X3T9.2/88-007

LASER MAGNETIC STORAGE
INTERNATIONAL COMPANY

OSI DIVISION

4425 ArrowsWest Drive
Colorado Springs, CO 80907-3489
Tel: (303) 593 7900
Fax: (303) 599 8713
Telex (910) 920 4908

From: Paul Boulay
To: All
Msg #63, 25-Nov-87 09:40cst
Subject: Relative Address Bit Unnecessarily Restricted??

As defined, the Relative Address Bit is restricted to use by linked commands. (SCSI-2 Rev.2, Sect. 6.2.5.) This restriction seems unnecessary. For instance, if a LUN has been reserved for the exclusive use of one initiator, no other device can intervene and foul the last address.

AH, but you say, the address is not held in the device but in the controller, and a controller is used for several different LUN's. Even though you have one LUN locked up, an access to a different LUN by some other initiator will indeed trash the address that your next command is relative to.

Well, I say that that is a silly excuse rooted in the bad old days when controllers were magic and had next to no RAM. Today it seems perfectly reasonable to allocate, oh say, 4 or 5 bytes to the last address used by each of the LUN's.

The problem is letting the application know if it is OK to use a Relative Address outside a set of linked commands. The controller may be storing the address once for all LUN's or may have the space for per LUN or even per initiator / LUN combination. (In this case the unit reservation assumed above would not be necessary. Does any of this sound at all familiar? I thought so.) Or, the application may have the (warranted) assumption that there are no other initiators on this bus. (Are we practicing Big Brotherism?)

When (not if) we create an implemented options page we should have a byte giving the method used for storing last address and one for the mode select parameters. The options I can think of that apply to both of these are: once for all LUNs, once per LUN, and per LUN / initiator ID. (For the mode select parameters, it might be better to differentiate by type of parameter or by page. This to allow device parameters to follow the device (e.g., sectors per track) and others to follow the initiator (e.g. max burst length).

The next question is: why bother? One answer is a configuration with a new application and an enlightened peripheral but a host adapter that does not support linking. (Since we already have intelligent peripherals, how about renaissance peripherals?)

Am I missing some architectural reason why this restriction should exist? Perhaps nobody really cares because only a very few take the search data commands seriously. (I'll take mine with a grain of salt and a tall Alka Selser, thanks.)

To generalize, (for that is our duty,) we need to root out the restrictions that exist in the standard because of former implementation constraints and models. As least we need to understand where they are and why we should live with them in the standard.