X3T9.2/87-46

Date: 15 March 1987

To: ANSI X3T9.2 Working Group

From: Paul R. Nitza
Emulex Corporation
(714) 662-5600

Subj: SCSI Command Stacking/Queuing and Auto Sense Proposals

Attached to this letter are 3 different proposals for inclusion in the SCSI-2 draft specification. Although somewhat interrelated I would like these proposals to be considered separately. The enclosure contains the following proposals:

Auto Sense and Mode Sense/Select Target Options Page
Queued Commands
Stacked Commands

Some of the information contained in the Command Queuing and Stacking proposals was taken from a document (X3T9.2/86-82) submitted by Jerry Houlder of Magnetic Peripherals and I thank him for referring me to that document and for the use of his material.

I am submitting these proposals to the working group for discussion and revision before consideration by the full plenary committee.

191

CC Bill Caldwell
Wendy Isbell
Devon Worrell
Kim Le

Auto Sense and Mode Select/Sense Target Options Page Proposal

6.3.1 Auto Sense on Check Condition Option

The Auto Sense on Check Condition option, when enabled, allows the Target to automatically return Sense data when a Check Condition occurs. This option can be enabled/disabled using the Auto Sense bit in the Target Options Mode Select page. When disabled, the Target terminates a command normally, even when a Check Condition occurs, by sending the Status byte followed by the Command Complete message. The Initiator should issue a Request Sense command to receive the Sense Data for any error that occurs. When the Auto Sense option is enabled and an error occurs during a command, the Target will send the Check Condition Status byte, change the phase to Data In, send the Sense Data then complete the command by sending a Command Complete message (status phase, data in phase, message in phase). The Target will send all of the available Sense Data or up to the number of bytes specified in the Mode Select Target Options page, whichever is less.

8.1.7.4 Mode Select/Sense Target Options Page

The Target Options page defines various options which may be supported by the Target device. The Target may support only one set of these parameters for all Initiators or it may support a set of parameters for each Initiator on the bus.

Bit / Byte	7	6	5	4	3	2	1	0
0	Rsvd	Rsvd	Page Code (??h)					
1	Parameter Length (04h)							
2	Cache Control							
3	Command Stacking Control							
4	Auto Sense Control							
5	Auto Sense Allocation Length							

The Cache Control field allows the Initiator to enable/disable caching for any logical unit and allows the Target to report which logical units support caching. This field is organized as a bit mask with bit 0 corresponding to logical unit 0 and bit 7 corresponding to logical unit 7. In the Mode Select command these bits enable/disable caching for the respective logical unit. If the bit is set to one, caching is enabled for that

logical unit. If the bit is set to zero, caching is disabled for that logical unit. In the Mode Sense command Changeable Values option, this field reports whether or not the respective logical unit supports caching. If the bit is set to one, the logical unit supports caching and if the bit is zero the Target does not support caching for that logical unit.

The Command Stacking Control field allows the Initiator to enable/disable command stacking (see section 6.1.6) for any logical unit and allows the Target to report which logical units support command stacking. This field is organized as a bit mask with bit 0 corresponding to logical unit 0 and bit 7 corresponding to logical unit 7. In the Mode Select command these bits enable/disable command stacking for the respective logical unit. If the bit is set to one, command stacking is enabled for that logical unit. If the bit is set to zero, command stacking is disabled for that logical unit. In the Mode Sense command Changeable Values option, this field reports whether or not the respective logical unit supports command stacking. If the bit is set to one the logical unit supports command stacking and if the bit is zero the Target does not support command stacking for that logical unit. When the Command Stacking option is disabled the Target will not send a Command Tag message to the Initiator.

The Auto Sense Control field allows the Initiator to enable/disable the Auto Sense option (see section 6.3.1) for any logical unit and allows the Target to report which logical units support the Auto Sense option. This field is organized as a bit mask with bit 0 corresponding to logical unit 0 and bit 7 corresponding to logical unit 7. In the Mode Select command these bits enable/disable the Auto Sense option for the respective logical unit. If the bit is set to one, Auto Sense on Check Condition is enabled for that logical unit. If the bit is set to zero, Auto Sense is disabled for that logical unit. In the Mode Sense command Changeable Values option, this field reports whether or not the respective logical unit supports the Auto Sense option. If the bit is set to one the logical unit supports the Auto Sense option and if the bit is zero the Target does not support the Auto Sense option for that logical unit. When the Auto Sense option is disabled the Command Stacking option is automatically disabled if it is supported by the Target.

The Auto Sense Data Allocation Length field specifies the maximum number of Auto Sense Data bytes (see section 6.3.1) that the Target can return. If the Initiator sets this field to zero the Target will set the Auto Sense Data Allocation Length to its default value. Any other value (1 through 255) in this field specifies the maximum number of bytes the Target is to return during the Auto Sense sequence. The Target will return all available Sense Data or up to the number of bytes specified in this field, whichever is less. Note: This field does not affect the number of bytes returned during a normal Request Sense command.

6.1.4 Non-Queued/Stacked Command Handling

Targets that do not implement command queuing or stacking shall not accept another command for the same logical unit, from the same or different Initiators, when it is already processing a command for that logical unit. If the same or a different Initiator sends a command for a logical unit, while the Target is processing a command for that logical unit, the Target will return a Busy status to the Initiator. The Initiator that receives the Busy status will have to resend the command at a later time in order for the command to be executed.

Implementors Note: The Target may receive the Identify message and/or Command Descriptor Block before returning the Busy status.

If the Target implements the Soft Reset option and a reset occurs while it is processing a command and the same Initiator sends another command to the same logical unit, the Target shall clear the original command and perform the new command (see section 5.2.2.2 number 3). Note: This condition only occurs after a reset is detected by the Target.

6.1.5 Command Queuing Option

Command Queuing is the ability of the Target to accept multiple commands from different Initiators for the same logical unit. The Target shall process all commands from different initiators in the order they are received unless two or more of the same type of commands are received (2 or more reads, or 2 or more writes, etc) without any intervening commands of a different type. When the Target has received two or more of the same type of commands it may rearrange the commands to perform device access optimization. If a command has the link bit set it is considered a different type no matter what type of command precedes or follows it. As an example, if 2 read commands followed by a test unit ready were received by the Target, the 2 read commands could be processed in any order as long as both are executed before the test unit ready command is processed. However, if a read followed by a test unit ready followed by another read was received, the Target could not rearrange the read commands and shall process them in the order they were received.

For commands from a given Initiator to be queued the Initiator shall support Arbitration, Reselection, send an Identify message after Selection, support disconnects and shall not reject the first disconnect message sent by the Target. If any one of these conditions are not true the Target cannot queue the command from that Initiator and shall respond as defined in section 6.1.4.

The number of commands that may be queued for any one logical unit is a Target dependent parameter. The Target may choose to implement full command queuing (i.e. one command from each Initiator for a total of seven commands) or limited queuing (less than seven commands are accepted for a single logical unit). The Target may even implement limited queuing on one logical unit and no or full queuing on another logical unit or any other combination it so desires. If the Target implements limited queuing and the command queue is full when another command is received, the Target shall respond to this new command as defined in section 6.1.4.

If a Target implements both command queuing and linked commands (see section 6.2.6), the Target shall complete execution of all linked commands from a given Initiator before it begins execution of the next command in the queue. New commands from different Initiators may still be added to the command queue while the linked command is being processed.

If the Soft Reset option is implemented the Target will not clear the command queue when a reset occurs. The Target will attempt to complete any queued commands as defined in section 5.2.2.2.

If the Hard Reset option is implemented the Target will clear the command queue when a reset occurs. The Initiator will detect this condition when it receives a Unit Attention on the next command sent to the Target. When this condition occurs the Initiator shall assume any outstanding queued commands have not been executed and must be resent to the Target.

When a Bus Device Reset message is received the Target will clear the command queue of all outstanding commands. The Initiator(s) will detect this condition when it receives a Unit Attention on the next command sent to the Target. When this condition occurs the Initiator(s) shall assume any outstanding queued commands have not been executed and must be resent to the Target.

Implementors Note: The Initiator must have a command time out to detect when a Bus Device Reset message was sent to a Target from another Initiator.

6.1.6 Command Stacking Option

Command Stacking is the ability of the Target to accept multiple commands from the same Initiator for the same logical unit. If the Target supports the Command Stacking Option it shall also support the Auto Sense on Check Condition option (see para 6.3.1). If the Auto Sense option is disabled by the Initiator the Target shall also disable command stacking.

Multiple commands stacked from a single Initiator shall not take precedence over commands received from another Initiator for the same logical unit. The Target shall process all commands to a logical unit, from the same or different Initiators, in the order in which they are received unless two or more of the same type of commands are received (2 reads, or 2 writes, etc) without any intervening commands. When the Target has received two or more of the same type of commands, from the same or different Initiators, it may rearrange the commands to perform device access optimization.

For commands to be stacked the Initiator shall support Arbitration, Reselection, send an Identify message after Selection, support disconnects and shall not reject the first disconnect message sent by the Target. If any one of these is not true the Target cannot stack the command from that Initiator and shall respond as defined in section 6.1.4.

Targets that support command stacking will return a Command Tag message (see section 5.5.7) whenever a command is received from an Initiator. The Command Tag message associates an unique ID tag with each command to identify that particular command while it is being processed. The Target will send this message to the Initiator after the Selection phase has been complete but before the first disconnect. Normally the Target will send the Command Tag message after the Identify Message or Command Descriptor Block has been received. The Initiator must save this ID tag for subsequent reconnections to identify which command the Target is currently processing. The Target will again send this Command Tag message immediately after the Identify message during a reselection to identify the command it is currently processing. If the Command Tag message is rejected by the Initiator the Target shall not stack the command and will process it as defined in section 6.1.4.

Implementors Note: If the first Command Tag message after a reset is rejected by the Initiator the Target should assume that the Initiator does not support command stacking and should not try to stack commands for that Initiator.

When a Check Condition is reported for any command the Target will use the Auto Sense option (see section 6.3.1) to return the Sense Data for the command which had the Check Condition. The Target can then continue executing commands from the same or different Initiators, if any are available.

The number of commands that may be stacked for a logical unit is a Target dependent parameter. The Target may implement command stacking on one logical unit and no stacking on another logical unit or any other combination it so desires. When the command stack becomes full and another command is received from the same Initiator, the Target shall respond to this new command as defined in section 6.1.4.

If a Target implements both command stacking and linked commands (see section 6.2.6), the target shall complete execution of all linked commands before it begins execution of the next command from the same or a different Initiator. New commands from the same Initiator (or different Initiators if command queuing is implemented) may still be added to the command queue/stack while the linked command is being processed.

If the Soft Reset option is implemented the Target will not clear any stacked commands when a reset occurs. The Target will attempt to complete any stacked commands as defined in section 5.2.2.2.

If the Hard Reset option is implemented the Target will clear any stacked commands when a reset occurs. The Initiator will detect this condition when it receives a Unit Attention on the next command sent to the Target. When this condition occurs the Initiator shall assume any outstanding stacked commands have not been executed and must be resent to the Target.

When a Bus Device Reset message is received the Target will clear any stacked commands. The Initiator will detect this condition when it receives a Unit Attention on the next command sent to the Target. When this condition occurs the Initiator shall assume any outstanding stacked commands have not been executed and must be resent to the Target.

Implementors Note: The Initiator must have a command time out to detect when a Bus Device Reset message was sent to a Target from another Initiator.

When an Abort message is received the Target will clear all stacked commands for the Initiator which sent the Abort message. See section 5.5.2 for further information on the Abort message.

5.5.7 Command Tag Message (Optional)

Table 5-8: Command Tag

Byte	Value	Description
0	01h	Extended message
1	02h	Extended message length
2	03h	Command Tag message code
3	xxh	Command Entry Identifier

The Command Tag message (table 5-8) is optional and is sent by the Target to the Initiator when the Target supports the Command Stacking option (see section 6.1.6). The message is used in conjunction with the normal Identify and possibly the Extended Identify message to assign a unique tag to a command that has been stacked by the Target. The Target will send this message to this Initiator when it has received a command from the Initiator. The Command Entry Identifier is used to uniquely identify up to 256 commands stacked from the same Initiator. The Target will again send this message immediately after the Identify message during a reselection to identify to the Initiator which command is currently being processed.

If the Initiator receives this message and does not support command stacking it shall send a Message Reject message to the Target. Upon receiving a Message Reject message in response to a Command Tag message the Target shall not stack commands for that Initiator.

5.5.8 Extended Abort Message (Optional)

Table 5-8: Extended Abort

Byte	Value	Description
0	01h	Extended message
1	02h	Extended message length
2	04h	Extended Abort message code
3	xxh	Command Entry Identifier

The Extended Abort message (table 5-8) is optional and is sent by the Initiator to the Target to abort a command that has been stacked for processing by the Target (see section 6.1.6). The Command Entry Identifier is the unique command identifier sent

by the Target in the Command Tag message (see section 5.5.7). The Extended Abort message will only affect the command identified in the Command Entry Identifier. Any other stacked commands from the Initiator will not be aborted.

When this message is received the Target will clear the identified command from the command stack and any pending data and/or status for that command. No status or ending message will be sent for the identified command. Pending status, data or commands for other stacked operations will not be cleared. If the Target is currently connected for the command specified in the Command Entry Identifier (Target has sent a Command Tag message with the same Command Entry Identifier), the Target will go to the Bus Free phase and start executing the next command (from the same or different Initiators), if one is available.

If the Target receives this message and does not support command stacking or does not have a command with the specified Command Entry Identifier it shall send a Message Reject message to the Initiator.