LSI LOGIC®

Date: 04 July 1999
To: T10 Technical Committee
From: Ralph Weber, LSI Logic Alternate Member of T10
Subj: SAM-2 Changes For Queuing

During the May General SCSI Working Group meeting I was asked to investigate two SAM-2 issues related to queuing and propose changes for inclusion in SAM-2. The issues were:

1. Clarifying that only tagged tasks can receive the TASK SET FULL status; and
2. Allowing protocols such as SBP-2 and FCP-2 to not provide for untagged tasks.

This proposal addresses those issues plus another concern I noticed while reviewing SAM-2 with respect to those issues.

All clause references in this proposal are based on SAM-2 revision 10.

Before proceeding to the issues and proposals, I wish to comment on one statement made during the May meeting. The statement was, "It seems that TASK SET FULL should be implemented regardless." While the statement might be true when the implementation is an FCP-2 device, it is not true for all SCSI devices. A parallel SCSI device is allowed to implement only untagged tasks and in such an implementation the TASK SET FULL status need not (indeed should not) be supported.

Issue #1 - Clarification of Task object constituents

This is the issue that I noticed while investigating the other two issues.

The problem concerns the definition of a Task. The glossary definition of a Task (3.1.96) is, "An object within the logical unit representing the work associated with a command or group of linked commands." Yet the architecture model description of a Task (4.9) says nothing about work to be done by the logical unit being a constituent of a Task. This seems like a serious oversight and it should be corrected by modifying the first few sentences in 4.9 as follows:

Old text:

> The Task object represents either a Tagged Task or an Untagged Task without regard for the tagged or untagged nature of the Task. A Tagged Task is composed of a Tagged Task Identifier (see 4.9.2) and a Task Attribute (see 7.6). An Untagged Task is composed of a Untagged Task Identifier (see 4.9.2) and implicitly a SIMPLE task attribute (see 7.6). For convenience, Task Identifier (see 4.9.2) refers to either a Tagged Task Identifier or an Untagged Task Identifier without regard for the tagged or untagged nature of the task.

New text:

> The Task object represents either a Tagged Task or an Untagged Task without regard for the tagged or untagged nature of the Task. The composition of a Task includes the work to be performed by the logical unit in response to a command or group of linked commands. A Tagged Task is composed of work to be performed by the logical unit, a Tagged Task Identifier (see 4.9.2) and a Task Attribute (see 7.6). An Untagged Task is composed of work to be performed by the logical unit, a Untagged Task Identifier (see 4.9.2) and implicitly a SIMPLE task attribute (see 7.6). For convenience, Task Identifier (see 4.9.2) refers to either a Tagged Task Identifier or an Untagged Task Identifier without regard for the tagged or untagged nature of the task.

Issue #2 - Clarification TASK SET FULL status usage

The problem is that the definition of the TASK SET FULL status (5.2) does not state that it can only be returned for tagged tasks. Wording changes to correct this oversight were proposed at the May meeting. The following proposal is in the spirit of the changes proposed in May, but with slight modifications.

Old text:

> **TASK SET FULL.** This status shall be implemented if the logical unit supports the creation of tagged tasks (see 4.9). This status shall be returned when the logical unit receives a command and does not have enough resources to enter the associated task in the task set.

New text:

> **TASK SET FULL.** This status shall be implemented if the logical unit supports the creation of tagged tasks (see 4.9). This status shall be returned when the logical unit receives a ~~command~~ tagged task and does not have enough resources to enter ~~the associated task~~ it in the task set. This status shall not be returned for untagged tasks.

New text minus strikeouts:

> **TASK SET FULL.** This status shall be implemented if the logical unit supports the creation of tagged tasks (see 4.9). This status shall be returned when the logical unit receives a tagged task and does not have enough resources to enter it in the task set. This status shall not be returned for untagged tasks.

Note that the proposed changes are slightly dependent on the changes proposed for issue #1. However, it probably is possible to make the proposed changes without making the changes proposed for issue #1.

Issue #3 - Removal of requirement that protocols defined untagged tasks

I approached development of this proposal by searching SAM-2 for all appearances of "untagged task" and I found two places appearing to need changes.

In defining the ability to not support untagged tasks, I tried to maintain some statement of when a protocol is required to include untagged tasks written in such a way that parallel SCSI will not be allowed to drop untagged task support.

The first site needing changes is the unconditional requirement in 4.9 that untagged tasks be supported by every SCSI protocol.

Old text:

> Every SCSI protocol shall support tagged and untagged tasks. Support for the creation of tagged tasks by a logical unit, however, is a logical unit implementation option.

New text (a complete replacement for the old text):

> Every SCSI transport protocol shall support tagged tasks. If a SCSI transport protocol defines the Tagged Task Identifier (see 4.9.2) in field having no other purpose or usage than containing the Tagged Task Identifier value, then that SCSI transport protocol shall support untagged tasks. If a SCSI transport protocol defines the Tagged Task Identifier based on transport property or field that is not specifically defined to be a Tagged Task Identifier value, then that SCSI transport protocol is not required to support untagged tasks. If the SCSI transport protocol upon which a SCSI device operates supports untagged tasks, then the SCSI device may be implemented to use only untagged tasks.
>
> Note *n* Initiators that support a variety of SCSI transport protocols may encounter concurrently SCSI devices that support only untagged tasks and SCSI devices that support only tagged tasks.

My only concern about this wording is that it might be construed to require untagged support in the SPI_L_Q information unit.

The second site needing changes is in the description of the unit attention condition (5.6.5). The text concerns the handling of a REQUEST SENSE command when a unit attention condition is pending.

The entire text on the subject is as follows:

> If a request for sense data is received from an initiator with a pending unit attention condition (before the logical unit establishes the auto contingent allegiance or contingent allegiance condition), then the logical unit shall either:
>
> a) Report any pending sense data and preserve the unit attention condition on the logical unit; or,
> b) Report the unit attention condition.
>
> If the second option is chosen (reporting the unit attention condition), the logical unit may discard any pending sense data and may clear the unit attention condition for that initiator.

> If the logical unit has already generated the auto contingent allegiance or contingent allegiance condition for the unit attention condition, the logical unit shall perform the second action listed above. If NACA for the REQUEST SENSE command is zero and the command is untagged the contingent allegiance condition shall be cleared.

The problem is in the last paragraph. Since there no longer is a guarantee that an initiator is able to send and untagged command, it appears that a previously existing capability is being taken away by elimination of the requirement for transport protocol support for untagged tasks. I propose a rather simplistic approach to address this.

Old text:

> If NACA for the REQUEST SENSE command is zero and the command is untagged the contingent allegiance condition shall be cleared.

New text:

> If NACA for the REQUEST SENSE command is zero ~~and the command is untagged~~ the contingent allegiance condition shall be cleared.

New text minus strikeouts:

> If NACA for the REQUEST SENSE command is zero the contingent allegiance condition shall be cleared.

In all honesty, I think this specific wording change has a 'bull in the china shop' feel. I am not certain about the motivation for the existing text and so have little confidence in the correctness of removing it.