

Date: March 18, 1987
To: XJT9.2
From: Jim McGrath
Company: Quantum Corporation
Subject: Command Queuing Proposal

This is a proposal for the inclusion of a command queuing feature in the SCSI 2 standard.

Command queuing would be implemented by expanding the multiple thread of command execution concept permitted by SCSI (re: appendix C of the SCSI standard). Currently a thread of execution is defined by the combination of Initiator address, Target address, and Logical Unit Number. No two currently active threads can possess the same triplet of values. Adding a fourth component, the SEQUENCE NUMBER, to this triplet will form a quadruple that can be used to uniquely identify multiple threads involving the same Initiator/Target/LUN. This allows for command queuing.

Specifically, it is proposed that a new extended message, the COMMAND SEQUENCE IDENTIFICATION message, be established. This message would be sent from the initiator to the target after the sending of the IDENTIFY message, but before entering Command Phase. In this manner the initiator would assign a sequence number to the command that is currently not in use for this Initiator/Target/LUN triplet. Upon reconnection this message, containing the same sequence number, will be sent from the target to the initiator after the sending of the IDENTIFY message, but before entering Data Phase. This will re-establish the logical connection with the command execution thread.

The precise queuing discipline implemented by the target shall be Vendor Unique. However, the discipline adopted must maintain the temporal order of command execution necessary to insure that the final contents of the logical blocks is the same as if the commands were executed in the order in which they arrived (e.g. a Write of LBA 0, followed by a Read of LBA 0, cannot be executed in reverse order). However, partial command execution that does not violate this restriction shall be allowed (e.g. a Write of LBA 1 and LBA 2, followed by a Read of LBA 0 and LBA 1, can be executed as a Read of LBA 0, a Write of LBA 1 and LBA 2, followed by a Read of LBA 1).

The following are the specific modifications proposed to the SCSI 2 Working Draft Proposal of 10/31/86:

Replace Table 5-4 with the following:

Table 5-4: Extended Message Codes

Code (y)	Description
00h	MODIFY DATA POINTER (Optional)
01h	SYNCHRONOUS DATA TRANSFER REQUEST (Optional)
02h	EXTENDED IDENTIFY (Optional)
03h	COMMAND SEQUENCE IDENTIFICATION (Optional)
04h - 7Fh	Reserved
80h - FFh	Vendor Unique

Add the following as the new section 5.5.6:

5.5.6 COMMAND SEQUENCE IDENTIFICATION (Optional)

Table 5-8: COMMAND SEQUENCE IDENTIFICATION

Byte	Value	Description
0	01h	Extended message
1	03h	Extended message length
2	03h	COMMAND SEQUENCE IDENTIFICATION code
3	x	Sequence number
4	00h	Reserved

The COMMAND SEQUENCE IDENTIFICATION message (Table 5-8) is optional and may be sent by a target or an initiator. It may be used in conjunction with the normal IDENTIFY message to provide a unique identifier for each of several commands currently active for an Initiator/Target/LUN combination. By allowing multiple commands to be active for a given Initiator/Target/LUN combination, this enables commands to be queued at the target and various Vendor Unique queue service algorithms to be employed. Up to 255 such commands (sequence numbered 1 to 255) may be active for each Initiator/Target/LUN combination at any given time. Sequence number 0 is reserved for commands requiring immediate execution.

Insert the following as the first words of the first sentence of paragraph 5 in section 6: "Unless command queuing is implemented (see 6.4), "

148

Insert the following as section 6.4, renumbering all subsequent sections:

6.4 Command Queuing

Command queuing may be implemented through the use of the COMMAND SEQUENCE IDENTIFICATION message. This message may only be sent under one of two circumstances. First, from the initiator to the target after the sending of the IDENTIFY message, but before entering Command Phase. Second, from the target to the initiator after the sending of the IDENTIFY message, but before entering Data Phase. Any attempt to send this message at any other time, or to send two such messages during an appropriate period of time, shall be ignored, but shall result in an CHECK CONDITION status for that command. [sense code and key to be supplied later - jpm]

The sequence number specified for the command shall act to distinguish it from other commands issued by that initiator to that LUN. Any sequence number from 1 to 255 may be used for this purpose. Any attempt to assign a sequence number that is still being used by a queued command to a new command shall result in a CHECK CONDITION for that new command, with no modifications having been made to the data on the logical unit. [sense code and key to be supplied later - jpm]

Command queuing is only supported for systems that implement disconnect/reconnect. The sequence number contained in the COMMAND SEQUENCE IDENTIFICATION message sent by the target to the initiator during reconnection informs the initiator of which command is returning data, status, or message information.

Note that any command may be preceded by a COMMAND SEQUENCE IDENTIFICATION message. Specifically, a REQUEST SENSE command may be preceded by such a message. When this is done the sense data reported shall be that of the most recently completed command having that sequence number. Thus non zero sense data must be kept for every command until either: 1) its sequence number is reused, 2) a REQUEST SENSE is performed for that command, or 3) a command is issued from the initiator that is not preceded by a COMMAND SEQUENCE IDENTIFICATION message.

If any commands are queued in the target for execution and a subsequent command is received that is not preceded by a COMMAND SEQUENCE IDENTIFICATION message, then all of the commands in the queue shall be executed before this new command, and during this time no other commands shall be accepted for queuing. Any attempt to issue a subsequent command before the last command has been executed will be treated if queuing was not implemented for this system (see 6).

The commands may be queued by the target according to a Vendor Unique queuing discipline. However, any command with a sequence number of 0 shall be executed immediately, regardless of the queuing discipline. Also, any queuing discipline must preserve temporal order of command execution necessary to insure that the final contents of the logical blocks is the same as if the commands were executed in the order in which they arrived (e.g. a Write of LBA 0, followed by a Read of LBA 0, cannot be executed in reverse order). Partial command execution that does not violate this restriction shall be allowed (e.g. a Write of LBA 1 and LBA 2, followed by a Read of LBA 0 and LBA 1, can be executed as a Read of LBA 0, a Write of LBA 1 and LBA 2, followed by a Read of LBA 1).