until after the command descriptor block has been received.

7.4.2 Interlocked Asynchronous Event Notification

Notification of an asynchronous event is performed using the SEND command with the AEN bit set to one. The information identifying the condition being reported shall be returned during the DATA OUT phase of the SEND command (see SCSI-3 Command Set).

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 processor that initiated the I/O process.

The asynchronous event notification protocol can be used to notify processor devices that a system resource has become available. If a target 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 to SCSI devices that return processor device type with an AENC bit of one in response to an INQUIRY command. The INQUIRY command should be issued 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 device deems it appropriate or when an event occurs that may invalidate the current information. (See SYNCHRONOUS DATA TRANSFER REQUEST message for examples of these events.)

Each SCSI device that returns processor device type with an AENC 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. An SCSI device returning CHECK CONDITION status is issued a REQUEST SENSE command. This clears any pending unit attention condition. An SCSI device that returns processor device type with an AENC bit of one and returns GOOD status when issued a TEST UNIT READY command shall accept a SEND command with an AEN bit of one.

An SCSI device which can use asynchronous event notification at initialization time should provide means to defeat these notifications. This can be done with a switch or jumper wire. Devices which 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 device conducts a survey with INQUIRY commands to be sure that the devices on the SCSI bus are appropriate destinations for SEND commands with an AEN bit of one. (The devices on the bus or the SCSI ID assignments may have changed.)

7.4.3 Unexpected Reselection

An unexpected reselection occurs if an SCSI device attempts to reconnect to an I/O process for which a nexus does not exist. An SCSI device should respond to an unexpected reselection by sending an ABORT message.

7.4.4 Unit Attention Condition

The target shall generate a unit attention condition for each initiator on each valid logical unit whenever the target has been reset by a BUS DEVICE RESET message, a hard reset condition, or by a power-on reset. For dual port implementations, the unit attention condition for:

(1) the hard reset condition,
(2) the BUS DEVICE RESET message, and

(3) the BUS DEVICE RESET OTHER PORT message

only affects the initiators on one port as described in sections 5.1.2.1, 6.1, and 6.4. The target shall also generate a unit attention condition on the affected logical unit(s) for each initiator whenever one of the following events occurs:

a) A removable medium may have been changed.
b) The mode parameters in effect for this initiator have been changed by another initiator.
c) The version or level of microcode has been changed.
d) Tagged commands queued for this initiator were cleared by another initiator.
e) INQUIRY data has been changed.
f) The mode parameters in effect for the initiator have been restored from non-volatile memory.
g) A change in the condition of a synchronized spindle.
h) Any other event occurs that requires the attention of the initiator.

Targets may queue unit attention conditions on logical units. After the first unit attention condition is cleared, another unit attention condition may exist (e.g., a power on condition followed by a microcode change condition).

137

The unit attention condition shall persist on the logical unit for each initiator until that initiator clears the condition as described in the following paragraphs.

If an INQUIRY command is received from an initiator to a logical unit with a pending unit attention condition (before the target generates the contingent allegiance condition), the target shall perform the INQUIRY command and shall not clear the unit attention condition. If the INQUIRY command is received after the target has generated the contingent allegiance condition for a pending unit attention condition, then the unit attention condition on the logical unit shall be cleared, and the target shall perform the INQUIRY command.

If any other command is received after the target has generated the contingent allegiance condition for a pending unit attention condition, the unit attention condition on the logical unit shall be cleared, and if no other unit attention condition is pending the target shall perform the command. If another unit attention condition is pending the target shall not perform the command and shall generate another contingent allegiance condition.

If a REQUEST SENSE command is received from an initiator with a pending unit attention condition (before the target generates the contingent allegiance condition), then the target shall either:

1) report any pending sense data and preserve the unit attention condition on the logical unit, or,
2) report the unit attention condition, may discard any pending sense data, and clear the unit attention condition on the logical unit for that initiator.

If the target has already generated the contingent allegiance condition for the unit attention condition, the target shall perform the second action listed above.

If an initiator issues a command other than INQUIRY or REQUEST SENSE while a unit attention condition exists for that initiator (prior to generating the contingent allegiance condition for the unit attention condition), the target shall not perform the command and shall report CHECK CONDITION status unless a higher priority status as defined by the target is also pending (e.g., BUSY or RESERVATION CONFLICT).

If after generating the contingent allegiance condition for a pending unit attention condition, the next command received from that initiator on the logical unit is not REQUEST SENSE, then that command shall be performed and the unit attention condition shall be cleared for that initiator on the logical unit and the sense data is lost (see 6.6).

If a target becomes a temporary initiator to issue a SEND command with an AEN bit of one, which informs the initiator (temporary target) of the unit attention condition, and the SEND command completes with GOOD status, then the target shall clear the unit attention condition for that initiator on the logical unit (see 6.6.5).

ANNEX A

(informative)

Single Command Example

An I/O process containing one untagged READ command is used in this section to illustrate a simple I/O process on the SCSI bus. This example does not include error or exception conditions.

The initiator has one set of active pointers that includes a command pointer, a data pointer, and a status pointer. In addition, the initiator has one set of saved pointers for each I/O process that it is able to concurrently manage. The initiator sets up the saved pointers to point to the appropriate bytes for the I/O process and copies the saved pointers to the active pointers. It then arbitrates for the SCSI bus, and upon winning arbitration, selects the target. Once the target is selected, the target assumes control of the I/O process.

During the SELECTION phase, the initiator asserts the ATN signal to inform the target that the initiator wishes to send a message. The target enters the MESSAGE OUT phase and transfers the IDENTIFY message from the initiator. This message informs the target of which logical unit is to be used. At this point, an I_T_L nexus has been established for the I/O process. This nexus associates the initiator's pointers with the I/O process.

The target switches to the COMMAND phase and transfers the command descriptor block from the initiator. In this case, the command descriptor block contains a READ command. The target interprets the command and switches to the DATA IN phase, transfers the data, switches to STATUS phase, sends GOOD status, switches to MESSAGE IN phase, and transfers a COMMAND COMPLETE message. After successfully sending the COMMAND COMPLETE message, the target goes to the BUS FREE phase by releasing the BSY signal and the I/O process ends.

ANNEX B

Informative

Disconnect Example

In the above single command example, the length of time necessary to obtain the data may require a time-consuming physical positioning operation. In order to improve system throughput, the target may disconnect from the initiator, thereby freeing the SCSI bus to allow other I/O process to occur.

After the target has received the READ command (and has determined that there will be a delay), it disconnects from the SCSI bus by sending a DISCONNECT message and by going to the BUS FREE phase.

After the target retrieves the requested data from the peripheral device it arbitrates for the SCSI bus. Upon winning arbitration, it reselects the initiator and sends an IDENTIFY message to the initiator via the MESSAGE IN phase. This revives the I_T_L nexus so that the initiator can retrieve the correct set of pointers for the I/O process. The initiator restores the active pointers to their most recent saved values (which, in this case, are the initial values) and the target continues (as in the single command example) to finish the I/O process.

If target wishes to disconnect after transferring part of the data (e.g., while crossing a cylinder boundary), it may do so by sending a SAVE DATA POINTER message and a DISCONNECT message to the initiator and then disconnecting. When reconnection is completed, the current data pointer is restored to its value immediately prior to the SAVE DATA POINTER message.

On those occasions when an error or exception condition occurs and the target elects to repeat the information transfer, the target may repeat the transfer by either issuing a RESTORE POINTERS message or by disconnecting without issuing a SAVE DATA POINTER message. When reconnection is completed, the most recent saved pointer values are restored.

ANNEX C

(informative)

Linked Command Example

An I/O process may contain multiple commands "linked" together. Upon completing a linked command successfully, the target automatically proceeds to the next linked command for the I/O process. All commands in a series of linked commands are addressed to the same nexus and are part of a single I/O process.

The commands are not entirely independent. When using the relative address bit (see SCSI-3 Command Set), the address of the last logical block accessed by one of the commands is available to the next command. Thus one can search for a particular data pattern using a SEARCH DATA command and then read the logical block containing the data pattern with a READ command linked to the SEARCH DATA command. One can also read a logical block at a specified displacement from the block containing the data pattern.

A LINKED COMMAND COMPLETE or LINKED COMMAND COMPLETE (WITH FLAG) message is sent from the target to the initiator to indicate that a linked command completed. The initiator then updates the saved pointers for the nexus so that subsequent transfers from the target reference the next command of the series. Command processing of linked and single commands is similar except that relative addressing is permitted in linked commands.

For example, a successful completion of a SEARCH DATA EQUAL command causes the target to continue with the linked READ command from the initiator. If the relative address bit in the READ command has been set to one, and the address field of the READ command is set to zero, the target transfers the successfully searched block to the initiator.

ANNEX D

(informative)

An example of I/O process queuing benefits from the consideration of the execution of a number of commands. After each command, the state of the queue kept in the target is shown to indicate the function actually performed by the queuing.

Typical Sequences for Tagged Queuing

An I/O process using tagged queuing uses the following sequences for normal execution. The initiator first arbitrates for the SCSI bus, and after successfully obtaining the SCSI bus, selects the appropriate SCSI device. The ATN signal is asserted during the SELECTION phase to indicate that a MESSAGE OUT phase is requested by the initiator. The first message byte transferred is an IDENTIFY message. The ATN signal continues to be asserted during the MESSAGE OUT phase to indicate that the initiator has another message. The second message byte transferred is the first byte of the appropriate queue tag message, in this case a SIMPLE QUEUE TAG message. The third and last message byte is transmitted containing the second byte of the queue tag message, the queue tag. As it is transferred, the ATN signal is negated to indicate that no more message bytes are available. The target then transfers the command descriptor block. Assuming the command requires disconnection, the target transmits a DISCONNECT message to the initiator and then enters the BUS FREE phase. The target places the command, identified by the I_T_L_Q nexus, at the appropriate place in the command queue.

When the target removes I/O processes from the queue for execution, a physical latency period may occur. At the end of this period, when the target is prepared to transfer the appropriate data, the target begins an ARBITRATION phase and, upon winning, enters a RESELECTION phase. After a successful reselection, the target sends the IDENTIFY message followed by a SIMPLE QUEUE TAG message with the queue tag value originally sent by the initiator. The initiator uses the I_T_L_Q nexus to identify the correct set of pointers and control blocks associated with the I/O process and to establish the necessary conditions for data transfer. The target begins data transfer. When the data transfer is successfully completed, the target returns GOOD status and terminates the I/O process with a COMMAND COMPLETE message.

Example of Tagged Queuing

An example of the execution of five queued I/O processes is described to demonstrate how tagged queuing operates. ABORTagged I/O processes are from one initiator to a single logical unit of a single target. The five I/O processes are defined in Table 22. The target is a direct-access device. At the time the I/O processes are first being executed, it is assumed that the actuator is in position to access logical block 10000.

Table 22: Commands in Order Received by Target

Command	Queue Tag Message	Queue Tag Value	Logical Block Address	Transfer Length	Status
READ	SIMPLE	01h	10000	1000	Queued
READ	SIMPLE	02h	100	1	Queued
READ	ORDERED	03h	1000	1000	Queued
READ	SIMPLE	04h	10000	1	Queued
READ	SIMPLE	05h	2000	1000	Queued

The optimum order would require that those blocks close to the actuator position be the first blocks accessed, followed by those increasingly far from the actuator position. However, the command with queue tag 03h is an ordered I/O process, so that all simple I/O processes transferred previously must be executed before, while all simple I/O processes transferred after the ordered I/O process must be executed after the ordered I/O process.

If a target supports an optimizing algorithm the actual order in which the I/O processes are executed could be as shown in Table 23.

Table 23: Commands in Order of Execution

Command	Queue Tag Message	Queue Tag Value	Logical Block Address	Transfer Length	Status
READ	SIMPLE	01h	10000	1000	Queued
READ	SIMPLE	02h	100	1	Queued
READ	ORDERED	03h	1000	1000	Queued
READ	SIMPLE	05h	2000	1000	Queued
READ	SIMPLE	04h	10000	1	Queued

I/O processes with queue tag values 01h and 02h are executed in the order received since the actuator is already in position to execute I/O process 01h. I/O process 02h must be executed before I/O process 04h or 05h because the ordered I/O process 03h was transmitted after I/O processes 01h and 02h but before I/O processes 04h and 05h. I/O process 03h is then executed after I/O process 02h. The I/O processes 04h and 05h are executed after the ordered I/O process 03h. I/O process 05h is executed before I/O process 04h because the actuator is in position to access block 2000 after executing I/O process 03h. I/O process 04h is executed last.

As an example of the operation of the HEAD OF QUEUE TAG I/O process, consider that a new I/O process, identified by a HEAD OF QUEUE TAG message with a queue tag of 08h, is transmitted to the

target while the ordered I/O process 03h is being executed. The I/O process 03h continues execution, but the new HEAD OF QUEUE TAG I/O process is placed in the queue for execution before all subsequent I/O processes. In this case, the queue for execution after the ordered I/O process 03h was executed would appear as shown in Table 24.

Table 24: Modified by HEAD OF QUEUE TAG Message

Command	Queue Tag Message	Queue Tag Value	Logical Block Address	Transfer Length	Status
READ	ORDERED	03h	1000	1000	Executing
READ	HEAD OF QUEUE	08h	0	8	Queued
READ	SIMPLE	05h	2000	1000	Queued
READ	SIMPLE	04h	10000	1	Queued

To obtain maximum performance gains using tagged queuing requires careful implementation of the queuing algorithms in the target. In addition, initiators should allow a maximum number of simple I/O processes to be executed with a minimum number of ordered I/O processes. RESERVE and RELEASE commands, SET LIMITS commands, and appropriate software locking conventions should be used to guarantee the proper relationship between the commands executed and the data stored on the peripheral devices. These conventions are not defined by this standard.