Document: T10/00-359r8 Date: November 13, 2001

To: T10 Committee Membership

From: Edward A. Gardner, Ophidian Designs

Subject: Unit Attention Issue

Previous revisions of this proposal discussed the detailed interaction between unit attention conditions and autosense. They concluded that when we introduced autosense delivery in FCP, we lost a significant piece of functionality. SCSI-2 ensured that an initiator could, if it so chose, guarantee it was aware of any unit attention conditions before issuing a command. That is, the protocol was interlocked between unit attention and command issuance. That capability still exists in SPI-n today provided data group transfers are used. It has been lost in all other SCSI protocols through their support of autosense. Note that SAM describes autosense delivery as an option that can be enabled or disabled on a command by command basis.

The January 17, 2001 CAP working group recommended that we use a bit in the Control mode page to control whether autosense clears a unit attention condition. This document describes the specific changes to SAM-2 and SPC-3 to accomplish that recommendation.

Revision 3 of this document includes changes from the July, 2001 CAP working group. It also references sam2r19.pdf instead of sam2r18.pdf.

Revision 4 of this document includes changes from the September 2001 CAP working group.

Revision 5 of this document is identical to revision 4, except that it references sam2r20.pdf instead of sam2r19.pdf.

Revision 6 of this document is identical to revision 4 and revision 5, except that it references spc3r01.pdf instead of spc3r00.pdf.

Revision 7 of this document addresses comments made by Ralph Weber in 01-299r1.

Revision 8 of this document includes the change requested by the November 2001 CAP working group. This reverses an earlier change requested by the July 2001 CAP working group, changing a "may" back to the "shall" present sam2r20 (last paragraph of sub-clause 5.8.5).

Black text denotes text that is unchanged from sam2r20.pdf and spc3r01.pdf. Red text and ~~cross outs~~ denote changes from sam2r20.pdf and spc3r01.pdf that were described in revision 3 of this proposal. Blue text and ~~cross outs~~ denote changes from sam2r20.pdf and spc3r01.pdf that are different from those described in revision 3 of this proposal. Green shaded text denotes editorial comments that are not part of the changes described by this proposal. Finally, a second copy of the entire proposal, without all the color-coded formatting, appears at the end. Some may find that simpler to read.

This document also corrects what I believe is an error in SAM-2. This error comes from 97-225, which was approved and incorporated into sam2r05. I believe the error resulted from a trivial mistake by the author of 97-225 that was not caught by the working group. However this cannot be verified, as the author of 97-225 is no longer available.

The error concerns the situation when a unit attention condition has not yet been reported to an initiator (no ACA or CA exists) and the initiator issues an ordinary command (e.g. a READ or WRITE). SAM-2 currently specifies that the target **shall** return ACA ACTIVE (NACA=1) or BUSY (NACA=0) status. This is absurd. In the first case no ACA condition exists, yet the target is being required to return ACA ACTIVE. In the second case the target is being required to return BUSY for no reason. The proper response is for the target to return CHECK CONDITION and report the unit attention condition. This was the required response in sam2r04 and all prior versions back to SCSI-1.

Changes to SAM-2 (sam2r20.pdf).

Sub-clause 5.8.4.2, Asynchronous Event Reporting, top of page 68, pdf page 91, third paragraph on that page:

An exception condition encountered after command completion ~~or unit attention condition~~ shall be reported to a specific initiator once per occurrence of the event causing it. The logical unit may choose to use an asynchronous event report or to return CHECK CONDITION status on a subsequent command, but not both. Notification of an exception condition encountered after command completion shall be ~~returned~~ reported only to the initiator or initiators that sent the affected task or tasks.

The only technical change this proposal makes to the above paragraph is to remove the requirement that a unit attention condition shall be reported to a specific initiator only once. Interlocking unit attention conditions may require that the same unit attention condition be reported multiple times. The requirement is simply removed here, rather than revising it, as sub-clause 5.8.5 fully describes the rules for reporting and clearing unit attention conditions. Any statement here would be redundant or contradictory.

All other changes shown to the above paragraph are editorial and may be included, ignored or revised at the SAM-2 editor's discretion. Note that if an exception condition affects multiple tasks, multiple initiators may have sent those tasks.

Sub-clause 5.8.4.3, Autosense, middle of page 68, pdf page 91, first two paragraphs of that sub-clause:

Autosense is the automatic return of sense data to the application client coincident with the completion of a SCSI command under the conditions described below. ~~The return of sense data in this way is equivalent to an explicit command from the application client requesting sense data immediately after being notified that an ACA or CA condition has occurred.~~ Inclusion of autosense support in a SCSI protocol standard is optional.

If supported by the protocol and logical unit and requested by the Execute Command remote procedure call (see 5.1), the device server shall only return sense data in this manner coincident with the completion of a command with a status of CHECK CONDITION. After autosense data is sent, sense data except sense data associated with a unit attention condition when the UAINTLCK bit equals one and ~~the sense data and~~ the CA (NACA equals zero), if any, shall then be cleared. Autosense shall not affect ACA (NACA equals one), see 5.8.1, or sense data associated with a unit attention condition when the UAINTLCK bit equals one.

This proposal makes two technical changes to the above paragraphs.

The first technical change is to eliminate the statement that autosense is equivalent to an explicit REQUEST SENSE command (sentence deleted from first paragraph). This is not true when unit attention is interlocked, as autosense does not clear a unit attention while REQUEST SENSE does. This statement is deleted rather than revised as the behavior of autosense is fully described without it (see 5.8.5).

The second technical change is to eliminate the requirement that sense data shall be cleared after autosense data is sent (changes to second paragraph). When unit attention is interlocked, sense data describing the same unit attention condition may need to be sent (as autosense data) many times, and therefore cannot be cleared. The blue text in the second paragraph is wording suggested by Ralph Weber in 01-299r1.

Clause 5.8.5, Unit Attention condition, bottom of page 68 (pdf page 91) through most of page 69 (pdf page 92), entire contents of that sub-clause:

5.8.5 Unit Attention condition

Each logical unit shall generate a unit attention condition whenever the logical unit has been reset as described in 5.8.7 or by a power-on reset. In addition, a logical unit shall generate a unit attention condition for each initiator whenever one of the following events occurs:

a) A removable medium may have been changed;

b) The mode parameters in effect for this initiator have been changed by another initiator;

c) The version or level of microcode has been changed;

d) Tasks for this initiator were cleared by another initiator;

e) INQUIRY data has been changed;

f) The logical unit inventory has been changed;

g) The mode parameters in effect for the initiator have been restored from non-volatile memory;

h) A change in the condition of a synchronized spindle; or

i) Any other event requiring the attention of the initiator.

Logical units may queue unit attention conditions. After the first unit attention condition is cleared, another unit attention condition may exist (e.g., a power on condition followed by a microcode change condition).

A unit attention condition shall persist on the logical unit for each initiator until that initiator clears the condition as described in the following paragraphs.

If an INQUIRY command enters the enabled task state, the logical unit shall perform the INQUIRY command and shall neither report nor clear any unit attention condition. ~~or a REPORT LUNS command is received from an initiator to a logical unit with a pending unit attention condition (i.e., before the logical unit generates the ACA or CA condition), the logical unit shall perform the command and shall not process the unit attention condition. If the~~

If a REPORT LUNS command enters the enabled task state, the logical unit shall perform the REPORT LUNS command and shall not report any unit attention condition. The logical unit shall clear any unit attention condition ~~was~~ established in response to a change in the logical unit inventory~~, the unit attention condition shall be cleared~~ for all logical units for the initiator that sent the REPORT LUNS command.~~ In all other cases, the INQUIRY or REPORT LUNS command shall not clear the unit attention condition.~~ The logical unit shall not clear any other unit attention condition.

The September 2001 CAP working group requested that the last sentence of the old text be restored for REPORT LUNS. The sentence in blue above is the result, reworded to match the style of the rest of the paragraph.

If a REQUEST SENSE command enters the enabled task state while a unit attention condition exists for the initiator that sent the REQUEST SENSE command, ~~and the unit attention interlock (uaintlck) bit in the logical unit’s control mode page contains zero (see SPC-3) request for sense data is received from an initiator with a pending unit attention condition (i.e., 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~~ all unit attention conditions on the logical unit; or,

b) Report ~~the~~ a unit attention condition for the initiator that sent the REQUEST SENSE command. The logical unit may discard any pending sense data and shall clear the reported unit attention condition for that initiator.

~~If the logical unit reports a unit attention condition (the second option above) option b) is chosen (i.e., reporting the unit attention condition), the logical unit may discard any pending sense data and may clear the reported unit attention condition for that initiator.~~

The September 2001 CAP working group directed that the above paragraph should be moved into item b above it and that the word “may” be deleted from “may clear the reported unit attention condition”. One of the words “may”, “should” or “shall” is clearly called for, if “may” is to be removed I believe the intent was to substitute “shall”. Note that some may consider removal of the “may” to be a technical change unrelated to this proposal.

The September 2001 CAP working group suggested the alternate wording (for item b) "report and clear a unit attention condition…". To me that leaves room for miss-interpreting this as saying that a UA is reported and a UA is cleared, but not necessarily the same UA. I prefer the above wording.

The September 2001 CAP working group asked why REQUEST SENSE was handled differently when UAINTLCK is zero or one. The primary reason is the "may" that has just been removed. When UAINTLCK is zero, we formerly said, "may clear the reported UA". When UAINTLCK is one, we formerly said, "shall clear the reported UA". The secondary reason was requiring that a unit attention condition be reported (requiring option b above). However, UAINTLCK is only expected to be one in autosense environments, which imply that there will never be pending sense data (i.e. option a is never applicable). As a consequence, with the removal of the "may" I see no reason to describe REQUEST SENSE differently when UAINTLCK is zero or one and have combined the two cases.

If the logical unit has already generated the ACA or CA condition for a ~~the~~ unit attention condition, the logical unit shall report the unit attention condition (the second action above). ~~perform the second action listed above. If NACA for the REQUEST SENSE command is zero and the command is untagged the CA condition shall be cleared.~~

~~If a REQUEST SENSE command enters the enabled task state while a unit attention condition exists for the initiator that sent the REQUEST SENSE command, and the unit attention interlock (UAINTLCK) bit in the logical unit's control mode page contains one (see SPC-3), then the logical unit shall report a unit attention condition for the initiator that sent the REQUEST SENSE command. The logical unit shall clear the reported unit attention condition for that initiator.~~

If ~~an initiator issues~~ a command other than INQUIRY, REPORT LUNS, or REQUEST SENSE enters the enabled task state while a unit attention condition exists for the ~~that~~ initiator that sent the command, ~~(prior to generating the ACA or CA condition for the unit attention condition),~~ the logical unit shall not perform the command and shall report CHECK CONDITION status. The logical unit shall provide sense data that reports a unit attention condition for the initiator that sent the command. ~~ACA ACTIVE (NACA equals one, see 5.2.3) or BUSY (NACA equals zero) status.~~

The above paragraph is the one that contained the error resulting from 97-225.

If a logical unit reports a unit attention condition with autosense or with an asynchronous event report, and the unit attention interlock (UAINTLCK) bit in the logical unit's control mode page ~~contains~~ is zero, ~~successfully sends an asynchronous event report informing the initiator of the unit attention condition,~~ then the logical unit shall ~~may shall~~ clear the reported unit attention condition for that initiator on the logical unit (see 5.8.4.2, 5.8.4.3 and SPC-3). If the unit attention interlock (UAINTLCK) bit ~~contains~~ is one, the logical unit shall not clear unit attention conditions reported with autosense or an asynchronous event report.

The November 2001 CAP working group changed the "may" back to "shall" in the first sentence of the above paragraph.

Changes to SPC-3 (spc3r01.pdf).

Clause 8.3.6, Control mode page, table 159, bottom of page 201, pdf page 222:

Define a reserved bit as UAINTLCK. I suggest bit 4 or 5 of byte 4.

Clause 8.3.6, Control mode page, middle of page 203, pdf page 224, add the following paragraph:

The unit attention interlock (UAINTLCK) bit controls the clearing of unit attention conditions reported with autosense or asynchronous event reporting (see SAM-2). A ~~unit attention interlock (~~UAINTLCK~~)~~ bit of zero specifies that the logical unit ~~shall~~ clears any unit attention condition reported with autosense or asynchronous event reporting~~ (see SAM-2)~~. A ~~unit attention interlock (~~UAINTLCK~~)~~ bit of one specifies that the logical unit shall not clear any unit attention condition reported with autosense or asynchronous event reporting. ~~When~~

~~the unit attention interlock (uaintlck) bit contains is one, issuing~~ A REQUEST SENSE command ~~shall~~ clears any unit attention condition that it reports.

The July 2001 CAP working group requested that the last sentence in the above paragraph be added. The September 2001 CAP working group made some comments on that sentence that I don't understand, but seemed to conclude that it should remain. However, since the behavior of REQUEST SENSE no longer depends upon the state of UAINTLCK, it did need some rewording. I would prefer to delete that final sentence, as it is redundant with the material in SAM-2. The second sentence was slightly reworded as recommended by Ralph Weber in 01-299r1.

The remainder of this document is a second copy of the proposed changes, without all the color-coded formatting. Some may find it easier to read than the above.

Changes to SAM-2 (sam2r20.pdf).

Sub-clause 5.8.4.2, Asynchronous Event Reporting, top of page 68, pdf page 91, third paragraph on that page:

> An exception condition encountered after command completion shall be reported to a specific initiator once per occurrence of the event causing it. The logical unit may choose to use an asynchronous event report or to return CHECK CONDITION status on a subsequent command, but not both. Notification of an exception condition encountered after command completion shall be reported only to the initiator or initiators that sent the affected task or tasks.

Sub-clause 5.8.4.3, Autosense, middle of page 68, pdf page 91, first two paragraphs of that sub-clause:

> Autosense is the automatic return of sense data to the application client coincident with the completion of a SCSI command under the conditions described below. Inclusion of autosense support in a SCSI protocol standard is optional.
>
> If supported by the protocol and logical unit and requested by the Execute Command remote procedure call (see 5.1), the device server shall only return sense data in this manner coincident with the completion of a command with a status of CHECK CONDITION. After autosense data is sent, sense data except sense data associated with a unit attention condition when the uaintlck bit equals one and the CA (NACA equals zero), if any, shall then be cleared. Autosense shall not affect ACA (NACA equals one), see 5.8.1, or sense data associated with a unit attention condition when the uaintlck bit equals one.

Clause 5.8.5, Unit Attention condition, bottom of page 68 (pdf page 91) through most of page 69 (pdf page 92), entire contents of that sub-clause:

5.8.5 Unit Attention condition

> Each logical unit shall generate a unit attention condition whenever the logical unit has been reset as described in 5.8.7 or by a power-on reset. In addition, a logical unit shall generate a unit attention condition for each initiator whenever one of the following events occurs:
>
> a) A removable medium may have been changed;
>
> b) The mode parameters in effect for this initiator have been changed by another initiator;
>
> c) The version or level of microcode has been changed;
>
> d) Tasks for this initiator were cleared by another initiator;
>
> e) INQUIRY data has been changed;
>
> f) The logical unit inventory has been changed;

g) The mode parameters in effect for the initiator have been restored from non-volatile memory;

h) A change in the condition of a synchronized spindle; or

i) Any other event requiring the attention of the initiator.

Logical units may queue unit attention conditions. After the first unit attention condition is cleared, another unit attention condition may exist (e.g., a power on condition followed by a microcode change condition).

A unit attention condition shall persist on the logical unit for each initiator until that initiator clears the condition as described in the following paragraphs.

If an INQUIRY command enters the enabled task state, the logical unit shall perform the INQUIRY command and shall neither report nor clear any unit attention condition.

If a REPORT LUNS command enters the enabled task state, the logical unit shall perform the REPORT LUNS command and shall not report any unit attention condition. The logical unit shall clear any unit attention condition established in response to a change in the logical unit inventory for all logical units for the initiator that sent the REPORT LUNS command. The logical unit shall not clear any other unit attention condition.

If a REQUEST SENSE command enters the enabled task state while a unit attention condition exists for the initiator that sent the REQUEST SENSE command, then the logical unit shall either:

a) Report any pending sense data and preserve all unit attention conditions on the logical unit; or,

b) Report a unit attention condition for the initiator that sent the REQUEST SENSE command. The logical unit may discard any pending sense data and shall clear the reported unit attention condition for that initiator.

If the logical unit has already generated the ACA or CA condition for a unit attention condition, the logical unit shall report the unit attention condition (the second action above).

If a command other than INQUIRY, REPORT LUNS, or REQUEST SENSE enters the enabled task state while a unit attention condition exists for the initiator that sent the command, the logical unit shall not perform the command and shall report CHECK CONDITION status. The logical unit shall provide sense data that reports a unit attention condition for the initiator that sent the command.

If a logical unit reports a unit attention condition with autosense or with an asynchronous event report, and the unit attention interlock (UAINTLCK) bit in the logical unit's control mode page is zero, then the logical unit shall clear the reported unit attention condition for that initiator on the logical unit (see 5.8.4.2, 5.8.4.3 and SPC-3). If the unit attention interlock (UAINTLCK) bit is one, the logical unit shall not clear unit attention conditions reported with autosense or an asynchronous event report.

Changes to SPC-3 (spc3r01.pdf).

Clause 8.3.6, Control mode page, table 159, bottom of page 201, pdf page 222:

Define a reserved bit as UAINTLCK. I suggest bit 4 or 5 of byte 4.

Clause 8.3.6, Control mode page, middle of page 203, pdf page 224, add the following paragraph:

The unit attention interlock (UAINTLCK) bit controls the clearing of unit attention conditions reported with autosense or asynchronous event reporting (see SAM-2). A UAINTLCK bit of zero specifies that the logical unit clears any unit attention condition reported with autosense or asynchronous event reporting. A UAINTLCK bit of one specifies that the logical unit shall not clear any unit attention condition reported with autosense or asynchronous event reporting. A REQUEST SENSE command clears any unit attention condition that it reports.