

To: X3T9.2 Committee (SCSI)

From: Gerry Houlder IMPRIMIS/CDC

Subject: 16 bit SCSI-3 Protocol

I saw a message from Jeff Stai on the SCSI BBS stating that he would like to eliminate the Wide Transfer Request message and require all data transfers to be 2 bytes wide. This is interesting, and brings another thought to mind. If 16 bit data transfers are required, why not require everything else to also be 16 bits wide?

ARBITRATION - 16 bit arbitration will allow 16 devices on the bus, a feature considered desirable for SCSI-3.

SELECTION - this must also be 16 bit if 16 devices are allowed. Also, require Message Out phase after Selection (for Identify Msg) even if Attention line isn't active.

COMMAND - With 2 byte wide transfers, less transfers are needed. All CDBs are even byte sizes already and LUN field gets reclaimed.

STATUS - A 2 byte status will make room for new ending status conditions. Some messages (Linked Command Complete, Linked Command Complete with Flag, Autosense Data Follows, others?) could be eliminated if their function was moved to status.

MESSAGES - With 2 byte messages, much redefinition is needed but many message functions can be combined into new and more efficient messages. Examples:

IDENTIFY - Combine current Identify and Queue Tag messages into one two byte message. I suggest:

Queue Tag Identifier	- 8 bits
Queue Handling (Head of, Ordered, Simple)	- 2 bits
LUN/TAR Identifier	- 4 bits
Disconnect Privilege	- 1 bit
IDENTIFY message bit	- 1 bit

NOTE: If we still insist on controlling disconnects more exactly (not during data phase, etc.) we may want a 2 bit disconnect privilege field. However, we would have to shorten the Queue Tag to 7 bits to get another bit (or decrease the LUN field?). Another way is to say that disconnect after Command phase is always allowed and the Disc Priv bit only applies to data/status part of the command. A third way is to require robust initiator designs that can tolerate disconnects at any point. Then we don't even need a bit because disconnect is always allowed. Or is the initiator still allowed to Message Reject a target request for disconnect?

TARGET DESIRED DISCONNECT - Second byte shows reason for the disconnect and combines function of other messages, such as:

Will reconnect and complete later	- All bits cleared
Do Save Data Pointer before disconnect (and will reconnect later)	- 1 bit
Command completed (no reconnect)	- 1 bit
Command terminated (no reconnect)	- 1 bit
Initiate Recovery (no reconnect, ECA start)	- 1 bit

INITIATOR DESIRED DISCONNECT - Second byte shows reason and specifies continuation/termination actions, such as:

Reconnect later to complete command	- all bits = 0
Command control field (this initiator)	- 2 bits
Abort current command (01)	
Abort all commands (this initiator) (11)	
Terminate current command (10)	
Clear Reservations (this initiator)	- 1 bit
General Reset (this initiator)	- 1 bit
Affects items not specified in previous 3 bits	
Affect Other Initiators Also	- 1 bit
Affect Other LUNs/Target Routines Also	- 1 bit
Release ECA Recovery	- 1 bit
Initiate ECA Recovery	- 1 bit

PROTOCOL EXCEPTION (can be either an IN and OUT message) - Second byte shows the exception condition and/or retry action, such as:

No operation if all bits = 0	
SCSI Parity Error Detected	- 1 bit
Other Error Detected	- 1 bit
Retry Desired (used with either bit above)	- 1 bit
Message Reject	- 1 bit
Restore Pointers	- 1 bit

WIDE TRANSFER REQUEST can be eliminated; IGNORE WIDE RESIDUE can be eliminated if all devices are required to use even block sizes or even numbers of bytes in variable block transfers. A way to enforce even byte transfers is to have the CDB Transfer Length be a word (instead of byte) count.

SYNCHRONOUS TRANSFER REQUEST can be redefined as a two word (4 byte) message and will have a "reserved" byte left over. Perhaps it could also contain the "Wide Transfer Request" function in case we see a need to do multi-cable SCSI-3? If so, Ignore Wide Residue must come back. I think that function should be bundled into the PROTOCOL EXCEPTION message previously described. A 3 bit "ignore count" should do, especially if expansion is in word (2 byte) increments and the ignore count is in words.

MODIFY DATA POINTER can be defined as a 3 word message if a full 4 byte offset is needed (as it exists now) or it could be a 2 word message if we could get by with a 3 byte offset value. I lean toward the 2 word message because the really long transfers don't get any benefit from this message anyway. The message is used mostly for "first sector up" operation that is advantageous only for transfers less than a couple of cylinders long.

The INITIATOR DISCONNECT, TARGET DISCONNECT, and PROTOCOL EXCEPTION messages described above suggest an 8 bit message code and 8 modifier bits. To allow more functions to be added (INIT DISCON is already full and PROTOCOL EXCEPTION will be if a 3 bit Ignore Wide Residue field is added) the message code field could be shortened to 6 bits, allowing 2 more bits for describing reasons.

The obvious disadvantage of such a redefinition is that new SCSI Protocol Chips are needed. The advantage would be much lower overhead, which will lead to higher performance and better bus utilization.

This is also incompatible with a single connector, 2 byte wide SCSI that implements SCSI-2 protocol. This suggests the need to chose different connectors for these two alternatives if both become real. Perhaps the existing 68 pin B cable connector could be used for the "2 byte wide SCSI-2" and a similar size ribbon connector for this "2 byte wide SCSI-3" bus. That ought to keep people from plugging the wrong kinds of devices together!

I hope these ideas will be considered for SCSI-3. There is a real need to address the higher performance that is required of future systems. The extra overhead of carrying one byte wide command, status, and message transfers must be addressed as well as just allowing faster transfers during the data phase.

One more final thought: how about using synchronous transfers for all information phases and at all times for two byte wide SCSI-3? This would decrease overhead but would require "remembering" the agreement through resets and/or a default offset and period until renegotiation can be accomplished.