92-231 R0 231

SCSI-2 OPTIONS SELECTION DOCUMENT R00

December 5, 1992

To : X3T9.2 Committee
From : Gary R. Stephens
IBM Corporation
9000 S. Rita Rd.
Tucson, Az 85744
(602) 799-2246
Internet: GSTEPHEN@TUCVM2.VNET.IBM.COM

Subject: SCSI-2 Options Selection Document

This document lists items extracted from the SCSI-2 standard that a product must implement and those optional items in the SCSI-2 standard from which an implementation may chose. The SCSI-2 standard is not a device specification. but rather a it is a listing of the SHALLs and guidelines for the MAYs it presents. Only the MAYs are contained in this document. It is the responsibility of a product development team to make the selections and identify those selections for each option.

This document serves as a checklist of pertinent areas of the SCSI-2 standard which affect a product. The selections made should form the basis of a product literature document for the final device. Options related to device classes other than tape and medium changers are not included at present.

This version covers sections 1-7. and 9 of the SCSI-2 standard.

Gary R. Stephens

92-231 R8

TABLE OF CONTENTS

92-231 R0

Structure of the Checklist

The base document for this selection list is ANSI X3T9.2/86-109, Revision 10h. All prior revisions are obsolete.

This document serves as a checklist of pertinent areas of the SCSI-2 standard which affect a product. The selections made form the basis of a product literature document for the final device. Options related to device classes other than tape and medium changers are not included at present.

The order of presentation is by chapter from the SCSI-2 document. Sections are in order through Section 7. The remaining sections have been reordered with appropriate headings added. In general, options selected in the earlier sections may affect selections in later chapters. For example, a decision to support only asynchronous information transfer on a single byte wide cable means that a product need not implement either the Synchronous Data Transfer Request or the Wide Data Transfer Request messages.

SCSI-2 Foreword, Scope, and Glossary

2.1 Foreword

No optional items identified that are not described later.

2.2 Scope

2.2.1 Provision to support SCSI-1 Device Attachments

Any SCSI-2 device may implement options that were available only in SCSI-1 and are now obsolete in SCSI-2. Refer to the SCSI-1 standard and the SCSI-2 standard foreword for differences.

2.2.1.1 Single Initiator Option

2.2.1.2 Non-arbitrating Systems Option

2.2.1.3 Non-extended Sense

2.2.1.4 Reservation Queuing

2.2.2 Multiple Initiator Systems

See provisions for Contingent Allegiance. Extended Contingent Allegiance management. the Unit Attention Condition. and the MODE SELECT/MODE SENSE commands.

See "4.2.0.2 Reset Condition (5.2.2)" on page 8.

2.3 Glossary

No optional items identified that are not described later.

SCSI-2 Section 4, Physical Characteristics

This section identifies options of the physical interface in a SCSI-2 system. These external characteristics and electrical characteristics determine the systems where a SCSI device can attach.

3.1 Physical Description, 4.1.

The A/B cable system is mutually exclusive with the P-cable proposed for SCSI-3 Parallel Interface.

3.1.0.1 68-Conductor B Cable

3.1.0.2 Cable Termination

The SCSI bus is terminated on both ends. If a SCSI device supplies termination, and the termination is not selectable. then only two such devices can exist on the same SCSI bus.

If imbedded termination is supplied it is recommended that it be selectable on the external interface of the SCSI device (e.g., a DIP switch). Jumpers internal to the device are not recommended since this requires opening the device to make the selection.

3.1.0.3 68-Conductor P Cable

SCSI-3 plans to provide a 68-conductor cable to replace the A/B cable system for 2-byte wide systems.

See the draft SCSI-3 Parallel Interface (SPI) document.

This option is mutually exclusive with the A/B cable options.

3.1.0.4 Driver/Receiver Options

A product model chooses Single-Ended or Differential drivers and receivers. Products using Single-Ended and Differential are electrically incompatible. In SCSI-2, Single-Ended drivers and receivers are not recommended if the fast synchronous data transfer option is to be selected (see "3.8 Fast Synchronous Transfer Option, 4.8." on page 5). (See "3.4.0.1 Single-Ended vs Differential" on page 5.)

- Many device manufacturers provide a model of each type.
- Other manufacturers provide an external DIP switch to make an installation time decision.
- External converters have been produced with mixed results because of bus timings.

Total cable length limitations are:

- Single-Ended

  6 Meters including internal device cabling, excluding stubs (See "3.4.0.1 Single-Ended vs Differential" on page 5.)
- Differential

  25 Meters including internal device cabling, excluding stubs (See "3.4.0.1 Single-Ended vs Differential" on page 5.)

3.2 Cable Rrequirements, 4.2.

3.2.0.1 Cable Impedance (Except Fast Synchronous)

Cable selection. along with termination, are the most difficult areas for single-ended systems to manage. In closed systems these items can be matched. In open systems attachment. cable impedance matching is difficult to obtain.

See SCSI-3 committee recommendations on cable impedance and other cable issues and apply those to SCSI-2 systems where applicable.

See "3.2.0.5 Fast Synchronous Data Transfer Cable Impedance (4.2.3)" for fast synchronous data transfer cable impedance requirements. (See "3.4.0.1 Single-Ended vs Differential" on page 5.) (See "3.8 Fast Synchronous Transfer Option, 4.8." on page 5.)

3.2.0.2 Conductor Size

Terminator power is 28 AWG minimum. Other signals may be smaller.

3.2.0.3 Stub Lengths (4.2.1, 4.2.2)

Single-ended stub lengths are 0.1 m maximum; differential stub lengths are 0.2 m maximum. (See "3.4.0.1 Single--Ended vs Differential" on page 5.)

3.2.0.4 Stub Clustering (4.2.1)

Single-ended stub spacing 0.3 m or greater; differential stub spacing is not specified. (See "3.4.0.1 Single-Ended vs Differential" on page 5.)

3.2.0.5 Fast Synchronous Data Transfer Cable Impedance (4.2.3)

See "3.8 Fast Synchronous Transfer Option, 4.8." on page 5 to determine whether these cable requirements apply. (See "3.4.0.1 Single-Ended vs Differential" on page 5.)

3.3 Connector Requirements, 4.3.

3.3.0.1 Shielded vs Non-Shielded Connectors

This selection affects the connector form chosen. The choices are:

- Non-Shielded Alternative 1 - A Cable
- Non-Shielded Alternative 2 - A Cable
- Non-Shielded - B Cable

  This is for either a B-Cable or a SCSI-3 P-Cable. The pin out arrangement is different for a B-cable than for a P-cable.
- Shielded Alternative 1 - A Cable
- Shielded Alternative 2 - A Cable
- Shielded - B Cable

This is for either a B-Cable or a SCSI-3 P-Cable. The pin out arrangement is different for a B-cable than for a P-cable.

3.4 Electrical Description. 4.4.

3.4.0.1 Single-Ended vs Differential

The selection made here is critical to all other cable and connector issues.

3.4.0.2 Single-Ended Termination Alternatives (4.4.1)

Active termination is recommended.

3.4.0.3 Terminator Power (4.4.3)

Each SCSI device identified as a principal initiator shall supply terminator power. A principal initiator is a SCSI device whose role is mainly to initiate operations in target devices.

3.5 SCSI Bus, 4.5.

No options identified in this section.

3.6 SCSI Signals, 4.6.

See "3.1 Physical Description. 4.1." on page 3 to determine signals which must be implemented.

3.6.0.1 Reset Signal (RST)

Each use of the Reset signal must be identified. (See "4.1.0.8 Wide Data Transfer (5.1.5.3)" on page 6.)

3.6.0.2 Driving Signals (4.6.1, 4.6.2)

A decision must be made on those signals which may or may not be or-tied for each implementation.

3.7 SCSI Bus Timing, 4.7.

No options identified in this section.

3.8 Fast Synchronous Transfer Option, 4.8.

This option is normally selected.

(See "3.1.0.4 Driver/Receiver Options" on page 3.)

Selection of this option affects cable selection, driver/receiver selection and bus timing selections.

All devices shall implement asynchronous information transfer and may negotiate synchronous data transfer rates (either fast or normal). The message system is affected by this selection.

SCSI-2 Section 5, Logical Characteristics

4.1 SCSI Bus Phases, 5.1.

4.1.0.1 Unexpected Disconnect

This target method for relinquishing the SCSI bus is not recommended, but there are some instances where it can be useful. Each instance of use must be documented since it affects initiator management of I/O processes.

(See "------ ----- ----- ----- -----" on page ---.)
(See "4.1.0.9 MESSAGE OUT Phase (5.1.9.2)".)

4.1.0.2 ARBITRATION Phase

A device in target role can be totally passive relative to the SCSI bus. If the device does not disconnect then it does not participate in the ARBITRATION phase. If disconnection is utilized, this phase must be implemented. (See "4.7.0.3 DISCONNECT (In)" on page 12.)

4.1.0.3 RESELECTION Phase

If disconnection is utilized, this phase must be implemented. (See "4.7.0.3 DISCONNECT (In)" on page 12.)

Initiators should not raise ATN during this phase. See "4.6.3 IDENTIFY (Out)" on page 9.

4.1.0.4 Selection Time-out Procedure

There are two options available. Normally option 2 is selected.

4.1.0.5 Reselection Time-out Procedure

There are two options available. Normally option 2 is selected.

4.1.0.6 Information Transfer Phases (5.1.5)

For initiators, the exact procedure for reporting parity errors must be specified for each phase and each condition.

For targets, the exact procedure for recovery from parity errors must be specified for each phase for each condition.

4.1.0.7 Synchronous Data Transfer (5.1.5.2)

This option is normally selected.

(See "3.8 Fast Synchronous Transfer Option, 4.8." on page 5.) (See "------ ----- ----- ----- -----" on page ---.)

4.1.0.8 Wide Data Transfer (5.1.5.3)

This option is required if the B-cable or the SCSI-3 P-cable is selected.

(See "3.1.0.3 68-Conductor P Cable" on page 3.) (See "3.1.0.1 68-Conductor B Cable" on page 3.)

(See "------ ----- ----- ----- -----" on page --- and "------ ----- ----- ----- -----" on page ---.)

One target recovery technique is Unexpected disconnect. (See "4.1.0.1 Unexpected Disconnect" on page 6.)

One initiator recovery technique is sending an INITIATOR DETECTED ERROR message.

One initiator recovery technique is asserting the reset signal. For multiple initiator environments, this option should be used sparingly.

4.1.0.9 MESSAGE OUT Phase (5.1.9.2)

The exact procedure for recovery from parity errors must be specified for this phase. A retry procedure is defined, but seldom implemented by initiators. The procedure is not dynamically selectable.

There are two additional alternatives at logical level:

- Return CHECK CONDITION Status with the sense data sense key set to ABORTED COMMAND and the additional sense set to MESSAGE ERROR.
- Perform an unexpected disconnect. (See "4.1.0.1 Unexpected Disconnect" on page 6.)

The exact procedure for every message must be specified; there may be different selections for different messages or it may turn out all the same.

4.1.0.10 Acting on Messages

A target may act on messages one at a time or as a group. Certain messages require that the target enter the BUS FREE phase following receipt, acceptance and execution of certain messages. This is not interpreted as an unexpected disconnect by initiators. (See "4.1.0.1 Unexpected Disconnect" on page 6.)

The exact procedure for every message must be specified: there may be different selections for different messages or it may turn out all the same.

4.2 SCSI Bus Conditions, 5.2.

4.2.0.1 Attention Condition (5.2.1)

There are six procedures to be specified for this condition which vary according to the phase and may be different for each phase.

- COMMAND phase
- DATA Phase
- STATUS Phase
- MESSAGE IN Phase
- MESSAGE OUT Phase

  (See "4.1.0.9 MESSAGE OUT Phase (5.1.9.2)" on page 7.)
- RESELECTION Phase

See "4.6.3 IDENTIFY (Out)" and "4.1.0.3 RESELECTION Phase" on page 6.

4.2.0.2 Reset Condition (5.2.2)

This condition is detected by all SCSI devices on the SCSI bus.

The ability to assert the RST signal must be stated.

If the RST signal is asserted, the conditions under which it is asserted must be defined. A SCSI device which is a pure initiator has no mechanism to declare its reset implementation.

The implementation option used after detecting the reset event is reflected in the INQUIRY command data.

For devices detecting the RST signal, there are two implementation options:

- Hard Reset Alternative

  This is a more likely selection in single initiator systems. Note that the SCSI bus type can change with each power on cycle for each device (i.e., single initiator now; multiple initiator later), or even while powered on by an initiator powering on late.

- Soft Reset Alternative

  This is probably most useful in Multiple Initiator Systems.

  The procedure following a COMMAND COMPLETE message transfer must be specified.

  The procedure following a SAVE DATA POINTER message transfer must be specified.

The Hard Reset and Soft Reset alternatives are mutually exclusive within a SCSI bus (system). All devices on a single SCSI bus must implement the same option. See "------ ----- ----- ----- -----" on page --- for the SftRe bit description.

The ability to respond to certain commands following a Hard Reset action must be specified, especially the time delay to respond.

4.3 SCSI Bus Phase Sequences, 5.3.

No options identified in this section.

4.4 SCSI Pointers, 5.4.

No options identified in this section.

4.5 Message System Description, 5.5.

The message system is probably the "ugliest" part of SCSI to implement. The number of variations possible is enormous since no minimum working set is defined. This area causes the most attachment difficulty between SCSI devices.

4.5.0.1 SCSI-1 Provisions

Several protocols changed from SCSI-1 involve the message system. If SCSI-1 provisions are to be implemented. they must be fully specified and must not conflict with normal SCSI-2 operation.

See "2.2.1 Provision to support SCSI-1 Device Attachments" on page 2. See "------ ----- ----- ----- -----" on page ---.

4.6 Mandatory Target Messages

4.6.1 ABORT (Out)

4.6.1.1 Recovery Procedure

The exact actions taken by the target must be specified to prevent unknown position resulting in the initiator. Clear specification under all situations is required.

4.6.2 BUS DEVICE RESET (Out)

4.6.2.1 Recovery Procedure

The exact state of the device must be defined under all conditions after execution of this message.

4.6.3 IDENTIFY (Out)

4.6.3.1 DiscPriv Field

This bit is normally set by all initiators for every I/O Process. For tagged queued I/O Processes, the action to be taken when it is NOT set to 1 must be specified.

4.6.3.2 LUNTAR Field

Support for LUNTAR=1 is optional in all of the SCSI system. The procedure to handle the condition when not supported, but the bit is set. must be specified. (See "5.5 Command Processing Exceptions" on page 21.)

4.6.3.3 LUNTRN Field

(See "5.5 Command Processing Exceptions" on page 21.)

4.6.3.4 Recovery Procedure

The Reselection sequence has a hole in it concerning immediate execution of messages as received. When a target reselects for a nexus, it must transmit the IDENTIFY message and optionally a queue tag message to properly identify a nexus. If the IDENTIFY message is edited on the fly, the procedure for MESSAGE IN Phase requires the initiator to raise the ATN signal on the exact message. The target may not get the nexus fully identified.

A similar situation occurs if ATN is raised during RESELECTION Phase since the target must go to MESSAGE IN Phase to identify the nexus. If ATN is active. the target may respond to the ATN signal before fully identifying the nexus.

The exact procedure for both conditions must be specified.

4.6.4 INITIATOR DETECTED ERROR (Out)

4.6.4.1 Recovery Procedure

The precise recovery procedure must be specified for INITIATOR DETECTED ERROR following each phase. See "4.2.0.1 Attention Condition (5.2.1)" on page 7.

4.6.5 MESSAGE REJECT (In)

4.6.5.1 Usage Procedures

The exact conditions under which this message is transferred must be specified.

4.6.6 INITIATOR DETECTED ERROR (Out)

4.6.6.1 Usage Procedure

Each instance of use of this message must be specified. Target actions must be checked thoroughly since the target is given wide latitude for its next action.

4.6.7 MESSAGE REJECT (Out)

4.6.7.1 Recovery Procedure

This is perhaps the most difficult message to deal with since it involves a dialog which executes only in the message system. The procedures must be clearly specified for handling this message following transmission of each target implemented message to the initiator.

4.7 Optional Target Messages

- ABORT TAG (Out)

  See "4.7.0.1 ABORT TAG (Out)" on page 12.

- CLEAR QUEUE (Out)

  See "4.7.0.2 CLEAR QUEUE (Out)" on page 12.

- DISCONNECT (In)

See "4.7.0.3 DISCONNECT (In)".

- DISCONNECT (Out)

  See "4.7.0.4 DISCONNECT (Out)".

- IDENTIFY (In)

  See "4.7.0.5 IDENTIFY (In)".

- IGNORE WIDE RESIDUE (In)

  See "4.1.0.8 Wide Data Transfer (5.1.5.3)" on page 6.

- INITIATE RECOVERY (In)

  See "4.7.0.7 INITIATE RECOVERY (In)" on page 13.

- INITIATE RECOVERY (Out)

  See "4.7.0.7 INITIATE RECOVERY (In)" on page 13 and "5.6 Asynchronous Event Notification" on page 21.

- LINKED COMMAND COMPLETE (In)

  See "5.2.1.2 Link Bit" on page 20.

- LINKED COMMAND COMPLETE with Flag (In)

  See "5.2.1.2 Link Bit" on page 20 and "5.2.1.3 Flag Bit" on page 20.

- MODIFY DATA POINTER (In)

  See "4.7.0.11 MODIFY DATA POINTER (In)" on page 13.

- HEAD OF QUEUE TAG (Out)

  See "4.7.0.12 HEAD OF QUEUE TAG (Out)" on page 13 and "5.9 Queued I/O Processes" on page 22.

- ORDERED QUEUE TAG (Out)

  See "4.7.0.13 ORDERED QUEUE TAG (Out)" on page 13 and "5.9 Queued I/O Processes" on page 22.

- SIMPLE QUEUE TAG (Out/In)

  See "4.7.0.14 SIMPLE QUEUE TAG (Out/In)" on page 13 and "5.9 Queued I/O Processes" on page 22.

- RELEASE RECOVERY (Out)

  See "4.7.0.15 RELEASE RECOVERY (Out)" on page 14.

- RESTORE POINTERS (In)

  See "4.7.0.16 RESTORE POINTERS (In)" on page 14.

- SAVE DATA POINTER (In)

  See "4.7.0.17 SAVE DATA POINTER (In)" on page 14.

- SYNCHRONOUS DATA TRANSFER REQUEST (Out/In)

  See "4.7.0.18 SYNCHRONOUS DATA TRANSFER REQUEST (Out/In)" on page 14.

- TERMINATE I/O PROCESS (Out)

  See "4.7.0.19 TERMINATE I/O PROCESS (Out)" on page 14.

- WIDE DATA TRANSFER REQUEST (Out/In)

  See "4.7.0.20 WIDE DATA TRANSFER REQUEST (Out/In)" on page 14.

4.7.0.1 ABORT TAG (Out)

If tagged queuing is implemented, this message must be implemented.

This message may be implemented when tagged queuing is not implemented; it is recommended.

See the comments in "4.6.1 ABORT (Out)" on page 9.

See "5.9 Queued I/O Processes" on page 22.

4.7.0.2 CLEAR QUEUE (Out)

If tagged queuing is implemented, this message must be implemented. This message requires careful coordination in multiple initiator systems. Its use should be restricted.

This message may be implemented when tagged queuing is not implemented; it is recommended.

See the comments in "4.6.1 ABORT (Out)" on page 9.

See "5.9 Queued I/O Processes" on page 22.

4.7.0.3 DISCONNECT (In)

If disconnection is used in a target, this message must be implemented. Disconnection is almost always implemented in SCSI-2 SCSI devices.

See the rules for use with the SAVE DATA POINTER message.

This message is normally implemented since it is efficient to release the bus during certain target operations.

4.7.0.4 DISCONNECT (Out)

Even if target disconnection is used, this message need not be implemented.

This message is recommended for implementation to reach a broader market.

4.7.0.5 IDENTIFY (In)

If disconnection is used in a target, this message must be implemented. Disconnection is almost always implemented in SCSI-2 devices.

See "4.6.3 IDENTIFY (Out)" on page 9, "4.7.0.3 DISCONNECT (In)" on page 12, and "4.7.0.4 DISCONNECT (Out)" on page 12.

4.7.0.6 IGNORE WIDE RESIDUE (In)

This asymmetric function permits targets to inform an initiator that an odd byte transfer has taken place on input.

For output, the assumption is that the transfers will all be even multiples of the current bus width (not true and a bug in SCSI-2) when the TERMINATE I/O PROCESS message is used. for normal output, the block length is specified by the initiator in either a MODE SELECT command or in the WRITE command.

This messages is recommended for implementation for SCSI devices which implement wide data transfer (see "4.1.0.8 Wide Data Transfer (5.1.5.3)" on page 6).

Further, it is recommended that targets implement this message as a Vendor Unique message in the outbound direction to cover the hole in SCSI-2 mentioned above. SCSI-3 proposed to remedy this omission.

See "------ ----- ----- ----- -----" on page --- also.

4.7.0.7 INITIATE RECOVERY (In)

Implemented only if extensive recovery operations are required that involve the initiator (e.g., buffered mode data recovery procedure). ;p Implement only if ECA is implemented in the SCSI device.

See "5.8 Extended Contingent Allegiance" on page 22 and "4.7.0.8 INITIATE RECOVERY (Out)" on page 13.

4.7.0.8 INITIATE RECOVERY (Out)

Implemented only if ECA is used.

See "4.7.0.7 INITIATE RECOVERY (In)" on page 13, "5.8 Extended Contingent Allegiance" on page 22, and "5.6 Asynchronous Event Notification" on page 21.

4.7.0.9 LINKED COMMAND COMPLETE (In)

This is normally implemented. See "5.2.1.2 Link Bit" on page 20.

4.7.0.10 LINKED COMMAND COMPLETE with Flag (In)

This is normally implemented. See "5.2.1.2 Link Bit" on page 20 and "5.2.1.3 Flag Bit" on page 20.

4.7.0.11 MODIFY DATA POINTER (In)

This message is usually not implemented by initiators. Its use is becoming more widely used. It is very useful in recovery operations that also involve one or more disconnects.

If implemented, it must be expected to be rejected by some initiators so a fall back position must be adopted, if or when rejected.

4.7.0.12 HEAD OF QUEUE TAG (Out)

If tagged queuing is implemented, this message must be implemented.

Tape drives usually have no queuing requirements to a single volume. However, with queueing being so popular, allowing initiators to use the messages for nexus identification is considered worthwhile. The queue depth can be controlled to prevent sequencing problems.

See "5.9 Queued I/O Processes" on page 22. See "------ ----- ----- ----- -----" on page ---.

4.7.0.13 ORDERED QUEUE TAG (Out)

If tagged queuing is implemented, this message must be implemented.

See "5.9 Queued I/O Processes" on page 22 and "4.7.0.12 HEAD OF QUEUE TAG (Out)" on page 13.

4.7.0.14 SIMPLE QUEUE TAG (Out/In)

If tagged queuing is implemented, this message must be implemented.

See "5.9 Queued I/O Processes" on page 22 and "4.7.0.12 HEAD OF QUEUE TAG (Out)" on page 13.

4.7.0.15 RELEASE RECOVERY (Out)

Implemented only if ECA is implemented and active. See "5.8 Extended Contingent Allegiance" on page 22 and "4.7.0.7 INITIATE RECOVERY (In)" on page 13.

4.7.0.16 RESTORE POINTERS (In)

This message is useful in recovery operations. It is less subtle than the MODIFY DATA POINTER message, but is usually more readily implemented by initiators. It is only good for recovery at the most recent disconnection or SAVE DATA POINTER message boundary.

This message should be implemented and every attempt made to permit partial block recovery in the tape devices. This is a more normal operating mode for initiators vs the MODIFY DATA POINTER message.

4.7.0.17 SAVE DATA POINTER (In)

If disconnection is used in a target during data transfer phase only, this message must be implemented.

This message implementation is recommended and almost mandatory.

4.7.0.18 SYNCHRONOUS DATA TRANSFER REQUEST (Out/In)

If synchronous or fast synchronous data transfer is implemented, this message must be implemented to activate it.

Implementation is recommended to achieve effective data rate goals.

See "4.1.0.7 Synchronous Data Transfer (5.1.5.2)" on page 6.

4.7.0.19 TERMINATE I/O PROCESS (Out)

For tape devices, this can be very useful to process writes of blocks of unknown lengths from the initiators point of view. This message causes the I/O process to terminate normally rather than abort.

Implementation is recommended. This message has analogies in IBM S/370 channel and IPI which are used today. Initiator support at present is limited, but it makes the product acceptable to a broader market.

4.7.0.20 WIDE DATA TRANSFER REQUEST (Out/In)

If wide data transfer is implemented, this message must be implemented to activate it.

Implementation is required for B-cable and P-Cable implementations. To achieve effective data rate goals. it must be implemented in these instances.

See "4.1.0.8 Wide Data Transfer (5.1.5.3)" on page 6. See "------ ----- ----- ----- -----" on page --- to determine extent of negotiation when wide data transfer is implemented.

4.8 Mandatory Initiator Messages

4.8.1 id=identa.IDENTIFY (In)

4.8.1.1 Recovery Procedure

It is recommended that initiators perform delayed message execution during a reselection sequence and that the ATN signal not be raised during RESELECTION Phase. See "------ ----- ----- ----- -----" on page ---.

4.9 Optional Initiator Messages

- ABORT (Out)

  See "4.9.0.1 ABORT (Out)".

- ABORT TAG (Out)

  See "4.9.0.2 ABORT TAG (Out)".

- BUS DEVICE RESET (Out)

  See "4.9.0.3 BUS DEVICE RESET (Out)".

- CLEAR QUEUE (Out)

  See "4.9.0.4 CLEAR QUEUE (Out)" on page 17.

- DISCONNECT (In)

  See "4.9.0.5 DISCONNECT (In)" on page 17.

- DISCONNECT (Out)

  See "4.9.0.6 DISCONNECT (Out)" on page 17.

- IGNORE WIDE RESIDUE (In)

  See "4.9.0.7 IGNORE WIDE RESIDUE (In)" on page 17.

- INITIATE RECOVERY (In)

  See "4.9.0.8 INITIATE RECOVERY (In)" on page 17.

- INITIATE RECOVERY (Out)

See "4.9.0.9 INITIATE RECOVERY (Out)".

- LINKED COMMAND COMPLETE (In)

  See "4.9.0.10 LINKED COMMAND COMPLETE (In)" on page 18.

- LINKED COMMAND COMPLETE with Flag (Out)

  See "4.9.0.11 LINKED COMMAND COMPLETE with Flag (In)" on page 18.

- MODIFY DATA POINTER (In)

  See "4.9.0.12 MODIFY DATA POINTER (In)" on page 18.

- HEAD OF QUEUE TAG (Out)

  See "4.9.0.13 HEAD OF QUEUE TAG (Out)" on page 18.

- ORDERED QUEUE TAG (Out)

  See "4.9.0.14 ORDERED QUEUE TAG (Out)" on page 18.

- SIMPLE QUEUE TAG (Out/In)

  See "4.9.0.15 SIMPLE QUEUE TAG (Out/In)" on page 18.

- RELEASE RECOVERY (Out)

  See "4.9.0.16 RELEASE RECOVERY (Out)" on page 18.

- RESTORE POINTERS (In)

  See "4.9.0.17 RESTORE POINTERS (In)" on page 19.

- SAVE DATA POINTER (In)

  See "4.9.0.18 SAVE DATA POINTER (In)" on page 19.

- SYNCHRONOUS DATA TRANSFER REQUEST (Out/In)

  See "4.9.0.19 SYNCHRONOUS DATA TRANSFER REQUEST (Out/In)" on page 19.

- TERMINATE I/O PROCESS (Out)

  See "4.9.0.20 TERMINATE I/O PROCESS (Out)" on page 19.

- WIDE DATA TRANSFER REQUEST (Out/In)

  See "4.9.0.21 WIDE DATA TRANSFER REQUEST (Out/In)" on page 19.

4.9.0.1 ABORT (Out)

This message is almost always implemented. It is used to terminate an I/O process when no status is required. Contrast with the TERMINATE I/O PROCESS message ("4.9.0.20 TERMINATE I/O PROCESS (Out)" on page 19).

Abort should be used carefully since no status is returned from the device following execution of the message. The exact logical or physical of a tape device is indeterminate. This is OK for a disk, but can be disastrous for a sequential device.

4.9.0.2 ABORT TAG (Out)

If tagged queuing is implemented, this message must be implemented.

This message may be implemented when tagged queuing is not implemented; it is recommended.

See "5.9 Queued I/O Processes" on page 22. See "4.9.0.1 ABORT (Out)" on page 16.

4.9.0.3 BUS DEVICE RESET (Out)

This message is almost always implemented. With the Soft Reset option for the Reset Condition. it is the only way to reset a wayward device.

Its use should be restricted to catastrophic situations.

4.9.0.4 CLEAR QUEUE (Out)

If tagged queuing is implemented, this message must be implemented.

This message may be implemented when tagged queuing is not implemented; it is recommended.

See "5.9 Queued I/O Processes" on page 22. See "4.9.0.1 ABORT (Out)" on page 16.

4.9.0.5 DISCONNECT (In)

This is a normal message from target devices and must be implemented. Disconnection is almost always implemented in SCSI-2 devices.

Implementation recommended.

4.9.0.6 DISCONNECT (Out)

This is a rare initiator message. It is used only to force a target device off the bus. Given the response rules available to target devices, it may not have the desired effect.

This is really a micro-management message. It is supposed to achieve better elapsed times for I/O processes which have some unspecified higher priority in the initiator. The target's reaction may not cause the desired results. Most targets will reject the message.

4.9.0.7 IGNORE WIDE RESIDUE (In)

This is not a popular initiator message. It is useful only for stream devices where the transfer length is suddenly and uncontrollably shortened to an odd length that is not a multiple of the current bus width. For tape reads. this can be important.

See "4.7.0.6 IGNORE WIDE RESIDUE (In)" on page 12 for a discussion of SCSI-2 on outbound transfers. This message, outbound, is proposed for SCSI-3 but its use would be vendor unique in a SCSI-2 implementation and would normally be rejected.

4.9.0.8 INITIATE RECOVERY (In)

Implemented only if extensive recovery operations are required that involve the initiator (e.g., buffered tape data recovery procedure).

This message requires coordination with each device manufacturer since the exact recovery procedure is vendor unique.

Implementation with the default response of MESSAGE REJECT built low into the protocol is not recommended. The optional response of an immediate RELEASE RECOVERY message is recommended from a higher level in the initiator protocol so that adaptations can be made if the function needs to be activated.

Implementation is recommended.

See "5.8 Extended Contingent Allegiance" on page 22.

4.9.0.9 INITIATE RECOVERY (Out)

Implemented only if ECA is active or used in any AEN operation.

Implementation is recommended.

See "4.7.0.7 INITIATE RECOVERY (In)" on page 13, "5.8 Extended Contingent Allegiance" on page 22, and "5.6 Asynchronous Event Notification" on page 21.

4.9.0.10 LINKED COMMAND COMPLETE (In)

This is normally implemented. See "5.2.1.2 Link Bit" on page 20.

Linking commands provides a way to extend an I/O process beyond one command. It has its place in multiple initiator systems where the RESERVE and RELEASE commands are not used.

4.9.0.11 LINKED COMMAND COMPLETE with Flag (In)

This is normally implemented. See "5.2.1.2 Link Bit" on page 20 and "5.2.1.3 Flag Bit" on page 20. See also "4.9.0.10 LINKED COMMAND COMPLETE (In)" on page 18.

4.9.0.12 MODIFY DATA POINTER (In)

This message is usually not implemented by initiators. Its use is becoming more widely used. It is very useful in recovery operations that also involve one or more disconnects and SAVE DATA POINTER messages.

This is normally implemented. The worst that can happen is that the initiator responds with a MESSAGE REJECT message.

Implement this message to support device recovery across multiple disconnects. This may prevent terminating user tasks since interface recovery is more robust.

4.9.0.13 HEAD OF QUEUE TAG (Out)

If tagged queuing is implemented, this message must be implemented.

See "5.9 Queued I/O Processes" on page 22. See "------ ----- ----- ----- -----" on page ---.

Target tolerance is more important (see "------ ----- ----- ----- -----" on page ---).

4.9.0.14 ORDERED QUEUE TAG (Out)

If tagged queuing is implemented, this message must be implemented.

See "5.9 Queued I/O Processes" on page 22. See "4.9.0.13 HEAD OF QUEUE TAG (Out)" on page 18.

4.9.0.15 SIMPLE QUEUE TAG (Out/In)

If tagged queuing is implemented, this message must be implemented.

See "5.9 Queued I/O Processes" on page 22. See "4.9.0.13 HEAD OF QUEUE TAG (Out)" on page 18.

4.9.0.16 RELEASE RECOVERY (Out)

Implemented only if ECA is implemented and active. See "5.8 Extended Contingent Allegiance" on page 22. See "4.9.0.8 INITIATE RECOVERY (In)" on page 17.

Most initiators do not activate ECA in targets which support it.

4.9.0.17 RESTORE POINTERS (In)

This is normally implemented. Use of this message is part of a target's normal repertoire of tools to support recovery at the device.

4.9.0.18 SAVE DATA POINTER (In)

If disconnection is used in a target during data transfer phase only, this message must be implemented.

This is normally implemented; practically mandatory.

4.9.0.19 SYNCHRONOUS DATA TRANSFER REQUEST (Out/In)

If synchronous or fast synchronous data transfer is implemented, this message must be implemented to activate it.

Almost mandatory to achieve effective data rate goals.

See "4.1.0.7 Synchronous Data Transfer (5.1.5.2)" on page 6.

4.9.0.20 TERMINATE I/O PROCESS (Out)

For tape devices, this can be very useful to process writes of blocks of unknown lengths from the initiators point of view. This message causes the I/O process to terminate normally rather than abort.

Most initiators do not implement this message.

This is a good message to gently stop an I/O process in a target. Target implementation is recommended and also for initiators.

4.9.0.21 WIDE DATA TRANSFER REQUEST (Out/In)

If wide data transfer is implemented, this message must be implemented to activate it.

Almost mandatory to achieve effective data rate goals.

See "4.1.0.8 Wide Data Transfer (5.1.5.3)" on page 6.

SCSI-2 Section 6, Commands and Status

5.1 Command Implementation Requirements

5.1.0.1 Reserved

The type of checking for reserved bits in all messages and commands must be specified. Since some SCSI-3 function may add definition to fields reserved in SCSI-2, the actions must be specified for EACH case in the appropriate section (e.g., each command description).

5.2 Command Descriptor Block

5.2.0.1 Operation Type Codes

The operation codes for optional. and vendor unique commands must be identified for each implemented command. The command description for each vendor unique command (not recommended) must be thorough, especially as it regards sense data responses and command length. Initiators only have product documantation as a guide if they implement the command.

The minimum length for vendor unique commands is 3 bytes (operation code, LUN field byte, and control byte).

5.2.0.2 Logical Unit Number

No use should be made of this field. SCSI-3 intends to reclaim the field in favor of the preferred IDENTIFY message. The LUN field will be 5-bits in SCSI-3 in the IDENTIFY message.

5.2.1 Control Field

5.2.1.1 Vendor Specific Field

No use should be made of this field.

5.2.1.2 Link Bit

Implementation recommended in target devices; optional in initiator devices.

See "------ ----- ----- ----- -----" on page ---.

5.2.1.3 Flag Bit

Implementation recommended in target devices; optional in initiator devices.

5.3 Status

There are no options in this section. Proper use of the status codes promote compatibility with other interfaces. especially BUSY, QUEUE FULL, and RESERVATION CONFLICT.

The recommended priority of status reporting is:

- Reservation Conflict
- Busy
- Queue Full
- Check Condition
- Command Terminated
- Intermediate or Intermediate-Condition Met
- Good or Condition Met

See "------ ----- ----- ----- -----" on page --- status values returned.

5.4 Command Examples

There are no options in this section.

5.5 Command Processing Exceptions

5.5.0.1 Programmable Operating Definition

Implementation not recommended unless special vendor unique functions are to be activated considerably outside the scope of the SCSI-2 standard. Functions of special value-add significance can hide behind this function.

See "------ ----- ----- ----- -----" on page ---.

5.5.0.2 Parameter Rounding

All use of this option must be fully specified for when and what happens in each instance.

5.6 Asynchronous Event Notification

5.6.0.1 Usage Procedure

Implementation recommended. It must be activated by an initiator, but it has an analog in the deferred unit check of OEMI. It also can help manage SCSI systems where the tape devices power on at a time later than the system or intermittently.

AEN use must be clearly specified for the conditions of use. A fall back position to REQUEST SENSE command deferred error must be implemented also.

5.7 Contingent Allegiance

5.7.0.1 Usage Procedure

This is not optional. The name has a direct equivalent meaning in almost all situations to the OEMI equivalent function.

5.8 Extended Contingent Allegiance

5.8.0.1 Usage Procedure

This is optional. The name has a direct equivalent meaning in almost all situations to the OEMI equivalent function.

5.9 Queued I/O Processes

5.9.0.1 Usage Procedure

Implementation recommended to extend scope of OEM attachment with less vendor specific code loads. The queue depth need only be one as it normally is for tape, but this use of queuing is very compatible with new SCSI-2 initiators.

5.9.0.2 Untagged I/O Processes

Required implementation.

5.9.0.3 Tagged I/O Processes

Highly recommended as noted above.

With a queue depth of one for each logical unit. questions of execution sequence are avoided when the various types of I/O processes are used.

5.10 Unit Attention

5.10.0.1 Usage Procedure

All uses of Unit Attention beyond those required must be fully documented.

Intentionally Left Blank

END OF DOCUMENT