X3T10/94-043R0
January 24, 1994

Date: January 24, 1994

From: Charles Monia
SAM Technical Editor

To: Members of X3T10

Subject: Results of January 11, 1994 SAM Working Group Review

References: (a) X3T10/94-028R0, Summary of Comments on SAM Revision 12
(b) X3T9.2/86-109 rev 10h, Draft proposed AMERICAN NATIONAL STANDARD X3.131-199x

Enclosures (a) Compilation of items cited from X3T10/94-028R0

On January 11, 1994, the general SCSI-3 Working Group discussed the review comments received for SAM, revision 12 along with the proposed responses prepared by the technical editor. This memo summarizes the status of each issue. The citations accompanying an item identify pertinent comments or responses from reference (a).

Comments and corrections may be submitted via the SCSI reflector or directly to me via mail, email or FAX as follows:

Mail: Digital Equipment Corporation
334 South Street
SHR 3-2/W4
Shrewsbury, MA 01545
EMAIL: monia@starch.enet.dec.com (internet)
Voice: 508-841-6757
FAX: 508-841-6100

Item 1: Policy for target return of autosense data.

The technical editor requested clarification on whether autosense data should be returned Only on the occurrence of an ACA condition or Whenever sense data is available -- that is on the occurrence of an ACA condition and at other times as specified in the SCSI-2 standard (see reference b, section 7.2.14, page 127)

See: Response to Sun comment T 014 on page 62.

Status: Open.

The technical editor will post this item for review on the SCSI reflector.

Item 2: Annex A, Task Set Boundaries.

Although this annex is identified an "informative", its inclusion is potentially confusing and may lead to non-compliant implementations.

See; (a) Sun comment T 066, page 80;
(b) Panasonic comment R2 on page 40.

Status: Annex A to be deleted.

The working group felt that the annex should be removed,. Since all interested parties were not present, the matter was referred to the plenary for discussion. Subsequently, the plenary voted to delete the annex.

As a related issue, Bob Snively of Sun requested that the technical editor include table 1 from the Annex C (queuing model working paper) in the main body of the document. The editor agreed to the inclusion of the technical information included in that table.

Item 2: Limitation on maximum length of vendor-specific command descriptor blocks.

In SCSI-2, there is no upper limit on the length of a vendor-specific command descriptor block. This is in conflict with the 16-byte upper limit for SCSI-3.

See: Seagate comment 52 on page 50.

Status: No change to SAM.

The working group consensus was that the limitation in CDB length was acceptable for SCSI-3.

Item 4: Bit-significance within the SCSI status byte.

Since new SCSI-3 status values no longer observe the bit conventions used in SCSI-2 there may be a compatibility problem with the installed base.

See: Seagate comment 55 on page 51.

Status: No change to SAM.

The working group agreed that removal of bit significance reflected a decision of the plenary and should, therefore, not be changed.

Item 5: Detection of duplicate task identifiers.

SAM states that the requirement for a target to detect a duplicate task identifier is established by each protocol specification. Duplicate tags should always be detected by a target, regardless of the protocol.

See: Western Digital comment on page 92 of a), re: SAM page 50, second paragraph, section 6.5.3.

Status: No change to SAM.

The working group concurred with Jeff Williams' (HP) view that the overhead for duplicate tag detection would be excessive for protocols implementing the maximum tag length (64 bits).

Item 6: SCSI-3 support for tagged queuing.

SAM incorrectly implies that queuing is optional in SCSI-3.

See: Hewlett-Packard comment 031 on page 9.

Status: SAM will be modified to state explicitly that queuing support is optional.

The working Group confirmed that queuing is optional in SCSI-3.

Item 7: Set of task management functions to be implemented when queuing is not supported.

SAM fails to state that certain task management functions, such as CLEAR TASK SET, are optional when the logical unit does not implement queuing.

See: Sun comment T 042 on page 70.

Status: Comment accepted.

SAM will be modified to state that support for the task management functions referenced in the comment is only required if the target supports tagged queuing.

Item 8: Service Delivery Transaction Ordering.

For a given sender and receiver, SAM requires the service delivery subsystem to deliver a series of transactions in the order in which they were sent. This requirement should be deleted since some implementations are incapable of preserving such order.

See: a) Hewlett Packard comment T 014 on page 4,
b) Panasonic comment R3 on page 40,
c) Sun comment T 007 on page 59,
d) Sun comment T 048 on page 74,
e) Sun comment T 062 on page 79.

Status: Comment accepted.

The working group felt that, while some protocol implementations may preserve the order of transactions, the only way for an application to guarantee ordering across all interconnects is to not use queuing when the sequence of requests must be maintained.

With one or two exceptions the working group supported the deletion of this requirement. The matter was subsequently put before the plenary with a similar result.

Item 9: Number of pending task management requests per I_T_L

The implementor's note in section 4.7.3.1, page 39 of SAM, advises against a protocol allowing multiple pending task management requests per I_T_L. This should be changed to a requirement prohibiting more than one pending task management request per I_T_L.

See: Western Digital comment on page 86 of (a) re: SAM page 35, section 4.7.3.1.

Status: No change to SAM

The working group felt that imposing the above restriction would unneccesarily constrain an implementation.

Item 10: CLEAR ACA while an ACA task is active.

SAM now specifies that a CLEAR ACA issued while an ACA is in effect and a task is in the Enabled state should unconditionally abort that task. In this case, a task abort should only occur if the Qerr bit was set.

See: Sun comment T 40 on page 70.

Status: Open.

The working group's instructions were to:

1) Obtain input from those not able to attend the working group. The technical editor will subsequently prepare a

proposal and solicit feedback via the reflector for resolution at the March meeting.

2) Revise SAM to specify that, if an ACA occurs for a command issued with the ACA flag set to 0, the ACA will be unconditionally cleared on receipt off the next command.

Item 11: Protocol specific arguments in remote procedure calls and service calls.

SAM makes no allowance for the inclusion of protocol-specific parameters within the procedure calls specified.

See: Hewlett Packard comment 17 on page 6,
Sun comment T 055 on page 77.

Status: SAM will be modified

The draft will state that an implementation interface may include protocol-specific arguments not described in SAM.

Item 12: Multiple target indentifiers

SAM allows a target device to have more than one identifier but does not address the consequences of such an implementation.

See: a) Sun comments T 008 and T 009 on page 80,
b) Western Digital comment on page 85, RE: SAM page 33, object definition 4

Status: No change to SAM

In the opinion of the working group, it is the responsibility of each protocol standard to fully define the behavior associated with multiple target identifiers.

The following extracts from X3T10/94-028R0 include both the comment and the original proposed reply (preceded by a right arrow).

Item 1:

Begin Sun comment:

T 014 Page 39, Section 6 and page 74, Section 9.1

The Autosense Data, as an output argument, is not defined elsewhere. In particular, it is missing from the definition of section 9.1, the Confirmation returned to the Application Client. The referenced clause should be defined.

>
> The autosense return flag indicates whether or not sense data was
> returned to the autosense buffer. Input from the working group is
> needed to determine the conditions under which autosense
> data is to be returned. ie. Can autosense data be returned for any command
> or only those which complete with a status of CHECK CONDITION or
> COMMAND TERMINATED?
>
> If the latter, then this flag is unnecessary.

End Sun Comment and response

Item 2

Begin Sin comment

T 066 Page 78-81, Annex A

This section should be removed from the standard. It offers nothing to the definitions of the standard.

It does, however, point out that there is some information missing in the definition of task set. I suggest that the information from Table 1 of annex C be included in the normative document, probably in section 4.7.3.2 or section 7.

>

> While the information in the body of the standard must agree with the
=> technical requirements set forth in the queuing model, the editor must
> have the latitude to select the clearest method of presentation.
>
> In that regard, I consider table 1 as illustrative material that

> supplements normative behavior described elsewhere.
>
> In my opinion, therefore, inclusion of table 1 is subject to editorial
> discretion.

End Sun comments

Extracted from Panasonic comments

R1. Though the SAM document has been through several major revisions there is still significant work needed for the document valuable to the industry. This is particularly the case with Annexes A and C which represent
a significant amount of committee effort but are not consistent with the remainder of the document. Concepts like queuing and terminology such as "execute", "task", "response", "confirmation" are not consistent. The document will confuse readers in its present state.

> SAM must, of course, be in full technical agreement with the queuing

> model passed by the committee. Any discrepancies or inconsistencies between
> the body of the document and the queuing model will be corrected. In that
> regard, please cite specific instances where the draft is either
> unclear or at variance with the queuing model.
>
> It is my understanding that once such corrections are made, annex C will be
> deleted. The committees' intent regarding the alternate task set descriptions
> in annex A, however, is not clear to me. I propose that the issue of
> whether or not to retain annex A be resolved in the working group."

R2. The document requires the use of "Per Logical Unit Task Set Boundaries" but discusses and provides for other implementations. This is very confusing. My experience is that these options in a standard will become requirements in the marketplace and therefore should be better documented in the standard. If the intent is to provide extensibility through these options it should be clearly stated.

>

> I believe the discussion of other alternatives in annex A was in
> accordance with the committees' wishes. Please see the previous response.
>

End Panasonic comments

Item 3

Begin Seagate comment

---

52) Should the Group 6 and 7 vendor specific commands now be limited to 16 or less bytes?

>
> I believe that issue should be discussed in the working group. Assuming
> the committee concurs with removing the CDB format defintions from SAM,
>
> I assume this would then become an issue to be addressed by the command
> standards.

End Seagate comment and response

Item 4:

Begin Seagate Comment
---
55) I remain concerned that there may be a conflict with the
installed base which had a presumption of bit significance and the use of the
BUSY bit in the Task Set Full Status code.

>
> Please feel free to raise the issue in the working group.
>

End Seagate Comment And Response

Item 5:

Begin Western Digital comment
---
pg 50, 6.5.3, 2nd pgf.: I don't understand the first sentence.
Why not always detect overlaps? Give a reason in a note or delete the
protocol specific provision and make it global.

>
> The reason for making it a protocol-specified requirement is due to
> the large tag space for protocols like FCP [I should have
> added "and hence the large perceived overhead to search for duplicate
> identifiers". In any event, I believe this is another item for further
> discussion].
>

End Western Digital Comment And Response

Item 6

Begin Hewlett Packard comment

#031 (E) Page 45, Section 6.3, Para 2

TASK SET FULL states that it is required if tagged tasks are

supported. Tagged tasks support is required, therefore remove
the statement about it being optional (the first sentence).

>
> As I understand it, device support for tagged tasks is still optional.
>
> If it is not optional, then devices such as tape drives will have to be
> modified for SCSI-3 compliance. ie. As a minimum, even if such
> devices are limited to one task per logical unit, they will have to
> understand queue tag messages and the like.
>
> I recommend discussing the matter at the next working group.
>

End Hewlett Packard comment and response

Item 7:

Begin Sun comment

T 042 Page 61, Section 7.4

The examples pertain only to those LUN's having a queue
management

[Editor's note: The examples show various task set management
scenarios]

capability. It is also possible that typical tape drives,
sequential data delivery devices (voice recorders, etc.),
mechanical devices (like media changers), and certain other devices may
not allow tagged queuing at all. In this case, the SAM
should absolutely not force the devices to perform queue
management, but rather should allow the synchronous delivery of the
tasks
under control of the initiator's application client.

This is allowed almost everywhere, but an example should be
included in section 7 to demonstrate that it is allowed.

In particular, such devices have a slightly different behavior
with respect to overfilling their very restricted task set. A
second task will be treated as an overlapped command, rather
than a Task Set Full condition. This must be made clear in
sections 6.3 and 6.5.3.

>
> Agreed.
>

End Sun comment and response

Item 8:

Begin Hewlett Packard comment

#014 (T) Page 31, Section 4.6, Para 1

You cannot require that the order be preserved in all cases for a given pair of devices. For example, if I run in fibre channel over a fabric and send two command frames, the order is only guaranteed if I request in-order transmission over the fabric. I may not want to do this for performance reasons. I think that you need to say that order may be "imposable" in the cases where ordered tasks are sent or some other ordering is required, but you cannot require it in all cases.

>
> I believe that, although some physical transports, such as Fibre
> Channel, may reorder data in transit, the ordering specified by the
> sender is restored before the transaction is presented to the consumer.
>
> In any event, ordering is implicit in SCSI-2 today. Therefore
> placing a new ordering burden on the application client (e.g., the
> device driver) may lead to implementations that break existing
> code that would otherwise be portable.

>
> I suggest this issue be left open for discussion
> at the next working group.
>

End Hewlett Packard comment and response

Begin Panasonic comment

R3. I am confused by the requirement in clause 4.6 that all transactions be received in the order they were sent. My understanding was that some of the SCSI-3 transports do not maintain order (i.e. P 1394, Fibre Channel and possibly SSA).
>
> I believe that, although some physical transports, such as Fibre
> Channel, may reorder data in transit, the ordering specified by the
> sender is restored before the transaction is presented to the consumer.
>
> In any event, ordering is implicit in SCSI-2 today. Therefore
> placing a new ordering burden on the application client (e.g., the
> device driver) may lead to implementations that break existing
> code that would otherwise be portable.
>
> I suggest this issue be left open for discussion
> at the next working group.
>

End Panasonic comment and response

Begin Sun Comment

T 007 Page 31, Section 4.6

The last paragraph of section 4.6 indicates that the service delivery transactions are received in the order in which they are sent for a given pair of source and destination devices. Fibre Channel and some other channels allow the proper operation of SCSI with out of order delivery of command information. This restriction should be modified to allow the out of order delivery of commands if operating system or channel conventions can guarantee the proper behavior of the SCSI targets. As an example, ordering of groups of commands can be enforced by the host adapter function or by management of individual commands with respect to the acknowledgment processing of commands requiring ordering.

>
> I had assumed that, while data may arrive out of order, the
> receiver would restore ordering before presenting the data to the
> consumer.
>
> I am concerned that relaxing the ordering requirement in the manner
> suggested will lead to implementations that break existing host code which
> depends on the implicit ordering provided by SIP/SPI.
>
> I would like to reopen this issue at the next working group.
>

End Sun Comment And Response

Begin Sun Comment

T 048 Page 70, section 8.2

What should the Device Server do if an Abort Task for a task with a certain task identifier overlaps or arrives shortly before the task with that task identifier? There is some uncertainty about whether the new task is the one to be aborted or is indeed a new task that is not to be aborted. This is theoretically possible with certain drivers and out-of-order delivery fabrics.

I would suggest that any task with the specified identifier be aborted if it arrives at the device server any time from before the task management function arrives until the service response from the task management function is transmitted by the device server. Any identical task after that time should be considered a new task. A recommendation that task identifiers be maintained unique for some time period after completion of the task would make this requirement more robust.

>

> In my opinion, the behavior specified in SAM should be based on a
> sequential delivery model. An SDS implementation

> may, of course, do what it likes provided the behavior visible to
> application clients, device servers and task managers is sequential.
>
> See the reply to item 007.
>

End Sun Comment And Response

Begin Sun Comment

T 062 Page 76, Section 9.2.1

The last sentence of the section is incorrect as written.
The initiator should assure that the blocks of data
are written into the buffer at the correct displacement
within the buffer, regardless of the order in which the
blocks were actually presented to the interface.

> No, the data must be written in the order received. Otherwise, an
> overlapping transfer that's written out of order will result in
> corrupted data.
>

End Sun Comment And Response

Item 9:

Begin Western Digital comment
---
pg 35, 4.7.3.1: "should" should be "shall", shouldn't it?

[Editor's note:

The referenced text states:

" There are no provisions for aborting, canceling or terminating
a task management function. Thus, an implementation should not
allow an initiator to have more than one pending task management
request per logical unit."}

>
> The present wording was added by request to eliminate what was thought
> to be an unnecessary behavioral restriction.
>
> I suggest reopening the issue at the upcoming meeting of X3T10.

End Western Digital comment and response

Item 10:

Begin Sun comment

T 040 Page 59, Section 7.3

The second sentence of the last paragraph may be overly general. The CLEAR ACA should only abort tasks if the QErr bit is set to one.

> This should be discussed at the next working group. The description
> in the specification reflects inputs from others in the
> working group.
>

End Sun comment and response

Item 11:

Begin Hewlett Packard comment

#017 (T) Page 38, Section 6, General

You also need an optional input and output arguments for protocol specific information. For example, FCP returns more information in status than just sense data and a status byte. It also returns residual byte counts. There is no place for this in the Execute Command primitive.

>
> The intent of the specification is to preserve software portability
> through protocol independence. In my opinion, protocol-specific
> parameters are, therefore, inappropriate for SAM interfaces.
>
>

End Hewlett Packard comment and response

Begin Sun comment

T 055 Page 73, Section 9.1

The SCSI Command Request may be excessively restrictive by defining a closed-ended list of request parameters. Is it implicit that a protocol private input and output parameter list is allowed, or should it be explicitly allowed.

This is also a valid question for the indication, response, and confirmation parameters.

>
> The goal of the architectural model is to specify behavior in a way
> that's protocol independent. Adding protocol- or implementation-specific
> parameters in the manner suggested above will seriously compromise
> that goal and make it impossible to write portable software or

> firmware.
>
> In my opinion, compromising portability will seriouslyjeopardize the
> industry's significant software and firmware investment. Failure to
> protect that investment will discourage migration to newer technologies.

End Sun comment and response

Item 12:

Begin Sun comment

E 008 Page 33, Section 4.7.1

Shouldn't the Initiator equation be:

Initiator = 0{Application Client} + 1{Initiator Identifier}1

If not, a considerable amount of additional information is required to indicate what the rules are for the execution of tasks across multiple independent ports.

>
> Multiple identifiers should be considered as nothing more than
> an alias for the same physical entity.
>
> I would like input from the working group on this issue. According
> to past feedback, multiple identifiers for the same entity were
> considered acceptable.
>

009 Page 34, Section 4.7.2

Shouldn't the Target equation be:

Target = 1{Logical Unit} + 1{Target Identifier}1

If not, a considerable amount of additional information is required to indicate what the rulse are for execution of tasks to a LUN having multiple target ports within a single task.

>
> See reply to comment 008.
>

End Sun comment and response

Begin Western Digital comment

. . .

Also (more importantly) all of the tools and services provided by SAM seem to allow for multiple Initiator and Target Identifiers only on the most simplified level: you can have more than one, but you can't relate one to the another within the same device. Given this, why don't we say:

Initiator = 1{Appl Client} + 1{Initiator Identifier}1

Likewise for target. The alternative is adding considerable
complexity to make multiple IDs fully functional.

>
> Multiple I/D's are supposed to be aliases representing the same physical
> device.
>
> Since this was added by request, I'd like some feedback from the
> working group before I modify the definition. In my opinion, the
> issue of multiple target identifiers is irrelevant to SAM.
> Although the behavioral model requires one unique identifier,
> a system could implement more than one without violating any
> architectural requirements.
>
>

End Western Digital comment and response