X3T9.2 / 91-105
July 13, 1991

To: X3T9.2

from: Edward A. Gardner
Digital Equipment Corp.

Proposal to Define Guidelines for Multi-Initiator, Enhanced Availability System Environments in SCSI-3

With its growing acceptance and success, SCSI is being used in an ever broader range of system environments. One particular environment in which Digital would like to use SCSI is primarily represented by multi-initiator systems. More generally, systems that enhance availability and reliability through redundant hardware or other special techniques. Examples of such systems include:

1. High availability RAID controllers, with duplicated RAID controller cards both connected to a shared array of disks.

2. High availability file servers, with duplicated file server hardware and software both connected to an array of disks.

3. Almost any system running a critical production application, where there is a requirement to repair or replace drives without having to take down the entire system or application.

We think such systems should be explicitly considered and addressed in he development of SCSI-3. Examples of some of the issues they raise appear at the end of this document.

However, before succumbing to the enticement of technical details, perhaps we should establish the procedural context in which this might take place. The first question is whether other members of X3T9.2 share our desire to see SCSI-3 address multi-initiator, enhanced availability systems. Such systems are becoming increasingly important in the industry at large, which argues for their being addressed by a mainstream open bus standard. But if no other system vendor is interested, it is probably a waste of X3T9.2's time to explicitly consider this.

The second question is how to incorporate this into the SCSI-3 documents. To a large extent options already exist to allow the desired behavior, and it's straightforward to address any oversights. System vendors can pick and choose from this option menu to achieve the desired result. However, different vendors may choose conflicting solutions to the same problem, resulting in incompatible devices, confused customers, and wasted engineering resources in the industry at large. We would like to avoid this if at all possible.

We would like to see common, compatible devices for multi-initiator, enhanced availability systems. We would like SCSI-3 to prescribe a specific set of solutions for the problems of such systems. One way to accomplish that is to make the desired behavior mandatory for all SCSI-3 devices. However, it may not be practical to do so. Many eatures that are desirable for multi-initiator, enhanced availability systems may be useless in other system environments, while adding an incremental burden of complexity, cost, and engineering effort. Some features might even conflict with goals of other system environments. For example, multi-initiator enhanced availability systems are more concerned with availability or responsiveness than power consumption, whereas battery powered portables are the reverse.

One possible way to deal with this would be to define a few system environments and give guidelines for compatibility with each. I would normally use the term "profiles" for these guidelines, but I've been told that "profile" has a very specific meaning in a standards context. Perhaps an appendix to some SCSI-3 document that is already being developed. Perhaps a separate document, but I'd prefer to avoid the procedural overhead of another document unless there are good reasons for one.

Digital is interested in furthering the development of industry standard guidelines for SCSI-3 devices in multi-initiator, enhanced availability systems. We would be interested in contributing to such guidelines if X3T9.2 wishes to persue them.

The following are a few of the items that might be addressed by multi-initiator, enhanced availability system environment guidelines. They are listed here as examples of the kinds of things such guidelines might include. The purpose of this memo is not to discuss them or try to reach an agreement, these are merely to illustrate the kinds of things that might be included. Many of these arise from a requirement to be able to promptly and efficiently determine whether a device is operating properly, has just appeared, or has just failed or disappeared.

1. A list of options that are required by the guidelines, such as tagged queuing, disconnect-reconnect page support, etc.

2. Minimum functional requirements for modes described by the disconnect-reconnect page (e.g., minimum requirements to avoid "bus hogging").

3. Identification of a preferred command for verifying that a device is still functioning properly and does not have a unit attention condition. TEST UNIT READY with a HEAD OF QUEUE TAG is a likely candidate. The main issues are that the command must execute without waiting for any other commands that may be queued, and there must be an upper bound on the time to complete the command when the device is unbroken (perhaps tens of milliseconds).

4. Command queue size or algorithm guidelines to ensure that all supported initiators can access the device for data transfers and can use the command described in the previous item regardless of outstanding data transfer commands.

5. An upper bound for an unbroken device to respond to selection (assert BSY during SELECTION phase) (perhaps tens of microseconds).

6. An upper bound for an unbroken device to provide its full SCSI-3 protocol functions after a reset.

7. An upper bound for an unbroken device to provide its full SCSI-3 protocol functions after a power failure. That is, an upper bound for a device's entire microcode and parameters to be available, not for media load or spin-up delays. If a device becomes partially functional quickly, and fully functional later, specific guidance on how an initiator can monitor this.

8. All of the above issues for devices that connect to multiple SCSI busses.