To: X3T10 X3T10/94-233r1

From: Bob Snively
Sun Microsystems Computer Company
Mail Stop UMPK 12-204
2550 Garcia Ave,
CA 94043-1100
(415) 786-6694

Date: November 22, 1994

Subject: Proposed improvements for multi-port multi-initiator environments

After significant study, the FC-AL ad hoc study group that has been preparing profiles for directly attached disk drives has determined that two new SCSI functions are required to properly support multi-host multi-port environments. This document explains the architecture that requires those functions, defines the functions, and provides additional text and descriptions for related task management functions.

Basic Concepts:

The basic concept for this proposal is that a multi-port target device or logical unit will treat each attached initiator in the same manner, whether the initiator accesses the target through the same port as other initiators or through an alternate port. It is the responsibility of each initiator's system software to determine that each initiator and each port are operating correctly and to agree with other initiators about what corrective actions must be taken if some element of the system is not operating correctly. The definition of this system software is outside the SCSI-3 standards, but the standards provide powerful tools for the system software to determine the identity and status of the various SCSI devices and to perform recovery actions if one of the devices fails.

A second important concept for this proposal is that the failure of one part of a SCSI system should not affect the capability of the remaining portions of the system to continue operating normally. In addition, those portions of the system that are still operating successfully should be able to recover resources that have been locked to the failing parts of the system.

Description of proposed architecture:

A particular SCSI-2 target can be identified by its serial number (using the INQUIRY command). Particular SCSI-3 targets can be identified by serial number or (where available) the World Wide Name of the device. Using this information, the software of an initiator can determine what devices are managed by that initiator. Any agreements among the initiators about the usage of particular devices and extents of data within a device are made in a manner not specified by the SCSI standards.

The normal SCSI commands of the particular device type are then used to reserve, release, and access each particular device according to the agreements among the initiators. If an initiator or a path from an initiator to a target device fails, then the other initiators having operating paths to the target should continue operating normally. The operating initiators should also have the capability to perform the necessary resets and over-riding operations to gain control of whatever resources in the target device were dedicated to the failing initiator.

Two new SCSI functions are required to allow an operating initiator to take over the resources dedicated to a failing initiator.

1) Priority Reserve:

The Priority Reserve function uses a modifier in the RESERVE command to force a logical unit to yield any existing reservations that conflict with the other fields in the RESERVE command to the initiator that is generating the Priority Reserve function. The new initiator takes over the reservation. Text describing this command must be added to SPC (X3T10/995D), clauses 7.20 and 7.21, describing the RESERVE(6) and RESERVE(10) commands.

2) ABORT TASK SET, OTHER INITIATOR

An initiator uses the ABORT TASK SET, OTHER INITIATOR (ATSOI) task management function to clear resources related to the initiator identified by an initiator unique identifier.

For compatibility with the SCSI-2 dual port function, the initiator unique identifier is null for SIP devices and the operation is assumed to apply to the alternate port. In this case, the task sets for all initiators on the alternate port are aborted.

For serial SCSI devices that have access to the initiator unique identifier of other initiators, the tag contains the identifier of the initiator whose task set is to be aborted. Only the tasks associated with the specified initiator are aborted, regardless of the port to which the initiator is attached. The initiator unique identifiers for a protocol may be a World Wide Name, an initiator address and process identifier, or some other appropriate value. This proposal does not define those identifiers at this time.

Text describing this function must be added to SAM (X3T10/994D), clause 7, describing the task management functions. Clarifications of the operation of other task management functions may also be required.
Proposed text changes to SPC, Revision 3:

Section 7.20:

1) Modify Table 54:

Bit 5 of Byte 1 in Table 54 will be defined as PriRsrv

2) Paragraph 2

"The RESERVE .... in multiple-initiator systems."

is changed to read:

"The RESERVE(6) and RELEASE(6) commands provide the basic mechanism for contention resolution in multiple-initiator systems by reserving or recovering
certain resources for the exclusive or shared use of the reserving initiator."

Create new section 7.20.5

7.20.5 Priority Reservations (Optional)

If the PriRsrv (Priority Reserve) bit is zero, this command is executed as described in previous paragraphs. Reservation conflicts with the command are recognized and reported as described in previous paragraphs.

If the PriRsrv bit is one, this command is executed as a priority reservation.
If the host is reserved by another initiator or initiators, any such conflicting reservations are released and a new reservation is created according to the new RESERVE(6) command. The command may perform a logical unit reservation, an extent reservation, or a third-party reservation. RESERVATION CONFLICT shall not be reported to a priority reservation.

The priority reservation is typically used for error recovery and may disrupt normal reservation protocols. The mechanisms that an initiator uses to determine that a priority reservation is allowed or required are outside the scope of this standard.
Section 7.21:

1) Modify Table 57:

Bit 5 of Byte 1 in Table 57 will be defined as PriRsrv

2) Paragraph 2

"The RESERVE(10) .... in multiple-initiator systems."

is changed to read:

"The RESERVE(10) and RELEASE(10) commands provide the basic mechanism for contention resolution in multiple-initiator systems by reserving or recovering certain resources for the exclusive or shared use of the reserving initiator."

Proposed text changes to SAM, Revision 16:

Section 7

An additional task management function is defined and placed between ABORT TASK SET and CLEAR ACA. The text defining the task set is:

"ABORT TASK SET, OTHER INITIATOR (Initiator unique identifier ||) - Abort all tasks in the task set for the initiator identified by the initiator unique identifier. The function shall be supported if the logical unit has multiple ports. The function is optional for logical units supporting a single port.

New section 7.n, placed between 7.2 and 7.3

7.n ABORT TASK SET, OTHER INITIATOR

Function Call:

Service Response =
ABORT TASK SET, OTHER INITIATOR (Initiator unique identifier ||)

Description:

This function shall be supported if the logical unit has multiple ports. The function is optional for logical units supporting a single port.

The task manager shall terminate all tasks in the task set that were created by the initiator identified by the initiator unique identifier.

The target shall perform an action equivalent to receiving a series of ABORT TASK requests from the initiator identified by the initiator unique identifier. All tasks from the identified initiator in the task set serviced by the logical unit shall be aborted. Tasks from other initiators or other task sets shall not be terminated. Previously established conditions, including MODE SELECT parameters and reservations shall not be changed by the ABORT TASK SET, OTHER INITIATOR function. Any ACA condition for the indicated initiator shall be cleared. ACA conditions for any other initiator shall not be changed.

For compatibility with dual port SCSI-2 implementations that do not have a mechanism for uniquely identifying an initiator, the initiator or initiators will be assumed to be all initiators on the other port of the logical unit.

Logical units that have access to an initiator unique identifier shall use that value to indicate the initiator for which the ABORT TASK SET, OTHER INITIATOR will be performed.