



X3T9.2/87-98  
Revision 1

August 17, 1987

To: John Lohmeyer, Chairman X3T9.2

From: Dan Davies

Subj: Proposed Buffered Mode for Deferred Error Handling

At the July meeting in Irvine, the X3T9.2 Working Group agreed to incorporate a new buffered write mode for sequential access devices. Buffered mode two is intended for multi-initiator systems to ensure that deferred write errors (errors that occur after GOOD status has been returned) are reported to any affected initiators.

Buffered mode one has no provision for identifying which initiator "owns" the data that remains in the buffer. Thus, the first initiator to discover the deferred error is responsible for the error recovery procedure, regardless of that initiator's affiliation with the unwritten data.

With buffered mode two, the target maintains identification of the initiator(s) that sent the data until the data has been successfully transferred to the medium. If a deferred write error occurs, the target will report the deferred error condition to any affected initiator(s). After the error is reported, any initiator with unwritten data in the buffer can recover or discard its data independent of another initiator's recovery action.

Reporting of this deferred error condition, and the associated recovery process, is included as Revision 1 of this proposal.

1) Buffered Mode Two MODE SELECT Wording

Add the following wording to the MODE SELECT command, page 9-20, replacing the last sentence at the bottom of the page:

"A buffered mode of two indicates that the target may report GOOD status on WRITE commands as soon as the data block has been transferred to the target's buffer. One or more blocks may be buffered prior to writing the block(s) to the medium. The target shall maintain identification of the initiator(s) that sent the data until the data has been successfully transferred to the medium. This mode is intended for multiple-initiator systems that require independent recovery paths for deferred write errors (see Deferred Error Condition, Section x.x.x, to be defined). Buffered modes of 3h through 7h are reserved."

2) Proposed Wording for Deferred Error Condition Reporting

Section 6.x.x. Deferred Error Condition

A deferred error condition shall begin whenever the target detects an error has occurred on a command that previously reported GOOD status. This condition shall persist for all affected initiator(s) until error recovery (if any) is completed, recovery is aborted, a BUS DEVICE RESET message is received, or "hard" reset condition occurs at the target.

If an initiator issues a non-medium access command while a deferred error condition is pending, the target shall perform the command and maintain the deferred error condition.  
[Maybe just INQUIRY and REQUEST SENSE should pass through ...?]

If an initiator issues a medium access [?] command while a deferred error condition is pending, the target will not perform the command and will report CHECK CONDITION status.

If the next command from the initiator is a REQUEST SENSE, then the ABORTED COMMAND Sense Key will be reported and the additional sense code will be set as appropriate.  
[ Possibly set the qualifier to the "owning" initiator(s)?

4C 00	Deferred Write Error	- No ID
4C 01	"	- Init ID 0
.....	"	.....
4C 80	"	- Init ID 7
.....	"	.....
4C 81	"	- Init ID 7,0

Specific error recovery and abort criteria is provided in the descriptions of the commands that can create the deferred error conditions (e.g. buffered WRITE, REWIND Immediate).

3) Proposed Wording for Deferred Write Error Recovery  
(Buffered mode one)

Section 9.x.x Deferred Write Error Condition  
[New section to be added to WRITE command]

A deferred write error condition occurs whenever the target detects that an error has occurred on a buffered write command that previously reported GOOD status.

If a deferred write error occurs in buffered mode one, the error will be reported to the first initiator issuing a buffer manipulating [?] command or the first initiator to respond to Asynchronous Event Notification (if supported).

After the error is reported to an initiator, subsequent commands from different initiators will result in a BUSY status until the deferred error condition is cleared.

The deferred write error condition in buffered mode one may be cleared by one of the following actions from the initiator receiving the deferred error indication:

- a) RECOVER BUFFERED DATA commands are issued to retrieve all buffered data that was not written to the medium.
- b) A REWIND or LOAD/UNLOAD command is issued to discard the unwritten buffered data. [Other positioning commands?]

The deferred error condition may also be cleared by a "hard" reset condition or a BUS DEVICE RESET message from any initiator.

4) Proposed Wording for Deferred Write Error Recovery  
(Buffered mode two)

If a deferred write error occurs in buffered mode two, the error will be reported to any initiator that issues a buffer manipulating command. Asynchronous Event Notification (if supported) will be attempted if a deferred write error is pending for any initiator(s) with unwritten data in the buffer.

After the error is reported, commands from initiators not involved in recovery will result in a BUSY status until the deferred error condition is cleared.

The deferred write error condition in buffered mode two may be cleared by the following action(s) from each initiator with unwritten data in the buffer:

- a) RECOVER BUFFERED DATA commands are issued by the initiator(s) involved in recovery to retrieve all data that was not written to the medium. (Note: Each initiator is responsible for recovering its own data in buffered mode two)
- b) An initiator issues a REWIND or LOAD/UNLOAD command to discard its unwritten buffered data. (Note: Each initiator is responsible for discarding its own data in buffered mode two)

The deferred error condition may also be cleared by a "hard" reset condition or a BUS DEVICE RESET message from any initiator.

5) Deferred Error Extended Sense Format

There have been concerns expressed about the extended sense format proposed for deferred error reporting. Compatibility with earlier implementations could be impacted by a target that arbitrarily responds to a REQUEST SENSE command with a previously reserved format. There is also a problem with reporting deferred error while another command is in process, particularly in multiple-initiator systems.

It may be possible to address both concerns by borrowing the desired data format concept proposed for the INQUIRY command. A desired data format of one would request sense format 7lh. This method ensures compatibility with earlier implementations while providing the tools for more sophisticated error reporting and recovery in future applications.

When a deferred write error is reported (as described in item two of this proposal), each initiator requesting deferred error format would only receive error information related to its recovery process. If a target receives a request for deferred error format from an initiator without a deferred error, a NO SENSE indication would be returned.