```
+---------------------------+ TM
|   |   |   |   |   |   |   |
| d | i | g | i | t | a | l |       INTEROFFICE MEMORANDUM
|   |   |   |   |   |   |   |
+---------------------------+
```

TO:	X3T11 Community	DATE:	21 April 1996
		FROM:	Doug Hagerman
CC:	X3T10 Community	DEPT:	Storage System Architecture
		PHONE:	508-841-2145
		INTERNET:	hagerman@starch.enet.dec.com

SUBJECT: Tape Support on Fibre Channel Arbitrated Loops

This memo was discussed at the T11 Working Group on Friday, April 12th. The group agreed that I should present it at the SCSI-3 Streaming Commands working group in May and return to T11 with a resolution in June.

The goal is to add the text, if any, to the Fibre Channel Arbitrated Loop (FC-AL) Private Loop Device Attach (PL-DA) that is needed to insure support for tape devices on FC-AL. The T11 working group recognizes that there is both a need for such support and also an opportunity for tape vendors to attempt to resolve any existing protocol or implementation oddities that may be characteristic of current SCSI tape drives.

Tape Issues

This paper describes one approach to adding SCSI-3 tape support to FC-PLDA. It includes a summary of the relevant tape and FC-AL issues, suggests a scenario for handling bit errors on the loop, and lists some features of SCSI and FC-AL that must be supported for this scenario to work properly.

1. References

This paper is based on the following documents:

FC-PH (Fibre Channel standard)
FC-AL (Fibre Channel Arbitrated Loop standard)
FCP (Fibre Channel Protocol for SCSI-3)
SAM (SCSI-3 Architectural Model)
FC-PLDA (Fibre Channel Private Loop Device Attach profile)

2. Error Handling

There is no frame-level error correcting code (ECC) in Fibre Channel, only an error detection code (EDC). There is no automatic retransmission of frames upon detected errors. The Private Loop Profile (hereafter: Profile) specifies that sequence retransmission on error shall not be used. Each SCSI command (including the associated data and returned status) is represented by one Fibre Channel exchange under the SCSI-3 Fibre Channel Protocol (FCP). Thus a single bit error detected anywhere in the exchange will cause the SCSI command to fail. Bit errors may occur in properly functioning FC-AL systems. They may be caused by random electronic events or by device swapping operations. These errors may occur at an average rate of up to 1 per 17 minutes on a properly functioning FC-AL system. This contrasts to parallel SCSI where parity errors are expected to occur only very rarely (weeks).

3. Acknowledgments and FCP Sequences

R_RDY is the Fibre Channel flow control acknowledgment, but it operates at the FC-1 level and is therefore not suitable for use as a frame acknowledgment. The Profile specifies that Class 3 communication shall be used, which is the datagram (no per-frame acknowledgment) method. The SCSI initiator expects an acknowledgment consisting of a valid FCP_RSP IU containing the completion status of the command.

Refer to FCP and FC-PLDA for details on the FCP model. For a write operation the method is that the WRITE command is sent in one sequence to the target, then the target returns an XFR_RDY, then the initiator sends a write data sequence (multiple FC frames). The XFR_RDY/write data sequence pair is repeated until all the data is transferred, then the target returns an FCP_RSP indicating completion of the transfer. An FC_RSP may also be sent earlier if an error is detected when transferring the data, but this can only be done when the target has “sequence initiative”, which is traded back and forth between the initiator and target.

4. Sequentiality

The FC-AL loop configuration guarantees the in-order delivery of frames because of electrical considerations. FCP requires that the exchange associated with a given SCSI command be identified by a unique X_ID. Sequence ID (SEQ_ID) use and optional reuse is specified by the Profile so as to guarantee the uniqueness of sequences. The Profile also specifies that continuously increasing Relative Offset be used in data frames. This combination of IDs allows the SCSI target to verify the correct sequentiality and completeness of a SCSI command and data transfer. An out-of-sequence frame causes an error recovery process that is similar to the bit error case.

5. Streaming and Buffers

Tape drives must stream data continuously to the media in order to maximize performance and capacity.

Data buffer memory may be large enough to hold the largest supported SCSI read or write command. This maximizes performance because continuous streaming can be guaranteed. This is because the transfer of data to or from the media and to or from the interconnect are independent. Typical modern DAT devices have 500 kiloBytes or more of buffer RAM, while DLT and other high-capacity high-performance drives may have over 2 MegaBytes of buffer.

If a write is encountered with more data than will fit in the buffer, the tape drive may be able to accept the data in units of logical blocks, each of which fits entirely inside the buffer. In addition, a drive may be able to write the data to the media in physical blocks that are smaller than the buffer. This gives high performance because continuous streaming may be possible. In this case the tape drive may be able to stop writing to the media at a block boundary, and then continue writing later.

6. Command Queueing and Internal Error Handling

SCSI-3 allows an initiator to issue additional commands before the first command has completed. The command queueing model applies to all device types including sequential devices such as tapes.

Modern tape drives use the queueing model and absorb incoming commands and data into a large input buffer. As soon as a command and its associated write data have been moved to the buffer, the target reports a success completion status to the initiator. The initiator considers the command to have completed at this time, regardless of whether the drive has actually completed writing the data to the tape media.

Eventually the drive attempts to write the data from the input buffer to the media. This may be before all the data has been transferred into the input buffer.

Depending on the sophistication of the drive, the target's internal controller may handle both the writing of the data and, if a media write error occurs, perform recovery operations including media repositioning and media write operation retries. If this recovery procedure is successful, any subsequent commands are handled properly and the initiator never finds out about the error. If after repeated attempts the internal controller is unable to write to the media, the target reports a deferred error which generally means that the media cartridge must be completely re-written. This is a rare occurrence.

Modern drives are block devices, and they write data to the media in individually addressable blocks. Thus it is possible for a drive to handle media write errors even if the input buffer is still accepting data from the interconnect.

Less sophisticated drives may report the error and rely upon the initiator to perform the recovery process. This may require a series of initiator commands to reposition the media to a previously-known position and then re-attempt the write.

7. Scenario: Interface Bit Error with Tape Drive with Large Buffer

Assuming that the tape drive can handle any errors that occur during the process of writing the data to the media, there remains the question of whether errors that occur on the Fibre Channel interconnect may cause difficulties. This is most likely to be a problem on a WRITE command.

Many modern SCSI tape devices have buffers large enough to hold multiple commands and the associated data. In these devices, if an interconnect bit error occurs in a data frame associated with a SCSI WRITE command to the tape drive, the following procedure is followed.

Previously accepted WRITE commands and data are stored in the input buffer and are in the process of being executed (written to the media) sequentially.

After a given WRITE command sequence is transferred, and the target has returned the XFR_RDY for that command, sequence initiative resides with the initiator.

Data frames making up a Write Data with SI transferred) are transferred from the initiator to the target.

A bit error occurs on the interconnect in middle of a frame.

Target receives erroneous frame. Target calculates EDC which does not match the one in the frame.

Target continues to receive frames until sequence initiative is transferred to target. (These frames are the ones that make up the remainder of the Write Data sequence that is in progress.)

Target returns an FCP_RSP response with RSP_CODE value of 01(hex), "FCP_DATA length different than BURST_LENGTH" and a SCSI status of "CHECK CONDITION".

Initiator terminates (does not send further sequences for) the remainder of the current command. Since the drive has not yet begun to write the data to the media, the drive deletes the failed command and all associated data from its input buffer.

Initiator reissues the offending command.

Assuming that the reissued command completes successfully, the initiator returns an appropriate status indication to the application program. The device begins to write the data to the media.

If the reissued command fails, the initiator retries the command for a vendor-specific number of times [perhaps 10?]. If these fail then the initiator is unable to transfer the command to the target and the initiator returns an appropriate status indication to the application program.

8. Implications

In order for the above model to work, all FC-PLDA tapes must comply with the following.

Tape drive must have a "large" buffer. It must be large enough to handle the command and data for the largest supported WRITE or READ command, and must be large enough to allow the tape to maintain streaming operation using the expected workloads andFibre Channel loop configuration environments.

Target must perform automatic repositioning and retries after failed media write operations.

9. Scenario: Interface Bit Error with Tape Drive with Small Buffer

Some applications [I'd like to see a list of them] perform single SCSI WRITE commands that transfer multiple megaBytes of data. In some cases transfers of this size may not fit in the input buffer of the tape drive. However, the tape device's logical block size may be negotiated using SSC mode pages so that a logical block fits in the buffer. My assumption is that the logical block size would be mapped to the FC-AL "Write Data with SI (sequence initiative) transferred (T6)" sequences. This needs to be discussed based on performance objectives.

In this case, if an interconnect bit error occurs in a data frame associated with a SCSI write command to the tape drive, the following procedure is followed.

After a given WRITE "Command/Task Mgmt with SI transferred (T1)" sequence is transferred, followed by the transfer of an XFR_RDY from the target back to the initiator, sequence initiative resides with the initiator.

Data frames are transferred using "Write Data with SI transferred (T6)" sequences. The target returns FCP_XFER_RDY sequences after each Write Data Sequence is transferred. After the first FCP_XFER_RDY is returned, the drive may begin to write the first block to the media. Meanwhile it continues to receive data from the next T6 sequence.

A bit error occurs on the interconnect in middle of a frame.

Target receives erroneous frame. Target calculates EDC which does not match the one in the frame.

Target continues to receive frames until sequence initiative is transferred to target. (Note that this may be several frames later, so the initiator doesn't know exactly which frame was transferred with error.)

Target returns an FCP_RSP response with RSP_CODE value of 01(hex), "FCP_DATA length different than BURST_LENGTH" and a SCSI status of "CHECK CONDITION".

Initiator terminates (does not send further sequences for) the remainder of the current command.

At this point the drive has written to the media part of the data for the command (those logical blocks of data contained in previously transmitted sequences), but is unable to obtain the remainder of the data for this sequence. There are several options that could be pursued; this should be decided by the SSC working group.

One option is that the initiator could issue a command to the drive telling it to "reposition to where you were before the start of the last command", then reissue the command. Another option is that the initiator could calculate the needed command in order to get the remainder of the data onto the media, and issue a "continuation" WRITE command. Another option is that the initiator could issue a series of commands to explicitly reposition the tape to wherever it thinks is the best place, then issue the correct commands to continue the write operation.

My hope is that the SCC working group will be able to agree upon a single approach to this situation. Documentation could be added to FC-PLDA to reflect this agreement, which would maximize the chance that tape drives will be supported on FC-AL in an industry-standard way.

I'm going to assume that the best approach is for the initiator to calculate what is needed to write the remainder of the data using a new WRITE command. This is based on the idea that an application doing long writes would not want to rewind the media and attempt to retry the original (failed) WRITE command because of the large transfer size.

10 Retransmission of Remaining Data to Drive with Small Buffer

The initiator knows that the data up to the last time it received an FCP_XFER_RDY from the target was transmitted successfully. After that point, any frame may have been the one in error. Thus the proper recovery point is the point at which the most recent FCP_XFER_RDY was received.

The responsibility for recovery resides partly in the initiator and partly in the target. The initiator calculates a new WRITE command with data length and offset values as appropriate to continue from the previously successful partial transfer, i.e. the last FCP_XFER_RDY. The target knows that the data received in the previous sequence was in error, so it resets the pointers in its input buffer back to where they were when the last FCP_XFER_RDY was sent. This is always possible if the logical block length is the same as the Write Data T6 sequence.

By this method, no repositioning of the media is required. The drive may continue to write to the media the data it had received up to the last FCP_XFER_RDY, and the only data that gets retransmitted on the interconnect is the frames in the failed Write Data sequence.