JTC 1

INTERNATIONAL STANDARD

ISO
9316

First edition
1989-07-15

Information processing systems — Small Computer System Interface (SCSI)

Systèmes de traitement de l'information — Interface système pour petits ordinateurs (SCSI)

Reference number
ISO 9316 : 1989 (E)

Contents

Annexes

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 9316 was prepared by Technical Committee ISO/TC 97, *Information processing systems.*

Annexes A, B, C, D, E and F are for information only.

International Organization for Standardization
Case postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

Information processing systems — Small Computer System Interface (SCSI)

1 Scope

This International Standard provides the mechanical, electrical, and functional requirements for a small computer input/output bus and command sets for peripheral device types commonly used with small computers.

The small computer system interface, described in this International Standard, is a local I/O bus that can be operated at data rates up to 4 Mbytes/s depending upon circuit implementation choices. The primary objective of the interface is to provide host computers with device independence within a class of devices. Thus, different disk drives, tape drives, printers, and even communication devices can be added to the host computer(s) without requiring modifications to generic system hardware or software. Provision is made for the addition of nongeneric features and functions through vendor unique fields and codes.

The interface uses logical rather than physical addressing for all data blocks. For direct access devices, each logical unit may be interrogated to determine how many blocks it contains. A logical unit may coincide with all or part of a peripheral device.

Provision is made for cable lengths up to 25 m using differential drivers and receivers. A single-ended driver and receiver configuration is defined for cable lengths of up to 6 m and is primarily intended for applications within a cabinet.

The interface protocol includes provision for the connection of multiple initiators (SCSI devices capable of initiating an operation) and multiple targets (SCSI devices capable of responding to a request to perform an operation). Optional 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 time to complete arbitration is independent of the number of devices that are contending and can be completed in less than 10 µs.

The physical characteristics are described in clause 4. There are two electrical alternatives: single-ended and differential. Single-ended and differential devices are electrically different and shall not be mixed on the same bus. In addition, there are several options: shielded or unshielded connectors may be used and parity may or may not be implemented.

Clause 5 describes the logical characteristics of the interface. An arbitration option is defined to permit multiple initiators and to permit concurrent I/O operations. All SCSI devices are required to be capable of operating with the defined asynchronous transfer protocol. In addition, an optional synchronous transfer protocol is defined. Clause 5 also specifies a message protocol for control of the interface. In most cases, messages are not directly apparent to the host computer software. Only one message, COMMAND COMPLETE, is mandatory; all others are optional and are not necessarily implemented. Note that some options (e.g., synchronous transfer) require the implementation of certain messages.

The SCSI command structure is specified in clause 6. Commands are classified as mandatory (M), extended (E), optional (O), or vendor unique (V). SCSI devices shall implement all mandatory commands defined for the appropriate device type and may implement other commands as well. Extended SCSI devices shall implement all extended plus all mandatory commands and may implement other commands as well. Extended SCSI devices contain commands that facilitate the writing of self-configuring software drivers that can "discover" all necessary attributes without prior knowledge of specific peripheral characteristics (such as storage capacity). Extended commands for direct access devices also implement a very large logical block address space (2^{32} blocks), although mandatory commands for direct access devices implement a somewhat smaller logical block address space (2^{21} blocks).

Clause 7 specifies those commands that have a consistent meaning for all device types.

Clauses 8 through 13 contain commands for direct-access (*e.g.*, magnetic disk), sequential-access (*e.g.*, magnetic tape), printer, processor, write-once-read-multiple (*e.g.*, optical disk), and read-only direct-access devices, respectively. The commands in each of these clauses are unique to the device type, or they have interpretations, fields, or features that are specific for the device type. Thus, for example, although the WRITE command is used for several device types, it has a somewhat different form for each type, with different parameters and meanings. Therefore, it is specified separately for each device type.

Clause 14 describes the status byte for all device types. Status is returned by targets at the end of each command.

2 Normative references

The following standards contain provisions, which through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 8482:1987, *Information processing systems — Data communication — Twisted pair multipoint interconnections.*

This International Standard defines codes which controllers may use to identify that storage media conform to the following storage media standards.

ISO 1863:1976, *Information processing — 9-track, 12,7 mm (0.5 in) wide magnetic tape for information interchange recorded at 32 rpmm (800 rpi).*

ISO 3788:1976, *Information processing — 9-track, 12,7 mm (0.5 in) wide magnetic tape for information interchange recorded at 63 rpmm (1 600 rpi), phase encoded.*

ISO 4057:1986, *Information processing — Data interchange on 6,30 mm (0.25 in) magnetic tape cartridge 63 bpmm (1 600 bpi) phase-encoded.*

ISO 5652:1984, *Information processing — 9-track, 12,7 mm (0.5 in) wide magnetic tape for information interchange -- Format and recording, using group coding at 246 cpmm (6 250 cpi).*

ISO 5654/1:1984, *Information processing — Data interchange on 200 mm (8 in) flexible disk cartridges using two-frequency recording at 13 262 ftprad, 1, 9 tpmm (48 tpi), on one side — Part 1: Dimensional, physical and magnetic characteristics.*

ISO 5654/2:1985, *Information processing — Data interchange on 200 mm (8 in) flexible disk cartridges using two-frequency recording at 13 262 ftprad, 1, 9 tpmm (48 tpi), on one side — Part 2: Track format.*

ISO 6596/1:1985, *Information processing — Data interchange on 130 mm (5.25 in) flexible disk cartridges using two-frequency recording at 7 958 ftprad, 1,9 tpmm (48 tpi), on one side — Part 1: Dimensional, physical and magnetic characteristics.*

ISO 6596/2:1985, *Information processing — Data interchange on 130 mm (5.25 in) flexible disk cartridges using two-frequency recording at 7 958 ftprad, 1,9 tpmm (48 tpi) on one side — Part 2: Track format.*

ISO 7065/1:1985, *Information processing — Data interchange on 200 mm (8 in) flexible disk cartridges using modified frequency modulation recording at 13 262 ftprad, 1,9 tpmm (48 tpi), on both sides — Part 1: Dimensional, physical and magnetic characteristics.*

ISO 7065/2:1985, *Information processing — Data interchange on 200 mm (8 in) flexible disk cartridges using modified frequency modulation recording at 13 262 ftprad, 1,9 tpmm 48 (tpi), on both sides — Part 2: Track format.*

ISO 7487/1:1985, *Information processing — Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 7 958 ftprad, 1,9 tpmm (48 tpi), on both sides — Part 1: Dimensional, physical and magnetic characteristics.*

ISO 7487/2:1985, *Information processing — Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 7 958 ftprad, 1,9 tpmm (48 tpi), on both sides — Part 2: Track format.*

ISO 7487/3:1986, *Information processing — Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 7 958 ftprad, 1,9 tpmm (48 tpi), on both sides — Part 3: Track format B.*

ISO 8063/1:1986, *Information processing — Data interchange on 6,30 mm (0.25 in) wide magnetic tape cartridge using IMFM recording at 252 ftpmm (6 400 ftpi) — Part 1: Mechanical, physical and magnetic properties.*

ISO 8063/2:1986, *Information processing — Data interchange on 6,30 mm (0.25 in) wide magnetic tape cartridge using IMFM recording at 252 ftpmm (6 400 ftpi) — Part 2: Track format and method of recording for data interchange in start/stop mode.*

ISO 8378/1:1986, *Information processing — Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 7 958 ftprad, 3,8 tpmm (96 tpi), on both sides — Part 1: Dimensional, physical and magnetic characteristics.*

ISO 8378/2:1986, *Information processing — Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 7 958 ftprad, 3,8 tpmm (96 tpi), on both sides — Part 2: Track format A.*

ISO 8378/3:1986, *Information processing — Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 7 958 ftprad, 3,8 tpmm (96 tpi), on both sides — Part 3: Track format B.*

ISO 8462/1:1986, *Information processing — Data Interchange on 6,30 mm (0.25 in) magnetic tape cartridge using GCR recording at 394 ftpmm (10 000 ftpi), 39 cpmm (1 000 cpi) — Part 1: Mechanical, physical and magnetic properties.*

ISO 8462/2:1986, *Information processing — Data Interchange on 6,30 mm (0.25 in) magnetic tape cartridges using GCR recording at 394 ftpmm (10 000 ftpi), 39 cpmm (1 000 cpi) — Part 2: Streaming mode.*

ISO 8630/1:1987, *Information processing -- Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 13 262 ftprad, on 80 tracks on each side — Part 1: Dimensional, physical and magnetic characteristics.*

ISO 8630/2:1987, *Information processing -- Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 13 262 ftprad, on 80 tracks on each side — Part 2: Track format A for 77 tracks.*

ISO 8630/3:1987, *Information processing -- Data interchange on 130 mm (5.25 in) flexible disk cartridges using modified frequency modulation recording at 13 262 ftprad, on 80 tracks on each side — Part 3: Track format B for 80 tracks.*

ISO 8860/1:1987, *Information processing - Data interchange on 90 mm (3.5 in) flexible disk cartridges using modified frequency modulation recording at 7 958 ftprad on 80 tracks on each side — Part 1: Dimensional, physical and magnetic characteristics.*

ISO 8860/2:1987, *Information processing -- Data interchange on 90 mm (3.5 in) flexible disk cartridges using modified frequency modulation recording at 7 958 ftprad on 80 tracks on each side — Part 2: Track format.*

3 Definitions, abbreviations and conventions

3.1 Definitions

For the purposes of this International Standard the following definitions apply.

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

3.1.2 command descriptor block (CDB): The structure used to communicate requests from an initiator to a target.

3.1.3 connect: The function that occurs when an initiator selects a target to start an operation.

3.1.4 disconnect: The function that occurs when a target releases control of the SCSI bus, allowing it to go to the BUS FREE phase.

3.1.5 initiator: An SCSI device (usually a host system) that requests an operation to be performed by another SCSI device.

3.1.6 INTERMEDIATE status: A status code sent from a target to an initiator upon completion of each command in a set of linked commands except the last command in the set.

3.1.7 logical unit: A physical or virtual device addressable through a target.

3.1.8 logical unit number: An encoded three-bit identifier for the logical unit.

3.1.9 one: A true signal value.

3.1.10 peripheral device: A peripheral that can be attached to an SCSI device (e.g., magnetic-disk, printer, optical-disk, or magnetic-tape).

3.1.11 reconnect: The function that occurs when a target selects an initiator to continue an operation after a disconnect.

3.1.12 reserved: The term used for bits, bytes, fields, and code values that are set aside for future standardization.

3.1.13 SCSI address: The octal representation of the unique address (0-7) assigned to an SCSI device. This address would normally be assigned and set in the SCSI device during system installation.

3.1.14 SCSI ID: The bit-significant representation of the SCSI address referring to one of the signal lines DB(7-0).

3.1.15 SCSI device: A host computer adapter or a peripheral controller or an intelligent peripheral that can be attached to the SCSI bus.

3.1.16 signal assertion: The act of driving a signal to the true state.

3.1.17 signal negation: The act of driving a signal to the false state or allowing the cable terminators to bias the signal to the false state (by placing the driver in the high impedance condition).

3.1.18 signal release: The act of allowing the cable terminators to bias the signal to the false state (by placing the driver in the high impedance condition).

3.1.19 status: One byte of information sent from a target to an initiator upon completion of each command.

3.1.20 target: An SCSI device that performs an operation requested by an initiator.

3.1.21 vendor unique: In this standard, this term indicates bits, fields, or code values that are vendor specific and are not defined by this standard.

3.1.22 zero: A false signal value.

3.2 Abbreviations

3.2.1 LSB. Least significant byte.

3.2.2 LUN. Logical unit number.

3.2.3 MSB. Most significant byte.

3.3 Editorial conventions

Certain words, terms, and phrases used in this standard have a specific meaning beyond the normal English meaning. These words, terms, and phrases are defined either in the definitions (see 3.1) or in the text where they first appear (*e.g.*, Arbitration Delay, see 4.7.1). Names of signals, phases, conditions, messages, commands, statuses, and sense keys are in all uppercase (*e.g.*, REQUEST SENSE). Lower case is used for words having the normal English meaning.

Hexidecimal numbers are indicated by the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F followed by the lower case "h" (*e. g.* 3F6Ah). All other numbers are decimal values.

4 Physical Characteristics

This section contains the physical definition of the SCSI. The connectors, cables, signals, terminators, and bus timing needed to implement SCSI are covered.

4.1 Physical Description

SCSI devices are daisy-chained together using a common cable. Both ends of the cable are terminated. All signals are common between all SCSI devices. Two driver/receiver alternatives are available:

a) Single-ended drivers and receivers, which allow a maximum cable length of 6 m (primarily for connection within a cabinet).

b) Differential drivers and receivers, which allow a maximum cable length of 25 m (primarily for connection outside of a cabinet).

4.2 Cable Requirements

An ideal impedance match with cable terminators implies a cable characteristic impedance of 132 Ω (singled-ended option) or 122 Ω (differential option). In general, cables with this high of a characteristic impedance are not available; however, impedances that are somewhat lower are satisfactory. A characteristic impedance of 100 Ω ± 10% is recommended for unshielded flat or twisted pair ribbon cable. A characteristic impedance greater than 90 Ω is preferred for shielded cables; however, most available cables have a somewhat lower characteristic impedance. To minimize discontinuities and signal reflections, cables of different impedances should not be used in the same bus. Implementations may require trade-offs in shielding effectiveness, cable length, the number of loads, transfer rates, and cost to achieve satisfactory system operation.

A mininimum conductor size of nominal cross section 0,080 42 mm^2 shall be employed to minimize noise effects and ensure proper distribution of optional terminator power.

4.2.1 Single-Ended Cable

A 50-conductor flat cable or 25-signal twisted- pair cable shall be used. The maximum cable length shall be 6 m.

A stub length of no more than 0,1 m is allowed off the mainline interconnection within any connected equipment.

SCSI bus termination may be internal to the SCSI devices that are at the ends of the cable.

4.2.2 Differential Cable

A 50-conductor cable or 25-signal twisted-pair cable shall be used. The maximum cable length shall be 25 m.

A stub length of no more than 0,2 m is allowed off the mainline interconnection within any connected equipment.

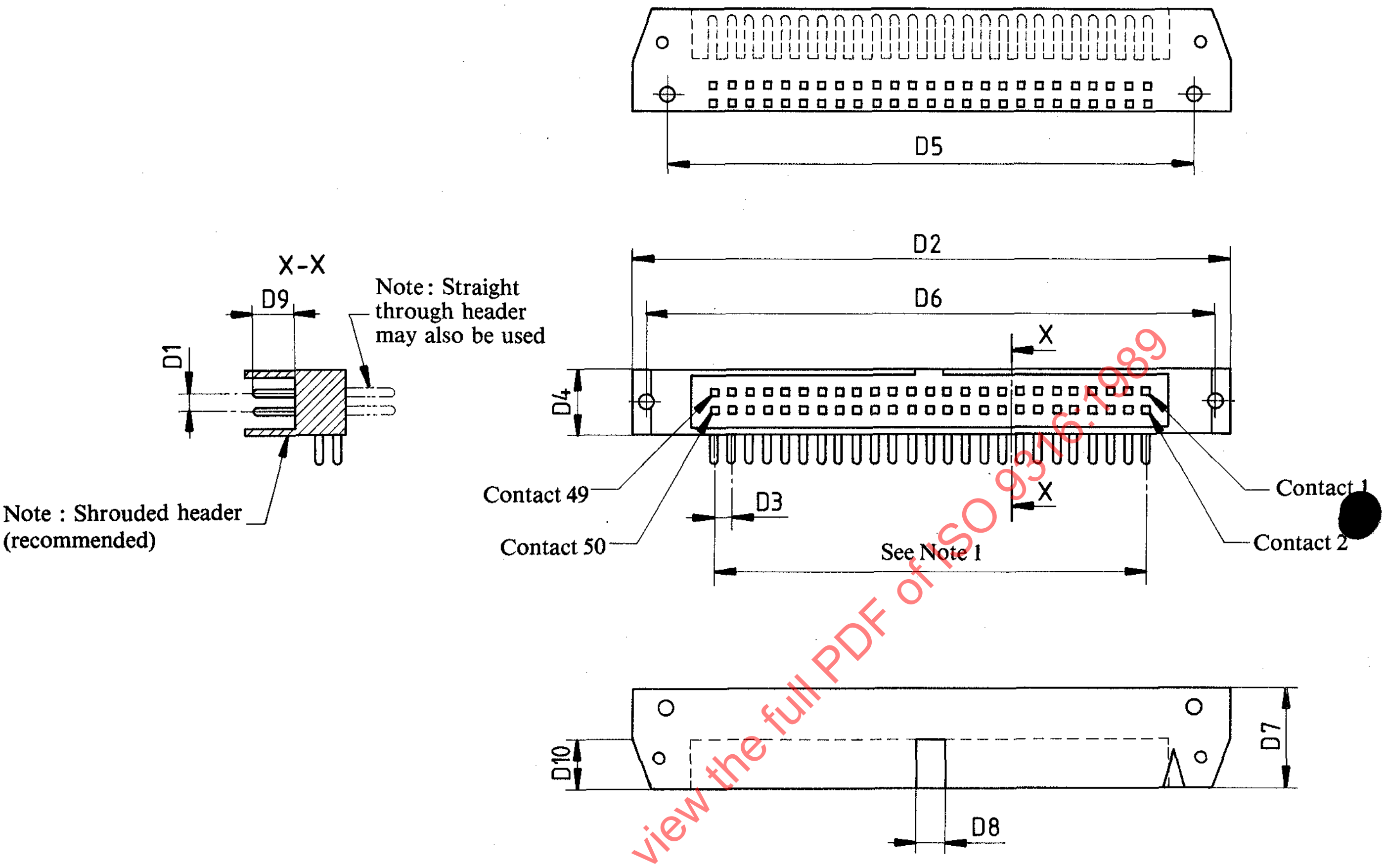
SCSI bus termination may be internal to the SCSI devices that are at the ends of the cable.

4.3 Connector Requirements

Nonshielded connectors are specified. The nonshielded connectors are typically used for in-cabinet applications. Annex D defines recommended shielded connectors and their pin assignments. These connectors are typically used for external applications where electromagnetic compatibility (EMC) and electrostatic discharge (ESD) protection may be required. Either type of connector may be used with the single-ended or differential drivers.

The nonshielded SCSI device connector (figure 1) shall be a 50-conductor connector consisting of two rows of 25 male contacts with adjacent contacts 2,54 mm (0,1 in) apart. A shroud and header body should be used. The nonmating portion of the connector is shown for reference only.

The nonshielded cable connector (figure 2) shall be a 50-conductor connector consisting of two rows of 25 female contacts with adjacent contacts 2,54 mm (0,1 in) apart. It is recommended that keyed connectors be used.


D5
D2
D6
X-X
D9
D1
Note: Straight through header may also be used
X
D4
Note : Shrouded header (recommended)
Contact 49
Contact 50
D3
X
See Note 1
Contact 1
Contact 2
D10
D7
D8

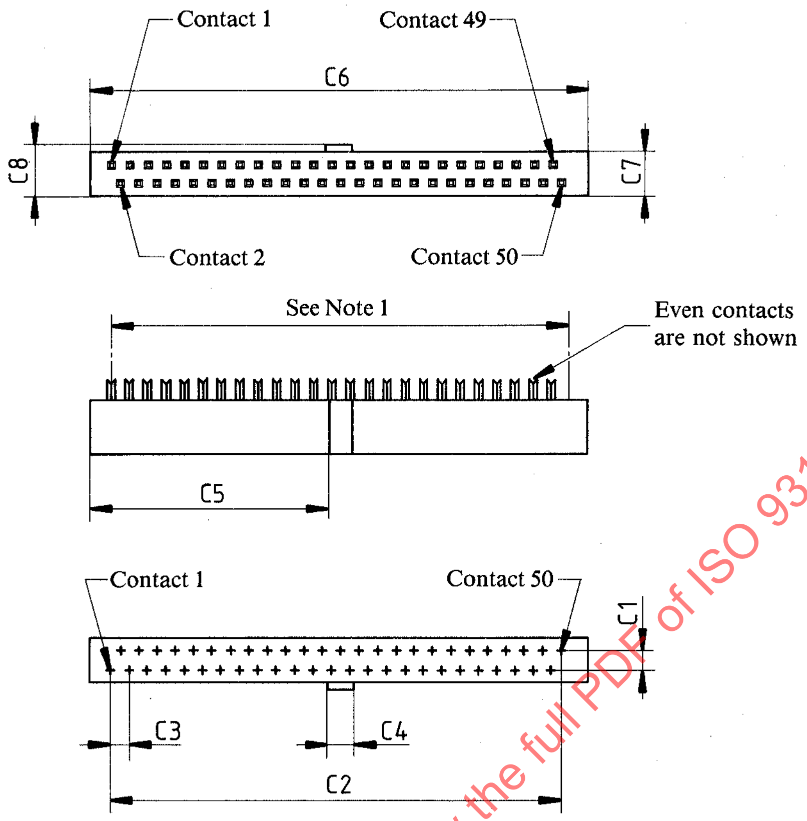
Dimensions	mm	in	Comments
D1	2,54	0,100	
D2	82,50	3,260	Reference only
D3	2,54	0,100	
D4	8,89	0,350	Reference only
D5	72,64	2,860	Reference only
D6	78,74	3,100	Reference only
D7	13,94	0,549	Reference only
D8	4,19 ± 0,25	0,165 ± 0,01	
D9	6,10	0,240	
D10	6,60	0,260	Reference only

NOTES :

1) Two rows of twenty five contacts on 2,54 mm (0,100 in) spacing = 60,96 mm (2,400 in).

2) Tolerances ± 0,127 mm (0,005 in) non-cumulative, unless specified otherwise.

Figure 1 — Nonshielded SCSI Device Connector



Dimensions	mm	in	Comments
C1	2,54	0,100	
C2	60,96	2,400	
C3	2,54	0,100	
C4	3,30	0,130	
C5	32,39	1,275	
C6	68,07	2,680	
C7	6,10	0,240	
C8	7,62	0,300	max.

NOTES :

1) Fifty contacts on 1,27 mm (0,05 in) staggered spacing = 62,23 mm (2,450 in) [reference only].

2) Tolerances ± 0,127 mm (0,005 in) non-cumulative, unless specified otherwise.

3) Connector cover and strain relief are optional.

Figure 2 — Nonshielded SCSI Cable Connector

The unshielded connector contact assignments shall be as shown in table 1 for single-ended drivers.

Table 1 – Single-Ended Connector Contact Assignments

Signal	Contact Number	Signal	Contact Number
GROUND	1	-DB(0)	2
GROUND	3	-DB(1)	4
GROUND	5	-DB(2)	6
GROUND	7	-DB(3)	8
GROUND	9	-DB(4)	10
GROUND	11	-DB(5)	12
GROUND	13	-DB(6)	14
GROUND	15	-DB(7)	16
GROUND	17	-DB(P)	18
GROUND	19	GROUND	20
GROUND	21	GROUND	22
GROUND	23	GROUND	24
OPEN	25	TERMPWR	26
GROUND	27	GROUND	28
GROUND	29	GROUND	30
GROUND	31	-ATN	32
GROUND	33	GROUND	34
GROUND	35	-BSY	36
GROUND	37	-ACK	38
GROUND	39	-RST	40
GROUND	41	-MSG	42
GROUND	43	-SEL	44
GROUND	45	-C/D	46
GROUND	47	-REQ	48
GROUND	49	-I/O	50

NOTE: The minus sign next to the signals indicates active low.

The unshielded connector contact assignments shall be as shown in table 2 for differential drivers.

Table 2 – Differential Connector Contact Assignments

Signal	Contact Number	Signal	Contact Number
SHIELD GROUND	1	GROUND	2
+DB(0)	3	-DB(0)	4
+DB(1)	5	-DB(1)	6
+DB(2)	7	-DB(2)	8
+DB(3)	9	-DB(3)	10
+DB(4)	11	-DB(4)	12
+DB(5)	13	-DB(5)	14
+DB(6)	15	-DB(6)	16
+DB(7)	17	-DB(7)	18
+DB(P)	19	-DB(P)	20
DIFFSENS	21	GROUND	22
GROUND	23	GROUND	24
TERMPWR	25	TERMPWR	26
GROUND	27	GROUND	28
+ATN	29	-ATN	30
GROUND	31	GROUND	32
+BSY	33	-BSY	34
+ACK	35	-ACK	36
+RST	37	-RST	38
+MSG	39	-MSG	40
+SEL	41	-SEL	42
+C/D	43	-C/D	44
+REQ	45	-REQ	46
+I/O	47	-I/O	48
GROUND	49	GROUND	50

NOTE: SHIELD GROUND is optional on some cables. Some shielded flat ribbon cables use contact 1 as a connection to the shield.

4.4 Electrical Description

For these measurements, SCSI bus termination is assumed to be external to the SCSI device. An SCSI device may have the provision for allowing optional internal termination.

4.4.1 Single-Ended Alternative

All assigned signals shall be terminated with 220 Ω to +5 V (nominal) and 330 Ω to ground at each end of the cable. (See figure 3.) All signals shall use open-collector or three-state drivers.

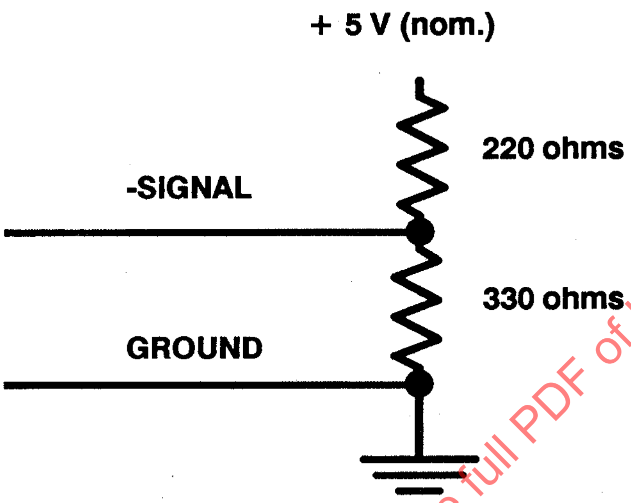


Figure 3 — Termination for Single-Ended Devices

4.4.1.1 Output Characteristics

Each signal driven by an SCSI device shall have the following output characteristics when measured at the SCSI device's connector:

Signal assertion = 0,0 V dc to 0,4 V dc
Minimum driver output capability = 48 mA (sinking) at 0,5 V dc
Signal negation = 2,5 V dc to 5,25 V dc

4.4.1.2 Input Characteristics

Each signal received by an SCSI device shall have the following input characteristics when measured at the SCSI device's connector:

Signal true = 0,0 V dc to 0,8 V dc
Maximum total input load = -0,4 mA at 0,4 V dc
Signal false = 2,0 V dc to 5,25 V dc
Minimum input hysteresis = 0,2 V dc

4.4.2 Differential Alternative.

All signals consist of two lines denoted +SIGNAL and -SIGNAL. A signal is true when +SIGNAL is more positive than -SIGNAL, and a signal is false when -SIGNAL is more positive than +SIGNAL. All assigned signals shall be terminated at each end of the cable as shown in figure 4.

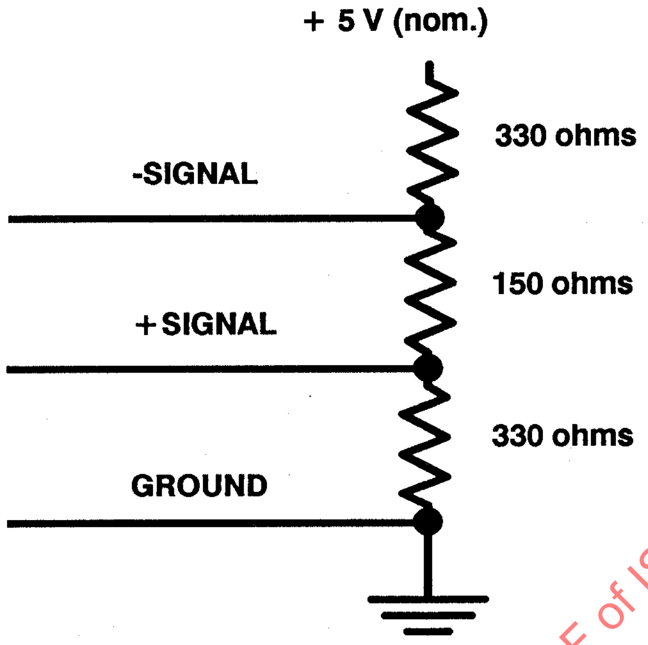


Figure 4 — Termination for Differential Devices

As an option, the DIFFSENS signal of the connector is reserved for an active high enable for the differential drivers. If a single-ended device or terminator is inadvertently connected, this signal is grounded, disabling the drivers. (See figure 5). This will prevent damage should single-ended and differential drivers be connected to the same cable.

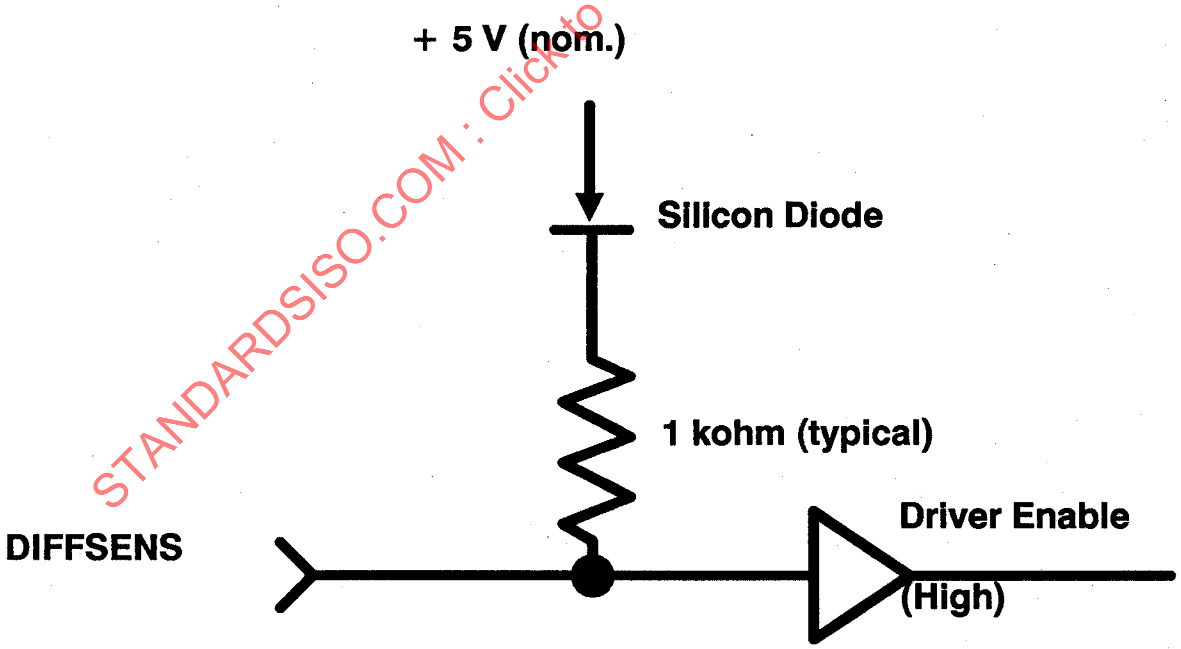


Figure 5 — Differential Driver Protection Circuit (Opt.)

4.4.2.1 Output Characteristics

Each signal driven by an SCSI device shall have the following output characteristics when measured at the SCSI device's connector:

V_{OL} (Low-level output voltage) = 2,0 V max. at I_{OL} (Low-level output current) = 55 mA.
V_{OH} (High-level output voltage) = 3,0 V min. at I_{OH} (High-level output current) = -55 mA.
V_{OD} (Differential voltage) = 1,0 V min. with common-mode voltage ranges from -7 V dc to +12 V dc.

V_{OL} and V_{OH} shall be as measured between the output terminal and the SCSI device's logic ground reference.

The output characteristics shall additionally conform to ISO 8482.

4.4.2.2 Input Characteristics

Each signal received by an SCSI device shall have the following input characteristics when measured at the SCSI device's connector:

I_I (Input current on either input) = ± 2,0 mA max.

NOTE: These characteristics include both receivers and passive drivers.

This requirement shall be met with the input voltage varying between -7 V dc and +12 V dc, with power on or off, and with the hysteresis equaling 35 mV, min.

The input characteristics shall additionally conform to ISO 8482.

4.4.3 Terminator Power (Optional)

Single-ended SCSI devices providing terminator power (TERMPWR) shall have the following characteristics:

V_{Term} = 4,0 V dc to 5,25 V dc
800 mA min. source drive capability
1,0 mA max. sink capability (except for the purposes of providing power to an internal terminator) with 1,0 A recommended current limiting (*e.g.*, a fuse).

Differential SCSI devices providing terminator power (TERMPWR) shall have the following characteristics:

V_{Term} = 4,0 V dc to 5,25 V dc
600 mA min. source drive capability
1,0 mA max. sink capability (except for the purposes of providing power to an internal terminator) with 1,0 A recommended current limiting (*e.g.*, a fuse).

The use of keyed connectors is recommended in SCSI devices that provide terminator power to prevent accidental grounding or misconnection of terminator power.

SCSI devices that supply terminator power shall do so through a diode or similar semiconductor that prevents the backflow of power to the SCSI device.

4.5 SCSI Bus

Communication on the SCSI bus is allowed between only two SCSI devices at any given time. There is a maximum of eight SCSI devices. Each SCSI device has an SCSI ID bit assigned as shown in figure 6.

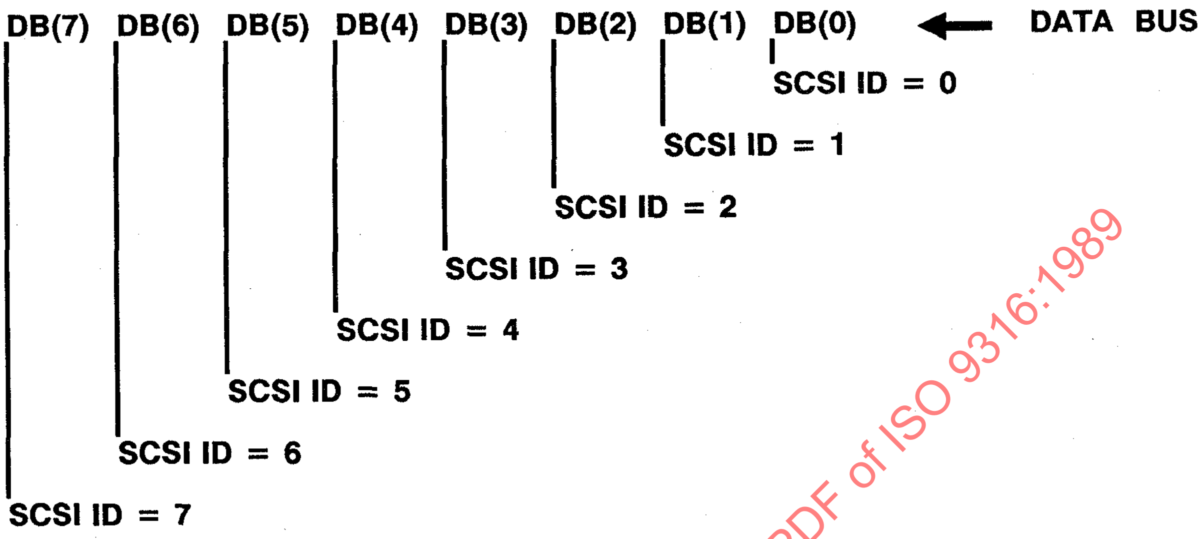
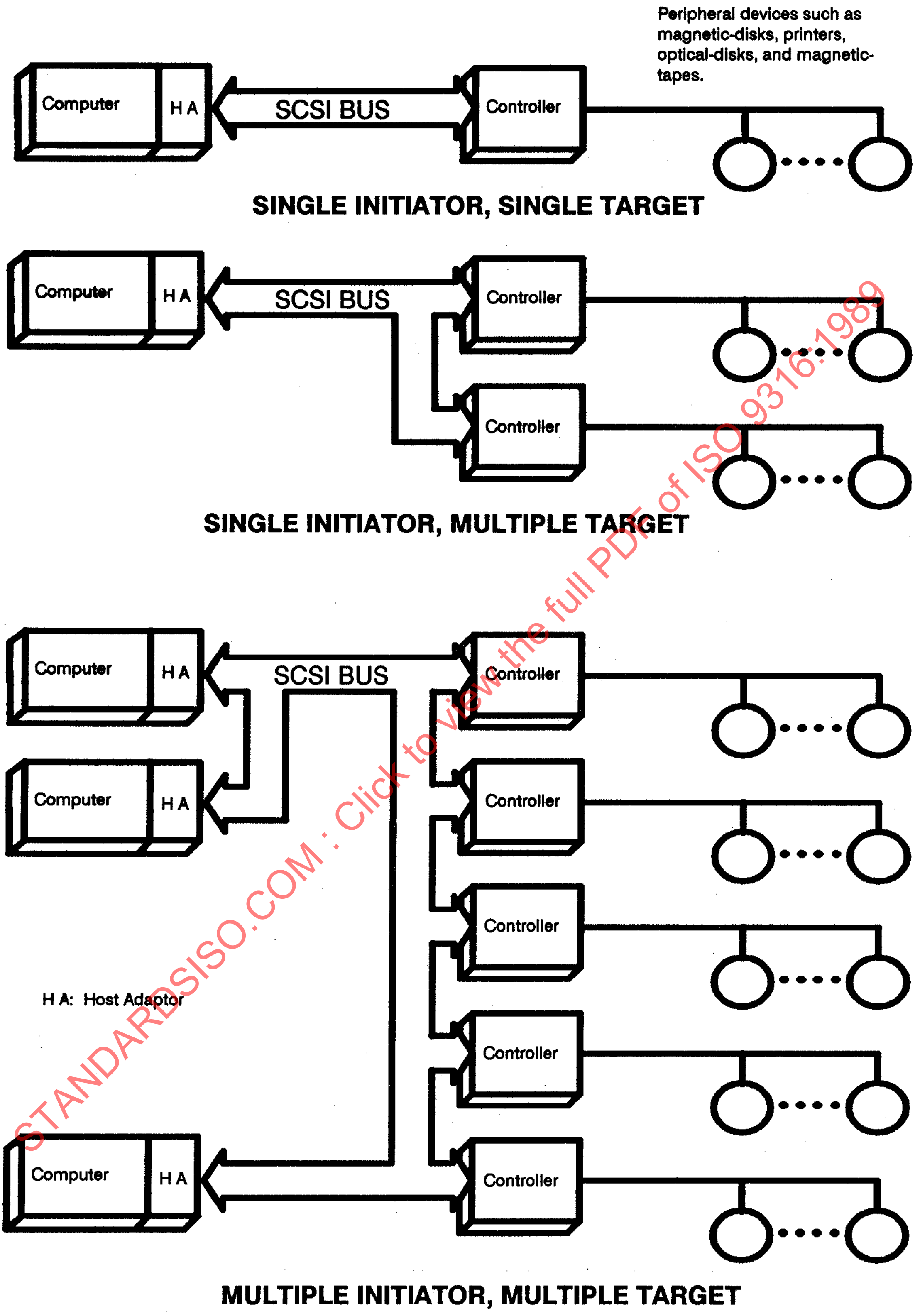


Figure 6 — SCSI ID Bits

When two SCSI devices communicate on the SCSI bus, one acts as an initiator and the other acts as a target. The initiator originates an operation and the target performs the operation. An SCSI device usually has a fixed role as an initiator or target, but some devices may be able to assume either role.

An initiator may address up to eight peripheral devices that are connected to a target. An option allows the addressing of up to 2,048 peripheral devices per target using extended messages. Three sample system configurations are shown in figure 7.



Peripheral devices such as magnetic-disks, printers, optical-disks, and magnetic-tapes.

SINGLE INITIATOR, SINGLE TARGET

SINGLE INITIATOR, MULTIPLE TARGET

H A: Host Adaptor

MULTIPLE INITIATOR, MULTIPLE TARGET

Figure 7 — Sample SCSI Configurations

Up to eight SCSI devices can be supported on the SCSI bus. They can be any combination of initiators and targets.

Certain SCSI bus functions are assigned to the initiator and certain SCSI bus functions are assigned to the target. The initiator may arbitrate for the SCSI bus and select a particular target. The target may request the transfer of COMMAND, DATA, STATUS, or other information on the DATA BUS, and in some cases it may arbitrate for the SCSI bus and reselect an initiator for the purpose of continuing an operation.

Information transfers on the DATA BUS are asynchronous and follow a defined REQ/ACK handshake protocol. One byte of information may be transferred with each handshake. An option is defined for synchronous data transfer.

4.6 SCSI Bus Signals

There are a total of eighteen signals. Nine are used for control and nine are used for data. (Data signals include the parity signal option). These signals are described as follows:

BSY (BUSY). An "OR-tied" signal that indicates that the bus is being used.

SEL (SELECT). A signal used by an initiator to select a target or by a target to reselect an initiator.

C/D (CONTROL/DATA). A signal driven by a target that indicates whether CONTROL or DATA information is on the DATA BUS. True indicates CONTROL.

I/O (INPUT/OUTPUT). A signal driven by a target that controls the direction of data movement on the DATA BUS with respect to an initiator. True indicates input to the initiator. This signal is also used to distinguish between SELECTION and RESELECTION phases.

MSG (MESSAGE). A signal driven by a target during the MESSAGE phase.

REQ (REQUEST). A signal driven by a target to indicate a request for a REQ/ACK data transfer handshake.

ACK (ACKNOWLEDGE). A signal driven by an initiator to indicate an acknowledgment for a REQ/ACK data transfer handshake.

ATN (ATTENTION). A signal driven by an initiator to indicate the ATTENTION condition.

RST (RESET). An "OR-tied" signal that indicates the RESET condition.

DB(7-0,P) (DATA BUS). Eight data-bit signals, plus a parity-bit signal that form a DATA BUS. DB(7) is the most significant bit and has the highest priority during the ARBITRATION phase. Bit number, significance, and priority decrease downward to DB(0). A data bit is defined as one when the signal value is true and is defined as zero when the signal value is false.

Data parity DB(P) is odd. The use of parity is a system option (*i.e.*, a system is configured so that all SCSI devices on a bus generate parity and have parity detection enabled, or all SCSI devices have parity detection disabled or not implemented). Parity is not valid during the ARBITRATION phase.

4.6.1 Signal Values

Signals may assume true or false values. There are two methods of driving these signals. In both cases, the signal shall be actively driven true, or asserted. In the case of OR-tied drivers, the driver does not drive the signal to the false state, rather the bias circuitry of the bus terminators pulls the signal false whenever it is released by the drivers at every SCSI device. If any driver is asserted, then the signal is true. In the case of non-OR-tied drivers, the signal may be actively driven false, or negated. In this International Standard, wherever the term negated is used, it means that the signal may be actively driven false, or may be simply released (in which case the bias circuitry pulls it false), at the option of the implementor. The advantage to actively drive signals false is that the transition from true to false occurs more quickly, and noise margins may be somewhat improved; this may permit somewhat faster data transfer.

4.6.2 OR-Tied Signals

The BSY and RST signals shall be OR-tied only. In the ordinary operation of the bus, these signals are simultaneously driven true by several drivers. No signals other than BSY, RST, and DB(P) are simultaneously driven by two or more drivers, and any signal other than BSY and RST may employ OR-tied or non-OR-tied drivers. DB(P) shall not be driven false during the ARBITRATION phase. There is no operational problem in mixing OR-tied and non-OR-tied drivers on signals other than BSY and RST.

4.6.3 Signal Sources

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

Table 3 – Signal Sources

	Signals				
			C/D, I/O,		
Bus Phase	**BSY**	**SEL**	**MSG, REQ**	**ACK/ATN**	**DB(7-0,P)**
BUS FREE	None	None	None	None	None
ARBITRATION	All	Winner	None	None	SCSI ID
SELECTION	I&T	Initiator	None	Initiator	Initiator
RESELECTION	I&T	Target	Target	Initiator	Target
COMMAND	Target	None	Target	Initiator	Initiator
DATA IN	Target	None	Target	Initiator	Target
DATA OUT	Target	None	Target	Initiator	Initiator
STATUS	Target	None	Target	Initiator	Target
MESSAGE IN	Target	None	Target	Initiator	Target
MESSAGE OUT	Target	None	Target	Initiator	Initiator

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

SCSI ID: A unique data bit (the SCSI ID) shall be driven by each SCSI device that is actively arbitrating; the other seven data bits shall be released (i.e., not driven) by this SCSI device. The parity bit (DB(P)) may be undriven or driven to the true state, but shall never be driven to the false state during this phase.

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

Initiator: If this signal is driven, it shall be driven only by the active initiator.

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

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

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

4.7 SCSI Bus Timing

Unless otherwise indicated, the delay-time measurements for each SCSI device, shown in 4.7.1 through 4.7.14, shall be calculated from signal conditions existing at that SCSI device's own SCSI bus connection. Thus, these measurements (except cable skew delay) can be made without considering delays in the cable.

4.7.1 Arbitration Delay (2,2 μs)

The minimum time an SCSI device shall wait from asserting BSY for arbitration until the DATA BUS can be examined to see if arbitration has been won. There is no maximum time.

4.7.2 Assertion Period (90 ns)

The minimum time that a target shall assert REQ while using synchronous data transfers. Also, the minimum time that an initiator shall assert ACK while using synchronous data transfers.

4.7.3 Bus Clear Delay (800 ns)

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

a) The BUS FREE phase is detected (BSY and SEL both false for a bus settle delay)

b) SEL is received from another SCSI device during the ARBITRATION phase

c) The transition of RST to true.

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

4.7.4 Bus Free Delay (800 ns)

The minimum time that an SCSI device shall wait from its detection of the BUS FREE phase (BSY and SEL both false for a bus settle delay) until its assertion of BSY when going to the ARBITRATION phase.

4.7.5 Bus Set Delay (1,8 μs)

The maximum time for an SCSI device to assert BSY and its SCSI ID bit on the DATA BUS after it detects BUS FREE phase (BSY and SEL both false for a bus settle delay) for the purpose of entering the ARBITRATION phase.

4.7.6 Bus Settle Delay (400 ns)

The time to wait for the bus to settle after changing certain control signals as called out in the protocol definitions.

4.7.7 Cable Skew Delay (10 ns)

The maximum difference in propagation time allowed between any two SCSI bus signals when measured between any two SCSI devices.

4.7.8 Data Release Delay (400 ns)

The maximum time for an initiator to release the DATA BUS signals following the transition of the I/O signal from false to true.

4.7.9 Deskew Delay (45 ns)

The minimum time required for deskew of certain signals.

4.7.10 Hold Time (45 ns)

The minimum time added between the assertion of REQ or ACK and the changing of the data lines to provide hold time in the initiator or target, respectively, while using synchronous data transfers.

4.7.11 Negation Period (90 ns)

The minimum time that a target shall negate REQ while using synchronous data transfers. Also, the minimum time that an initiator shall negate ACK while using synchronous data transfers.

4.7.12 Reset Hold Time (25 μs)

The minimum time for which RST is asserted. There is no maximum time.

4.7.13 Selection Abort Time (200 μs)

The maximum time that a target (or initiator) shall take from its most recent detection of being selected (or reselected) until asserting a BSY response. This timeout is required to ensure that a target (or initiator) does not assert BSY after a SELECTION (or RESELECTION) phase has been aborted. This is not the selection timeout period; see 5.1.3.5 and 5.1.4.2 for a complete description.

4.7.14 Selection Timeout Delay (250 ms, recommended)

The minimum time that an initiator (or target) should wait for a BSY response during the SELECTION (or RESELECTION) phase before starting the timeout procedure. Note that this is only a recommended time period. The specifications for the peripheral devices shall be consulted for the actual timing requirements.

4.7.15 Transfer Period (set during a MESSAGE phase)

The Transfer Period specifies the minimum time allowed between the leading edges of successive REQ pulses and of successive ACK pulses while using synchronous data transfers. (See 5.1.5.2 and 5.5.5.)

5 Logical characteristics

5.1 SCSI Bus Phases

The SCSI architecture includes eight distinct phases:

BUS FREE phase
ARBITRATION phase
SELECTION phase
RESELECTION phase
COMMAND phase
DATA phase
STATUS phase
MESSAGE phase

These phases (COMMAND, DATA, STATUS, MESSAGE) are collectively termed the information transfer phases.

The SCSI bus can never be in more than one phase at any given time. Unless otherwise noted in the following descriptions, signals that are not mentioned shall not be asserted.

5.1.1 BUS FREE Phase

The BUS FREE phase is used to indicate that no SCSI device is actively using the SCSI bus and that it is available for subsequent users.

SCSI devices shall detect the BUS FREE phase after SEL and BSY are both false for at least a bus settle delay. SCSI devices shall release all SCSI bus signals within a bus clear delay after BSY and SEL become continuously false for a bus settle delay. If an SCSI device requires more than a bus settle delay to detect the BUS FREE phase then it shall release all SCSI bus signals within a bus clear delay minus the excess time to detect the BUS FREE phase. The total time to clear the SCSI bus shall not exceed a bus settle delay plus a bus clear delay.

5.1.2 ARBITRATION Phase

The ARBITRATION phase allows one SCSI device to gain control of the SCSI bus so that it can assume the role of an initiator or target. Implementation of the ARBITRATION phase is a system option. Systems that do not implement this option can have only one initiator. The ARBITRATION phase is required for systems that use the RESELECTION phase.

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

a) The SCSI device shall first wait for the BUS FREE phase to occur. The BUS FREE phase is detected whenever both BSY and SEL are simultaneously and continuously false for a minimum of a bus settle delay. (Implementors Note: This bus settle delay is necessary because a transmission line phenomenon known as a “wire-OR glitch” may cause BSY to briefly appear false, even though it is being driven true.)

b) The SCSI device shall wait a minimum of a bus free delay after detection of the BUS FREE phase (i.e. after BSY and SEL are both false for a bus settle delay) before driving any signal.

c) Following the bus free delay in step b), the SCSI device may arbitrate for the SCSI bus by asserting both BSY and its own SCSI ID, however the SCSI device shall not arbitrate (i.e. assert BSY and its SCSI ID) if more than a bus set delay has passed since the BUS FREE phase was last observed. (Note: There is no maximum delay before asserting BSY and the SCSI ID following the bus free delay in step b) as long as the bus remains in the BUS FREE phase. However, SCSI devices that delay longer than a bus settle delay plus a bus set delay from the time when BSY and SEL first become false may fail to participate in arbitration when competing with faster SCSI devices.)

d) After waiting at least an arbitration delay (measured from its assertion of BSY) the SCSI device shall examine the DATA BUS. If a higher priority SCSI ID bit is true on the DATA BUS (DB(7) is the highest), then the SCSI device has lost the arbitration and the SCSI device may release its signals and return to Step a). If no higher priority SCSI ID bit is true on the DATA BUS, then the SCSI device has won the arbitration and it shall assert SEL. Any other SCSI device that is participating in the ARBITRATION phase has lost the arbitration and shall release BSY and its SCSI ID bit within a bus clear delay after SEL becomes true. An SCSI device that loses arbitration may return to step a).

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

The SCSI ID bit is a single bit on the DATA BUS that corresponds to the SCSI device's unique SCSI address. All other seven DATA BUS bits shall be released by the SCSI device. Parity is not valid during the ARBITRATION phase. During the ARBITRATION phase, DB(P) may be undriven or driven to the true state, but shall not be driven to the false state.

5.1.3 SELECTION Phase

The SELECTION phase allows an initiator to select a target for the purpose of initiating some target function (e.g., READ or WRITE command).

During the SELECTION phase the I/O signal shall be negated so that this phase can be distinguished from the RESELECTION phase.

5.1.3.1 Nonarbitrating Systems

In systems with the ARBITRATION phase not implemented, the initiator shall first detect the BUS FREE phase and then wait a minimum of a bus clear delay. Then, except in certain single initiator environments with initiators employing the single initiator option (see 5.1.3.4), the initiator shall assert the desired target's SCSI ID and its own initiator SCSI ID on the DATA BUS. After two deskew delays the initiator shall assert SEL.

5.1.3.2 Arbitrating Systems

In systems with ARBITRATION phase implemented, the SCSI device that won the arbitration has both BSY and SEL asserted and has delayed at least a bus clear delay plus a bus settle delay before ending the ARBITRATION phase. The SCSI device that won the arbitration becomes an initiator by not asserting the I/O signal. Except in certain single initiator environments with initiators employing the single initiator option (see 5.1.3.4), the initiator shall set the DATA BUS to a value which is the OR of its SCSI ID bit and the target's SCSI ID bit. The initiator shall then wait at least two deskew delays and release BSY. The initiator shall then wait at least a bus settle delay before looking for a response from the target.

5.1.3.3 All Systems

In all systems, the target shall determine that it is selected when SEL and its SCSI ID bit are true and BSY and I/O are false for at least a bus settle delay. The selected target may examine the DATA BUS in order to determine the SCSI ID of the selecting initiator unless the initiator employed the single initiator option (see 5.1.3.4). The selected target shall then assert BSY within a selection abort time of its most recent detection of being selected; this is required for correct operation of the timeout procedure. In systems with parity implemented, the target shall not respond to a selection if bad parity is detected. Also, if more than two SCSI ID bits are on the DATA BUS, the target shall not respond to selection.

At least two deskew delays after the initiator detects BSY is true, it shall release SEL and may change the DATA BUS.

5.1.3.4 Single Initiator Option

Initiators that do not implement the RESELECTION phase and do not operate in the multiple initiator environment are allowed to set only the target's SCSI ID bit during the SELECTION phase. This makes it impossible for the target to determine the initiator's SCSI ID.

5.1.3.5 Selection Timeout Procedure

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

a) Optionally, the initiator shall assert the RST signal (see 5.2.2).

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

5.1.4 RESELECTION Phase (Optional)

RESELECTION is an optional phase that allows a target to reconnect to an initiator for the purpose of continuing some operation that was previously started by the initiator but was suspended by the target, (*i.e.*, the target disconnected by allowing a BUS FREE phase to occur before the operation was complete).

5.1.4.1 RESELECTION

RESELECTION can only be used in systems that have ARBITRATION phase implemented.

Upon completing the ARBITRATION phase, the winning SCSI device has both BSY and SEL asserted and has delayed at least a bus clear delay plus a bus settle delay. The winning SCSI device becomes a target by asserting the I/O signal. The winning SCSI device shall also set the DATA BUS to a value that is the OR of its SCSI ID bit and the initiator's SCSI ID bit. The target shall wait at least two deskew delays and release BSY. The target shall then wait at least a bus settle delay before looking for a response from the initiator.

The initiator shall determine that it is reselected when SEL, I/O, and its SCSI ID bit are true and BSY is false for at least a bus settle delay. The reselected initiator may examine the DATA BUS in order to determine the SCSI ID of the reselecting target. The reselected initiator shall then assert BSY within a selection abort time of its most recent detection of being reselected; this is required for correct operation of the timeout procedure. In systems with parity implemented, the initiator shall not respond to a RESELECTION if bad parity is detected. Also, the initiator shall not respond to a RESELECTION if more than two SCSI ID bits are on the DATA BUS.

After the target detects BSY, it shall also assert BSY and wait at least two deskew delays and then release SEL. The target may then change the I/O signal and the DATA BUS. After the reselected initiator detects SEL false, it shall release BSY. The target shall continue asserting BSY until the target is ready to relinquish the SCSI bus.

NOTE: When the target is asserting BSY, a transmission line phenomenon known as a "wire-OR glitch" may cause BSY to appear false for up to a round-trip propagation delay following the release of BSY by the initiator. This is the reason why the BUS FREE phase is recognized only after both BSY and SEL are continuously false for a minimum of a bus settle delay. Cables longer than 25 m should not be used even if the chosen driver, receiver, and cable provide adequate noise margins, because they increase the duration of the glitch and could cause SCSI devices to inadvertently detect the BUS FREE phase.

5.1.4.2 RESELECTION Timeout Procedure

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

a) Optionally, the target shall assert the RST signal (see 5.2.2).

b) Optionally, the target shall continue asserting SEL and I/O and shall release all DATA BUS signals. If the target has not detected BSY to be true after at least a selection abort time plus two deskew delays, the target shall release SEL and I/O allowing the SCSI bus to go to the BUS FREE phase. SCSI devices that respond to RESELECTION shall ensure that the RESELECTION was still valid within a selection abort time of their assertion of BSY. Failure to comply with this requirement could result in an improper reselection (two initiators connected to the same target or the wrong initiator connected to a target).

5.1.5 Information Transfer Phases

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

The C/D, I/O, and MSG signals are used to distinguish between the different information transfer phases. (See table 4.) The target drives these three signals and therefore controls all changes from one phase to another. The initiator can request a MESSAGE OUT phase by asserting ATN, while the target can cause the BUS FREE phase by releasing MSG, C/D, I/O, and BSY.

The information transfer phases use one or more REQ/ACK handshakes to control the information transfer. Each REQ/ACK handshake allows the transfer of one byte of information. During the information transfer phases BSY shall remain true and SEL shall remain false. Additionally, during the information transfer phases, the target shall continuously envelope the REQ/ACK handshake(s) with C/D, I/O, and MSG in such a manner that these control signals are valid for a bus settle delay before the assertion of REQ of the first handshake and remain valid until the negation of ACK at the end of the last handshake.

Table 4 – Information Transfer Phases

Signal					
MSG	C/D	I/O	Phase Name	Direction Of Transfer	Comment
0	0	0	DATA OUT	Initiator to target	Data Phase
0	0	1	DATA IN	Initiator from target	
0	1	0	COMMAND	Initiator to target	
0	1	1	STATUS	Initiator from target	
1	0	0	*		
1	0	1	*		
1	1	0	MESSAGE OUT	Initiator to target	Message Phase
1	1	1	MESSAGE IN	Initiator from target	

Key: 0 = False, 1 = True, * = Reserved for future standardization.

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

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

If I/O is false (transfer to the target), the target shall request information by asserting REQ. The initiator shall drive DB(7-0,P) to their desired values, delay at least one deskew delay plus a cable skew delay and assert ACK. The initiator shall continue to drive DB(7-0,P) until REQ is false. When ACK becomes true at the target, the target shall read DB(7-0,P), then negate REQ. When REQ becomes false at the initiator, the initiator may change or release DB(7-0,P) and shall negate ACK. The target may continue the transfer by asserting REQ, as described above.

5.1.5.2 Synchronous Data Transfer (Optional)
Synchronous data transfer is optional, and may be used only in the data phase if previously agreed to by the initiator and target through the message system (see SYNCHRONOUS DATA TRANSFER REQUEST message, 5.5.5). The messages determine the use of synchronous mode by both SCSI devices and establish a REQ/ACK offset and a transfer period.

The REQ/ACK offset specifies the maximum number of REQ pulses that can be sent by the target in advance of the number of ACK pulses received from the initiator, establishing a pacing mechanism. If the number of REQ pulses exceeds the number of ACK pulses by the REQ/ACK offset, the target shall not assert REQ until the next ACK pulse is received. A requirement for successful completion of the data phase is that the number of ACK and REQ pulses be equal.

The target shall assert the REQ signal for a minimum of an assertion period. The target shall wait at least the greater of a transfer period from the last transition of REQ to true or a minimum of a negation period from the last transition of REQ to false before asserting the REQ signal.

The initiator shall send one pulse on the ACK signal for each REQ pulse received. The initiator shall assert the ACK signal for a minimum of an assertion period. The initiator shall wait at least the greater of a transfer period from the last transition of ACK to true or for a minimum of a negation period from the last transition of ACK to false before asserting the ACK signal.

If I/O is true (transfer to the initiator), the target shall first drive DB(7-0,P) to their desired values, wait at least one deskew delay plus one cable skew delay, then assert REQ. DB(7-0,P) shall be held valid for a minimum of one deskew delay plus one cable skew delay plus one hold time after the assertion of REQ. The target shall assert REQ for a minimum of an assertion period. The target may then negate REQ and change or release DB(7- 0,P). The initiator shall read the value on DB(7-0,P) within one hold time of the transition of REQ to true. The initiator shall then respond with an ACK pulse.

If I/O is false (transfer to the target), the initiator shall transfer one byte for each REQ pulse received. After receiving a REQ pulse, the initiator shall first drive DB(7-0,P) to their desired values, delay at least one deskew delay plus one cable skew delay, then assert ACK. The initiator shall hold DB(7-0,P) valid for at least one deskew delay plus one cable skew delay plus one hold time after the assertion of ACK. The initiator shall assert ACK for a minimum of an assertion period. The initiator may then negate ACK and may change or release DB(7-0,P). The target shall read the value of DB(7-0,P) within one hold time of the transition of ACK to true.

5.1.6 COMMAND Phase

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

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

5.1.7 Data Phase

The data phase is a term that encompasses both the DATA IN phase and the DATA OUT phase.

5.1.7.1 DATA IN Phase

The DATA IN phase allows the target to request that data be sent to the initiator from the target.

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

5.1.7.2 DATA OUT Phase

The DATA OUT phase allows the target to request that data be sent from the initiator to the target.

The target shall negate the C/D, I/O, and MSG signals during the REQ/ACK handshake(s) of this phase.

5.1.8 STATUS Phase

The STATUS phase allows the target to request that status information be sent from the target to the initiator.

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

5.1.9 Message Phase

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

5.1.9.1 MESSAGE IN Phase

The MESSAGE IN phase allows the target to request that message(s) be sent to the initiator from the target.

The target shall assert C/D, I/O, and MSG during the REQ/ACK handshake(s) of this phase.

5.1.9.2 MESSAGE OUT Phase

The MESSAGE OUT phase allows the target to request that message(s) be sent from the initiator to the target. The target may invoke this phase at its convenience in response to the ATTENTION condition (see 5.2.1) created by the initiator.

The target shall assert C/D and MSG and negate I/O during the REQ/ACK handshake(s) of this phase. The target shall handshake byte(s) in this phase until ATN goes false, unless an error occurs (see MESSAGE REJECT, 5.5.2).

If the target detects one or more parity error(s) on the message byte(s) received, it may indicate its desire to retry the message(s) by asserting REQ after detecting ATN has gone false and prior to changing to any other phase. The initiator, upon detecting this condition, shall resend all of the previous message byte(s) sent during this phase. When resending more than one message byte, the initiator shall assert ATN prior to asserting ACK on the first byte and shall maintain ATN asserted until the last byte is sent as described in 5.2.1.

If the target receives all of the message byte(s) successfully (i.e., no parity errors), it shall indicate that it does not wish to retry by changing to any information transfer phase other than the MESSAGE OUT phase and transfer at least one byte. The target may also indicate that it has successfully received the message byte(s) by changing to the BUS FREE phase (*e.g.*, ABORT or BUS DEVICE RESET messages).

5.1.10 Signal Restrictions between Phases

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

a) The BSY, SEL, REQ, and ACK signals shall not change.

b) The C/D, I/O, MSG, and DATA BUS signals may change. When switching the DATA BUS direction from out (initiator driving) to in (target driving), the target shall delay driving the DATA BUS by at least a data release delay plus a bus settle delay after asserting the I/O signal and the initiator shall release the DATA BUS no later than a data release delay after the transition of the I/O signal to true. When switching the DATA BUS direction from in (target driving) to out (initiator driving), the target shall release the DATA BUS no later than a deskew delay after negating the I/O signal.

c) The ATN and RST signals may change as defined under the descriptions for the ATTENTION condition (5.2.1) and RESET condition (5.2.2).

5.2 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.

5.2.1 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 at its convenience by performing a MESSAGE OUT phase.

The initiator creates the ATTENTION condition by asserting ATN at any time except during the ARBITRATION or BUS FREE phases.

The target may respond with the MESSAGE OUT phase.

The initiator shall keep ATN asserted if more than one byte is to be transferred. The initiator may negate the ATN signal at any time except it shall not negate the ATN signal while the ACK signal is asserted during a MESSAGE OUT phase. Normally, the initiator negates ATN while REQ is true and ACK is false during the last REQ/ACK handshake of the MESSAGE OUT phase.

5.2.2 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 asserting RST for a minimum of a reset hold time. During the RESET condition, the state of all SCSI bus signals other than RST is not defined.

All SCSI devices shall release all SCSI bus signals (except RST) within a bus clear delay of the transition of RST to true. The BUS FREE phase always follows the RESET condition.

The effect of the RESET condition on uncompleted commands, SCSI device reservations, and SCSI device operating modes is determined by whether the SCSI device has implemented the “hard” RESET option or the “soft” RESET option (one of which shall be implemented) as defined in 5.2.2.1 and 5.2.2.2.

5.2.2.1 "Hard" RESET Option.
SCSI devices that implement the "hard" RESET option, upon detection of the RESET condition, shall

a) clear all uncompleted commands;

b) release all SCSI device reservations;

c) return any SCSI device operating modes (MODE SELECT, PREVENT/ALLOW MEDIUM REMOVAL commands, etc) to their default conditions.

5.2.2.2 "Soft" RESET Option.
SCSI devices that implement the "soft" RESET option, upon detection of the RESET condition, shall

a) attempt to complete any uncompleted commands that were fully identified;

b) preserve all SCSI device reservations;

c) pPreserve any SCSI device operating modes (MODE SELECT, PREVENT/ALLOW MEDIUM REMOVAL commands, etc).

The "soft" RESET option allows a single initiator to reset the SCSI bus without disturbing the operation of other initiators in a multiple initiator system. To ensure proper operation the following conditions shall be met:

a) An initiator shall not consider a command to be fully identified until the IDENTIFY message 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.

b) A target shall consider a command to be fully identified when it successfully receives the IDENTIFY message.

c) If an initiator selects a logical unit for which there already is an active command for the same initiator, the target shall clear the original command and perform the new command.

d) If a target reselects an initiator to continue a command for which the initiator has no record, the initiator shall abort that command by sending the ABORT message.

e) An initiator shall consider a command to be completed when it negates ACK for a successfully received COMMAND COMPLETE message.

f) A target shall consider a command to be completed when it detects the false transition of ACK for the COMMAND COMPLETE message with the ATN signal false.

g) An initiator shall not negate ACK for the SAVE DATA POINTER message until it has actually saved the data pointer for the operation.

h) A target shall consider the data pointer to be saved when it detects the false transition of ACK for the SAVE DATA POINTER message with the ATN signal false.

i) If the RESET condition occurs between the time that the target asserts REQ for the SAVE DATA POINTER message and it detects the false transition of ACK, the target shall terminate the command with a CHECK CONDITION status. If extended sense is implemented, 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 f) or h), the target would normally switch to MESSAGE OUT phase and attempt to transfer a message byte. If the RESET condition occurs before it is able to successfully receive the message byte, the target shall assume that the initiator may not have successfully received the COMMAND COMPLETE message or the SAVE DATA POINTER message. In the case of COMMAND COMPLETE message, the target shall reselect the initiator and attempt to send the COMMAND COMPLETE message again. In the case of the SAVE DATA POINTER message, the target shall reselect the initiator and terminate the command as described in condition i).

5.3 SCSI Bus Phase Sequences

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

In all systems, the RESET condition can abort any phase and is always followed by the BUS FREE phase. Also, any other phase can be followed by the BUS FREE phase.

5.3.1 Nonarbitrating Systems

In systems where the ARBITRATION phase is not implemented, the allowable sequences shall be as shown in figure 8. The normal progression is from the BUS FREE phase to SELECTION, and from SELECTION to one or more of the information transfer phases (COMMAND, DATA, STATUS, or MESSAGE).

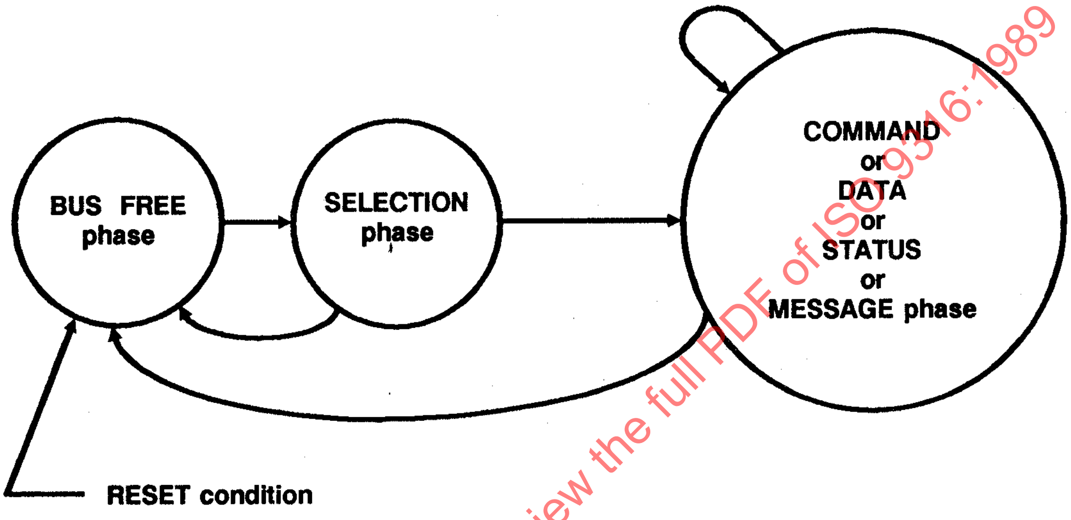


Figure 8 — Phase Sequence without Arbitration

5.3.2 Arbitrating Systems

In systems where the ARBITRATION phase is implemented, the allowable sequences shall be as shown in figure 9. The normal progression is from the BUS FREE phase to ARBITRATION, from ARBITRATION to SELECTION or RESELECTION, and from SELECTION or RESELECTION to one or more of the information transfer phases (COMMAND, DATA, STATUS, or MESSAGE).

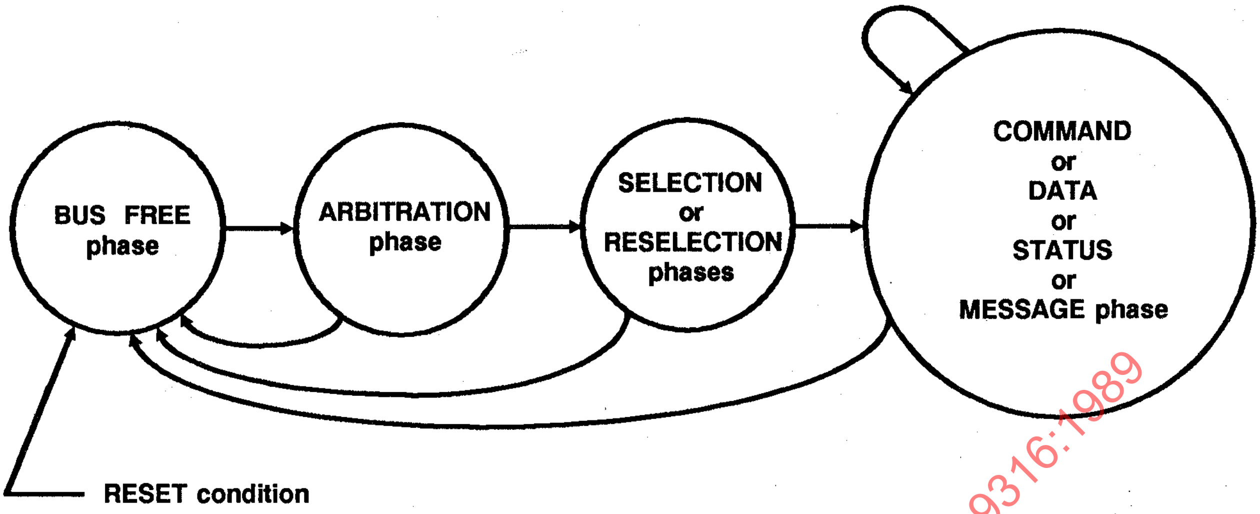

BUS FREE phase
ARBITRATION phase
SELECTION or RESELECTION phases
COMMAND or DATA or STATUS or MESSAGE phase
RESET condition

Figure 9 — Phase Sequence with Arbitration

5.3.3 All Systems

There are no restrictions on the sequences between information transfer phases. A phase type may even be followed by the same phase type (*e.g.*, a data phase may be followed by another data phase).

5.4 SCSI Pointers

Consider the system shown in figure 10 in which an initiator and target communicate on the SCSI bus in order to execute a command.

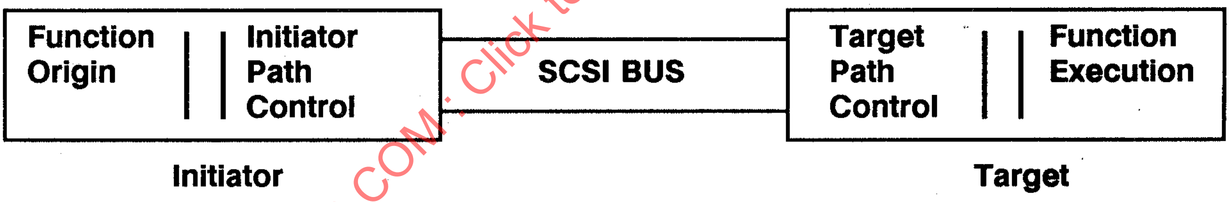


Figure 10 — Simplified SCSI System

The SCSI architecture provides for two sets of three pointers within each initiator. The pointers reside in the initiator path control. The first set of pointers are known as the current (or active) pointers. These pointers are used to represent the state of the interface and point to the next command, data, or status byte to be transferred between the initiator's memory and the target. There is only one set of current pointers in each initiator. The current pointers are used by the target currently connected to the initiator.

The second set of pointers are known as the saved pointers. There is one set of saved pointers for each command that is currently active (whether or not it is currently connected). The saved command pointer always points to the start of the command descriptor block (see 6.2) for the current command. The saved status pointer always points to the start of the status area for the current command. At the beginning of each command, the saved data pointer points to the start of the data area. It remains at this value until the target sends a SAVE DATA POINTER message (see 5.5.2) to the initiator. In response to this message, the initiator stores the value of the current data pointer into the saved data pointer. The target may restore the current pointers to their saved values by sending a RESTORE POINTERS message (see 5.5.2) to the initiator. The initiator moves the saved value of each pointer into the corresponding current pointer. Whenever an SCSI device disconnects from the bus, only the saved pointer values are retained. The current pointer values are restored from the saved values upon the next reconnection.

5.5 Message System Specification

The message system allows communication between an initiator and target for the purpose of physical path management.

5.5.1 Message Protocol

All SCSI devices shall implement the COMMAND COMPLETE message. A functional SCSI device can be constructed without using any of the other messages if the logical unit number is specified in the command descriptor block. The remainder of this section deals with the additional requirements on SCSI devices that support messages other than COMMAND COMPLETE.

SCSI devices indicate their ability to accommodate more than the COMMAND COMPLETE message by asserting or responding to the ATN signal. The initiator indicates this in the SELECTION phase by asserting ATN prior to the SCSI bus condition of SEL true, and BSY false. The target indicates its ability to accommodate more messages by responding to the ATTENTION condition with the MESSAGE OUT phase after going through the SELECTION phase.

For SCSI devices that support messages other than COMMAND COMPLETE, the first message sent by the initiator after the SELECTION phase shall be the IDENTIFY message. This allows the establishment of the physical path for a particular logical unit specified by the initiator. After the RESELECTION phase, the target's first message shall be IDENTIFY. This allows the physical path to be reestablished for the target's specified logical unit number. Under some exceptional conditions, an initiator may send the ABORT message or the BUS DEVICE RESET message instead of the IDENTIFY message, as the first message. Only one logical unit number shall be identified for any one selection sequence; a second IDENTIFY message with a new logical unit number shall not be issued before the SCSI bus has been released (BUS FREE phase).

Whenever a physical path is established in an initiator that can accommodate disconnection and reconnection, the initiator shall ensure that the active pointers of the physical path are equal to the saved pointers for that particular logical unit number. (An implied restore pointers operation occurs as a result of connect or reconnect.)

SCSI devices that implement any message other than the COMMAND COMPLETE message shall also implement the MESSAGE REJECT message.

5.5.2 Messages

The single byte messages (table 5) are listed along with their code values and their definitions.

Table 5 – Message Codes

Code	Type	Description	Direction
00h	M	COMMAND COMPLETE	In
01h	O	EXTENDED MESSAGE	In Out
02h	O	SAVE DATA POINTER	In
03h	O	RESTORE POINTERS	In
04h	O	DISCONNECT	In
05h	O	INITIATOR DETECTED ERROR	Out
06h	O	ABORT	Out
07h	O	MESSAGE REJECT	In Out
08h	O	NO OPERATION	Out
09h	O	MESSAGE PARITY ERROR	Out
0Ah	O	LINKED COMMAND COMPLETE	In
0Bh	O	LINKED COMMAND COMPLETE (WITH FLAG)	In
0Ch	O	BUS DEVICE RESET	Out
0Dh - 7Fh	R	Reserved Codes	
80h - FFh	O	IDENTIFY	In Out

Key: In = Target to initiator, Out = Initiator to target.

COMMAND COMPLETE 00h (Mandatory). This message is sent from a target to an initiator to indicate that the execution of a command (or series of linked commands) has terminated 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 BSY. The command may have been executed successfully or unsuccessfully as indicated in the status.

EXTENDED MESSAGE 01h (Optional). This message is sent from either the initiator or the target as the first byte of a multiple-byte message. (See 5.5.3 for descriptions of extended messages.)

SAVE DATA POINTER 02h (Optional). This message is sent from a target to direct the initiator to save a copy of the present active data pointer for the currently attached logical unit. (See 5.4 for a definition of pointers.)

RESTORE POINTERS 03h (Optional). This message is sent from a target to direct the initiator to restore the most recently saved pointers (for the currently attached logical unit) to the active state. Pointers to the command, data, and status locations for the logical unit shall be restored to the active pointers. 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 logical unit.

DISCONNECT 04h (Optional). This message is sent from a target to inform an initiator that the present physical path is going to be broken (the target plans to disconnect by releasing BSY), but that a later reconnect will be required in order to complete the current operation. If the initiator detects the BUS FREE phase (other than as a result of a RESET condition) without first receiving a DISCONNECT or COMMAND COMPLETE message, the initiator shall consider this as a catastrophic error condition. If the target intentionally creates this condition, the target shall clear the current command. This message shall not cause the initiator to save the data pointer. If DISCONNECT messages are used to break a long data transfer into two or more shorter transfers, then a SAVE DATA POINTER should be issued before each DISCONNECT message.

INITIATOR DETECTED ERROR 05h (Optional). This message is sent from an initiator to inform a target that an error (*e.g.*, parity error) has occurred that does not preclude the target from retrying the operation. 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.

ABORT 06h (Optional). This message is sent from the initiator to the target to clear the present operation. If a logical unit has been identified, all pending data and status for the issuing initiator from the affected logical unit shall be cleared, and the target shall go to the BUS FREE phase. Pending data and status for other initiators shall not be cleared. If a logical unit has not been identified, the target shall go to the BUS FREE phase. No status or ending message shall be sent for the operation. It is not an error to issue this message to an logical unit that is not currently performing an operation for the initiator.

MESSAGE REJECT 07h (Optional). This message is sent from either the initiator or target to indicate that the last message 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 ACK for the REQ/ACK handshake of the message that is to be rejected. 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 is rejected.

This message shall be implemented if any other optional messages are implemented.

NO OPERATION 08h (Optional). This 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.

MESSAGE PARITY ERROR 09h (Optional). This message is sent from the initiator to the target to indicate that one or more bytes in the last message it received had 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 ACK for the REQ/ACK handshake of the message that has the parity error. This provides an interlock so that the target can determine which message has the parity error.

LINKED COMMAND COMPLETE 0Ah (Optional). This 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.

LINKED COMMAND COMPLETE (WITH FLAG) 0Bh (Optional). This 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.

BUS DEVICE RESET 0Ch (Optional). This message is sent from an initiator to direct a target to clear all current commands on that SCSI device. This message forces the SCSI device to an initial state with no operations pending for any initiator. Upon recognizing this message, the target shall go to the BUS FREE phase.

Reserved 0Dh to 7Fh. These message codes are reserved for future standardization.

IDENTIFY 80h to FFh (Optional). These messages are sent by either the initiator or the target to establish the physical path connection between an initiator and target for a particular logical unit.

Bit 7. This bit is always set to one to distinguish these messages from the other messages.

Bit 6. This bit is only set to one by the initiator. When set to one, it indicates that the initiator has the ability to accommodate disconnection and reconnection.

Bits 5-3. Reserved.

Bits 2-0. These bits specify a logical unit number in a target.

Only one logical unit number shall be identified for any one selection sequence; a second IDENTIFY message with a new logical unit number shall not be issued before the bus has been released (BUS FREE phase).

When sent from a target to an initiator during reconnection, an implied RESTORE POINTERS message shall be performed by the initiator prior to completion of this message.

5.5.3 Extended Messages (Optional)

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 6 and 7, respectively.

Table 6 – Extended Message Format

Byte	Value	Description
0	01h	Extended message
1	n	Extended message length
2	y	Extended message code
3 to n+1	x	Extended message arguments

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 7. The extended message arguments are specified for the defined extended messages in 5.5.4 through 5.5.6.

Table 7 – Extended Message Codes

Code (y)	Description
00h	MODIFY DATA POINTER (Optional)
01h	SYNCHRONOUS DATA TRANSFER REQUEST (Optional)
02h	EXTENDED IDENTIFY (Optional)
03h - 7Fh	Reserved
80h - FFh	Vendor Unique

5.5.4 MODIFY DATA POINTER Message (Optional)

The MODIFY DATA POINTER message (table 8) 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.

Table 8 – MODIFY DATA POINTER

Byte	Value	Description
0	01h	Extended message
1	05h	Extended message length
2	00h	MODIFY DATA POINTER code
3	x	Argument
4	x	Argument
5	x	Argument
6	x	Argument (LSB)

5.5.5 SYNCHRONOUS DATA TRANSFER REQUEST Message (Optional)

A pair of SYNCHRONOUS DATA TRANSFER REQUEST messages (table 9) are exchanged between an initiator and a target whenever an SCSI device that can support synchronous data transfer recognizes that it has not communicated with the other SCSI device since receiving the last "hard" RESET condition or a BUS DEVICE RESET message. The SCSI devices may also exchange messages to establish synchronous data transfer when requested to do so. The message exchange establishes the transfer period and the REQ/ACK offset. The transfer period is the minimum time between leading edges of successive REQ pulses and of successive ACK pulses.

Table 9 – SYNCHRONOUS DATA TRANSFER REQUEST

Byte	Value	Description
0	01h	Extended message
1	03h	Extended message length
2	01h	SYNCHRONOUS DATA TRANSFER REQUEST code
3	m	Transfer period (m times 4 nanoseconds)
4	x	REQ/ACK offset

The REQ/ACK offset is the maximum number of REQ pulses that may be outstanding before its corresponding ACK pulse is received at the target. A REQ/ACK offset value of zero shall indicate asynchronous mode; a value of FFh shall indicate unlimited offset.

If the initiator recognizes that negotiation is required, it asserts ATN and, if the target implements message transfers, sends a SYNCHRONOUS DATA TRANSFER REQUEST message indicating an REQ/ACK offset and minimum transfer period. The REQ/ACK offset is chosen to prevent initiator buffer overflows, while the minimum transfer period is chosen to meet the data handling requirements of the initiator. The target responds in any of the following ways:

Target Response	Implied Agreement
a) REQ/ACK offset less than or equal to the requested value. Minimum transfer period equal to or greater than requested period.	REQ/ACK offset equal to target value. Minimum transfer period equal to the target value.
b) REQ/ACK offset equal to zero.	Asynchronous transfer.
c) MESSAGE REJECT.	Asynchronous transfer.

If the target recognizes that negotiation is required, it sends a SYNCHRONOUS DATA TRANSFER REQUEST message to the initiator. The REQ/ACK offset is selected to prevent buffer and offset counter overflows, while the minimum transfer period is chosen to meet the data handling requirements of the target. The initiator responds in any of the following ways if the target chooses an REQ/ACK offset equal to FFh:

Initiator Response	Implied Agreement
a) REQ/ACK offset equal to FFh. Minimum transfer period equal to. or greater than requested period.	REQ/ACK offset unlimited. Minimum transfer period equal to the initiator value.
b) REQ/ACK offset equal to zero.	Asynchronous transfer. The targer may renegotiate for an REQ/ACK offset less than FFh and greater than 00h.
c) MESSAGE REJECT.	Asynchronous transfer.

The initiator responds in any of the following ways if the target selects an REQ/ACK offset less than FFh.

Initiator Response	Implied Agreement
a) REQ/ACK offset less than or equal to the requested value. Minimum transfer period equal to or greater than requested period.	REQ/ACK offset equals initiator value. Minimum transfer period equal to the initiator value.
b) REQ/ACK offset equal to zero.	Asynchronous transfer.
c) MESSAGE REJECT.	Asynchronous transfer.

The implied 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. Renegotiation at every selection is not recommended, since a significant performance impact is likely. The default mode of data transfer is asynchronous mode. The default mode is entered at power on, after a BUS DEVICE RESET message, or after a "hard" RESET condition. The SYNCHRONOUS DATA TRANSFER REQUEST message exchange can only take place following a SELECTION phase that includes the SCSI IDs for both the initiator and the target. Violation of this rule may make data transfer impossible owing to disagreements among SCSI devices about the data transfer mode.

5.5.6 EXTENDED IDENTIFY Message (Optional)

The EXTENDED IDENTIFY message (table 10) is optional and may be sent by a target or an initiator. It may be used in conjunction with the normal IDENTIFY message in order to expand the logical unit number addressing in a target. The sub-logical unit number specifies the encoded eight-bit sub- logical unit number used to identify one of 256 sub-logical units within the logical unit. This allows up to 2048 units to be addressed on a single target.

Table 10 – EXTENDED IDENTIFY

Byte	Value	Description
0	01h	Extended message
1	02h	Extended message length
2	02h	EXTENDED IDENTIFY code
3	x	Sub-logical unit number

6 SCSI commands

This section defines the SCSI command structure and gives several examples.

The command definitions assume a data structure providing the appearance at the interface of a contiguous set of logical blocks of a fixed or explicitly defined data length. The SCSI device maps the physical characteristics of the attached peripheral devices to one of several logical structures defined by the device type code.

A single command may transfer one or more logical blocks of data. Multiple commands may be linked if they are sent to the same logical unit. A target may disconnect from the SCSI bus to allow activity by other SCSI devices while a logical unit is being prepared to transfer data.

Upon command completion (successful or unsuccessful), the target returns a status byte to the initiator. Since most error and exception conditions cannot be adequately described with a single status byte, one status code, CHECK CONDITION, indicates that additional information is available. The initiator may issue a REQUEST SENSE command to retrieve this additional information.

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.

6.1 Command Implementation Requirements.

The first byte of any SCSI command shall contain an operation code as defined in this International Standard. Three bits (bits 7 to 5) of the second byte of each SCSI command specify the logical unit if it is not specified using the IDENTIFY message (see 5.5.2). The last byte of all SCSI commands shall contain a control byte as defined in 6.2.6.

6.1.1 Reserved.

Reserved bits, fields, bytes, and code values are set aside for future standardization. Their use and interpretation will be specified by future extensions to this International 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 a CHECK CONDITION status and, if extended sense is implemented, the sense key shall be set to ILLEGAL REQUEST. It shall also be acceptable for a target to interpret the bit, field, byte, or code value in accordance with a future extension to this International Standard.

6.1.2 Operation Code Types

The Operation code types are defined in table 11.

Table 11 – Operation Codes

Operation Code Type	Description
M	Mandatory - Commands so designated shall be implemented in order to meet the minimum requirement of this International Standard.
E	Extended - Commands so designated shall be implemented in addition to mandatory commands to meet the extended requirement of this Internaional Standard.
O	Optional - Commands so designated, if implemented, shall be implemented as defined in this International Standard.
V	Vendor unique - 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 International Standard.

6.1.3 Unit Attention Condition

A unit attention condition for a logical unit shall begin for each initiator whenever the removable medium may have been changed or the target has been reset (by a BUS DEVICE RESET message or a "hard" RESET condition). The unit attention condition shall persist for each initiator until that initiator issues a command to the logical unit other than REQUEST SENSE or INQUIRY for which the target shall return CHECK CONDITION status. If the next command from that initiator to the logical unit (following the CHECK CONDITION status) is REQUEST SENSE, and if the target supports extended sense, then the UNIT ATTENTION sense key shall be returned. (If any command other than REQUEST SENSE is received, the unit attention condition is lost.)

If an INQUIRY command is received from an initiator with a pending unit attention condition (before the target reports CHECK CONDITION status), the target shall perform the INQUIRY command and shall not clear the unit attention condition.

If a REQUEST SENSE command is received from an initiator with a pending unit attention condition (before the target reports CHECK CONDITION status), then the target may either:

a) report any pending sense data and preserve the unit attention condition;

b) discard any pending sense data, report UNIT ATTENTION sense key, and clear the unit attention condition for that initiator.

If an initiator issues a command other than INQUIRY or REQUEST SENSE while a unit attention condition exists for that initiator, the target shall not perform the command and shall report CHECK CONDITION status.

6.2 Command Descriptor Block (CDB)

A request to a peripheral device is performed by sending a command descriptor block to the target. For several commands, the request is accompanied by a list of parameters sent during the DATA OUT phase. See the specific commands for detailed information.

The command descriptor block always has an operation code as the first byte of the command. This is followed by a logical unit number, command parameters (if any), and a control 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.

6.2.1 Operation Code.

The operation code (table 12) 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 clauses 7 to 13.

Table 12 – Operation Code

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

The group code specifies one of the following groups:

Group 0 - six-byte commands (see table 13)
Group 1 - ten-byte commands (see table 14)
Group 2 - reserved
Group 3 - reserved
Group 4 - reserved
Group 5 - twelve-byte commands (see table 15)

Group 6 - vendor unique
Group 7 - vendor unique

Table 13 – Typical CDB for Six-Byte Commands

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

Table 14 – Typical CDB for Ten-Byte Commands

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

Table 15 – Typical CDB for Twelve-Byte Commands

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

6.2.2 Logical Unit Number

The logical unit number addresses one of up to eight physical or virtual devices attached to a target. This method of addressing is provided for systems that do not implement the IDENTIFY message. A target that accepts an IDENTIFY message shall use the logical unit number specified within the message. In this case, the target shall ignore the logical unit number specified within the command descriptor block. (Implementors note: It is a good practice for initiators that implement the IDENTIFY message to specify the same logical unit number in the command descriptor block.)

6.2.3 Logical Block Address

The logical block address on logical units shall begin with block zero and be contiguous up to the last logical block on that logical unit.

Group 0 command descriptor blocks contain 21-bit logical block addresses. Groups 1 and 5 command descriptor blocks contain 32-bit logical block addresses.

The logical block concept implies that the initiator and target shall have previously established the number of data bytes per logical block. This may be established through the use of the READ CAPACITY command or the MODE SENSE command or by prior arrangement.

6.2.4 Relative Address Bit

The relative address (RelAdr) bit of the group 1 and group 5 commands is set to one to indicate that the logical block address portion of the command descriptor block is a two's complement displacement. This negative or positive displacement is to be added to the logical block address last accessed on the logical unit to form the logical block address for this command. This feature is only available when linking commands. The feature requires that a previous command in the linked group have accessed a block of data on the logical unit. (For an example of the operation of this function, see 6.3.3.)

6.2.5 Transfer Length

The transfer length specifies the amount of data to be transferred, usually the number of blocks. For several commands the transfer length indicates the requested number of bytes to be sent as defined in the command description. For these commands the transfer length field may be identified by a different name. See the following descriptions and the individual command descriptions for further information.

Commands that use one byte for transfer length allow up to 256 blocks of data to be transferred by one command. A transfer length value of 1 to 255 indicates the number of blocks that shall transferred. A value of zero indicates 256 blocks.

Commands that use two bytes for transfer length allow up to 65,535 blocks of data to be transferred by one command. In this case, a transfer length of zero indicates that no data transfer shall take place. A value of 1 to 65,535 indicates the number of blocks that shall be transferred.

For several commands more than two bytes are allocated for transfer length. Refer to the specific command description for further information.

The transfer length of the commands that are used to send a list of parameters to a target is called the parameter list length. The parameter list length specifies the number of bytes sent during the DATA OUT phase.

The transfer length of the commands that are used to return sense data (*e.g.*, REQUEST SENSE, INQUIRY, MODE SENSE, etc) to an initiator is called the allocation length. The allocation length specifies the number of bytes that the initiator has allocated for returned data. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available sense data have been transferred to the initiator, whichever is less.

6.2.6 Control Byte

The control byte is the last byte of every command descriptor block. A typical control byte is described in table 16.

Table 16 – Control Byte

Byte \ Bit	7	6	5	4	3	2	1	0
Last	Vendor Unique		Reserved				Flag	Link

Bit	Description
7 – 6	Vendor unique
5 – 2	Reserved
1	Flag bit - If the link bit is zero, then the flag bit shall be set to zero. If the link bit is one, and if the command terminates successfully, the target shall send LINKED COMMAND COMPLETE message if the flag bit is zero and shall send LINKED COMMAND COMPLETE (WITH FLAG) message if the flag bit is one. Typically, this bit is used to cause an interrupt in the initiator between linked commands.
0	Link bit - This bit is set to one to indicate that the initiator desires an automatic link to the next command upon successful completion of the current command. Implementation of linked commands is optional. If the link bit is one, targets that implement linked commands, upon successful termination of the command, shall return INTERMEDIATE status and shall then send one of the two messages defined by the flag bit (above). Targets that do not implement linked commands shall return a CHECK CONDITION status and, if extended sense is implemented, shall set the sense key to ILLEGAL REQUEST if either of the link and flag bits are set to one.

6.3 Command Examples

6.3.1 Single Command Example

A typical operation on the SCSI bus is likely to include a single READ command to a peripheral device. This operation is described in detail starting with a request from the initiator. This example assumes that no linked commands and no malfunctions or errors occur.

The initiator has active pointers and a set of stored pointers representing active disconnected SCSI devices (an initiator without disconnect capability does not require stored pointers). The initiator sets up the active pointers for the operation requested, arbitrates for the SCSI bus, and selects the target. Once this process is completed, the target assumes control of the operation.

The target obtains the command from the initiator (in this case, a READ command). The target interprets the command and executes it. In this case, the target gets the data from the peripheral device and sends it to the initiator. At the completion of the READ command, the target sends a status byte to the initiator. To end the operation, the target sends a COMMAND COMPLETE message to the initiator.

6.3.2 Disconnect Example

In the above single command example, the length of time necessary to obtain the data may require a time-consuming physical seek. In order to improve system throughput, the target may disconnect from the initiator, freeing the SCSI bus to allow other requests to be sent to other logical units. To do this, the initiator needs to be reselectable and capable of restoring the pointers upon reconnection. The target needs to be capable of arbitrating for the SCSI bus and reselecting the initiator.

After the target has received the READ command (and has determined that there will be a delay), it disconnects by sending a DISCONNECT message and releasing BSY.

When the data are ready to be transferred, the target reconnects to the initiator. As a result of this reconnection, the initiator restores the 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 operation. The initiator recognizes that the operation is complete when COMMAND COMPLETE message is received.

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 value 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.

6.3.3 Linked Command Example

The link function defines a relationship between commands that when combined with the relative address bit, allows previous operations to modify subsequent operations. Link makes high- performance functions possible by providing a relative addressing capability and allowing multiple command execution without invoking the functional component of the initiator.

If the desired data address (in the previously described READ command example) is unknown, but a search key defined as some particular bytes of a field is known, then by linking the READ command to a SEARCH DATA EQUAL command, the data can be quickly and effectively transferred to the initiator.

A LINKED COMMAND COMPLETE message is sent from the target to the initiator to indicate linked command completion. The initiator then updates the stored pointers so that subsequent requests from the target will reference the next command of the chain. Command processing of linked and single commands is similar except that relative addressing is permitted in linked commands. Linked commands shall be addressed to a single logical unit.

For example, the successful completion of a SEARCH DATA EQUAL command causes the target to fetch 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.

7 Command descriptions for all device types

7.1 Group 0 Commands for All Device Types

These commands shall be as listed in table 17.

Table 17 – Group 0 Commands for all Device Types

Operation Code	Type	Command Name	Subclause	Page
00h	O	TEST UNIT READY	7.1.1	44
01h	*			
02h	V			
03h	M	REQUEST SENSE	7.1.2	44
04h	*			
05h	*			
06h	V			
07h	*			
08h	*			
09h	V			
0Ah	*			
0Bh	*			
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	*			
10h	*			
11h	*			
12h	E	INQUIRY	7.1.3	48
13h	*			
14h	*			
15h	*			
16h	*			
17h	*			
18h	O	COPY	7.1.4	50
19h	*			
1Ah	*			
1Bh	*			
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	57
1Dh	O	SEND DIAGNOSTIC	7.1.6	58
1Eh	*			
1Fh	R			

Key:

M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.
* = These operation codes may have different meanings for specific types of peripheral devices. (See the appropriate section for further information.)

7.1.1 TEST UNIT READY Command

Peripheral Device Type: All
Operation Code Type: Optional
Operation Code: 00h

Table 18 – TEST UNIT READY Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The TEST UNIT READY command (table 18) provides a means to check if the logical unit is ready. This is not a request for a self test. If the logical unit would accept an appropriate medium-access command without returning CHECK CONDITION status, this command shall return a GOOD status.

7.1.2 REQUEST SENSE Command

Peripheral Device Type: All
Operation Code Type: Mandatory
Operation Code: 03h

Table 19 – REQUEST SENSE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The REQUEST SENSE command (table 19) requests that the target transfer sense data to the initiator.

The sense data shall be valid for a CHECK CONDITION status returned on the prior command. This sense data shall be preserved by the target for the initiator until retrieved by the REQUEST SENSE command or until the receipt of any other command for the same logical unit from the initiator that issued the command resulting in the CHECK CONDITION status. Sense data shall be cleared upon receipt of any subsequent command to the logical unit from the initiator receiving the CHECK CONDITION status. In the case of the single initiator option (see 5.1.3.4), the target shall assume that the REQUEST SENSE command is from the same initiator.

The allocation length specifies the number of bytes that the initiator has allocated for returned sense data. An allocation length of zero indicates that four bytes of sense data shall be transferred. Any other value indicates the maximum number of bytes that shall be transferred. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available sense data have been transferred to the initiator, whichever is less.

The REQUEST SENSE command shall return the CHECK CONDITION status only to report fatal errors for the REQUEST SENSE command. For example:

a) The target receives a nonzero reserved bit in the command descriptor block.

b) An unrecovered parity error occurs on the DATA BUS.

c) A target malfunction prevents return of the sense data.

If any nonfatal error occurs during the execution of the REQUEST SENSE command, the target shall return the sense data with GOOD status.

Following a fatal error on a REQUEST SENSE command, sense data may be invalid.

A target may implement the nonextended, the extended, or both sense data formats.

Note: Targets that implement both sense data formats may select the nonextended sense data format in response to an allocation length of zero. Other methods of selection are also feasible.)

7.1. 2.1 Nonextended Sense The format of the sense data is determined by the error class. Error classes 0 through 6 use the nonextended sense data format (table 20). Error class 7 is described in 7.1.2.2.

Table 20 – Nonextended Sense Data

Byte \ Bit	7	6	5	4	3	2	1	0
0	AdValid	Error Class			Error Code			
1	Vendor Unique			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							

The address valid (AdValid) bit indicates that the logical block address field contains valid information related to the error code.

The error class specifies a class of errors with error classes 0 through 6 being vendor unique. For these classes, the error code is vendor unique.

7.1. 2. 2 Extended Sense
Error class 7 specifies extended sense. Error code zero specifies the extended sense data format. Error code Fh specifies a vendor unique data format for extended sense. Error codes 1h through Eh are reserved.

The extended sense data format is shown in table 21.

Table 21 – Extended Sense Data

Byte \ Bit	7	6	5	4	3	2	1	0
0	Valid	Error Class (7)			Error Code (0)			
1	Segment Number							
2	Filemark	EOM	ILI	Reserved	Sense Key			
3	Information Byte (MSB)							
4	Information Byte							
5	Information Byte							
6	Information Byte (LSB)							
7	Additional Sense Length (n)							
8 to n + 7	Additional Sense Bytes							

The information bytes are not defined if the valid bit is zero. If the valid bit is one, the information bytes contain valid information as follows:

a) The unsigned logical block address associated with the sense key, for direct-access devices (Type 0), write-once read-multiple devices (Type 4), and read-only direct-access devices (Type 5).

b) The difference (residue) of the requested length minus the actual length in either bytes or blocks, as determined by the command, for sequential-access devices (Type 1), printer devices (Type 2), and processor devices (Type 3). (Negative values are indicated by two's complement notation.)

c) The difference (residue) of the requested number of blocks minus the actual number of blocks copied or compared for the current segment descriptor of a COPY, COMPARE, or COPY AND VERIFY command.

The segment number contains the number of the current segment descriptor if the extended sense is in response to a COPY, COMPARE, or COPY AND VERIFY command. Up to 256 segments are supported beginning with segment zero.

The filemark bit indicates that the current command has read a filemark. This bit is only used for sequential-access devices.

The end-of-medium (EOM) bit indicates that an end-of-medium condition (end- of-tape, beginning-of-tape, out-of-paper, etc) exists on a sequential access device or printer device. For sequential-access devices, this bit indicates that the unit is at or past the early-warning end-of-tape if the direction was forward or that the command could not be completed because beginning-of-tape was encountered if the direction was reverse. Direct-access devices shall not use this bit; instead, these devices shall report attempts to access beyond the end-of-medium as ILLEGAL REQUEST sense key (see table 22).

The incorrect length indicator (ILI) bit indicates that the requested logical block length did not match the logical block length of the data on the medium.

Table 22 – Sense Key Description

Sense Key	Description
0h	NO SENSE. Indicates that there is no specific sense key information to be reported for the designated logical unit. This would be the case for a successful command or a command that received a CHECK CONDITION status because one of the filemark, EOM, or ILI bits is set to one.
1h	RECOVERED ERROR. Indicates that the last command completed successfully with some recovery action performed by the target. Details may be determinable by examining the additional sense bytes and the information bytes.
2h	NOT READY. Indicates that the logical unit addressed cannot be accessed. Operator intervention may be required to correct this condition.
3h	MEDIUM ERROR. Indicates that the command terminated with a nonrecovered error condition that was probably caused by a flaw in the medium or an error in the recorded data.
4h	HARDWARE ERROR. Indicates that the target detected a nonrecoverable hardware failure (for example, controller failure, device failure, parity error, etc.) while performing the command or during a self test.
5h	ILLEGAL REQUEST. Indicates that there was an illegal parameter in the command descriptor block or in the additional parameters supplied as data for some commands (FORMAT UNIT, SEARCH DATA, etc). If the target detects an invalid parameter in the command descriptor block, then it shall terminate the command without altering the medium. If the target detects an invalid parameter in the additional parameters supplied as data, then the target may have already altered the medium.
6h	UNIT ATTENTION. Indicates that the removable medium may have been changed or the target has been reset. See 6.1.3 for more detailed information about the unit attention condition.
7h	DATA PROTECT. Indicates that a command that reads or writes the medium was attempted on a block that is protected from this operation. The read or write operation is not performed.
8h	BLANK CHECK. Indicates that a write-once read-multiple device or a sequential-access device encountered a blank block while reading or a write-once read-multiple device encountered a non-blank block while writing.
9h	Vendor unique. This sense key is available for reporting vendor unique conditions.
Ah	COPY ABORTED. Indicates a COPY, COMPARE, or COPY AND VERIFY command was aborted due to an error condition on the source device, the destination device, or both. (See 7.1.4.2 for additional information about this sense key.)
Bh	ABORTED COMMAND. Indicates that the target aborted the command. The initiator may be able to recover by trying the command again.
Ch	EQUAL. Indicates a SEARCH DATA command has satisfied an equal comparison.
Dh	VOLUME OVERFLOW. Indicates that a buffered peripheral device has reached the end-of-medium and data remains in the buffer that has not been written to the medium. A RECOVER BUFFERED DATA command(s) may be issued to read the unwritten data from the buffer.
Eh	MISCOMPARE. Indicates that the source data did not match the data read from the medium.
Fh	This sense key is reserved

The sense keys are described in table 22.

The additional sense length specifies the number of additional sense bytes to follow. If the allocation length of the command descriptor block is too small to transfer all of the additional sense bytes, the additional sense length is not adjusted to reflect the truncation.

The additional sense bytes contain command-specific, peripheral-device-specific data, or both kinds of data that further define the nature of the CHECK CONDITION status. The COPY, COMPARE, COPY AND VERIFY, and SEARCH DATA commands define a standard purpose for some of these bytes. Except as described in these commands, the additional sense bytes are vendor unique.

7.1.3 INQUIRY Command

Peripheral Device Type: All
Operation Code Type: Extended
Operation Code: 12h

Table 23 – INQUIRY Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The INQUIRY command (table 23) requests that information regarding parameters of the target and its attached peripheral device(s) be sent to the initiator.

The allocation length specifies the number of bytes that the initiator has allocated for returned INQUIRY data. An allocation length of zero indicates that no INQUIRY data shall be transferred. This condition shall not be considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available INQUIRY data have been transferred to the initiator, whichever is less.

The INQUIRY command shall return a CHECK CONDITION status only when the target cannot return the requested INQUIRY data.

Note: It is recommended that the INQUIRY data be returned even though the peripheral device may not be ready for other commands.)

If an INQUIRY command is received from an initiator with a pending unit attention condition (before the target reports CHECK CONDITION status), the target shall perform the INQUIRY command and shall not clear the unit attention condition. (See 6.1.3.)

The INQUIRY data (table 24) contains a five-byte header, followed by the vendor unique parameters, if any.

Table 24 – INQUIRY Data

Byte \ Bit	7	6	5	4	3	2	1	0
0	Peripheral Device Type							
1	RMB	Device-Type Qualifier						
2	ISO Version		ECMA Version			ANSI-Approved Version		
3	Reserved							
4	Additional Length (n)							
	Vendor Unique Parameters							
5 to n + 4	Vendor Unique Parameter Bytes							

The peripheral device type is shown in table 25.

Table 25 – Peripheral Device Type

Code	Description
00h	Direct-access device (e.g., magnetic disk)
01h	Sequential-access device (e.g., magnetic tape)
02h	Printer device
03h	Processor device
04h	Write-once read-multiple device (e.g., some optical disks)
05h	Read-only direct-access device (e.g., some optical disks)
06h - 7Eh	Reserved
7Fh	Logical unit not present
80h - FFh	Vendor unique

A removable medium (RMB) bit of zero indicates that the medium is not removable. A RMB bit of one indicates that the medium is removable.

The device-type qualifier is a seven-bit user-specified code. This code may be set with switches or by some other means by the target or peripheral device. SCSI devices that do not support this feature shall return all zero bits. This feature allows each user to assign unique codes to each specific type of peripheral device that is supported on the system being used. These codes may then be used by self-configuring software to determine what specific peripheral device is at each logical unit number. This is especially valuable for systems that support multiple types of removable medium.

The ISO version indicates the implemented version of this standard and is defined as shown in table 26.

Table 26 – ISO Version

Code	Description
0h	Version is unspecified.
1h	This version. This code shall be used by SCSI devices that claim to comply with this International Standard
2h - 3h	Reserved.

The usage of nonzero code values in the ECMA version and ANSI-approved version fields are defined by the European Computer Manufacturers Association and the American National Standards Institute, respectively. A zero code value in these fields shall indicate that the target does not claim compliance to the the ECMA version of SCSI (ECMA-111) or the ANSI-approved version of SCSI (ANSI X3.131-1986). Note that it is possible to claim compliance to more than one of these SCSI standards.

The additional length shall specify the length in bytes of the vendor unique parameters. If the allocation length of the command descriptor block is too small to transfer all of the vendor unique parameters, the additional length shall not be adjusted to reflect the truncation.

7.1.4 COPY Command

Peripheral Device Type: All
Operation Code Type: Optional
Operation Code: 18h

Table 27 – COPY Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Parameter List Length (MSB)							
3	Parameter List Length							
4	Parameter List Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The COPY command (table 27) provides a means to copy data from one logical unit to another or the same logical unit. The logical units may reside on the same SCSI device or different SCSI devices. Some SCSI devices that implement this command may not support copies to or from another SCSI device or third party copies (both logical units reside on other SCSI devices).

The parameter list length specifies the length in bytes of the parameters that shall be sent during the DATA OUT phase of the command. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

The COPY parameter list (table 28) begins with a four-byte header that contains the COPY function code and priority. Following the header is one or more segment descriptors.

Table 28 – COPY Parameter List

Byte \ Bit	7	6	5	4	3	2	1	0
0	Copy Function Code					Priority		
1	Vendor Unique							
2	Reserved							
3	Reserved							
	Segment Descriptor(s)							
0 to xx	Segment Descriptor 0 (See specific table for length)							
	⋮							
0 to xx	Segment Descriptor n (See specific table for length)							

The COPY function code defines a specific format for the segment descriptors. The COPY function codes are defined in table 29.

The priority field of the COPY parameter list establishes the relative priority of this COPY command to other commands being executed by the same target. All other commands are assumed to have a priority of 1. Priority 0 is the highest priority with increasing values indicating lower priorities.

The segment descriptor formats are determined by the COPY function code.

The segment descriptor format used for write-once read-multiple devices and for read-only direct-access devices shall be the same as for direct-access devices. The segment descriptor format used for printer devices and for processor devices shall be the same as for sequential-access devices. Thus a COPY from a write-once read-multiple device to a printer device uses the same segment descriptor format as for a COPY from a direct-access device to a sequential-access device. (See table 29.) The segment descriptor formats are described in tables 29 through 31. A maximum of 256 segment descriptors are permitted. The segment descriptors are identified by ascending numbers beginning with zero.

7.1.4.1 Errors Detected by the Managing SCSI Device

Two classes of unusual conditions may occur during execution of a COPY command. The first class consists of those unusual conditions detected by the SCSI device that received the COPY command and is managing the execution of the command. These conditions include parity errors while transferring the COPY command and status byte, invalid parameters in the COPY command, invalid segment descriptors, and inability of the SCSI device controlling the COPY functions to continue operating. In the event of such an unusual condition, the SCSI device managing the COPY shall

a) terminate the COPY command with a CHECK CONDITION status,

b) return the sense data in the extended sense format. The valid bit shall be set to one. The segment number shall contain the number of the segment descriptor being processed at the time the unusual condition is detected. The sense key shall contain the sense key code describing the unusual condition. The information bytes shall contain the difference between the number of blocks field in the segment descriptor being processed at the time of the failure and the number of blocks successfully copied. This number is the residue of unprocessed blocks remaining for the segment descriptor.

Table 29 – COPY Functions

Peripheral Device Type		COPY Function Code	Segment Descriptor Table	Comment
Source	Destination			
00h	01h	00h	table 30	Direct Access to Sequential Access
00h	02h	00h	table 30	
00h	03h	00h	table 30	
04h	01h	00h	table 30	
04h	02h	00h	table 30	
04h	03h	00h	table 30	
05h	01h	00h	table 30	
05h	02h	00h	table 30	
05h	03h	00h	table 30 15	
01h	00h	01h	table 30	Sequential Access to Direct Access
01h	04h	01h	table 30	
03h	00h	01h	table 30	
03h	04h	01h	table 30	
00h	00h	02h	table 31	Direct Access to Direct Access
00h	04h	02h	table 31	
04h	00h	02h	table 31	
04h	04h	02h	table 31	
05h	00h	02h	table 31	
05h	04h	02h	table 31	
01h	01h	03h	table 32	Sequential Access to Sequential Access
01h	02h	03h	table 32	
01h	03h	03h	table 32	
03h	01h	03h	table 32	
03h	02h	03h	table 32	
03h	03h	03h	table 32	

Peripheral device type:	00h	Direct-access device
	01h	Sequential-access device
	02h	Printer device
	03h	Processor device
	04h	Write-once read-multiple device
	05h	Read-only direct-access device
COPY function code:	00h	Direct access to sequential access
	01h	Sequential access to direct access
	02h	Direct access to direct access
	03h	Sequential access to sequential access
	04h – 0Fh	Reserved
	10h – 1Fh	Vendor unique

7.1.4.2 Errors Detected by a Target

The second class of errors consists of unusual conditions detected by the SCSI device transferring data at the request of the SCSI device managing the transfer. The SCSI device managing the COPY command detects unusual conditions by receiving a CHECK CONDITION status from one of the SCSI devices it is managing. It then shall recover the sense data associated with the unusual condition.

The SCSI device managing the COPY command may also be the source or destination SCSI device (or both). It shall distinguish between a failure of the management of the COPY and a failure of the data transfer being requested. It shall then create the appropriate sense data internally.

After recovering the sense data associated with the detected error, the SCSI device managing the COPY command shall

a) terminate the COPY command with a CHECK CONDITION status,

b) return the sense data in the extended sense format. The valid bit shall be set to one. The segment number shall contain the number of the segment descriptor being processed at the time the unusual condition is detected. The sense key shall be set to COPY ABORTED. The information bytes shall contain the difference between the number of blocks field in the segment descriptor being processed at the time of the failure and the number of blocks successfully copied. This number is the residue of unprocessed blocks remaining for the segment descriptor. The additional sense length shall specify the number of additional sense bytes.

The first additional sense byte shall specify the byte number, relative to the first byte of sense data of the beginning of the source logical unit's status byte and sense data. A zero value indicates that no status byte or sense data is being returned for the source logical unit. The first byte of the area pointed to by the first additional sense byte shall contain the status byte from the source logical unit. The subsequent bytes shall contain, unchanged, the sense data recovered from the source logical unit.

The second additional sense byte shall specify the byte number, relative to the first byte of sense data of the beginning of the destination logical unit's status byte and sense data. A zero value indicates that no status byte or sense data is being returned for the destination logical unit. The first byte of the area pointed to by the second additional sense byte shall contain the status byte from the destination logical unit. The subsequent bytes shall contain, unchanged, the sense data recovered from the destination logical unit.

7.1.4.3 COPY Function Code 00h and 01h

The format for the segment descriptors for COPY transfers between direct-access and sequential-access devices is specified in table 30. This format is required for COPY function codes 00h or 01h. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Source address and destination address fields specify the SCSI devices and the source LUN and destination LUN fields specify the logical units to use for this segment of the COPY command. Some SCSI devices may not support "third-party" COPY in which the copying SCSI device is not the source or destination device. Some SCSI devices only support COPY within the SCSI device and not to other SCSI devices. If an unsupported COPY operation is requested, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

The sequential-access device block-length field specifies the block-length to be used on the sequential-access logical unit during this segment of the COPY command. If this block-length is known by the SCSI device managing the COPY to be not supported, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. If the block-length is found to be invalid while executing a read or write operation to the sequential-access device, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to COPY ABORTED.

The direct-access device number of blocks field specifies the number of blocks in the current segment. A value of zero indicates that no blocks shall be transferred in this segment. The direct-access device logical block address specifies the starting logical block address on the logical unit for this segment.

Table 30 – Segment Descriptor for Codes 00h and 01h

Byte \ Bit	7	6	5	4	3	2	1	0
0	Source Address			Reserved		Source LUN		
1	Destination Address			Reserved		Destination LUN		
2	Sequential-Access Device Block-Length (MSB)							
3	Sequential-Access Device Block-Length (LSB)							
4	Direct-Access Device Number of Blocks (MSB)							
5	Direct-Access Device Number of Blocks							
6	Direct-Access Device Number of Blocks							
7	Direct-Access Device Number of Blocks (LSB)							
8	Direct-Access Device Logical Block Address (MSB)							
9	Direct-Access Device Logical Block Address							
10	Direct-Access Device Logical Block Address							
11	Direct-Access Device Logical Block Address (LSB)							

7.1.4.4 COPY Function Code 02h

The format for the segment descriptors for COPY transfers among direct-access devices is specified by table 31. This format is required for COPY function code 02h. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

The source address and destination address fields specify the SCSI devices and the source LUN and destination LUN specify the logical units to use for this segment of the COPY command. Some SCSI devices may not support "third-party" COPY in which the copying SCSI device is not the source or destination device. Some SCSI devices only support COPY within the SCSI device and not to other SCSI devices. If an unsupported COPY operation is requested, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

The source number of blocks field specifies the number of blocks to be transferred from the source device during command execution. The source logical block address field specifies the starting logical block address on the source device. The destination logical block address field specifies the starting logical block address on the destination device.

Table 31 – Segment Descriptor for COPY Function Code 02h

Byte \ Bit	7	6	5	4	3	2	1	0
0	Source Address			Reserved		Source LUN		
1	Destination Address			Reserved		Destination LUN		
2	Reserved							
3	Reserved							
4	Source Number of Blocks (MSB)							
5	Source Number of Blocks							
6	Source Number of Blocks							
7	Source Number of Blocks (LSB)							
8	Source Logical Block Address (MSB)							
9	Source Logical Block Address							
10	Source Logical Block Address							
11	Source Logical Block Address (LSB)							
12	Destination Logical Block Address (MSB)							
13	Destination Logical Block Address							
14	Destination Logical Block Address							
15	Destination Logical Block Address (LSB)							

7.1.4.5 COPY Function Code 03h

The format for the segment descriptors for COPY transfers among sequential-access devices is specified by table 32. This format is required for COPY function code 03h. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Table 32 – Segment Descriptor for COPY Function Code 03h

Byte \ Bit	7	6	5	4	3	2	1	0
0	Source Address			Reserved		Source LUN		
1	Destination Address			Reserved		Destination LUN		
2	Reserved							
3	Reserved							
4	Source Block Length (MSB)							
5	Source Block Length (LSB)							
6	Destination Block Length (MSB)							
7	Destination Block Length (LSB)							
8	Source Number of Blocks (MSB)							
9	Source Number of Blocks							
10	Source Number of Blocks							
11	Source Number of Blocks (LSB)							

Source address and destination address fields specify the SCSI devices and the source LUN and destination LUN fields specify the logical units to use for this segment of the COPY command. Some SCSI devices may not support “third-party” COPY in which the copying SCSI device is not the source or destination device. Some SCSI devices only support COPY within the SCSI device and not to other SCSI devices. If an unsupported COPY operation is requested, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

The source block-length field specifies the block-length of the source device for this segment of the COPY. A zero in this field indicates variable block-length. For nonzero values, this field shall match the logical unit's actual block-length. If block-length mismatches are detected by the SCSI device managing the COPY, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. If the mismatches are detected during the read operation by the COPY manager, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to COPY ABORTED.

The destination block-length field specifies the block-length to be used on the destination logical unit during the COPY. Destination block-length mismatches are handled in the same manner as source block-length mismatches.

The source number of blocks field specifies the number of blocks to be transferred from the source device during this segment. A value of zero indicates that no blocks shall be transferred.

7.1.5 RECEIVE DIAGNOSTIC RESULTS Command

Peripheral Device Type: All
Operation Code Type: Optional
Operation Code: 1Ch

Table 33 – RECEIVE DIAGNOSTIC RESULTS Command

Byte Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Allocation Length (MSB)							
4	Allocation Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RECEIVE DIAGNOSTIC RESULTS command (table 33) requests analysis data be sent to the initiator after completion of a SEND DIAGNOSTIC command (see 7.1.6).

The allocation length shall specify the number of bytes that the initiator has allocated for returned diagnostic data. An allocation length of zero indicates that no diagnostic data shall be transferred. Any other value indicates the maximum number of bytes that shall be transferred. The target terminates the DATA IN phase when allocation length bytes have been transferred or when all available diagnostic data have been transferred to the initiator, whichever is less.

The diagnostic data returned is vendor unique.

NOTE: Although diagnostic software is generally device-specific, this command and the SEND DIAGNOSTIC command provide a means to isolate the operating system software from the device-specific diagnostic software. Hence the operating system can remain device-independent. This also allows diagnostic software to be more easily ported to other operating systems.

7.1.6 SEND DIAGNOSTIC Command

Peripheral Device Type: All
Operation Code Type: Optional
Operation Code: 1Dh

Table 34 – SEND DIAGNOSTIC Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved		SelfTest	DevOfL	UnitOfL
2	Reserved							
3	Parameter List Length (MSB)							
4	Parameter List Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The SEND DIAGNOSTIC command (table 34) requests the target to perform diagnostic tests on itself, on the attached peripheral devices, or on both. This command is usually followed by a RECEIVE DIAGNOSTIC RESULTS command, except when the self test (SelfTest) bit is one.

The parameter list length specifies the length in bytes of the parameter list that shall be transferred during the DATA OUT phase. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. The parameter list is vendor unique.

A logical unit off-line (UnitOfL) bit of one enables write operations on user medium or operations that affect user visible medium positioning. An SCSI device off-line (DevOfL) bit of one enables diagnostic operations that may adversely affect operations to other logical units on the same target.

The logical unit off-line and SCSI device off-line bits are generally set by operating system software, while the parameter list is prepared by diagnostic application software. Thus, by preventing operations that are not enabled by these bits, the target assists the operating system in protecting its resources.

A self test bit of one directs the target to complete its default self test. If the self test is requested, the parameter list length shall be set to zero and no data shall be transferred. If the self test successfully passes, the command shall be terminated with a GOOD status; otherwise, the command shall be terminated with a CHECK CONDITION status and, if extended sense is implemented, the sense key shall be set to HARDWARE ERROR.

NOTE: See the note to 7.1.5.

7.2 Group 1 Commands for All Device Types

These commands shall be as listed in table 35.

Table 35 – Group 1 Commands for All Device Types

Operation Code	Type	Command Name	Section	Page
20h	V			
21h	V			
22h	V			
23h	V			
24h	V			
25h	*			
26h	V			
27h	V			
28h	*			
29h	V			
2Ah	*			
2Bh	*			
2Ch	V			
2Dh	V			
2Eh	*			
2Fh	*			
30h	*			
31h	*			
32h	*			
33h	*			
34h	R			
35h	R			
36h	R			
37h	R			
38h	R			
39h	O	COMPARE	7.2.1	60
3Ah	O	COPY AND VERIFY	7.2.2	61
3Bh	R			
3Ch	R			
3Dh	R			
3Eh	R			
3Fh	R			

Key:

O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.
* = These operation codes may have different meanings for specific types of peripheral devices. (See the appropriate section for further information.)

7.2.1 COMPARE Command

Peripheral Device Type: AI
Operation Code Type: Optional
Operation Code: 39h

Table 36 – COMPARE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Parameter List Length (MSB)							
4	Parameter List Length							
5	Parameter List Length (LSB)							
6	Reserved							
7	Reserved							
8	Reserved							
9	Vendor Unique		Reserved				Flag	Link

The COMPARE command (table 36) provides the means to compare data from one logical unit with another or the same logical unit in a manner similar to the COPY command.

This command functions in the same manner as the COPY command, except that the data from the source is compared on a byte-by-byte basis with the data from the destination. The parameter list transferred to the target is the same as for the COPY command. This parameter list contains the information to identify the logical units involved in the comparison and the length of the comparison. (See 7.1.4 for additional information about the COPY command.)

If the comparison is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE. The remaining fields in the extended sense shall be set as documented in the COPY command.

7.2.2 COPY AND VERIFY Command

Peripheral Device Type: All
Operation Code Type: Optional
Operation Code: 3Ah

Table 37 – COPY AND VERIFY Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			BytChk	Reserved
2	Reserved							
3	Parameter List Length (MSB)							
4	Parameter List Length							
5	Parameter List Length (LSB)							
6	Reserved							
7	Reserved							
8	Reserved							
9	Vendor Unique		Reserved				Flag	Link

The COPY AND VERIFY command (table 37) performs the same function as the COPY command, except that a verification of the data written to the destination logical unit is performed after the data is written. The parameter list transferred to the target is the same as for the COPY command. This parameter list contains the information to identify the logical units involved in the copy and the length of the copy. (See 7.1.4 for additional information about the COPY command.)

A byte check (BytChk) bit of zero causes the verification to be simply a medium verification (CRC, ECC, etc). A BytChk bit of one causes a byte-by- byte comparison of data written to the destination logical unit and the data read from the source logical unit.

If the comparison is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE. The remaining fields in the extended sense shall be set as documented in the COPY command.

7.3 Group 2 Commands for All Device Types

The Group 2 commands (operation codes 40h through 5Fh) are all reserved for future standardization.

7.4 Group 3 Commands for All Device Types

The Group 3 commands (operation codes 60h through 7Fh) are all reserved for future standardization.

7.5 Group 4 Commands for All Device Types

The Group 4 commands (operation codes 80h through 9Fh) are all reserved for future standardization.

7.6 Group 5 Commands for All Device Types

The Group 5 commands with operation codes A0h through AFh are vendor unique. The Group 5 commands with operation codes B0h through BFh are reserved for future standardization.

7.7 Group 6 Commands for All Device Types

The Group 6 commands (operation codes C0h through DFh) are all vendor unique.

7.8 Group 7 Commands for All Device Types

The Group 7 commands (operation codes E0h through FFh) are all vendor unique.

8 Command descriptions for direct-access devices

8.1 Group 0 Commands for Direct-Access Devices

The Group 0 commands for direct-access devices shall be as shown in table 38.

Table 38 – Group 0 Commands for Direct-Access Commands

Operation Code	Type	Command Name	Subclause	Page
00h	O	TEST UNIT READY	7.1.1	44
01h	O	REZERO UNIT	8.1.1	64
02h	V			
03h	M	REQUEST SENSE	7.1.2	44
04h	M	FORMAT UNIT	8.1.2	64
05h	V			
06h	V			
07h	O	REASSIGN BLOCKS	8.1.3	69
08h	M	READ	8.1.4	71
09h	V			
0Ah	M	WRITE	8.1.5	73
0Bh	O	SEEK	8.1.6	74
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	V			
10h	V			
11h	V			
12h	E	INQUIRY	7.1.3	48
13h	V			
14h	V			
15h	O	MODE SELECT	8.1.7	74
16h	O	RESERVE	8.1.8	77
17h	O	RELEASE	8.1.9	81
18h	O	COPY	7.1.4	50
19h	V			
1Ah	O	MODE SENSE	8.1.10	82
1Bh	O	START/STOP UNIT	8.1.11	84
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	57
1Dh	O	SEND DIAGNOSTIC	7.1.6	58
1Eh	O	PREVENT/ALLOW MEDIUM REMOVAL	8.1.12	85
1Fh	R			

Key:

M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

8.1.1 REZERO UNIT Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: 01h

Table 39 – REZERO UNIT Comand

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The REZERO UNIT command (table 39) requests that the target set the logical unit to a specific state. See vendor specifications for details.

8.1.2 FORMAT UNIT Command

Peripheral Device Type: Direct Access
Operation Code Type: Mandatory
Operation Code: 04h

Table 40 – FORMAT UNIT Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			FmtData	CmpLst	Defect List Format		
2	Vendor Unique							
3	Interleave (MSB)							
4	Interleave (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The FORMAT UNIT command (table 40) ensures that the medium is formatted so that all data blocks can be accessed. There is no guarantee that the medium has or has not been altered. In addition, the medium may be certified and control structures be created for the management of the medium and defects.

The FORMAT UNIT command is both a mandatory command and an extended command with multiple options (see table 41).

Table 41 – FORMAT UNIT Command Variations

Bit Reference						
4	3	2	1	0		
FmtData	CmpLst	Defect List Format			Command Type	Comments
0	X	X	X	X	Mandatory	Format with no defect data sent from the initiator to the target.
1	0	0	X	X	Extended	Format adding the defects specified in the defect list to the known defects. (See Table 41.)
1	1	0	X	X	Optional	Format using defects in the defect list as the full set of known defects. (See Table 41.)
1	0	1	0	0	Optional	Format adding the defects in the defect list to the known defects. (See Table 42.)
1	1	1	0	0	Optional	Format using the defects in the defect list as the full set of known defects. (See Table 42.)
1	0	1	0	1	Optional	Format adding the defects in the defect list to the known defects. (See Table 43.)
1	1	1	0	1	Optional	Format using the defects in the defect list as the full set of known defects. (See Table 43.)
1	X	1	1	0	Vendor unique	
1	0	1	1	1	Reserved	
1	1	1	1	1	Reserved	

X = 1 or 0 (i.e., don't care term).

The FORMAT UNIT command shall be rejected with RESERVATION CONFLICT status if any extent (see 8.1.8.2) in the specified logical unit is reserved.

A format data (FmtData) bit of one indicates that format data is supplied during the DATA OUT phase. The defect list included with this data specifies the defects that shall be entered into the defect map. The format of the defect list is determined by defect list format. A FmtData bit of zero indicates that the DATA OUT phase shall not occur (no defect data shall be supplied by the initiator).

A complete list (CmpLst) bit of one indicates the data supplied is the complete list of known defects. Any previous initiator-specified defect map or defect data shall be erased. The target may add to this list as it formats the medium. The result is to purge any previous initiator-specified defect list and to build a new defect list. A CmpLst bit of zero indicates that the data supplied is in addition to existing defect data using the current format. When using the block format, the defect list refers to the current block length (and not to the new block length, if it is different) and the defect list refers to current logical block addresses (not physical addresses). The target may add to this list as it formats the medium.

The defect list format field specifies additional information related to the defect list. (See table 41 for further information.)

The interleave field requests that the logical blocks be related in a specific fashion to the physical blocks to facilitate speed matching. An interleave value of zero requests that the target use its default interleave. An interleave value of one requests that consecutive logical blocks be placed in consecutive physical order. Values of two or greater are vendor unique.

The defect lists shown in tables 42, 43, and 44 contain a four-byte header followed by one or more defect descriptors. The length of the defect descriptors vary with the format of the defect list.

Table 42 – Defect List - Block Format

Byte	Defect List Header
0	Reserved
1	Reserved
2	Defect List Length (MSB)
3	Defect List Length (LSB)
	Defect Descriptor(s)
0	Defect Block Address (MSB)
1	Defect Block Address.
2	Defect Block Address
3	Defect Block Address (LSB)

The defect list length in each table specifies the total length in bytes of the defect descriptors that follow. In table 42, the defect list length is equal to four times the number of defect descriptors. In tables 43 and 44, the defect list length is equal to eight times the number of defect descriptors.

Each defect descriptor for the block format specifies a four-byte defect block address that contains the defect. The defect descriptors shall be in ascending order.

Table 43 – Bytes from Index Format

Byte	Defect List Header
0	Reserved
1	Reserved
2	Defect List Length (MSB)
3	Defect List Length (LSB)
	Defect Descriptor(s)
0	Cylinder Number of Defect (MSB)
1	Cylinder Number of Defect.
2	Cylinder Number of Defect (LSB)
3	Head Number of Defect
4	Defect Bytes from Index (MSB)
5	Defect Bytes from Index
6	Defect Bytes from Index
7	Defect Bytes from Index (LSB)

Each defect descriptor for the bytes from index format specifies the beginning of an eight-byte defect location on the medium. Each defect descriptor is comprised of the cylinder number of defect, the head number of defect, and the defect bytes from index. The defect descriptors shall be in ascending order. For determining ascending order, the cylinder number of defect is considered the most significant part of the address and the defect bytes from index is considered the least significant part of the address.

A defect bytes from index of FFFFFFFFh indicates that the entire track shall be reassigned.

Table 44 – Defect List - Physical Sector Format

Byte	Defect List Header
0	Reserved
1	Reserved
2	Defect List Length (MSB)
3	Defect List Length (LSB)
	Defect Descriptor(s)
0	Cylinder Number of Defect (MSB)
1	Cylinder Number of Defect.
2	Cylinder Number of Defect (LSB)
3	Head Number of Defect
4	Defect Sector Number (MSB)
5	Defect Sector Number
6	Defect Sector Number
7	Defect Sector Number (LSB)

Each defect descriptor for the physical sector format specifies a sector- size defect location comprised of the cylinder number of defect, the head number of defect, and the defect sector number. The defect descriptors shall be in ascending order. For determining ascending order, the cylinder number of defect is considered the most significant part of the address and the defect sector number is considered the least significant part of the address.

A defect sector number of FFFFFFFFh indicates that the entire track shall be reassigned.

8.1.3 REASSIGN BLOCKS Command

Peripheral Device Type: Direct Access and Write-Once Read-Multiple
Operation Code Type: Optional
Operation Code: 07h

Table 45 – REASSIGN BLOCKS Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The REASSIGN BLOCKS command (table 45) requests the target to reassign the defective logical blocks to an area on the logical unit reserved for this purpose.

The initiator transfers a defect list that contains the logical block addresses to be reassigned. The target shall reassign the physical medium used for each logical block address in the list. The data contained in the logical blocks specified in the defect list may be altered, but the data in all other logical blocks on the medium shall be preserved.

The effect of specifying a logical block to be reassigned that previously has been reassigned is to reassign the block again. Thus, over the life of the medium, a logical block can be assign to multiple physical addresses (until no more spare locations remain on the medium).

The REASSIGN BLOCKS defect list (table 46) contains a four-byte header followed by one or more defect descriptors. The length of each defect descriptor is four bytes.

The defect list length specifies the total length in bytes of the defect descriptors that follow. The defect list length is equal to four times the number of defect descriptors.

Table 46 – REASSIGN BLOCKS Defect List

Byte	Defect List Header
0	Reserved
1	Reserved
2	Defect List Length (MSB)
3	Defect List Length (LSB)
	Defect Descriptor(s)
0	Defect Logical Block Address (MSB)
1	Defect Logical Block Address.
2	Defect Logical Block Address
3	Defect Logical Block Address (MSB)

The defect descriptor specifies a four-byte defect logical block address that contains the defect. The defect descriptors shall be in ascending order.

If the logical unit has insufficient capacity to reassign all of the defective logical blocks, the command shall terminate with a CHECK CONDITION status and the sense key shall be set to MEDIUM ERROR. The logical block address of the first logical block not reassigned shall be returned in the information bytes of the sense data.

8.1.4 READ Command

Peripheral Device Type: Direct Access
Operation Code Type: Mandatory
Operation Code: 08h

Table 47 – READ Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	Vendor Unique		Reserved				Flag	Link

The READ command (table 47) requests that the target transfer data to the initiator.

The logical block address specifies the logical block at which the read operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data to transferred. A transfer length of zero indicates that 256 logical blocks shall be transferred. Any other value indicates the number of logical blocks that shall be transferred.

The most recent data value written in the addressed logical block shall be returned.

This command shall be terminated with a RESERVATION CONFLICT status if any reservation access conflict (see 8.1.8) exists and no data shall be read.

If any of the conditions listed in table 48 occur, this command shall be terminated with a CHECK CONDITION status, and if extended sense is implemented, the sense key shall be set as indicated in the following table. Table 48 does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Table 48 — READ Command Check Condition Status

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST
Target reset or medium change since last command from this initiator	UNIT ATTENTION
Unrecoverable read error	MEDIUM ERROR
Recovered read error	RECOVERED ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND

When an invalid logical block address causes ILLEGAL REQUEST to be set, the extended sense information bytes shall be set to the logical block address of the first invalid address.

8.1.5 WRITE Command

Peripheral Device Type: Direct Access
Operation Code Type: Mandatory
Operation Code: 0Ah

Table 49 – WRITE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	Vendor Unique		Reserved				Flag	Link

The WRITE command (table 49) requests that the target write the data transferred by the initiator to the medium.

The logical block address specifies the logical block at which the write operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data to transferred. A transfer length of zero indicates that 256 logical blocks shall be transferred. Any other value indicates the number of logical blocks that shall be transferred.

This command shall be terminated with a RESERVATION CONFLICT status if any reservation access conflict (see 8.1.8) exists and no data shall be written.

If any of the conditions listed in table 50 occur, this command shall be terminated with a CHECK CONDITION status, and if extended sense is implemented, the sense key shall be set as indicated in the table. Table 50 does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Table 50 — Write Command Check Condition Status

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST
Target reset or medium change since last command from this initiator	UNIT ATTENTION
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND

When an invalid logical block address causes ILLEGAL REQUEST to be set, the extended sense information bytes shall be set to the logical block address of the first invalid address. In this case, no data shall be written on the logical unit.

8.1.6 SEEK Command

Peripheral Device Type: Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 0Bh

Table 51 – SEEK Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The SEEK command (table 51) requests that the logical unit seek to the specified logical block address.

8.1.7 MODE SELECT Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: 15h

Table 52 – MODE SELECT Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Parameter List Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SELECT command (table 52) provides a means for the initiator to specify medium, logical unit, or peripheral device parameters to the target.

The parameter list length specifies the length in bytes of the MODE SELECT parameter list that shall be transferred during the DATA OUT phase. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

The MODE SELECT parameter list (table 53) contains a four-byte header, followed by zero or more block descriptors, followed by the vendor unique parameters, if any.

Table 53 – MODE SELECT Parameter List

Byte	MODE SELECT Header
0	Reserved
1	Medium Type
2	Reserved
3	Block Descriptor Length
	Block Descriptor(s)
0	Density Code
1	Number of Blocks (MSB)
2	Number of Blocks
3	Number of Blocks (LSB)
4	Reserved
5	Block Length (MSB)
6	Block Length
7	Block Length (LSB)
	Vendor Unique Parameter(s)
0 to n	Vendor Unique Parameter Byte(s)

Table 54 – Direct-Access Medium-Type Codes

Code Value	Medium Type				
00h 01h 02h	Default medium type (currently mounted medium type) Flexible disk, single-sided; unspecified medium Flexible disk, double-sided; unspecified medium				
	Flexible Disks				
	Diameter mm (inch)	Bit Density Bits/Radian	Track Density /mm (/inch)	Number Of Sides	Reference Standard
05h	200 (8,0)	6 631	1,9 (48)	1	ISO 5654/1:1984, ISO 5654/2:1985
06h	200 (8,0)	6 631	1,9 (48)	2	See Note
09h	200 (8,0)	13 262	1,9 (48)	1	None
0Ah	200 (8,0)	13 262	1,9 (48)	2	ISO 7065/1:1985, ISO 7065/2:1985
0Dh	130 (5,25)	3 979	1,9 (48	1	ISO 6596/1:1985, ISO 6596/2:985
12h	130 (5,25)	7 958	1,9 (48)	2	ISO 7487/1:1985, ISO 7487/2:1985, ISO 7487/3:1984
16h	130 (5,25)	7 958	3,8 (96)	2	ISO 8378/1:1986, ISO 8378/2:1986, ISO 8378/3:1986
1Ah	130 (5,25)	13 262	3,8 (96)	2	ISO 8630/1:1987, ISO 8630/2:1987, ISO 8630/3:1987
1Eh	90 (3,5)	7 958	5,3 (135)	2	ISO 8860/1:1987, ISO 8860/2:1987
	Direct-Access Magnetic Tapes				
Code Value	Width mm (inch)	Tracks	Density ftpmm (ftpi)		
40h	6,3 (0,25)	12	394 (10 000)		
44h	6,3 (0,25)	24	394 (10 000)		
80h – FFh	Vendor unique				
All others	Reserved				
NOTE See Annex F for additional standards that may be applicable.					

Code values for the medium type field are defined in table 54.

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors shall be included in the parameter list. This condition shall not be considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a density code, a number of blocks, and a block length.

Code values for the density code field shall be assigned as follows:

00h	Default (only one density supported)
01h	Flexible disk, single density
02h	Flexible disk, double density
02h – 7Fh	Reserved
80h – FFh	Vendor unique

The number of blocks field specifies the number of logical blocks on the medium that meet the density code and block length in the block descriptor. A number of blocks of zero indicates that all of the remaining logical blocks of the logical unit shall have the medium characteristics specified by the block descriptor.

The block length specifies the length in bytes of each logical block described by the block descriptor.

8.1.8 RESERVE Command

Peripheral Device Type:	Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type:	Optional
Operation Code:	16h

Table 55 – RESERVE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			3rdPty	Third Party Device ID			Extent
2	Reservation Identification							
3	Extent List Length (MSB)							
4	Extent List length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RESERVE command (table 55) is used to reserve logical units or, if the extent reservation option is implemented, extents within logical units for the use of the initiator. If third-party reservation option is implemented, the logical units or extents may be reserved for another specified SCSI device. The RESERVE and RELEASE commands provide the basic mechanism for contention resolution in multiple-initiator systems.

8.1.8.1 Logical Unit Reservation.

If the extent bit is zero, this command shall request that the entire logical unit be reserved for the exclusive use of the initiator until the reservation is superseded by another valid RESERVE command from the initiator that made the reservation or until released by a RELEASE command from the same initiator, by a BUS DEVICE RESET message from any initiator, or by a "hard" RESET condition. A logical unit reservation shall not be granted if the logical unit or any extent is reserved by another initiator. It shall be permissible for an initiator to reserve a logical unit that is currently reserved by that initiator. If the extent bit is zero, the reservation identification and the extent list length shall be ignored.

If the logical unit, or any extent within the logical unit is reserved for another initiator, the target shall respond by returning RESERVATION CONFLICT status.

If, after honoring the reservation, any other initiator then subsequently attempts to perform any command on the reserved logical unit other than a RELEASE command, which shall be ignored, then the command shall be rejected with RESERVATION CONFLICT status.

8.1.8.2 Extent Reservation (Optional).
The reservation identification provides a means for an initiator to identify each extent reservation. This allows an initiator in a multiple tasking environment, to have multiple reservations outstanding. The reservation identification is used in the RELEASE command to specify which reservation is to be released. It is also used in superseding RESERVE commands to specify which reservation is to be superseded.

If the extent reservation option is implemented, then the extent release option (see 8.1.9.2) shall also be implemented. These options permit multiple extents within the logical unit to be reserved, each with a separate reservation type.

If the extent bit is one, and the extent reservation option is implemented, then the target shall process the reservation request as follows:

a) The extent list shall be checked for the number of extents in the reservation request. If the extent list length is zero, no reservations shall be changed and this condition shall not be treated as an error. If the extent list contains more extents than are supported on the logical unit, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. If the extent list contains more extents than are currently available on the logical unit, then the target shall return RESERVATION CONFLICT status.

b) The extent list shall be checked for valid extent logical block addresses. If any logical block address is invalid for this logical unit, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. The extent list shall be checked for invalid extent overlaps (as defined by reservation type) with other extent descriptors in the extent list and if invalid overlaps are found, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

c) If the requested reservation does not conflict with any active or previously requested reservation, the extents specified shall be reserved until superseded by another valid RESERVE command from the initiator that made the reservation or until released by a RELEASE command from the same initiator, by a BUS DEVICE RESET message from any initiator, or by a "hard" RESET condition. If either of the last two conditions occur, the next command (except for INQUIRY or REQUEST commands) from each initiator shall be terminated with a CHECK CONDITION status and the sense key shall be set to UNIT ATTENTION.

d) If the reservation request conflicts with a reservation already active, then the target shall return RESERVATION CONFLICT status.

If the extent bit is one, and the extent reservation option is not implemented, then the RESERVE command shall be rejected with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

Table 56 – Data Format of Extent Descriptors

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved					RelAdr	Reservation Type	
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Logical Block Address (MSB)							
5	Logical Block Address							
6	Logical Block Address							
7	Logical Block Address (LSB)							

The size of the extent list shall be defined by the extent list length parameter. The extent list shall consist of zero or more descriptors as shown in Table 55. Each extent descriptor defines an extent beginning at the specified logical block address for the specified number of blocks. If the number of blocks is zero, the extent shall begin at the specified logical block address and continue through the last logical block address on the logical unit.

The reservation type field shall determine the type of reservation to be effected for each extent. Four types of reservations are possible as follows:

DB(1)	DB(0)	Reservation Type
1	0	Read Exclusive
0	1	Write Exclusive
1	1	Exclusive Access
0	0	Read Shared

Read Exclusive: While this reservation is active, no other initiator shall be permitted read operations to the indicated extent. This reservation shall not inhibit write operations from any initiator or conflict with a write exclusive reservation; however, read exclusive, exclusive access, and read shared reservations that overlap this extent shall conflict with this reservation.

Write Exclusive: While this reservation is active, no other initiator shall be permitted write operations to the indicated extent. This reservation shall not inhibit read operations from any initiator or conflict with a read exclusive reservation from any initiator. This reservation shall conflict with write exclusive, exclusive access, and read shared reservations that overlap this extent.

Exclusive Access: While this reservation is active, no other initiator shall be permitted any access to the indicated extent. All reservation types that overlap this extent shall conflict with this reservation.

Read Shared: While this reservation is active, no write operations shall be permitted by any initiator to the indicated extent. This reservation shall not inhibit read operations from any initiator or conflict with a read shared reservation. Read exclusive, write exclusive, and exclusive access reservations that overlap with this extent shall conflict with this reservation.

If the relative address bit is one, the logical block address shall be treated as a two's complement displacement. This displacement shall be added to the logical block address last accessed on the logical unit to form the logical block address for this extent. This feature is only available when linking commands and requires that a previous command in the linked group has accessed a logical block on the logical unit; if not, the RESERVE command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

If an initiator attempts a command to a logical block that has been reserved and that access is prohibited by the reservation, the command shall not be performed and the command shall be terminated with a RESERVATION CONFLICT status. If a reservation conflict precludes any part of the command, none of the command shall be performed. COPY commands shall be terminated with a CHECK CONDITION status and the sense key shall be set to DATA PROTECT if any part of the copy operation is prohibited by an active reservation. If any extent in a logical unit is reserved in any way, a FORMAT UNIT command shall be rejected with a RESERVATION CONFLICT status.

8.1.8.3 Third Party Reservation (Optional)

The third-party reservation option for the RESERVE command allows an initiator to reserve a logical unit or extents within a logical unit for another SCSI device. This option is intended for use in multiple-initiator systems that use the COPY command. Any target that implements the third-party reservation option shall also implement the third-party release option (see 8.1.9.3).

If the third-party (3rdPty) bit is zero, then the third-party reservation option is not requested. If the 3rdPty bit is one and the third-party reservation option is implemented, then the RESERVE command shall reserve the specified logical unit or extents for the SCSI device specified in the third- party device ID field. The target shall preserve the reservation until it is superseded by another valid RESERVE command from the initiator that made the reservation or until it is released by the same initiator, by a BUS DEVICE RESET message from any initiator, or a “hard” RESET condition. The target shall ignore any attempt to release the reservation made by any other initiator.

If the 3rdPty bit is one and the third-party reservation option is not implemented, then the target shall reject the RESERVE command with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

8.1.8.4 Superseding Reservations

An initiator that holds a current reservation may modify that reservation by issuing another RESERVE command to the same logical unit and, if the extent bit is one, using the same reservation identification. The superseding RESERVE command shall release the previous reservation state when the new reservation request is granted. The previous reservation shall not be modified if the new reservation request cannot be granted. If the superseding reservation cannot be granted because of conflicts with a previous active reservation (other than the reservation being superseded), then the target shall return RESERVATION CONFLICT status.

NOTE: Superseding reservations are principally intended to allow the SCSI device ID to be changed on a reservation using the third-party reservation option. This capability is necessary for certain situations when using the COPY command.

8.1.9 RELEASE Command

Peripheral Device Type: Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 17h

Table 57 – RELEASE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			3rdPty	Third Party Device ID			Extent
2	Reservation Identification							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The RELEASE command (table 57) is used to release previously reserved logical units, or, if the extent release option is implemented, previously reserved extents within logical units. It is not an error for an initiator to attempt to release a reservation that is not currently active. In this case, the target returns GOOD status without altering any other reservation.

8.1.9.1 Logical Unit Release

If the extent bit is zero, this command shall cause the target to terminate all logical unit and extent reservations that are active from the initiator to the specified logical unit.

8.1.9.2 Extent Release (Optional)

If the extent bit is one and the extent release option is not implemented, then the RELEASE command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. This option shall be implemented if the extent reservation option (see 8.1.8.2) is implemented.

If the extent bit is one and the extent release option is implemented, this command shall cause any reservation from the requesting initiator with a matching reservation identification to be terminated. Other reservations from the requesting initiator shall remain in effect.

8.1.9.3 Third Party Release (Optional)

The third-party release option for the RELEASE command allows an initiator to release a logical unit or extents within a logical unit that were previously reserved using the third- party reservation option (see 8.1.8.3). This option shall be implemented if the third-party reservation option is implemented. This option is intended for use in multiple-initiator systems that use the COPY command.

If the third-party (3rdPty) bit is zero, then the third-party release option is not requested. If the 3rdPty bit is one and the target implements the third-party release option, then the target shall release the specified logical unit or extents, but only if the reservation was made using the third- party reservation option by the initiator that is requesting the release for the same SCSI device as specified in the third-party device ID field.

If the 3rdPty bit is one and the target does not implement the third-party release option, then the target shall terminate the command with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

8.1.10 MODE SENSE Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: 1Ah

Table 58 – MODE SENSE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SENSE command (table 58) provides a means for a target to report its medium, logical unit, or peripheral device parameters to the initiator. It is a complementary command to the MODE SELECT command for support of medium that may contain multiple block lengths or densities.

The allocation length specifies the number of bytes that the initiator has allocated for returned MODE SENSE data. An allocation length of zero indicates that no MODE SENSE data shall be transferred. This condition shall not be considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available MODE SENSE data have been transferred to the initiator, whichever is less.

The MODE SENSE data (table 59) contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Table 59 – MODE SENSE Data

Byte \ Bit	7	6	5	4	3	2	1	0
0	Sense Data Length							
1	Medium Type							
2	WP	Reserved						
3	Block Descriptor Length							
	Block Descriptor(s)							
0	Density Code							
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Reserved							
5	Block Length (MSB)							
6	Block Length							
7	Block Length (LSB)							
	Vendor Unique Parameter(s)							
0 to n	Vendor Unique Parameter Byte(s)							

The sense data length specifies the length in bytes of the following MODE SENSE data that is available to be transferred during the DATA IN phase. The sense data length does not include itself.

Code values for the medium-type field are defined in table 54.

A write protected (WP) bit of zero indicates that the medium is write enabled. A WP bit of one indicates that the medium is write protected.

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors shall be included in the parameter list. This condition shall not be considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a density code, a number of blocks, and a block length.

Code values for the density code field shall be assigned as follows:

00h	Default (only one density supported)
01h	Flexible disk, single density
02h	Flexible disk, double density
02h – 7Fh	Reserved
80h – FFh	Vendor unique

The number of blocks field specifies the number of logical blocks of the medium that meets the density code and block length in the block descriptor. A number of blocks of zero indicates that all of the remaining logical blocks of the logical unit have the medium characteristics specified by the block descriptor.

The block length specifies the length in bytes of each logical block.

8.1.11 START/STOP UNIT Command

Peripheral Device Type: Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 1Bh

Table 60 – START/STOP UNIT Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Immed
2	Reserved							
3	Reserved							
4	Reserved							Start
5	Vendor Unique		Reserved				Flag	Link

The START/STOP UNIT command (table 60) requests that the target enable or disable the logical unit for further operations.

An immediate (Immed) bit of one indicates that status shall be returned as soon as the operation is initiated. An Immed bit of zero indicates that status shall be returned after the operation is completed.

A start bit of one requests the logical unit be made ready for use. A start bit of zero requests that the logical unit be stopped.

8.1.12 PREVENT/ALLOW MEDIUM REMOVAL Command

Peripheral Device Type: Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 1Eh

Table 61 – PREVENT/ALLOW MEDIUM REMOVAL Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							Prevent
5	Vendor Unique		Reserved				Flag	Link

The PREVENT/ALLOW MEDIUM REMOVAL command (table 61) requests that the target enable or disable the removal of the medium in the logical unit.

A prevent bit of one shall inhibit mechanisms that normally allow removal of the medium. A prevent bit of zero shall allow removal of the medium.

This prevention of medium removal condition shall terminate upon receipt of a PREVENT/ALLOW MEDIUM REMOVAL command with the prevent bit set to zero, or by the receipt of a BUS DEVICE RESET message from any initiator or by a "hard" RESET condition.

8.2 Group 1 Commands for Direct-Access Devices.
The Group 1 commands for direct-access devices shall be as shown in table 62.

Table 62 – Group 1 Commands for Direct-Access Devices

Operation Code	Type	Command Name	Section	Page
20h	V			
21h	V			
22h	V			
23h	V			
24h	V			
25h	E	READ CAPACITY	8.2.1	87
26h	V			
27h	V			
28h	E	READ	8.2.2	89
29h	V			
2Ah	E	WRITE	8.2.3	90
2Bh	O	SEEK	8.2.4	92
2Ch	V			
2Dh	V			
2Eh	O	WRITE AND VERIFY	8.2.5	93
2Fh	O	VERIFY	8.2.6	94
30h	O	SEARCH DATA HIGH	8.2.7.1	96
31h	O	SEARCH DATA EQUAL	8.2.7.2	96
32h	O	SEARCH DATA LOW	8.2.7.3	98
33h	O	SET LIMITS	8.2.8	98
34h	R			
35h	R			
36h	R			
37h	R			
38h	R			
39h	O	COMPARE	7.2.1	60
3Ah	O	COPY AND VERIFY	7.2.2	61
3Bh	R			
3Ch	R			
3Dh	R			
3Eh	R			
3Fh	R			

Key:

M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

8.2.1 READ CAPACITY Command

Peripheral Device Type: Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type: Extended
Operation Code: 25h

Table 63 – READ CAPACITY Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Reserved							
8	Vendor Unique		Reserved					PMI
9	Vendor Unique		Reserved				Flag	Link

The READ CAPACITY command (able 63) provides a means for the initiator to request information regarding the capacity of the logical unit.

A partial medium indicator (PMI) bit of zero indicates that the information returned in the READ CAPACITY data shall be the logical block address and block length (in bytes) of the last logical block of the logical unit. The logical block address in the command descriptor block shall be to set zero for this option.

A PMI bit of one indicates that the information returned shall be the logical block address and block length (in bytes) of the last logical block address after which a substantial delay in data transfer will be encountered. This logical block address shall be greater than or equal to the logical block address specified in the command descriptor block. (Implementors note: This function is intended to assist storage management software in determining whether there is sufficient space on the current track, cylinder, etc. to contain a frequently accessed data structure such as a file directory or file index.)

The eight bytes of READ CAPACITY data shown in table 64 shall be sent during the DATA IN phase of the command.

Table 64 – READ CAPACITY Data

Byte	Description
0	Logical Block Address (MSB)
1	Logical Block Address
2	Logical Block Address
3	Logical Block Address (LSB)
4	Block Length (MSB)
5	Block Length
6	Block Length
7	Block Length (LSB)

8.2.2 READ Command

Peripheral Device Type: Direct Access
Operation Code Type: Extended
Operation Code: 28h

Table 65 – READ Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The READ command (table 65) requests that the target transfer data to the initiator.

The logical block address specifies the logical block at which the read operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered as an error. Any other value indicates the number of logical blocks that shall be transferred.

The most recent data value written in the addressed logical block shall be returned.

This command shall be terminated with a RESERVATION CONFLICT status if any reservation access conflict (see 8.1.8) exists and no data shall be read.

If any of the conditions shown in table 66 occur, this command shall return a CHECK CONDITION status and the sense key shall be set as indicated. Table 66 does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Table 66 – READ Command Check Condition Status

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST
Target reset or medium change since last command from this initiator	UNIT ATTENTION
Unrecoverable read error	MEDIUM ERROR
Recovered read error	RECOVERED ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND

When an invalid logical block address causes ILLEGAL REQUEST to be set, the extended sense information bytes shall be set to the logical block address of the first invalid address.

8.2.3 WRITE Command

Peripheral Device Type: Direct Access
Operation Code Type: Extended
Operation Code: 2Ah

Table 67 – WRITE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The WRITE command (table 67) requests that the target write the data transferred by the initiator to the medium.

The logical block address specifies the logical block at which the write operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered as an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

This command shall be terminated with a RESERVATION CONFLICT status if any reservation access conflict (see 8.1.8) exists and no data shall be written.

If any of the conditions shown in table 68 occur, this command shall be terminated with a CHECK CONDITION status and the sense key shall be set as indicated in the following table. Table 68 does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Table 68 — Write Command Check Condition Status

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST
Target reset or medium change since last command from this initiator	UNIT ATTENTION
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND

When an invalid logical block address causes ILLEGAL REQUEST to be set, the extended sense information bytes shall be set to the logical block address of the first invalid address. In this case, no data shall be written on the logical unit.

8.2.4 SEEK Command

Peripheral Device Type: Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 2Bh

Table 69 – SEEK Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Reserved							
8	Reserved							
9	Vendor Unique		Reserved				Flag	Link

The SEEK command (table 69) requests that the logical unit seek to the specified logical block address.

8.2.5 WRITE AND VERIFY Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: 2Eh

Table 70 – WRITE AND VERIFY Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			BytChk	RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The WRITE AND VERIFY command (table 70) requests that the target write the data transferred from the initiator to the medium and then verify that the data is correctly written.

A byte check (BytChk) bit of zero causes the verification to be simply a medium verification (CRC, ECC, etc). A BytChk bit of one causes a byte-by-byte compare of data written on the peripheral device and the data transferred from the initiator. If the compare is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE.

The logical block address specifies the logical block at which the write operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered as an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

8.2.6 VERIFY Command

Peripheral Device Type: Direct Access
Operation Code Type: Optional
Operation Code: 2Fh

Table 71 – VERIFY Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			BytChk	RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Verification Length (MSB)							
8	Verification Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The VERIFY command (table 71) requests that the target verify the data written on the medium.

A byte check (BytChk) bit of zero causes the verification to be simply a medium verification (CRC, ECC, etc). A BytChk bit of one causes a byte-by- byte compare of data on the medium and the data transferred from the initiator. If the compare is unsuccessful, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE.

The logical block address specifies the logical block at which the verify operation shall begin.

The verification length specifies the number of contiguous logical blocks of data that shall be verified. A transfer length of zero indicates that no logical blocks shall be verified. This condition shall not be considered as an error. Any other value indicates the number of logical blocks that shall be verified.

8.2.7 SEARCH DATA Commands

Peripheral Device Type: Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 30h, 31h, or 32h

Table 72 – SEARCH DATA Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Invert	Reserved		SpnDat	RelAdr
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Transfer Length (MSB)							
8	Transfer Length (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The SEARCH DATA commands (table 72) search one or more logical blocks for equality or inequality to a data pattern. The concept of records within a logical block is used to allow multiple records within a logical block to be searched.

The invert bit determines whether the search condition is to be inverted. See 8.2.7.1 through 8.2.7.3 for a description of the search conditions for the individual SEARCH DATA commands.

A spanned data (SpnDat) bit of zero indicates that each record shall be wholly contained within a single block. Any space at the end of a block that is smaller than the record length is ignored by the SEARCH DATA commands. A SpnDat bit of one indicates that records span block boundaries. Thus, a record may start in one block and end in the next or a subsequent block.

A transfer length of zero indicates that no data shall be searched. This condition shall be treated the same as an unsatisfied search.

A link bit of zero indicates a nonlinked command and if the search is satisfied, the command shall be terminated with a CONDITION MET status. A REQUEST SENSE command can then be issued to determine the logical block address and record offset of the matching record. If the search is not satisfied and no error occurs, the command shall be terminated with GOOD status.

A link bit of one indicates a command is linked to the SEARCH DATA command and if the search is satisfied, CONDITION MET status is returned and the next command is executed. If the RelAdr bit in the next command is one, the logical block address of the next command is used as a displacement from the logical block address at which the search was satisfied. If a linked search is not satisfied, the command is terminated with a CHECK CONDITION status. A REQUEST SENSE command may then be issued.

A REQUEST SENSE command following a satisfied SEARCH DATA command shall

a) return a sense key of EQUAL if the search was satisfied by an exact match. If the search was satisfied by an inequality then a sense key of NO SENSE shall be returned,

b) return the valid bit set to one,

c) return the logical block address of the logical block containing the first matching record in the information bytes

d) return the record offset of the matching record in the first four bytes of additional sense bytes.

A REQUEST SENSE command following an unsatisfied SEARCH DATA command shall

a) return a sense key of NO SENSE, if no errors occurred during the command execution,

b) return the valid bit set to zero.

The SEARCH DATA parameter list (table 73) contains a fourteen-byte header, followed by one or more search argument descriptors.

The logical record length field specifies the record length in bytes.

The first record offset field specifies the number of bytes that shall be ignored in the first logical block before the search begins. The value in the first record offset field shall not exceed the length of the logical block. Subsequent logical blocks shall be searched beginning with the first byte in the logical block. This permits one or more records to be skipped initially.

The number of records field specifies the maximum number of records that shall be searched by this command. An unsatisfied search shall terminate when the number of records or the number of blocks (from the command descriptor block) have been exhausted.

The search argument length specifies the length in bytes of all the search argument descriptors that follow. Since the pattern length can vary, there is no fixed multiple of the search argument descriptor to determine the search argument length.

The search argument descriptors specify one or more search conditions to execute within a single record in order to satisfy the search. Each search argument descriptor is made up of a displacement, a pattern length, and a pattern.

The displacement field specifies the displacement in bytes of the first byte of the data to be compared from the start of the logical record.

The pattern length field specifies the length in bytes of the pattern that follows.

The pattern specifies the data to compare to the logical record.

8.2.7.1 SEARCH DATA HIGH Command
The SEARCH DATA HIGH command (table 72, operation code 30h) shall be satisfied by the first logical record searched that contains data that satisfies all of the search argument descriptor(s). If the invert bit in the command descriptor block is zero, the search argument descriptor(s) shall be satisfied by data in the logical record being greater than the data in the pattern. If the invert bit is one, the search argument descriptor(s) shall be satisfied by data in the logical record being less than or equal to the data in the pattern. (See 8.2.7.)

8.2.7.2 SEARCH DATA EQUAL Command
The SEARCH DATA EQUAL command (table 72, operation code 31h) shall be satisfied by the first logical record searched that contains data that satisfies all of the search argument descriptor(s). If the invert bit in the command descriptor block is zero, the search argument descriptor(s) shall be satisfied by data in the logical record being equal to the data in the pattern. If the invert bit is one, the search argument descriptor(s) shall be satisfied by data in the logical record being not equal to the data in the pattern. (See 8.2.7.)

Table 73 – SEARCH DATA Parameter List

Byte	Parameter List Header
0	Logical Record Length (MSB)
1	Logical Record Length
2	Logical Record Length
3	Logical Record Length (LSB)
4	First Record Offset (MSB)
5	First Record Offset
6	First Record Offset
7	First Record Offset (LSB)
8	Number of Records (MSB)
9	Number of Records
10	Number of Records
11	Number of Records (LSB)
12	Search Argument Length (LSB)
13	Search Argument Length (LSB)
	Search Argurment Descriptors
0	Displacement (MSB)
1	Displacement
2	Displacement
3	Displacement (LSB)
4	Pattern Length (MSB)
5	Pattern Length (LSB)
6 to n	Pattern

8.2.7.3 SEARCH DATA LOW Command.
The SEARCH DATA LOW command (table 72, operation code 32h) shall be satisfied by the first logical record searched that contains data that satisfies all of the search argument descriptor(s). Ifthe invert bit in the command descriptor block is zero, the search argument descriptor(s) shall be satisfied by data in the logical record being less than the data in the pattern. If the invert bit is one, the search argument descriptor(s) shall be satisfied by data in the logical record being greater than or equal to the data in the pattern. (See 8.2.7.)

8.2.8 SET LIMITS Command
Peripheral Device Type: Direct Access, Write-Once Read-Multiple, and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 33h

Table 74 – SET LIMITS Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			RdInh	WrInh
2	Logical Block Address (MSB)							
3	Logical Block Address							
4	Logical Block Address							
5	Logical Block Address (LSB)							
6	Reserved							
7	Number of Blocks (MSB)							
8	Number of Blocks (LSB)							
9	Vendor Unique		Reserved				Flag	Link

The SET LIMITS command (table 74) defines the range within which subsequent linked commands may operate. A second SET LIMITS command may not be linked to a chain of commands in which a SET LIMITS command has already been issued.

A read inhibit (RdInh) bit of one indicates that read operations within the range are inhibited. A write inhibit (WrInh) bit of one indicates that write operations within the range are inhibited.

The logical block address specifies the starting address for the range. The number of blocks specifies the number of logical blocks within the range. A number of blocks of zero indicates that the range shall extend to the last logical block on the logical unit.

Any attempt to access outside of the restricted range or any attempt to perform an inhibited operation within the restricted range shall not be performed. The command shall be terminated with CHECK CONDITION status and, if extended sense is implemented, the sense key shall be set to DATA PROTECT. A second SET LIMITS command within a linked list of commands shall be rejected with CHECK CONDITION status and, if extended sense is implemented, the sense key shall be set to DATA PROTECT.

9 Group 0 command descriptions for sequential-access devices

The Group 0 commands for sequential-access devices shall be as shown in table 75.

Table 75 – Group 0 Commands for Sequential-Access Devices

Operation Code	Type	Command Name	Subclause	Page
00h	O	TEST UNIT READY	7.1.1	44
01h	M	REWIND	9.1	100
02h	V			
03h	M	REQUEST SENSE	7.1.2	44
04h	R			
05h	E	READ BLOCK LIMITS	9.2	101
06h	V			
07h	V			
08h	M	READ	9.3	102
09h	V			
0Ah	M	WRITE	9.4	103
0Bh	O	TRACK SELECT	9.5	104
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	O	READ REVERSE	9.6	105
10h	M	WRITE FILEMARKS	9.7	106
11h	O	SPACE	9.8	107
12h	E	INQUIRY	7.1.3	48
13h	O	VERIFY	9.9	108
14h	O	RECOVER BUFFERED DATA	9.10	109
15h	O	MODE SELECT	9.11	110
16h	O	RESERVE UNIT	9.12.1	113
17h	O	RELEASE UNIT	9.12.2	114
18h	O	COPY	7.1.4	50
19h	O	ERASE	9.13	114
1Ah	O	MODE SENSE	9.14	115
1Bh	O	LOAD/UNLOAD	9.15	117
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	57
1Dh	O	SEND DIAGNOSTIC	7.1.6	58
1Eh	O	PREVENT/ALLOW MEDIUM REMOVAL	9.16	118
1Fh	R			

Key:
M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.
* = These operation codes may have different meanings for specific types of peripheral devices. (See the appropriate section for further information.)

9.1 REWIND Command

Peripheral Device Type: Sequential Access
Operation Code Type: Mandatory
Operation Code: 01h

Table 76 – REWIND Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Immed
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The REWIND command (table 76) requests that the target rewind the logical unit to the beginning-of-medium or load-point.

An immediate (Immed) bit of one indicates that status shall be returned as soon as the operation is initiated. An Immed bit of zero indicates that status shall be returned after the operation is complete.

9.2 READ BLOCK LIMITS Command

Peripheral Device Type: Sequential Access
Operation Code Type: Extended
Operation Code: 05h

Table 77 – READ BLOCK LIMITS Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The READ BLOCK LIMITS command (table 77) requests that the target's capability for block length limits be returned for the logical unit. The READ BLOCK LIMITS data shown in table 78 shall be sent during the DATA IN phase of the command.

Table 78 – READ BLOCK LIMITS Data

Byte	Description
0	Reserved
1	Maximum Block Length (MSB)
2	Maximum Block Length
3	Maximum Block Length (LSB)
4	Minimum Block Length (MSB)
5	Minimum Block Length (LSB)

If the maximum block length equals the minimum block length, only fixed-length blocks of the length indicated are supported. Otherwise, variable-length blocks are supported. For variable-length blocks, if the maximum block length equals zero, no upper limit is specified.

9.3 READ Command

Peripheral Device Type: Sequential Access
Operation Code Type: Mandatory
Operation Code: 08h

Table 79 – READ Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Fixed
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The READ command (table 79) transfers one or more block(s) to the initiator beginning with the next block on the logical unit. The fixed bit specifies both the meaning of the transfer length field and whether fixed-length or variable-length block(s) are to be transferred.

If the fixed bit is zero, a single block shall be transferred with the transfer length specifying the maximum number of bytes the initiator has allocated for the returned data. If the actual block length is different from the specified transfer length, a CHECK CONDITION status shall be sent to the initiator and the incorrect length indicator (ILI) bit and valid bit in extended sense shall be set to one. The information bytes in extended sense shall be set to the difference (residue) between the requested transfer length and the actual block length. Targets that do not support negative residues shall set the ILI bit to one and the residue to zero when the actual block length is larger than the transfer length. In any case, no more than transfer length bytes shall be transferred to the initiator and the medium shall be positioned after the block (end-of-medium side).

If the fixed bit is one, the transfer length specifies the number of blocks to be transferred to the initiator. This form of the READ command is valid only if the logical unit is currently operating in fixed block mode. A logical unit is in fixed block mode when either of the following conditions are true:

a) The logical unit reports the same value for minimum block length and maximum block length in response to the READ BLOCK LIMITS command. In this case, the current block length is the value returned.

b) The logical unit has been instructed to use fixed-length blocks with the MODE SELECT command. In this case, the current block length is the block length defined in the MODE SELECT command.

Otherwise, the logical unit is in variable block mode. The target may implement fixed block mode, variable block mode, or both modes. If the fixed bit does not match the current mode, or the mode indicated by the fixed bit is not implemented, the target shall reject the command by returning a CHECK CONDITION status and by setting the sense key to ILLEGAL REQUEST.

A successful READ command with the fixed bit equal to one shall transfer the current block length times the transfer length bytes of data to the initiator. Upon termination of the READ command, the medium shall be positioned after the last block transferred (end-of-medium side).

If the fixed bit is one and if a block is read that is larger or smaller than the current block length, a CHECK CONDITION status shall be returned to the initiator. The ILI bit and the valid bit in extended sense shall be set to one. The information bytes shall be set to the difference (residue) between the requested transfer length and the actual number of blocks read (not including the incorrect length block). Upon termination, the medium shall be positioned after the incorrect length block (end-of-medium side).

If a logical unit reads a filemark during a READ command, it shall send a CHECK CONDITION status to the initiator and shall set the filemark bit in extended sense. Upon termination, the medium shall be positioned after the filemark (end-of-medium side). If the fixed bit is one, the target shall set the valid bit to one and the information bytes shall be set to the difference (residue) between the requested transfer length and the actual number of blocks read (not including the filemark).

If a logical unit encounters the physical end-of-medium during a READ command, the target shall return a CHECK CONDITION status to the initiator and shall set the end-of-medium (EOM) bit to one in extended sense. The sense key shall be set to MEDIUM ERROR. If the fixed bit is one, the target shall set the valid bit to one and the information bytes to the difference (residue) between the requested transfer length and the actual number of blocks successfully read. The medium position following this condition is not defined.

When the transfer length is zero, no data shall be transferred and the current position on the logical unit shall not be changed. This condition shall not be considered as an error.

9.4 WRITE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Mandatory
Operation Code: 0Ah

Table 80 – WRITE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Fixed
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The WRITE command (table 80) transfers one or more block(s) from the initiator to the current position on the logical unit. The fixed bit specifies both the meaning of the transfer length field and whether fixed-length or variable-length block(s) are to be transferred.

If the fixed bit is zero, a single block shall be transferred from the initiator and shall be written to the logical unit beginning at the current medium position. The transfer length specifies the length of the block to be written (in bytes). The requested block length shall be within the minimum and maximum block length range (returned by the READ BLOCK LIMITS command, 9.2). If this condition is not met, a CHECK CONDITION status shall be returned and the sense key shall be set to ILLEGAL REQUEST and no data shall be written. Upon successful termination, the medium shall be positioned after the block written by this command (end-of-medium side).

If the fixed bit is one, the transfer length field specifies the number of block(s) to be transferred to the logical unit beginning at the current medium position. This form of the WRITE command is valid only if the logical unit is currently operating in fixed block mode (see 9.3). Upon termination, the medium shall be positioned after the block(s) written by this command (end-of-medium side).

The target may implement fixed block mode, variable block mode, or both modes. If the fixed bit does not match the current mode, or the mode indicated by the fixed bit is not implemented, the target shall reject the command by returning a CHECK CONDITION status and by setting the sense key to ILLEGAL REQUEST.

If the early warning end-of-medium condition is encountered while writing, an attempt to finish writing any buffered data may be made. The command shall terminate with a CHECK CONDITION status and the EOM bit in extended sense shall be set to one. If any data remains in the target's buffer, then the sense key shall be set to VOLUME OVERFLOW. If the fixed bit is one and the logical unit is not buffered (buffered mode of the MODE SENSE command is zero), then the valid bit in extended sense shall be set to one and the information bytes shall be set to the difference (residue) between the requested transfer length and the actual number of blocks written to the medium. If the fixed bit is one and the logical unit is buffered (buffered mode of the MODE SENSE command is one), then the valid bit shall be set to one and the information bytes shall be set to the total number of blocks not written (the number of blocks not transferred from the initiator plus the number of blocks remaining in the target's buffer). Note that in this case it is possible for the value in the information bytes to exceed the transfer length.

When the transfer length is zero, no data shall be transferred and the current position on the logical unit shall not be changed. This condition shall not be considered as an error.

9.5 TRACK SELECT Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 0Bh

Table 81 – TRACK SELECT Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Fixed
2	Reserved							
3	Reserved							
4	Track Value							
5	Vendor Unique		Reserved				Flag	Link

The TRACK SELECT command (table 81) requests that the track specified in the track value field be selected.

9.6 READ REVERSE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 0Fh

Table 82 – READ REVERSE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Fixed
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The READ REVERSE command (table 82) functions identically to the READ command except that medium motion is in the reverse direction. Thus, the block(s) and bytes within the block(s) are transferred in the reverse order and the medium position upon termination is before the last block read (beginning-of-medium side). This command shall terminate with a CHECK CONDITION status and the EOM bit in extended sense shall be set to one if beginning-of-medium or load-point is encountered. The sense key shall be set to NO SENSE. If the fixed bit is one, then the valid bit shall be set to one and the information bytes shall contain the difference (residue) of the requested transfer length and the actual number of blocks transferred before beginning-of-medium or load-point was encountered.

Filemark handling is the same as in the READ command except that the medium position upon command termination shall be before the filemark (beginning-of- medium side).

If the transfer length is zero, no data shall be transferred and the current position on the logical unit shall not be changed. This condition shall not be considered as an error.

The target may implement fixed block mode, variable block mode, or both modes. If the fixed bit does not match the current mode, or the mode indicated by the fixed bit is not implemented, the target shall reject the command by returning a CHECK CONDITION status and by setting the sense key to ILLEGAL REQUEST.

9.7 WRITE FILEMARKS Command

Peripheral Device Type: Sequential Access
Operation Code Type: Mandatory
Operation Code: 10h

Table 83 – WRITE FILEMARKS Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Number of Filemarks (MSB)							
3	Number of Filemarks							
4	Number of Filemarks (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The WRITE FILEMARKS command (table 83) causes the specified number of filemarks to be written beginning at the current medium position on the logical unit. A zero in this field indicates that no filemarks are to be written.

This command is also used to force any buffered data (see buffered mode in the MODE SENSE command, 9.14) to be written. This command shall not return a GOOD status unless all buffered data blocks and the filemarks (if any) are correctly written on the medium.

If the early warning end-of-medium condition is encountered while writing, an attempt to finish writing any buffered data may be made. The command shall terminate with a CHECK CONDITION status and the EOM bit in extended sense shall be set to one. If any filemarks remain to be written, then the sense key shall be set to VOLUME OVERFLOW. If the logical unit is not buffered (buffered mode of the MODE SENSE command is zero), then the valid bit in extended sense shall be set to one and the information bytes shall be set to the number of unwritten filemarks. If the logical unit is buffered (buffered mode of the MODE SENSE command is one), then the valid bit shall be set to one and the information bytes shall be set to the total number of blocks not written (the number of unwritten filemarks plus the number of blocks remaining in the target's buffer). Note that in this case it is possible for the value in the information bytes to exceed the transfer length.

9.8 SPACE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 11h

Table 84 – SPACE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			Code	
2	Count (MSB)							
3	Count							
4	Count (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The SPACE command (table 84) provides a variety of positioning functions that are determined by the code and count. Both forward (toward end-of- medium) and reverse (toward beginning-of-medium) positioning are provided, although some SCSI devices may only support a subset of this command. Such SCSI devices shall return a CHECK CONDITION status and set the sense key to ILLEGAL REQUEST in response to any attempt to invoke a function that is not supported.

The code is defined as follows:

DB(1)	DB(0)	Description
0	0	Blocks
0	1	Filemarks
1	0	Sequential Filemarks
1	1	Physical End-of-Data

When spacing over blocks or filemarks, the count field specifies the number of blocks or filemarks to be spaced over. A positive value N in the count field shall cause forward medium movement over N blocks or filemarks ending on the end-of-medium side of the last block or filemark. A zero value in the count field shall cause no medium movement. A negative value -N (2's complement notation) in the count field shall cause reverse medium movement over N blocks or filemarks ending on the beginning-of-medium side of the last block or filemark.

If a filemark is encountered while spacing over blocks, medium movement shall be stopped. The medium shall be positioned on the end-of-medium side of the filemark if movement was in the forward direction and on the beginning-of-medium side of the filemark if movement was in the reverse direction. A CHECK CONDITION status shall be sent to the initiator and the filemark and valid bits in extended sense shall be set to one. The information bytes shall be set to the difference (residue) in the requested count and the actual number of blocks spaced over (not including the filemark).

If the physical end-of-medium is encountered while spacing forward over blocks or filemarks, the target shall return a CHECK CONDITION status to the initiator and shall set the end-of-medium (EOM) bit in extended sense to one. The sense key shall be set to MEDIUM ERROR. The target shall set the valid bit to one and the information bytes to the difference (residue) between the requested count and the actual number of blocks or filemarks spaced over.

If beginning-of-medium or load-point is encountered while spacing over blocks or filemarks in the reverse direction, the target shall return a CHECK CONDITION status to the initiator and shall set the end-of-medium (EOM) bit in extended sense to one. The sense key shall be set to NO SENSE. The target shall set the valid bit to one and the information bytes to the difference (residue) between the requested count and the actual number of blocks or filemarks spaced over.

When spacing over sequential filemarks, the count field is interpreted as follows:

a) A positive value N shall cause forward medium movement to the first occurrence of N or more consecutive filemarks stopping after the Nth filemark.

b) A zero value shall cause no medium movement.

c) A negative value -N (2's complement notation) shall cause reverse medium movement to the first occurrence of N or more consecutive filemarks stopping on the beginning-of-medium side of the Nth filemark.

When spacing to physical end-of-data, the count field is ignored. Forward medium movement shall occur until the logical unit encounters physical end-of-data as defined by the sequential-access device. Some sequential-access devices define physical end-of-data as an erased area on the medium; however, other definitions are not precluded. Targets that implement this function shall leave the medium positioned such that a subsequent WRITE command would append data to the last recorded information on the medium.

9.9 VERIFY Command

Peripheral Device Type:	Sequential Access
Operation Code Type:	Optional
Operation Code:	13h

Table 85 – VERIFY Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			BytCmp	Fixed
2	Verification Length (MSB)							
3	Verification Length							
4	Verification Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The VERIFY command (table 85) verifies one or more block(s) beginning with the next block on the logical unit. The fixed bit specifies both the meaning of the verification length field and whether fixed-length or variable-length block(s) are to be verified.

A byte compare (BytCmp) bit of zero indicates that the verification shall be simply a medium verification (CRC, ECC, etc). No data shall be transferred between the initiator and target. A byte compare bit of one indicates that a byte-by-byte compare of the data on the medium, and the data transferred from the initiator shall be performed by the target. Data shall be transferred from the initiator to the target as in a WRITE command.

A fixed bit of zero requests that the next block of the logical unit be verified. The verification length specifies the number of bytes to verify. A fixed bit of one requests verification length blocks be verified beginning with the next logical block on the logical unit. This form of the VERIFY command is only valid if the logical unit is currently in fixed block mode as defined in the READ command. If the data does not compare (byte compare bit equals one), the command shall terminate with a CHECK CONDITION status and the sense key shall be set to MISCOMPARE. If the fixed bit is one, the valid bit shall be set to one and the information bytes shall be set to the difference (residue) between the verification length and the actual number of blocks successfully verified. The medium shall be positioned after the block containing the miscompare (end-of-medium side).

The target may implement fixed block mode, variable block mode, or both modes. If the fixed bit does not match the current mode, or the mode indicated by the fixed bit is not implemented, the target shall reject the command by returning a CHECK CONDITION status and by setting the sense key to ILLEGAL REQUEST.

The VERIFY command shall terminate when the verification length has been satisfied, when a filemark is encountered, or when physical end-of-medium is encountered. The status and sense data for each of these conditions are handled the same as in the READ command. Upon completion of the VERIFY command, the medium shall be positioned after the last block from which data was verified or after the filemark, if encountered.

When the verification length is zero, no data shall be verified and the current position on the logical unit shall not be changed. This condition shall not be considered as an error.

9.10 RECOVER BUFFERED DATA Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 14h

Table 86 – RECOVER BUFFERED DATA Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Fixed
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RECOVER BUFFERED DATA command (table 86) is used to read data that has been transferred to an SCSI device buffer but has not been written on the medium. It is normally only used to recover from error or exception conditions that make it impossible to write the buffered data on the medium.

This command functions similarly to the READ command except that the data is transferred from the SCSI device buffer instead of the medium. The order in which block(s) are transferred is the same as if they would have been transferred to the medium. One or more RECOVER BUFFERED DATA commands may be used to read the unwritten buffered data.

Refer to the READ command (9.3) for a definition of the fixed bit and the transfer length field.

If an attempt is made to recover more logical blocks of data than are contained in the SCSI device buffer, the command shall be terminated with a CHECK CONDITION status. The EOM bit in extended sense shall be set to one. If the fixed bit is one, the valid bit shall be set to one and the information bytes shall be set to the difference (residue) between the requested transfer length and the actual number of blocks transferred.

9.11 MODE SELECT Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 15h

Table 87 – MODE SELECT Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Parameter List Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SELECT command (table 87) provides a means for the initiator to specify medium, logical unit, or peripheral device parameters to the target.

The parameter list length specifies the length in bytes of the MODE SELECT parameter list that shall be transferred during the DATA OUT phase. A zero parameter list length indicates that no data shall be transferred. This condition shall not be considered as an error.

The MODE SELECT parameter list shown in table 88 contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Table 88 – MODE SELECT Parameter List

Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved	Buffered Mode			Speed			
3	Block Descriptor Length							
	Block Descriptor(s)							
0	Density Code							
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Reserved							
5	Block Length (MSB)							
6	Block Length							
7	Block Length (LSB)							
	Vendor Unique Parameter(s)							
0 – n	Vendor Unique Parameter Byte(s)							

A buffered mode of zero indicates that the target shall not report a GOOD status on WRITE commands until the data blocks are actually written on the medium. A buffered mode of one indicates that the target may report a GOOD status on WRITE commands as soon as the data block has been transferred to the SCSI device buffer. One or more blocks may be buffered prior to writing the block(s) to the medium. Buffered modes of 2h through 7h are reserved.

Code values for the speed field shall be assigned as follows:

0h	Default (Use the peripheral device's default speed).
1h	Use the peripheral device's lowest speed.
2h – Fh	Use increasing peripheral device speeds.

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors are included in the parameter list. This condition shall not be considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a density code, a number of blocks, and a block length.

Code values for the density code field are defined in table 89.

Table 89 — Sequential-Access Density Codes

Code Value	Density								
00h	Default (peripheral device's default or only density)								
	Magnetic Tape								
	Width		Tracks	Density		Code	Type	ISO Reference Standard	Note
	mm	(inch)		bpmm	(bpi)				
01h	12,7	(0,5)	9	32	(800)	NRZI	R	1863 : 1976	2
02h	12,7	(0,5)	9	63	(1 600)	PE	R	3788 : 1976	2
03h	12,7	(0,5)	9	246	(6 250)	GCR	R	5652 : 1984	2
04h	6,3	(0,25)	4/9	315	(8 000)	GCR	C		1
05h	6,3	(0,25)	4/9	315	(8 000)	GCR	C	8462-1,2 : 1986	1
06h	12,7	(0,5)	9	126	(3 200)	PE	R		2
07h	6,3	(0,25)	4	252	(6 400)	IMFM	C	8063-1 : 1984	1
08h	3,81	(0,15)	4	315	(8 000)	GCR	CS		1
09h	12,7	(0,5)	18	1491	(37 871)	GCR	C		2
0Ah	12,7	(0,5)	22	262	(6 667)	MFM	C		1
0Bh	6,3	(0,25)	4	63	(1 600)	PE	C	4057 : 1979	1
0Ch	12,7	(0,5)	24	500	(12 690)	GCR	C		1
0Dh	12,7	(0,5)	24	999	(25 380)	GCR	C		1
80h - FFh	Vendor unique								
All others	Reserved								

Key:	Code	Type
NRZI	Non Return to Zero, change on ones	R Reel-to-Reel
GCR	Group Code Recording	C Cartridge
PE	Phase Encoded	CS Cassette
IMFM	Inverted Modified Frequency Modulation	
MFM	Modified Frequency Modulation	

NOTES

1 Serially Recorded.
2 Parallel Recorded.
3 See Annex F for additional standards that may be applicable.

The number of blocks field specifies the number of logical blocks on the medium that meet the density code and block length in the block descriptor. A number of blocks of zero indicates that all of the remaining logical blocks of the logical unit shall have the medium characteristics specified by the block descriptor.

The block length specifies the length in bytes of each logical block described by the block descriptor. A block length of zero indicates that the length shall be variable.

9.12 RESERVE UNIT and RELEASE UNIT Commands

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 16h and 17h, respectively

Table 90 – RESERVE UNIT and RELEASE UNIT Commands

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			3rdPty	Third Party Device ID			Reserved
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The RESERVE UNIT and RELEASE UNIT commands both use the command descriptor block shown in table 90.

9.12.1 RESERVE UNIT Command

The RESERVE UNIT command (table 90, operation code 16h) shall reserve the specified logical unit for the exclusive use by the requesting initiator or, if third-party reservation option is implemented, to another specified SCSI device.

The reservation shall remain in effect until superseded by another RESERVE UNIT command from the initiator that made the reservation or until released by a RELEASE UNIT command from the same initiator, or a BUS DEVICE RESET message from any initiator, or a "hard" RESET condition. The occurrence of the last two conditions is indicated by a sense key of UNIT ATTENTION on the next command following the condition. It is not an error to issue this command to a logical unit that is currently reserved to the requesting initiator.

If the logical unit is previously reserved by another initiator, then the target shall return RESERVATION CONFLICT status.

If, after honoring the reservation, any other initiator then subsequently attempts to perform any command on the reserved logical unit other than a RELEASE UNIT command, which shall be ignored, then the command shall be rejected with a RESERVATION CONFLICT status.

The third-party reservation option for the RESERVE UNIT command allows an initiator to reserve a logical unit for another SCSI device. This option is intended for use in multiple-initiator systems that use the COPY command. Any target that implements the third-party reservation option shall also implement the third-party release option (see 9.12.2).

If the third-party (3rdPty) bit is zero, then the third-party reservation option is not requested. If the 3rdPty bit is one and the third-party reservation option is implemented, then the RESERVE UNIT command shall reserve the specified logical unit for the SCSI device specified in the third-party device ID field. The target shall preserve the reservation until superseded by another RESERVE UNIT command from the initiator that made the reservation or until released by the same initiator, by a BUS DEVICE RESET message from any initiator, or by a "hard" RESET condition. The target shall ignore (i.e., return GOOD status) any attempt made by any other initiator to release the reservation.

If the 3rdPty bit is one and the third-party reservation option is not implemented, then the target shall reject the RESERVE UNIT command with a CHECK CONDITION status and a sense key of ILLEGAL REQUEST.

An initiator that holds a current reservation may modify that reservation (e.g., switch third-parties) by issuing another RESERVE UNIT command to the same logical unit. The superseding RESERVE UNIT command shall release the previous reservation state only when the new reservation is granted..

9.12.2 RELEASE UNIT Command The RELEASE UNIT command (table 90, operation code 17h) shall release the logical unit if it is currently reserved by the requesting initiator.

It is not an error to attempt to release a logical unit that is not currently reserved to the requesting initiator. However, it shall not be released if it is reserved by another initiator.

The third-party release option for the RELEASE UNIT command allows an initiator to release a logical unit that was previously reserved using the third-party reservation option (see 9.12.1). This option shall be implemented if the third-party reservation option is implemented. This option is intended for use in multiple-initiator systems that use the COPY command.

If the third-party (3rdPty) bit is zero, then the third-party release option is not requested. If the 3rdPty bit is one and the target implements the third-party release option, then the target shall release the specified logical unit, but only if the reservation was made using the third-party reservation option by the initiator that is requesting the release and for the same SCSI device as specified in the third-party device ID field.

If the 3rdPty bit is one and the target does not implement the third-party release option, then the target shall terminate the command with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

9.13 ERASE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 19h

Table 91 – ERASE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Long
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The ERASE command (table 91) causes part or all of the remaining medium to be erased beginning from the current medium position. As used here, “erased” means either the medium shall be erased or a pattern shall be written on the medium that appears as gap to the target.

The distance to be erased is controlled by the long bit. A long bit of one indicates that all remaining medium on the logical unit shall be erased. A long bit of zero indicates that a peripheral device specified portion of the medium shall be erased. Normally, short erases are used to create an extended gap for software controlled error recovery or for support of “update in place” functions. The medium position following an ERASE command with a long bit of one is not defined by this standard.

NOTE Some targets may reject ERASE commands with the long bit set to one if the medium is not positioned at the beginning-of-medium.

9.14 MODE SENSE Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optiona
Operation Code: 1Ah

Table 92 – MODE SENSE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SENSE command (table 92) provides a means for a target to report its medium, logical unit, or peripheral device parameters to the initiator. It is a complementary command to the MODE SELECT command (see 9.11) for support of a medium that may contain different densities, such as half-inch tapes.

The allocation length specifies the number of bytes that the initiator has allocated for returned MODE SENSE data. An allocation length of zero indicates that no MODE SENSE data shall be transferred. This condition shall not be considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available MODE SENSE data have been transferred to the initiator, whichever is less.

The MODE SENSE data (table 93) contains a four-byte header, followed by zero or more eight-byte block descriptors, followed by the vendor unique parameters, if any.

Table 93 – MODE SENSE Data

Byte \ Bit	7	6	5	4	3	2	1	0
0	Sense Data Length							
1	Medium Type							
2	WP	Buffered Mode			Speed			
3	Block Descriptor Length							
	Block Descriptor(s)							
0	Density Code							
1	Number of Blocks (MSB)							
2	Number of Blocks							
3	Number of Blocks (LSB)							
4	Reserved							
5	Block Length (MSB)							
6	Block Length							
7	Block Length (LSB)							
	Vendor Unique Parameter(s)							
0 to n	Vendor Unique Parameter Byte(s)							

The sense data length specifies the length in bytes of the following mode sense data that is available to be transferred during the DATA IN phase. The sense data length does not include itself.

Code values for the medium type field shall be assigned as follows:

00h	Default (Only one medium type supported)
01h – 7Fh	Reserved
80h – FFh	Vendor unique

A write protected (WP) bit of zero indicates that the medium is write enabled. A write protected bit of one indicates that the medium is write protected.

A buffered mode of zero indicates that the target does not report a GOOD status on WRITE commands until the data blocks are actually written on the medium. A buffered mode of one indicates that the target may report a GOOD status on WRITE commands as soon as the data block has been transferred to the SCSI device buffer. One or more blocks may be buffered prior to writing the block(s) to the medium. Buffered modes of 2h through 7h are reserved.

Code values for the speed field shall be assigned as follows:

0h	Default (only one speed supported)
1h	Lowest peripheral device speed
2h – Fh	Increasing peripheral device speeds

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight and does not include the vendor unique parameters, if any. A block descriptor length of zero indicates that no block descriptors shall be included in the parameter list. This condition shall not be considered as an error.

Each block descriptor specifies the medium characteristics for all or part of a logical unit. Each block descriptor contains a density code, a number of blocks, and a block length.

Code values for the density code field are defined in table 89.

The number of blocks field specifies the number of logical blocks on the medium that meet the density code and block length in the block descriptor. A number of blocks of zero indicates that all of the remaining logical blocks of the logical unit have the medium characteristics specified by the block descriptor.

The block length specifies the length in bytes of each logical block described by the block descriptor. A block length of zero indicates that the length is variable.

9.15 LOAD/UNLOAD Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 1Bh

Table 94 – LOAD/UNLOAD Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Immed
2	Reserved							
3	Reserved							
4	Reserved						Re-Ten	Load
5	Vendor Unique		Reserved				Flag	Link

The LOAD/UNLOAD command (table 94) requests that the target enable or disable the logical unit for further operations. This command may also be used to request the re-tension function on peripheral devices that support this function.

A load bit of one indicates that the medium on the logical unit shall be loaded and positioned to the beginning-of-medium or load-point as determined by the peripheral device. A load bit of zero indicates that the medium on the logical unit shall be positioned for removal from the peripheral device.

Status shall be returned after the medium is positioned unless the immediate (Immed) bit is one. If the Immed bit is one, status may be returned as soon as the command has been accepted.

A re-tension (Re-Ten) bit of one indicates that the medium on the addressed logical unit shall be correctly tensioned before the LOAD/UNLOAD command is completed. This is an optional function intended for use by those peripheral devices that support the re-tension function.

9.16 PREVENT/ALLOW MEDIUM REMOVAL Command

Peripheral Device Type: Sequential Access
Operation Code Type: Optional
Operation Code: 1Eh

Table 95 – PREVENT/ALLOW MEDIUM REMOVAL Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							Prevent
5	Vendor Unique		Reserved				Flag	Link

The PREVENT/ALLOW MEDIUM REMOVAL command (table 95) requests that the target enable or disable the removal of the medium in the logical unit.

A prevent bit of one shall inhibit mechanisms that normally allow removal of the medium. A prevent bit of zero shall allow removal of the medium.

This prevention of medium removal condition shall terminate upon receipt of a PREVENT/ALLOW MEDIUM REMOVAL command with the prevent bit set to zero, or by the receipt of a BUS DEVICE RESET message from any initiator or by a “hard” RESET condition.

10 Group 0 command descriptions for printer devices

The Group 0 commands for printer devices shall be as shown in table 96.

Table 96 – Group 0 Commands for Printer Devices

Operation Code	Type	Command Name	Subclause	Page
00h	O	TEST UNIT READY	7.1.1	44
01h	V			
02h	V			
03h	M	REQUEST SENSE	7.1.2	44
04h	O	FORMAT	10.1	120
05h	V			
06h	V			
07h	V			
08h	V			
09h	V			
0Ah	M	PRINT	10.2	121
0Bh	O	SLEW AND PRINT	10.3	121
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	V			
10h	O	FLUSH BUFFER	10.4	122
11h	V			
12h	E	INQUIRY	7.1.3	48
13h	V			
14h	O	RECOVER BUFFERED DATA	10.5	123
15h	O	MODE SELECT	10.6	124
16h	O	RESERVE UNIT	10.7.1	125
17h	O	RELEASE UNIT	10.7.2	126
18h	O	COPY	7.1.4	50
19h	V			
1Ah	O	MODE SENSE	10.8	126
1Bh	O	STOP PRINT	10.9	128
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	57
1Dh	O	SEND DIAGNOSTIC	7.1.6	58
1Eh	R			
1Fh	R			

Key:

M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

10.1 FORMAT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 04h

Table 97 – FORMAT Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved			Format Type	
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The FORMAT command (table 97) provides a means for the initiator to specify forms or fonts to printers that support programmable forms or fonts. The format information sent is vendor unique since it is peripheral-device specific.

The format type field specifies the type of format information to be transferred from the initiator to the target. This field is defined as follows:

DB(1)	DB(0)	Format Type
0	0	Set Form
0	1	Set Font
1	0	Vendor Unique
1	1	Reserved

The transfer length specifies the length in bytes of format information that shall be sent during the DATA OUT phase. A transfer length of zero indicates that no format information shall be sent. This condition shall not be considered as an error.

10.2 PRINT Command

Peripheral Device Type: Printer
Operation Code Type: Mandatory
Operation Code: 0Ah

Table 98 – PRINT Command

Byte Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The PRINT command (table 98) transfers the specified number of bytes from the initiator to the target to be printed. The data sent is application dependent.

The transfer length specifies the length in bytes of data that shall be sent during the DATA OUT phase. A transfer length of zero indicates that no data shall be sent. This condition shall not be considered as an error.

10.3 SLEW AND PRINT Command

Peripheral Device Type:Printer
Operation Code Type:Optional
Operation Code:

Table 99 – SLEW AND PRINT Command

Byte Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Channel
2	Slew Value							
3	Transfer Length (MSB)							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The SLEW AND PRINT command (table 99) transfers the specified number of bytes from the initiator to the target to be printed. The data sent is application dependent. This command is provided for printers that do not support forms control information imbedded within the print data.

The transfer length specifies the length in bytes of data that shall be sent during the DATA OUT phase. A transfer length of zero indicates that no data shall be sent. This condition shall not be considered as an error.

If the channel bit is zero, the slew value specifies the number of lines the form shall be advanced before printing. A value of 255 indicates that the form shall be advanced to the first line of the next form before printing. If the channel bit is one, the slew value specifies the forms control channel number to which the form shall be advanced prior to printing the data.

If the channel bit is one, and the channel option is not implemented, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

10.4 FLUSH BUFFER Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 10h

Table 100 – FLUSH BUFFER Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The FLUSH BUFFER command (table 100) provides a means for an initiator to ensure that the data have been successfully printed prior to releasing the peripheral device. This is useful for applications that wish to handle any error or exception conditions (e.g., end-of-medium) prior to termination of the application.

When all buffered data are actually printed the command shall be terminated with a GOOD status. If it is not possible to finish printing all of the buffered data (due to an error or exception condition on the peripheral device), then this command shall be terminated with a CHECK CONDITION status and the appropriate sense key.

10.5 RECOVER BUFFERED DATA Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 14h

Table 101 – RECOVER BUFFERED DATA Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RECOVER BUFFERED DATA command (table 101) returns to the initiator the data that has been sent to the target, but not yet printed.

This command is normally used only to recover from error or exception conditions that make it impossible to print the buffered data. The order in which the data is transferred from the target to the initiator is the same as it was when the data was previously transferred using the PRINT command or SLEW AND PRINT command. Data that is transferred by this command is deleted from the target data buffer. One or more RECOVER BUFFERED DATA commands may be used to return the unprinted buffered data.

If an attempt is made to recover more data than is contained in the buffer, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to NO SENSE. In addition, the EOM, the valid, and the ILI bits in extended sense shall be set to one. The information bytes shall be set to the difference (residue) between the transfer length and the actual number of bytes returned.

The transfer length specifies the maximum length in bytes of data that shall be transferred during the DATA IN phase. A transfer length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

10.6 MODE SELECT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 15h

Table 102 – MODE SELECT Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Parameter List Length							
5	Vendor Unique	Reserved					Flag	Link

The MODE SELECT command (table 102) provides a means for the initiator to specify medium, logical unit, or peripheral device parameters to the target.

The parameter list length specifies the length in bytes of the MODE SELECT parameter list that shall be transferred during the DATA OUT phase. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. The MODE SELECT parameter list (table 103) contains a four-byte header, followed by the vendor unique parameters, if any.

Table 103 – MODE SELECT Parameter List

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved	Buffered Mode			Reserved			
3	Reserved							
	Vendor Unique Parameter(s)							
0 to n	Vendor Unique Parameter Byte(s)							

A buffered mode of zero indicates that the target shall not report a GOOD status on PRINT commands or SLEW AND PRINT commands until the data are actually printed. A buffered mode of one indicates that the target may report a GOOD status on PRINT commands or SLEW AND PRINT commands as soon as the data have been transferred to the

SCSI device buffer. The data from one or more commands may be buffered prior to printing. Buffered modes of 2h through 7h are reserved.

10.7 RESERVE UNIT and RELEASE UNIT Commands

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 16h and 17h, respectively

Table 104 – RESERVE UNIT and RELEASE UNIT Commands

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			3rdPty	Third Party Device ID			Reserved
2	Reserved							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The RESERVE UNIT and RELEASE UNIT commands both use the command descriptor block shown in table 104.

10.7.1 RESERVE UNIT Command

The RESERVE UNIT command (table 104, operation code 16h) shall reserve the specified logical unit for the exclusive use by the requesting initiator or, if third-party reservation option is implemented, to another specified SCSI device.

The reservation shall remain in effect until superseded by another RESERVE UNIT command from the initiator that made the reservation or until released by a RELEASE UNIT command from the same initiator, or a BUS DEVICE RESET message from any initiator, or a "hard" RESET condition. The occurrence of the last two conditions is indicated by a sense key of UNIT ATTENTION on the next command following the condition. It is not an error to issue this command to a logical unit that is currently reserved to the requesting initiator.

If the logical unit is previously reserved by another initiator, then the target shall return RESERVATION CONFLICT status.

If, after honoring the reservation, any other initiator then subsequently attempts to perform any command on the reserved logical unit other than a RELEASE UNIT command, which shall be ignored, then the command shall be rejected with a RESERVATION CONFLICT status.

The third-party reservation option for the RESERVE UNIT command allows an initiator to reserve a logical unit for another SCSI device. This option is intended for use in multiple-initiator systems that use the COPY command. Any target that implements the third-party reservation option shall also implement the third-party release option (see 10.7.2).

If the third-party (3rdPty) bit is zero, then the third-party reservation option is not requested. If the 3rdPty bit is one and the third-party reservation option is implemented, then the RESERVE UNIT command shall reserve the specified logical unit for the SCSI device specified in the third-party device ID field. The target shall preserve the reservation until superseded by another RESERVE UNIT command from the initiator that made the reservation or until released by the same initiator, by a BUS DEVICE RESET message from any initiator, or by a "hard" RESET condition. The target shall ignore (i.e., return GOOD status) any attempt made by any other initiator to release the reservation.

If the 3rdPty bit is one and the third-party reservation option is not implemented, then the target shall reject the RESERVE UNIT command with a CHECK CONDITION status and a sense key of ILLEGAL REQUEST.

An initiator that holds a current reservation may modify that reservation (e.g., switch third-parties) by issuing another RESERVE UNIT command to the same logical unit. The superseding RESERVE UNIT command shall release the previous reservation state only when the new reservation is granted.

10.7.2 RELEASE UNIT Command

The RELEASE UNIT command (table 104, operation code 17h) shall release the logical unit if it is currently reserved by the requesting initiator.

It is not an error to attempt to release a logical unit that is not currently reserved to the requesting initiator. However, it shall not be released if it is reserved by another initiator.

The third-party release option for the RELEASE UNIT command allows an initiator to release a logical unit that was previously reserved using the third-party reservation option (see 10.7.1). This option shall be implemented if the third-party reservation option is implemented. This option is intended for use in multiple-initiator systems that use the COPY command.

If the third-party (3rdPty) bit is zero, then the third-party release option is not requested. If the 3rdPty bit is one and the target implements the third-party release option, then the target shall release the specified logical unit, but only if the reservation was made using the third-party reservation option by the initiator that is requesting the release and for the same SCSI device as specified in the third-party device ID field.

If the 3rdPty bit is one and the target does not implement the third-party release option, then the target shall terminate the command with a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

10.8 MODE SENSE Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 1Ah

Table 105 – MODE SENSE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Reserved							
3	Reserved							
4	Allocation Length							
5	Vendor Unique		Reserved				Flag	Link

The MODE SENSE command (table 105) provides a means for a target to report its medium, logical unit, or peripheral device parameters to the initiator. It is a complementary command to the MODE SELECT command.

The allocation length specifies the number of bytes that the initiator has allocated for returned MODE SENSE data. An allocation length of zero indicates that no MODE SENSE data shall be transferred. This condition shall not be considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. The target shall terminate the DATA IN phase when allocation length bytes have been transferred or when all available MODE SENSE data have been transferred to the initiator, whichever is less.

The MODE SENSE data (table 106) contains a four-byte header, followed the vendor unique parameters, if any.

Table 106 – MODE SENSE Data

Byte \ Bit	7	6	5	4	3	2	1	0
0	Sense Data Length							
1	Reserved							
2	Reserved	Buffered Mode			Reserved			
3	Reserved							
	Vendor Unique Parameter(s)							
0 to n	Vendor Unique Parameter Byte(s)							

The sense data length specifies the length in bytes of the following MODE SENSE data that is available to be transferred during the DATA IN phase. The sense data length does not include itself.

A buffered mode of zero indicates that the target does not report a GOOD status on PRINT commands or SLEW AND PRINT commands until the data are actually printed. A buffered mode of one indicates that the target may report a GOOD status on PRINT commands or SLEW AND PRINT commands as soon as the data have been transferred to the SCSI device buffer. The data from one or more commands may be buffered prior to printing. Buffered modes of 2h through 7h are reserved.

10.9 STOP PRINT Command

Peripheral Device Type: Printer
Operation Code Type: Optional
Operation Code: 1Bh

Table 107 – STOP PRINT Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				Retain
2	Vendor Unique							
3	Reserved							
4	Reserved							
5	Vendor Unique		Reserved				Flag	Link

The STOP PRINT command (table 107) is used to halt printing on buffered devices in an orderly fashion.

A retain bit of zero requests that the target data buffer be discarded; otherwise, the unprinted data is retained. The unprinted data may be recovered by use of the RECOVER BUFFERED DATA command, if supported. A subsequent PRINT command or SLEW AND PRINT command shall cause the remaining unprinted and unrecovered data to be printed followed by the data transferred by the subsequent command. The point at which printing is suspended by this command is peripheral-device specific and is not defined by this International Standard.

11 Group 0 command descriptions for processor devices

The Group 0 commands for processor devices shall be as shown in table 108.

Table 108 – Group 0 Commands for Processor Devices

Operation Code	Type	Command Name	Subclause	Page
00h	O	TEST UNIT READY	7.1.1	44
01h	V			
02h	V			
03h	M	REQUEST SENSE	7.1.2	44
04h	V			
05h	V			
06h	V			
07h	V			
08h	O	RECEIVE	11.1	122
09h	V			
0Ah	M	SEND	11.2	122
0Bh	V			
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	V			
10h	V			
11h	V			
12h	E	INQUIRY	7.1.3	48
13h	V			
14h	V			
15h	V			
16h	V			
17h	V			
18h	O	COPY	7.1.4	50
19h	R			
1Ah	R			
1Bh	R			
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	57
1Dh	O	SEND DIAGNOSTIC	7.1.6	58
1Eh	R			
1Fh	R			

Key:

M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

11.1 RECEIVE Command

Peripheral Device Type: Processor Devices
Operation Code Type: Optional
Operation Code: 08h

Table 109 – RECEIVE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Allocation Length (MSB)							
3	Allocation Length							
4	Allocation Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The RECEIVE command (table 109) transfers data from the target to the initiator.

The allocation length specifies the number of bytes that the initiator has allocated for the returned data. An allocation length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. Any other value indicates the maximum number of bytes that shall be transferred. 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.

11.2 SEND Command

Peripheral Device Type: Processor Devices
Operation Code Type: Mandatory
Operation Code: 0Ah

Table 110 – SEND Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Reserved				
2	Transfer Length (MSB)							
3	Transfer Length							
4	Transfer Length (LSB)							
5	Vendor Unique		Reserved				Flag	Link

The SEND command (table 110) transfers data from the initiator to the target.

The transfer length specifies the length in bytes of data that shall be sent during the DATA OUT phase. A transfer length of zero indicates that no data shall be sent. This condition shall not be considered as an error.

12 Command descriptions for write-once read-multiple devices

12.1 Group 0 Commands for Write-Once Read-Multiple Devices.

The Group 0 commands for write-once read-multiple devices shall be as shown in table 111.

Table 111 – Group 0 Commands for WORM Devices

Operation Code	Type	Command Name	Subclause	Page
00h	O	TEST UNIT READY	7.1.1	44
01h	O	REZERO UNIT	8.1.1	64
02h	V			
03h	M	REQUEST SENSE	7.1.2	44
04h	R			
05h	V			
06h	V			
07h	O	REASSIGN BLOCKS	8.1.3	69
08h	O	READ	12.1.1	133
09h	V			
0Ah	O	WRITE	12.1.2	134
0Bh	O	SEEK	8.1.6	74
0Ch	V			
0Dh	V			
0Eh	V			
0Fh	V			
10h	V			
11h	V			
12h	E	INQUIRY	7.1.3	48
13h	V			
14h	V			
15h	O	MODE SELECT	12.1.3	136
16h	O	RESERVE	8.1.8	77
17h	O	RELEASE	8.1.9	81
18h	O	COPY	7.1.4	50
19h	V			
1Ah	O	MODE SENSE	12.1.4	138
1Bh	O	START/STOP UNIT	8.1.11	84
1Ch	O	RECEIVE DIAGNOSTIC RESULTS	7.1.5	57
1Dh	O	SEND DIAGNOSTIC	7.1.6	58
1Eh	O	PREVENT/ALLOW MEDIUM REMOVAL	8.1.12	85
1Fh	R			

Key:

M = Command implementation is mandatory.
E = Command implementation is required for SCSI devices that support device-independent self-configuring software.
O = Command implementation is optional.
R = Operation code is reserved for future standardization.
V = Operation code is available for vendor unique commands.

12.1.1 READ Command

Peripheral Device Type: Write-Once Read-Multiple and Read-Only Direct Access
Operation Code Type: Optional
Operation Code: 08h

Table 112 – READ Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	Vendor Unique		Reserved				Flag	Link

The READ command (table 112) requests that the target transfer data to the initiator.

The logical block address specifies the logical block at which the read operation shall begin.

The transfer length specifies the number of contiguous logical blocks of data to be transferred. A Transfer Length of zero indicates that 256 logical blocks shall be transferred. Any other value indicates the number of logical blocks that shall be transferred.

This command shall be terminated with a status of RESERVATION CONFLICT if any reservation access conflict (see 8.1.8) exists and no data shall be transferred.

If any of the conditions indicated in table 113 occur, this command shall be terminated with a CHECK CONDITION status, and if extended sense is implemented, the sense key shall be set as indicated in table 113. Table 113 does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Table 113 – READ Command Check Condition Status

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST
Target reset or medium change since last command from this initiator	UNIT ATTENTION
Unrecoverable read error	MEDIUM ERROR
Attempt to read a blank or previously unwritten block	RECOVERED ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND
When an invalid logical block address causes ILLEGAL REQUEST to be set, the extended sense information bytes shall be set to the logical block address of the first invalid address. When an attempt to read a blank or previously unwritten block causes RECOVERED ERROR to be set, the extended sense information bytes shall be set to the logical block address of the first blank block encountered. The data read up to that block shall be transferred.	

12.1.2 WRITE Command

Peripheral Device Type: Write-Once Read-Multiple
Operation Code Type: Optional
Operation Code: 0Ah

Table 114 – WRITE Command

Byte \ Bit	7	6	5	4	3	2	1	0
0	Operation Code							
1	Logical Unit Number			Logical Block Address (MSB)				
2	Logical Block Address							
3	Logical Block Address (LSB)							
4	Transfer Length							
5	Vendor Unique		Reserved				Flag	Link

The WRITE command (table 114) requests that the target write the data transferred from the initiator to the medium.

The logical block address specifies the logical block at which the write operation shall begin.