Seagate

Jan. 5, 1993 Document X3T9.2/93-012

TO: X3T9.2 Committee (SCSI)

FROM: Gerry Houlder

SUBJECT: Clarification of TTD, CIOP message operation

The Target Transfer Disable (TTD) and Continue I/O Process (CIOP) messages were added to SCSI to help disk array devices control when a particular drive can reconnect to transfer data to the array controller. From the descriptions of these messages and my recollection of the discussions of these messages in earlier meetings, it is clear that the messages were intended to work in the following manner:

1) The initiator selects the target with IDENTIFY message, optional QUEUE TAG message, TTD message, sends the command bytes (a READ command, for example), receives DISCONNECT message in, and BUS FREE results.

2) After target has data ready, it reselects the initiator, sends IDENTIFY message and optional QUEUE TAG message, sends DISCONNECT message, and (if ATTENTION signal isn't active) goes to BUS FREE.

3) The initiator selects the target with IDENTIFY message, optional QUEUE TAG message, CIOP message, receives the data the target has ready for this command, receives STATUS byte, COMMAND COMPLETE message, and BUS FREE results.

The documentation of these messages gives some guidelines on when the TTD and CIOP messages are illegal, but I have found an instance that is not either defined or disallowed. That situation is as follows:

1) The initiator selects the target with IDENTIFY message, optional QUEUE TAG message, TTD message, sends the command bytes (a READ command, for example), receives DISCONNECT message in, and BUS FREE results. This is the same as step 1) above.

2) The initiator selects the target with IDENTIFY message, optional QUEUE TAG message, CIOP message. The target recognizes the CIOP as legal (there is a valid TTD associated with this I/O process) but doesn't have data ready yet. It is logical for the target to send DISCONNECT message at this point, but there are no guidelines on how the pending TTD operation should be affected.

There are several "reasonable" ways for the target to proceed from this point. I want the committee to choose the one way it should work.

A) The "ignore early CIOP" method. If the CIOP comes before the target has the data, the target sends DISCONNECT message and goes bus free. When data is ready, the target reselects, identifies the nexus, and disconnects again. The initiator has to either 1) assert ATN and send MESSAGE REJECT to reject the disconnection attempt, or 2) send another CIOP message to retrieve the data.
ADVANTAGE: Easiest method for target to implement.
DISADVANTAGE: Lower performance than methods B or C.

B) The "override outstanding TTD" method. If the CIOP comes before the target has the data, the target sends DISCONNECT message and goes bus free. When data is ready the target reselects, identifies the nexus, and presents the data as if there wasn't any outstanding TTD message. If all of the requested data is transferred at this time, the command ends with status byte and COMMAND COMPLETE message. If only part of the requested data is transferred, things get messier. The target transfers some data and disconnects. Now another CIOP message (either before or after the target reselects to indicate that it has more data) must be sent by the initiator to get the next chunk of data. This continues until all data has been transferred.
ADVANTAGE: Better performance than method A because the overhead of sending the CIOP message can occur while the target is still getting the data.
DISADVANTAGE: Most difficult for target to implement.

C) The "cancel outstanding TTD" method. If the CIOP comes before the target has the data, the target sends DISCONNECT message and goes bus free. When data is ready the target reselects, identifies the nexus, and presents the data as if there wasn't any outstanding TTD message. If all of the requested data is transferred at this time, the command ends with status byte and COMMAND COMPLETE message. If only part of the requested data is transferred, the process continues as if no TTD message had ever been sent. The target transfers some data and disconnects. Now the target reselects and transfers the next chunk of data at its own convenience. This continues until all data has been transferred.
ADVANTAGES: Better performance than method A (for the same reason that method B has better performance) and easier for target to implement than method B.
DISADVANTAGES: More difficult for target to implement than method A and loss of the advantages of TTD operation for large data transfers.

I recommend method B.