

Date: Oct,31 1993

X3T9.2/93-164r0

To: X3T9.2 Committee (SCSI)

From: George Penokie (IBM)

Subject: Comments on SAM Rev 12

Note: Comments marked with an * what I consider to be non-editorial issues, questions, etc.. I am having trouble seeing how I can vote yes on forwarding SAM with as many *s as I have.

- 1 * There is a fundamental change between the Queueing Model and the way SAM defines Queueing. It is that the Queueing Model assumes the target is the 'owner/manager' of the task sets. This allows the possibility of task set boundaries other than on a per Logical Unit with no change to rules of the queuing model other than the boundary definition.

SAM defines the 'owner/manager' of task sets to be the device server. There is one device server per Logical Unit. And the device server has no knowledge of or control over other logical units within a target. Because of this there is no way a device server could manage a task set which crosses logical units.

There is nothing 'wrong' with the way SAM has defined the task set control because the 'one task set per logical unit' was what the committee voted to support. I am only pointing out that with SAM it will require a much greater effort to change to a different or support for multiple task set boundaries. In other words we are burning bridges.

- 2 * Through out SAM there are references to the confirmation of services and the response to services. This is confusing: Are services confirmed or responded to?
- 3 Page 10 Section 2: TBD Must be removed
- 4 Page 11 Section 3.1.11: The sentence 'A command has than has..' should be changed to 'A command that has...'
- 5 Page 12 Section 3.1.21: The sentence '...for the next in a series...' is not clear and should be changed to '...for the next task in a series...'
- 6 * Page 12 Section 3.1.26: This definition implies that application clients are part of initiators. I do not agree. application clients are the originators of commands and initiators do not originate commands.
Page 23 section 4.2.1: Again refers to the application client residing in initiator devices.
- 7 Page 12 Section 3.1.28: What is an integrated path? I can find no definition of it.
- 8 Page 12 Section 3.1.30: The sentence '...commands executed by a single task..' is not correct. Tasks do not execute things. I suggest changing 'executed' to 'contained within' or 'bounded by'.
- 9 Page 13 Section 3.1.41: The sentence '...devices in a

- domain.' should be changed to '...devices within a domain.'
- 10 Page 14 Section 3.1.61: Why is this definition even in SAM? None of the other protocols are defined. If this definition stays then all the other protocols need to be defined. If it stays the the definition should be 'A protocol defined by the SIP standard.' nothing more.
 - 11 Page 14 Section 3.1.63: This is a GS definition. Remove everything after '...device delivery subsystem.'
 - 12 Page 15 Section 3.1.73: What is a 'constituent system'? What is a 'related system'? What has this definition have to do with SCSI?
 - 13 Page 15 Section 3.1.74: The sentence '...the data was addressed.' should be '...the data is associated.'
 - 14 Page 15 sections 3.1.77 and 3.1.79: The first word of both sentences should be 'An'.
 - 15* Page 15 section 3.1.84: The logical unit does not define the boundaries of the task set. The sentence '...grouped by the logical unit so...' should be '...grouped so...'
 - 16 Page 15 section 3.1.85: Task slots do not 'represent' tasks. I think think of task slots 'holding' tasks.
 - 17 Page 19 figure 1: The is a line missing on the 'section' block.
 - 18* Page 33 section 4.7.1: Pending is being used but it is not clear what is meant. What does not mean to have a 'pending task management function'? Depending on the definition of 'pending' it may or may not be possible to have pending task management functions.
 - 19 There seems to be an inconsistency in the usage of 'task' and 'command'. If I read the definitions of each it would not be possible for the application client to deal with tasks. But for linked commands I know this not the case because there are many commands which make up a single task. So if I ignore the definitions then the description of 'Application Client' on page 33 is OK. But now I go to page 34 and read the description of 'Logical Unit' and it talks about commands not tasks. How is anyone supposed to understand this.
 - 20 Page 34 section 4.7.2: The sentence in 'Logical Unit' '...request are directed.' should be '...requests are directed.'
 - 21 Page 35 section 4.7.3: The sentence in 'Device Server' should be '...SCSI commands and manages the task set.'
 - 22 Page 35 section 4.7.3.1: The last sentence in the second para. the 'should' should be 'shall'. And here is the undefined pending task management again.
 - 23 Page 36 section 4.7.3.2: In the 'Tagged task ' section: Why not make life a little less complicated and change 'initiator-specified component (tag)' to 'tag' or at least 'initiator-specified tag'.

- 24 Page 39 section 6: In the 'autosense data' section there is an undefined cross reference.
- 25 Page 39 section 6: In the 'Austosense Return Flag' section there is a missing ')'
- 26 Page 39 section 6: In the 'Status' section: When is the status undefined?
- 27 Pages 40,41, 42, 43, and 44: All the Tables on these pages are messed up.
- 28* Page 44 table 8:The committee requested that this table be changed to remove the possibility that anyone would interpret status codes as being bit sensitive but coding the statuses as hex values.
- 29* Page 45 section 6.3: In the 'ACA Active' section the two places where 'command' is stated should be 'task'.
- 30* Page 46 section 6.4.1: Bullet number 2 talks about a task being created when it is received by the device server. Again; how can the application client know about some that does not exist?
- 31* Page 49 section 6.5.2.1: Paragraph 2: The sentence '...was created by the initiator...' should be changed to '...was created for the initiator...'. An ACA is not created by an initiator, it is created for an initiator.
- 32* Page 49-50 section 6.5.22: This is section is a disaster. There appears to be an attempt to redefine how SCSI-2 CA should work and it is not right. I will argue that it is a bad idea to try to define a SCSI-2 function is SCSI-3. We should just say 'see SCSI-2'. Look at the 'Queueing Model document to see how this is handled'
- 33* Page 50 section 6.5.3: This section is not reflect the committees desires for overlapped commands. See section 2.2 of the queueing model for the latest information.
- 34 Page 50 section 6.5.4: Bullet b - The sentence ' ...the logical unit...' should be '...the logical unit...'
- 35* Page 52 section 6.5.7: Is Asynchronous event reporting per target?
- 36 Page 53 section 6.6: In the first paragraph there is a '[to be specified]': What is that all about?
- 37 Page 53 section 6.6: Bullet h: What is 'Any other event'? That statement could imply that an unit attention could legally occur on any event. (eg command complete, check condition, etc.)
- 38* Page 54 section 7.1: The 'Ordered Blocked Boundary' section looks to be an attempt to combine the 'Ordered Blocking' and 'Task Set Boundaries' concepts. These two concepts are important to understand and should be separately defined.
- 39 Page 55 section 7.2: In the 'task abort' section and elsewhere the message names have changed from Queue to Task. I do not think this is a good idea and yes in know task is

the 'correct' term but if this change is made it will be forever confusing.

- 40 Page 56 section 7.2 last sentence: The sentences read '... to the commands it completes...' What is the 'it' this section of the sentence is referring to?
- 41 Page 58 table 58: In the Dormant row under the 'All prev. HOQ and previous ORDERED tasks ended' column 'tasls' should be 'tasks'.
- 42 Pages 61,63,65, and 67: The Black background with white lettering is in many cases almost unreadable.
- 43* Page 73 section 9.1: The send SCSI command looks allot like the execute command service response but some of the parameters are in different locations. Are there supposed to be two different services that are almost the same? If so why?

Subject: Bob Snively's SAM comments
To: SCSI Reflector List <scsi@WichitaKS.NCR.COM>
Date: Tue, 2 Nov 93 8:45:28 MST
From: John Lohmeyer <jlohme@ncr-mpd.ftcollinsco.NCR.COM>
Cc: Bob Snively <Bob.Snively@eng.sun.com>
Extracted-To: SCSI_Reflector

I received the following comments from Bob Snively on the SAM letter ballot (to forward to X3T9 for first public review). He asked me to distribute his comments to the SCSI Reflector. (I did not assign a document number as Larry maintains the document register.)

John

"Bob Snively wrote:"

John,

My response to the SAM ballot is negative for the reasons defined in the following memo. I will be bringing my signed ballot with me to the meeting next week. With your permission, I will send out my comments on the reflector, or you could do so after assigning a document number.

Bob Snively

To: John Lohmeyer , X3T9.2 Chair
From: Robert Snively
Date: October 26, 1993
Subject: Review of SAM, X3T9.2/994D, Revision 12

Dear Mr. Lohmeyer,

I regret to inform you that I must vote to not forward the SAM document. The following changes to the standard would change my vote to an affirmative vote.

T 046 Page 70, sections 8.2 - 8.7
(Modification required to change vote to affirmative).

It is clear from the document that each protocol shall be required to provide a mechanism to perform each of the task management functions. In addition to this, it must be made absolutely clear which of the task management functions are optional for a SCSI device to implement and which ones are required. Some task management functions may only be required if certain other optional capabilities are allowed.

Terminate Task is an example of a function that is always optional.

Clear Task Set is an example of a function that is only required if the task set elects the definition:

Task Set = {0(Tagged Task) + 0(Untagged Task)}

Clear Auto Contingent Allegiance is an example of a function

that is only required if the ACA bit is allowed to be set to one.

These are probably best placed in a new paragraph under each Task Management Function entitled "Service Response"

A typical case would be for Terminate Task:

Service Response:

Function Complete: Indicates Terminate Task Function was accepted and will be attempted by Device Server

Function Rejected: Indicates Terminate Task Function not implemented by Device Server

Failure: Indicates Terminate Task Function could not be delivered to Device Server

A contrasting case would be for Clear Task Set:

Service Response:

Function Complete: Indicates Clear Task Set Function was accepted and will be attempted by Device Server.

Function Rejected: This response is only allowed for Device Servers that reject all Tagged Tasks.

Failure: Indicates Clear Task Set Function could not be delivered to Device Server.

Table of desired optionality:

Task Management Function	Opt/Rqrd	Conditions
Abort Task	Rqrd	
Abort Task Set	Opt supported	Rqrd if Tagged Tasks
Clear ACA	Opt supported	Rqrd if ACA bit = 1
Clear Task Set	Opt supported	Rqrd if Tagged Tasks
Target Reset	Rqrd	
Terminate Task	Opt	

T 068 Page 86-94, Annex C
(Modification required to change vote to affirmative).

This section is redundant with and disagrees with the normative standard. It must be removed from the standard. It does not contain the necessary information about the management of tasks during the various states, information which is correctly included in the draft.

The problems related to Annex C indicate information discrepancies and describe an appropriate correction for each discrepancy.

In addition to those key changes, I believe that the following editorial and technical changes should be included to make the document clearer, less subject to misinterpretation, and would correct some technical problems. Technical problems are preceded by a T, while editorial recommendations are preceded by an E. The problems are ordered by page number. The critical problems are duplicated in the section above this paragraph.

E 001 Page 9 and 10, Section 1.0

The reference document identifications should be completed, including at least the official title and project draft number.

E 002 Page 11, Section 3.1.11

The sentence should be corrected in the area: "command has than has ended".

E 003 Page 16, Section 3.1.90

The word "word" should be re-examined. If all goes well, a search throughout the document will reveal that the word is either never used or can always be replaced with something like "4-byte value", avoiding ambiguity.

E 004 Page 16, Section 3.3

From context, it appears that lower case is used both for words having their normal English language meaning and for words other than SCSI commands, statuses, and acronyms. The last sentence should either be deleted or the distinction between lower case and upper case significance made clear.

E 005 Page 23, Section 4.2.1 and Page 33, Section 4.7.1

Page 23 indicates in the next to the last paragraph that an application client controls one or more tasks. Page 33 indicates in the definition of application client that there is one application client for each task or pending task management function.

It is my impression that there is a one-to-one relationship between application clients and tasks and that each new task is created by a new instance of an application client. That would make page 33 correct and page 23 incorrect. This should be clarified in both locations and corrected in one.

E 006 Page 24, Figure 4

Figure 4 uses a layering model which does not refer to the layering terms defined immediately below it. I believe that the LLP and ULP terms should be properly assigned to the figure. That either requires relabeling or may require that the central layer be broken into a ULP layer and an LLP layer.

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.

E 008 Page 33, Section 4.7.1

Shouldn't the Initiator equation be:

$$\text{Initiator} = 0\{\text{Application Client}\} + 1\{\text{Initiator Identifier}\}1$$

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.

E 009 Page 34, Section 4.7.2

Shouldn't the Target equation be:

$$\text{Target} = 1\{\text{Logical Unit}\} + 1\{\text{Target Identifier}\}1$$

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

T 010 Page 35, Section 4.7.3

LUN 0 should always exist, at least to the point of being able to respond to TIO, INQUIRY, and REQUEST SENSE. The LUN need not be installed.

T 011 Page 36, Section 4.7.3.2

The Task Set should have only one Untagged Task per initiator according to previous definitions. The definitions in this document allow more than one, since an untagged task can also have an identifier. If it has an identifier, isn't it tagged?

T 012 Page 37, Section 4.7.3.2

The last sentence implies that the only mechanism known at the service layer and at the communication level might be the Task Identifier. In fact, the Task Identifier known at the service layer may have an arbitrary mapping to the Task Identifier information actually passed across the protocol interface. As an example, CAM knows a service request by the pointer to the CAM Control Block, but a SIP implementation uses the initiator identifier, target identifier, LUN, and Tag to perform the identification of the same IO Process. This should be clarified.

T 013 Page 39, Section 6

The Data-In Buffer Pointer is an input accompanying the Execute Command, not an output. It identifies the buffer that will be used when the Data In Delivery Service is executed.

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.

T 015 Section 6 and Section 9

The Command Model and the Service Model appear to be

partially uncorrelated and redundant. I would suggest that Section 6.1 be replaced by Sections 9.1 and 9.2, with appropriate complementary information retained from section 6.1. That would become a new section 6.

I would then suggest that section 9.3 be moved to become the new section 7.

The sections 6.1 - 6.6 would become the new section 8 and section 7 would be renumbered to section 9.

This would restructure the document to present the model information in conjunction with the service model that implements it.

A summary of the chapter titles would be:

5: Procedural model for commands and task management functions

6: SCSI Command Model and Services

6.1: Data Transfer Services

7: SCSI Task Management Model and Services

8: SCSI Command and Protocol

9: Task Set Management

10: Task Management Functions

If this cannot be done, a great deal of editorial work must be done to show the relationship between the Command Model and the Service Model and to correlate them.

E 017 Page 39, Section 6

The Autosense Return Flag should be labeled the Autosense Returned Flag.

E 018 Page 40, Tables 1, 2, and 3.

Using unjustified, uncentered, courier type would make this work better. Alternatively, it should be presented using the table functionality of the word processor.

T 019 Page 43, Section 6.3

Status is also not presented after Abort Task and Abort Task Set.

T 020 Page 47, Section 6.4.2

In the execution of linked commands, step 5, the accessing of the second command, is normally created by actions of the target. This is especially obvious in SIP, where the target manages the command phase of the second command directly. It is also true for SBP, where the target fetches new commands from initiator memory and for the FCP, where initiative is passed to the initiator to enable the output of the command information.

T 021 Page 48, Section 6.5.1

The Programmable Operating Definition is not a characteristic

of SAM or of any protocol. It can actually be used to influence only those things that are visible at the command set level, and so should only be described in the command set. In particular (ref paragraph 2), the SCSI compliance level, SCSI specification level, and other parameters should be explicitly banned from changing, since these parameters reflect details like the use or non-use of an Identify message, the use of extended Identify messages, and the width of data transfers, items which may extend into the requesting and the responding services.

Among the other parameters which should not be changeable are included:

Vendor Identification: Changing this parameter allows fraudulent replacement of disk drives with non-conforming disk drives.

Vendor Serial Number: Changing this parameter allows fraudulent warranty repairs to be sought.

The technical solution to this problem is to remove the section from SAM and to place it in the appropriate section of the SPC document. The CHANGE DEFINITION command should then be described in such a manner as to prohibit modifications to anything other than command level behavior and to prohibit modifications that create the opportunity for fraud.

T 022 Page 48, Section 6.5.1

The text, when taken over to SPC, should contain a precise indication of the state of the machine necessary to accept and execute a CHANGE DEFINITION command. That state should be that no tasks for any initiator are present in the task set of the logical unit. This can be accomplished by reserving and quiescing the logical unit through normal command set operations. The CHANGE DEFINITION command should be rejected if any tasks are in the task set for the destination LUN.

The text, when taken over to SPC, should contain a precise indication of when the new definition becomes active. It should become active at the time that the service response indicating successful completion of the CHANGE DEFINITION command is transmitted from the LUN.

E 023 Page 49, Section 6.5.2.2

The definition of "hard reset" should be documented.

E 024 Page 50, Section 6.5.3

According to Annex C and SCSI-2, the proper response for overlapped commands should be a sense key of "ILLEGAL REQUEST", not "ABORTED COMMAND".

E 025 Page 50, Section 6.5.4

Item B, typo: "unti" should be "until"

T 026 Page 52, Section 6.5.7

It should be clearly indicated that AEN is an optional behavior, both by the target and by the initiator. While a protocol is required to architect the capability, no device should be

required to implement the capability.

T 027 Page 52, Section 6.5.7

In the last paragraph of the page, the text indicates that AEN should be reported only once per occurrence of the causing event. In fact, for errors that are generic and may influence the operation of any attached initiators, the AEN should be presented to all attached initiators. The text should be modified to clearly indicate that AEN should be offered only once to the initiator related to the command causing the failure, but is allowed to be offered to every initiator if it is unclear to the target which initiator will be affected by the failure.

T 028 Page 52, Section 6.5.7

AEN is defined in SCSI-2 as sharply limited in its capabilities. It is always executed as an initiator-to-target operation, where the target is identifiable through the INQUIRY command as a Processor Type device. The information is sent as a SEND command to LUN 0 of the device.

In particular, AEN SEND cannot be generated to a target that identifies itself as other than a Processor Type device because the command overlaps the decode of a WRITE(6) in other command sets.

Text should be installed in section 6.5.7 describing this characteristic of AEN.

E 029 Page 53, Section 6.6

The events, items a-h, should include additionally those unitemized events in the first paragraph, including TARGET RESET, hard reset, and power-on reset.

T 030 Page 54, Section 7.1

The definition of "current task" and therefore "pending task" varies somewhat from the definition contained in Annex C. In Annex C, the current task is defined as a task that is moving data to or from the physical transport, meaning in the case of disk drives and tapes, that data is being moved to or from the magnetic storage medium. In fact, the term "current task" is not used in section 7 and should probably be dropped. Of the three terms used, only one has meaning in the clarified task management structure, and that is Ordered Blocking Boundary. That term should probably be moved into section 7.2 and section 7.1 should be collapsed out of existence.

E 031 Page 54, Section 7.2

First paragraph, undetermined information missing in second sentence.

E 032 Page 55, Section 7.2

"All Previous tasks complete" event needs to be clarified. In particular, it cannot occur for simultaneously enabled tasks and is only true at ordered blocking boundaries.

E 033 Page 55, Section 7.2

The definition of the QErr bit must be obtained from SPC and installed here, since it is not clear what meaning it will have without it's being defined.

T 034 Page 55, Section 7.2

Task abort should include other causes of task termination, including hard reset and power on reset.

T 035 Page 55, Section 7.2

There does not appear to be a state for tasks that are on-going behind the scenes, including immediate commands and tasks that have been aborted as far as the SCSI interface is concerned, but are continuing on to completion to bring the logical unit to a known state. These may not actually be treated as tasks, but rather as peer applications started by tasks. In any case, mention of their allowability should be made in section 7.2.

T 036 Page 57, Section 7.3

Restricted reordering appears to be defined incompatibly with SCSI-2. In SCSI-2, the restricted reordering only requires that the final value of all data observable on the medium shall have exactly the same value as it would have if the commands had been executed in the same received sequence without using tagged queueing (which in SCSI-2, meant "received synchronously".) SAM appears to require the execution in precisely the specified order in the case of restricted reordering. This is an overly strict requirement and should be modified to the SCSI-2 definition. As an example, that means that READ operations having the SIMPLE task attribute performed under restricted ordering may be executed in any order.

This must also be corrected in paragraph 2 of page 58, same section.

E 037 Page 57, Section 7.3

The state diagram applies to each single task, but does not express the relationship among tasks. This should be clarified in the title and explanations.

E 038 Page 58, Section 7.3

The last paragraph before Table 9 has a typographical error in the last sentence. The word "be" should be deleted.

T 039 Page 58, Section 7.3

The restrictions on head of queue task enabling are more correct than those in section 7 of Annex C, which are probably unenforceable.

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.

T 041 Page 60, Section 7.3

During an ACA condition, only properly offered ACA tasks are enabled. All other tasks are rejected. This is not made clear in the text or table.

Alternatively, it may be required for a task to enter the enabled state for it to be rejected. If this interpretation is correct, the state diagram should be modified to contain it.

T 042 Page 61, Section 7.4

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

T 043 Page 63/64, Section 7.4.2

It is not clear that tasks 4,5, and 6 should be enabled before task 7 (a head of queue task) is completed. I would not expect any new tasks to be enabled until all head of queue tasks are done. See step 3 in the sequence.

T 044 Page 67, Section 7.4.4, step 1

The text should read:

- 1) Since restricted ordering is in effect, simple task 1 can be enabled. If the effects of enabling simple task 2 would cause a violation of the restricted reordering algorithm, it must remain dormant until simple task 1 is completed.

It is likely that, if both simple task 1 and simple task 2 are read operations on a disk drive, they can both be enabled and still meet the restricted ordering requirements.

T 045 Page 68, Section 8

SIP only provides an indication that the task management function has been delivered, not that it has been completed. It is possible that the Function Complete service response is stepping beyond the original definitions from SCSI and should be softened to an accepted delivery indication.

T 046 Page 70, sections 8.2 - 8.7
(Modification required to change vote to affirmative).

It is clear from the document that each protocol shall be required to provide a mechanism to perform each of

the task management functions. In addition to this, it must be made absolutely clear which of the task management functions are optional for a SCSI device to implement and which ones are required. Some task management functions may only be required if certain other optional capabilities are allowed.

Terminate Task is an example of a function that is always optional.

Clear Task Set is an example of a function that is only required if the task set elects the definition:

Task Set = {0(Tagged Task) + 0(Untagged Task)}

Clear Auto Contingent Allegiance is an example of a function that is only required if the ACA bit is allowed to be set to one.

These are probably best placed in a new paragraph under each Task Management Function entitled "Service Response"

A typical case would be for Terminate Task:

Service Response:

Function Complete: Indicates Terminate Task Function was accepted
and will be attempted by Device Server
Function Rejected: Indicates Terminate Task Function not
implemented by Device Server
Failure: Indicates Terminate Task Function could
not be delivered to Device Server

A contrasting case would be for Clear Task Set:

Service Response:

Function Complete: Indicates Clear Task Set Function was
accepted and will be attempted by Device Server.
Function Rejected: This response is only allowed for Device
Servers that reject all Tagged Tasks.
Failure: Indicates Clear Task Set Function could not
be delivered to Device Server.

Table of desired optionality:

Task Management Function	Opt/Rqrd	Conditions
Abort Task	Rqrd	
Abort Task Set	Opt supported	Rqrd if Tagged Tasks
Clear ACA	Opt supported	Rqrd if ACA bit = 1
Clear Task Set	Opt supported	Rqrd if Tagged Tasks
Target Reset	Rqrd	
Terminate Task	Opt	

T 047 Page 70, section 8.2, 8.3

Should the Abort Task or the Abort Task Set function including

a task with the ACA attribute also abort the ACA condition?

At present, the answer is it should not be.

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.

T 049 Page 70, section 8.4

When does the Clear ACA take effect? I propose that it becomes effective immediately upon the arrival of the Clear ACA request.

T 050 Page 71, section 8.5

The phrase "All data for all terminated tasks shall be cleared." is ambiguous. I assume that this includes both state information and the contents and identification of data buffers. In particular, if a WRITE operation was active when a Clear Task Set was taking place, the data that had been transmitted to the disk, even if incomplete, would be the data returned by a subsequent READ. The buffered data, even if incomplete, would not be returned by the READ.

T 051 Page 71, Section 8.6

The definition of hard reset is not present in the SAM document.

T 052 Page 71, Section 8.6

In SIP, the target address, rather than the Logical Unit, is the destination of a TARGET RESET function. It is probably reasonable to change the definition to LU Identifier, but this is definitely a change in the structure of SCSI. I would further propose that for those devices having a hierarchical Logical Unit structure as described in the RAID model, that the TARGET RESET be defined to reset not only the specified Logical Unit, but all Logical Units depending from that Logical Unit in the hierarchy.

T 053 Page 72, Section 8.7

The definition of corrupting the medium should be clarified. In particular, none of the Abort functions are expected to stop so quickly that incorrect CRC is written on the recording medium. All of the functions, including Terminate Task, leave the medium in an ambiguous state in that the operation is incomplete. Properly implemented, Terminate Task could provide the necessary information to make it possible for the

host to determine the nature of the medium corruption, but cannot avoid the corruption.

T 054 Page 72, Section 8.7

In the third paragraph, item 1, there is a discussion of the residue of an allocation length or parameter list under the conditions of a Terminate Task. Terminate task has some meaning with respect to data transfer tasks, where there is a long operation which will be interrupted. For tasks that are instantaneously executed or which require the complete parameter list for execution (MODE SELECT and similar commands), the meaning of Terminate Task should be clarified. I believe that the command should either not begin execution or should complete execution regardless of how many bytes have been transferred.

If the task completes successfully, it should not indicate TASK TERMINATED status, but should indicate function complete (GOOD status).

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.

T 056 Page 73, Section 9.1

I would have expected the Data In Buffer Pointer and the Autosense Buffer Pointer to be inputs to the requests, since they are pre-allocated areas that are being made known to the executor of the services at both ends of the link.

T 057 Page 73, Section 9.1

The autosense data returned in the area defined by the Autosense Buffer Pointer may include protocol dependent wrapper information and status in addition to the sense data.

T 058 Page 75, Section 9.2

The application client's buffer is not actually "a single, physically contiguous block of memory". It is actually a contiguous block of virtual memory or a logically contiguous block of memory.

T 059 Page 75, Section 9.2

The device server is described as "must have the ability to transfer such data in increments smaller" than the total extent to be transferred. In fact, there is no such requirement, but should be reworded to indicate:

"The model also requires that the protocol be capable of transferring data in increments smaller than the extent specified in the command descriptor. The device server may have the requirement to transfer data in such smaller increments."

T 060 Page 75, Section 9.2

In the definition of Command Byte Count, it should additionally be clarified that verification of complete and correct information transfer may be impossible when data transfer segments overlap. A protocol should be allowed to prohibit overlapped transfers.

E 061 Page 76, Section 9.2.1

Third line, missing comma.

T 062 Page 76, Section 9.2.1

The last sentence of the section is incorrect as written. The initiator should assure that the 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.

T 063 Page 76, Section 9.2.2

The Device Server Buffer Pointer is managed by the Device Server making the request and is not necessarily an input or output parameter of the request.

T 064 Page 76, Sections 9.2.1 and 9.2.2

The transfers should not be considered successful unless the number of bytes transferred exactly matches the requested byte count. In the case of a write, an incomplete byte count can only occur in the block transferring the highest offset byte of data and indicates an incorrect length, which must be posted back to the SCSI initiator according to the rules for incorrect length. In the case of a read, such a mismatch in byte count is an error in the transfer process.

T 065 Page 77, Section 9.3

This section appears to be redundant with section 8. See problem 15.

The "result" parameter codes disagree with those defined in Page 68, section 8.

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.

E 067 Page 82-85, Annex B

This section should be removed from the standard. Subsequent updates should be documented with an appropriate cover letter to the draft standard and not within the standard document itself.

T 068 Page 86-94, Annex C
(Modification required to change vote to affirmative).

This section is redundant with and disagrees with the normative standard. It must be removed from the standard. It does not contain the necessary information about the management of tasks during the various states, information which is correctly included in the draft.

The following problems indicate information discrepancies and describe an appropriate correction for each discrepancy.

T 069 Annex C

The description of the ACA control bit in Annex C is incomplete and has been correctly supplanted by the normative information contained in the body of the draft.

T 070 Page 87, Annex C, section 1.1

"The target shall manage....all other task sets." is the information that should be transferred to section 4.7.3.2 as described in problem 066

T 071 Page 87, Annex C, section 2.1

This section is not correct and is supplanted by the draft.

T 072 Page 88, Annex C, section 2.1.4

"If the target accepts ... complete the current task." is correct information not presently explicit in the body of the draft. It should be added to section 8.4.

T 073 Page 89, Annex C, section 2.2

The list of task terminations may be an appropriate addition to section 7. Section 7 is a little vague about the mechanisms by which a task is ended.

T 074 Page 89, Annex C, section 3.1

The second paragraph indicates that more than one current task may exist at a time. This really means that more than one task may be enabled (and therefore in some state of execution) at one time.

The third paragraph indicates that more than one task may be active on a physical transport system at once.

While implicit in the body of the draft, it might be helpful to make these points explicit, possibly in section 7.3.

T 075 Page 91, Annex C, Table 1

The information contained in this table should be included in section 4.7.3.2 or section 7 as proposed in problem 66.

To: scsi@WichitaKS.NCR.COM
Subject: SAM Rev 12 Comments
Date: Wed, 03 Nov 93 13:24:54 MST
From: Jeff Williams <jlw@hpbs1506.boi.hp.com>
Extracted-To: SCSI_Reflector

SAM Review Comments
Revision 12

#001 (E) Page 2, Ed Note

Remove the editors note.

#002 (E) Page 3, 3.4 and 4.7.3.2

Fix formatting of the lines (make the17 and36 on the same lines as the 3.4 and 4.7.3.2).

#003 (E) Page 9, Section 1, Bullet 10

SSA-SCSI should be SSP. Remove the question mark.

#004 (E) Page 10, Section 1, Para 6

Add a definition to the SMC section as was done in all of the previous sections (e.g. SPC, SBC, SSC).

#005 (E) Page 10, Section 1, Para 12

Again, not SSA-SCSI, but SSP. Also, add that SSP runs over the SSA-PH protocol.

#006 (E) Page 10, Section 1, Para 14

SPI is the SCSI-3 Parallel Interface, not Interconnect.

#007 (I) Page 10, Section 2

This section must be filled in. I have a problem forwarding any document with a TBD section. This should be a minor item to correct.

#008 (I) Page 11, Section 3.1, General

Use the agreed upon definitions from 92-141R8 wherever applicable.

#009 (E) Page 19, Figure 1

The "Section" box is missing a piece of the box.

#010 (E) Page 20, Section 3.5, Para 2

Shouldn't the "[input-1,input-2,...]" be "{input}" indicating zero or more input parameters. This is consistent with the notation in the previous section.

#011 (E) Page 21, Section 4.2, Para 2

"...and receives a completion response. or a failure notification." should be "...and receives a completion response or a failure notification.".

#012 (T) Page 23, Section 4.2.1, Para 2 and 5

Paragraph 2 states that there is one application client for each task. Paragraph 5 states that an application client controls one or more tasks. Which one is correct? I thought that the first case is correct - that there is one and only one task per application client.

#013 (T) Page 27, Section 4.3.3

I do not think that we need an LLP Service. First, it is not a good model in that a ULP should never control an LLP at another node. Second, the premise that data transfer is not a ULP request is not true. Data has to go somewhere in the control of the ULP. The ULP does not have to give up control to the LLP for this transfer to take place. I do not see the need for this section.

#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.

#015 (T) Page 35, Section 4.7.3.1, Para 1

The task manager controls the execution and "responds to" task management functions. It does not control execution "in response to" task management functions. The latter implies that a task management function is always required to control a task, which is not the case.

#016 (E) Page 38, Section 5, Para 2

The "Object Identifier" is not included in the description in section 3.5. Also, this appears to be a duplication of the description in section 3.5. Remove this section if it is a duplicate.

#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 ExecuteCommand primitive.

#018 (T) Page 39, Section 6, Para 4

The last sentence states that the buffer is undefined for a status such as CHECK CONDITION. This is not true in all cases. I can return data from a read and then run into an error and return CHECK CONDITION. The data is valid in the buffer and may even contain the block in error if the TB is set. I may also have recovered all data via ECC and returned the data along with a CHECK CONDITION indicating that the data was recovered with ECC. In either case the data is valid.

#019 (E) Page 39, Section 6, Para 5

Remove the "???". Replace it with 6.5.8.

#020 (E) Page 39, Section 6, Para 6

Typo. No close paren on the paragraph.

#021 (T) Page 39, Section 6, Para 11

Again, the buffers may be valid even though a CHECK CONDITION status was returned. Remove this restriction.

#022 (T) Page 40, Section 6.1.1, Para 1

The wording from SCSI-2 to this document changed so that I am no longer required to check any reserved fields! In SCSI-2, it states that, "A target that receives [reserved fields] that are non zero...". In SAM it states, "A logical unit that detects a reserved field that is not zero..." These are totally different! According to SAM, I don't have to do anything unless I check and detect the non-zero fields. This is less restrictive than SCSI-2.

#023 (T) Page 40, Section 6.2, Para 3

Are there specific commands which will alter the medium when errors are present in the command descriptor block or parameter list? This was not allowed in SCSI-2. Why is it allowed here?

#024 (E) Various

Clean up the Tables to align correctly.

#025 (E) Page 42, Section 6.2.1, Para 3

Add references to tables 1, 2, and 4 like was done for table 3.

#026 (E) Page 42, Section 6.2.1

Why aren't logical block address, transfer length, parameter list length, and allocation length described?

#027 (T) Page 43, Section 6.2.2, Para 2

The flag bit is optional even if the link bit is supported. There is a new requirement in this paragraph which states that the flag bit is required if the link bit is supported. This was not required in SCSI-2.

#028 (E) Page 43, Section 6.2.2, Para 5

Typo. Extra period in the last sentence.

#029 (T) Page 43, Section 6.3, Para 2

The list should also include ABORT TASK and ABORT TASK SET.

#030 (T) Page 43, Section 6.3

SCSI-2 allowed the status to be sent prior to the CDB. This clause was removed in SAM. It should be included in appropriate wording for the protocols (e.g. don't use terms like status phase).

#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).

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

Also add into the TASK SET FULL status that the task is discarded.

#033 (T) Page 48, Section 6.5.2

There is a paragraph from the 92-141R8 document missing from this clause. This paragraph described the reaction to an ACA condition being created within an ACA by an ACA tagged task. Include this paragraph.

#034 (E) Page 49, Section 6.5.2.1, Para 3

This is only the case if the ACA bit was set in the control byte. If the ACA bit was clear, this should be an illegal queue type. I assume that the ACA bit means that the error handling is just like in the SCSI-2 standard which means that the next command clears the CA. By the way, auto sense should be treated as though it was a Request Sense after the CA went into effect thus

clearing the CA.

#035 (I) Page 49, Section 6.5.2.2, Last set of bullets

If the ACA bit was zero, then the error handling is strictly SCSI-2 and should not be defined further. The ACA bit was meant to provide a backwards compatibility. Let's not redefine what that means. I think all that needs to be said is that auto sense is treated as a command following the CA condition. This should be stated in the section on auto sense.

No additional information should be given for an ACA bit cleared.

#036 (E) Page 50, Section 6.5.3

This entire section should be replaced with the section from 92-141R8. There are a number of additional reasons why the tags may be reused. In addition, the use of the two ASCs is explained in 92-141R8.

#037 (E) Page 50, Section 6.5.4, Para 5

General comment. Whenever sense data is discussed, mention it in general terms of the sense key, additional sense code, etc. Do not say, "In response to a Request Sense command...." since auto sense makes this an invalid statement.

#038 (E) Page 50, Section 6.5.4, Para 6

Typo. "Logical unti" should be "Logical unit".

#039 (E) Page 53, Section 6.5.8

State that the auto sense acts as an implied Request Sense command in order to eliminate the confusion on handling the ACA bit cleared in the control byte.

#040 (E) Page 53, Section 6.6, Para 1

Why is hard reset condition "[to be specified]"?

#041 (E) Page 53, Section 6.6, Para 9

Bullet (h) should be "Any other event that requires the attention of the initiator". Leaving it as "Any other event" tells me to generate a Unit Attention when anything happens!

#042 (E) Page 54, Section 6.6, Para 1

What if the Inquiry has been sent after the ACA was created? In SCSI-2 it was cleared. Is that what happens here also? I assume that the Inquiry is treated like any other command and that an ACA ACTIVE status is sent if it is not tagged

with an ACA Tag Type.

#043 (I) Page 54, Section 7.1, Para 2

The definition of suspended information should be the one from 92-141R8.

#044 (E) Page 54, Section 7.1, Para 3

The definition of current task should be the one from 92-141R8.

#045 (E) Page 54, Section 7.1, Para 4

The definition of pending task should be the one from 92-141R8.

#046 (T) Page 55, Section 7.2, Para 11 (Dormant)

What is execution mean? This has no meaning to me. Can I bring in write data for a write command? Is this executing?

#047 (T) General Comment.

All of the text associated with Head of Queue, Ordered, and Simple has been removed from SAM. This should not have happened! The text was succinct. It described exactly what was to complete and when. We have now gone back into this nebulous "execution" of tasks model that failed in SCSI-2. I cannot vote to forward anything that allows this model to exist.

#048 (T) Page 57, Figure 22 and Page 58, Para 2

SIMPLE TASKS do not wait for previous SIMPLE TASKS to complete before executing when restrictive reordering is applied. They are only restricted if an overlap occurs on the data set being operated upon. The restriction also only applies to write CDBs.

#049 (E) Page 57, Figure 22

A note is needed stating that entry to the figure is restricted by an ACA condition. For example, if an ACA is in effect, the HEAD OF QUEUE tasks are not allowed to be submitted to the task set. If this note is missing, then the figure implies that HEAD OF QUEUE may coexist with ACA tagged tasks during an ACA.

#050 (T) Page 57, Figure 22

What can be done in each of these states? Can I bring in write data in the Dormant state? Can I bring in write data in the Held state?

#051 (T) Page 61, Section 7.4.1, Figure 24

Depending upon your definition of executing, I do not agree

with the boundaries applied. I can bring in data for an ordered write command when simples are being operated upon.

#052 (T) Page 67

Restrictive reordering only applies to commands with overlapping LBA ranges. It does not apply to all simples unless they meet the overlapped criteria.

#053 (E) Page 70, Section 8.3, Para 5

Add statement that the target shall guarantee that no further responses from the aborted tasks are sent to the initiator.

#054 (E) Page 70, Section 8.4

What happens on a CLEAR ACA when an ACA does not exist?

#055 (T) Page 74, Section 9.2

Why is this an LLP service? The LLP has no idea where the data is located. It is a confirmed service between two ULPs.

#056 (T) Page 75, Section 9.2, Para 1

The assumption that the client buffer is contiguous is not necessarily true. How I put my data in my buffer is completely outside the scope of this standard. It is true that the data buffer is a logically contiguous space, but not physically.

The buffer also does not have to be large enough to physically hold the data. I may page out data to some other space or consume the data as it is read into a buffer.

Jeffrey L. Williams
Disc Controllers Lab
Disc Mechanism Division
Hewlett-Packard Co.

HP TelNet: 396-5030
Telephone: (1) (208) 396-5030
Facsimile: (1) (208) 396-6858
ARPANET: jlw@hpdmd48.boi.hp.com

Date: Wed, 3 Nov 93 16:11:17 PST
From: Jeff Stai <stai@dt.wdc.com>
To: scsi@WichitaKS.NCR.COM
Subject: Yet Another SAM Rev 12 Commentary
Cc: john.lohmeyer@FtCollinsCO.NCR.COM, monia@starch.enet.dec.com
Extracted-To: SCSI_Reflector

Well, seeing as how everyone else is dumping their comments on the reflector, I figured I better get mine in while it was hot! I have a feeling, however, that my comments are an attempt to throw rocks at trees, and there is this big forest in front of me....;-)

jeff stai
western digital lsi, oem storage products
stai@dt.wdc.com
714-932-7644

=====

Comments on Letter Ballot for SAM
11/03/93 jd stai

This posting constitutes my comments on SAM rev 12 (X3T9.2/994D R12). My vote to forward is NO. SAM is a heck of an effort (thanks, Charles!), but it just isn't ready yet. What's the rush? Releasing it for public review when it is incomplete and not yet consistent will just confuse things and probably make us look foolish.

I'm going to choose not to try and identify technical versus editorial. That's not my job...;-)

pg 11, 3.1: You define request, response, and indication, but there is no definition for 'confirmation'. You also define lower level protocol without defining upper level protocol.

pg 11, 3.1.11: "...has than has..."

pg 14, 3.1.61: You shouldn't need a definition of SIP. Likewise for 3.1.63. Delete them. In any case, the definition given for SIP is true for ANY protocol!

pg 21, 4.2, 2nd line: The "...single request-response transaction." should be given an adjective like "virtual" or "effective", to differentiate from the "actual" transaction that passes through the service delivery subsystem.

pg 22, Figure 2: This figure is good, in fact it is the cornerstone for the rest of the document. As such, certain conventions should be used within the figure which are then used in all subsequent figures. For instance, the exact same dashed line style for the "virtual" server request/response transaction should be used in all subsequent figures. I note that subsequent figures use different line styles, which made it difficult to follow what was really being passed to whom.

pg 33, obj. def. 4: Hard to see how an initiator can exist without at least 1 Application Client. Also, the definition of Initiator Identifier (in words) says that the "content" of the object is protocol specific, yet the formal definition above says that it contains a value between 0..2^64-1. Which is it, or did you mean "encoding" as was said for the Target Identifier on the next page?

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.

Another point: Where does it say that the ID may be less than 2^{64} for a given protocol (like, say, SPI/SIP)?

pg 34, 4.7.2: Why even mention encoding for Target ID?

pg 35, 4.7.3: A consequence of having more than one Target ID is that a logical unit can have more than one Logical Unit Identifier....

pg 35, 4.7.3.1: "should" should be "shall", shouldn't it?

pg 35-36, 4.7.3.x: The Device Server contains the Task Set (never mind George's objections regarding the Queuing Model, which I have come to agree with). From an object-oriented point of view, it seems that the Task Set and Task Manager need to at least be peers to allow the Task Manager to operate on the Task Set. With the Task Set being within the Device Server, I don't see how the Task Manager can do that without directly interacting with the Device Server.

So, we open a can of worms, between this and the Queuing Model. It seems that the hierarchy is at fault, not the objects themselves. So:

- Put the Task Manager, Target ID, and Logical Unit(s), under the Target.

- Put the Task Set under the Task Manager (keep Task under Task Set). This allows the logical units to interact with the Task manager as peers, without actually touching the Task Set. This also hides the Task Set organization from the rest of the model (as per the Queuing Model in Annex C).

- Logical Unit now has Device Server and LUN below it. Device Server should be:

Device Server = 1{Device Impl.}

- Another argument in favor of a single Task Manager per Target is that the Task Manager receives Task management requests like "TARGET RESET". Clearly a target level function. All other requests operate directly on the Task Set. QED.

pg 37, obj def 8: Under "Tag", it says: "An initiator shall not assign the same tag...". If you allow more than one ID per initiator, then the constraint is by ID, not by initiator. No change necessary if initiators can have only one ID.

pg 37, last sentence: The word "automatically" is not necessary.

pg 38, 6.0: SOMEWHERE, it should say that Execute Command is a virtual request of some sort (a ULP transaction as back on Fig 5). This is presented as something real and is terribly confusing. Perhaps getting rid of it (at least as a formal entity) is best.

pg 39, Task attribute: These are defined nowhere. If you follow the references, you get into a loop back to Obj Def 8, where they are listed and not defined.

pg 39, pointers: How can the server assign the data in and autosense pointers in the client? These should be passed from the client as places for the server to put them.

pg 39, Data-in pointer: The contents of the buffer can be perfectly well defined for other status values, like CHECK or TERMINATED.

pg 39, command complete: same comment as for data-in pointer.

pg 39, service response: I think the list is incomplete for linked commands. See later comment on pg 43.

pg 39, autosense data: missing clause reference.

pg 40-41, tables 1-3: It seems inappropriate to show LBAs and such here. These belong in SBC. The tables here should be generic; say "Parameters" in each byte that is not otherwise occupied.

pg 42, table 4: What possible use is a 16 byte CDB that has all bytes reserved for all commands sets? Either replace "Reserved" with "Parameters", or nuke them!

pg 43, 6.2.2: The notification when the flag bit is set should be part of the service response of the Execute Command. The service responses should be:

- command complete: The command has finished.
- linked command complete: The linked command has finished and the next command should be delivered via an Execute Command request.
- linked command complete with flag: The linked command (with flag bit set in the CDB) has finished and the next command should be delivered via an Execute Command request.
- etc.

These responses should be a piece of cake even for serial protocols, the easiest way being to copy the flag and link bits into the payload of the status packet returned.

pg 44, INTERMEDIATE: The list of status codes at the end of the first sentence should include TASK SET FULL and BUSY, because these can occur on the first command of a link. Also, COMMAND TERMINATED could occur anytime by action of the initiator, which should break the link.

pg 45, ACA ACTIVE: There should be a sentence for recommended initiator action, as there is for BUSY.

pg 45, ACA ACTIVE, item c): "attribute" should be "Task Attribute". Confused me, anyway.

pg 46, 6.4.2: Reference SBC in pgf 2.

pg 47, step 4: The "response" is not defined. I know what it SHOULD be! See pg 43 comment above.

pg 47, general comment on linking: It is apparent that the application client is expected to issue another Execute Command for the next

command in the link. I didn't see where it actually says this (what clause), or that the Task Identifier and Task Attribute must be the same for each command in the link.

pg 48, 6.5.1: In the cold light of day, this seems to belong in SPC. READ CAPACITY?

pg 48-49, 6.5.2: I like "faulting command"; makes it clear what you are talking about. May I suggest "faulted initiator" for the initiator that issued the faulting command? It isn't always clear which initiator you are talking about. For example, pgf. 2 under 6.5.2.1.

pg 49, 6.5.2.1, item b). Criteria a) under 6.5.2.2 does not apply to a new command. Be more specific. This clause needs lotsa work.

pg 50, pgf. at top beginning "When an auto...": Both sentences seem to say the same thing, sort of; i.e., the ACA flag of the first CDB that caused the ACA remains in effect for the duration of the ACA, even if you get more ACAs during the current ACA. Or something like that. Or did I miss something?

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.

pg 50, 6.5.4, item b): "unti"

pg 51, 6.5.6, 1st sentence: Just say sense data is made available when an ACA occurs.

pg 51, 6.5.6, 2nd sentence: "created" not "returned". Returning is done via REQUEST SENSE, etc.

pg 52, 6.5.7, 2nd sentence: Remove "automatically". Can be confused with autosense.

pg 53, 6.5.8: The items on clearing sense data belong under 6.5.6.

pg 53, 6.5.8: Nowhere does it say that autosense is optional for a protocol (like SIP/SPI).

pg 53, 6.6, item h): This has been truncated relative to SCSI-2. As it appears, I should do unit attention whenever ANYTHING happens. The full SCSI-2 sentence should be retained: "Any other event occurs that requires the attention of the initiator."

pg 56: I believe you meant: "... all tasks shall behave as if they became Enabled upon entering the Task Set."

pg 57-59, 7.3: I would like to propose using the convention used in IEEE P1394 for describing states and transitions. Be glad to help generate the words for you. The nice thing about it is that each and every transition and state are fully defined, and you can find each in the text. Everything is mixed up here.

pg 57, 7.3: HEAD OF QUEUE (HOQ) tasks enter enabled. Is the intent to have undefined re-ordering if several HOQs come in at once, since they will all be enabled? Or should there be a mechanism to put HOQs into dormant? Or have I stepped in something horrible that the Queue Model SSWG tiptoed around?

pg 59, last pgf.: Give the figure and table reference numbers. Easier to read.

pg 60: Table 10 is missing.

pg 67, 7.4.4: I don't see anything under the description of restrictive reordering that requires only one task enabled at a time, or am I (once again) missing something? If only one at a time is allowed, then HQs must be able to start Dormant, since you can't bump the already enabled task.

pg 68, 8.0: ABORT TASK SET, ABORT TASK, CLEAR AUTO..., and CLEAR TASK SET need Initiator Identifier as an argument. Also pages 70-71.

pg 68, 8.0: TARGET RESET argument should be Target Identifier. More could be sent in a given protocol, but that's all that is REQUIRED to do the job at the SAM level.

pg 69: Add Initiator Identifier and Target Identifier to argument list.

pg 70, 8.2, last pgf.: "A response of Func. Comp. indicates that the task no longer exists." This is in conflict with your definition of Function Complete on page 68.

pg 70, 8.3: Replace "terminate" with "abort". Terminate has a different meaning, as in 8.7.

pg 70, 8.4: The last sentence is incorrect. A service delivery failure is always possible. Delete the sentence.

pg 71, first line: "may" should be "shall", maybe?

pg 71, 8.5: All occurrences of "terminated" should be "aborted".

pg 72 on, 9.0: The text formatting makes it very unclear where things begin and end. Highlighting and formatting should be applied as in previous pages.

pg 73, 9.1: It is just not clear what these are relative to Execute Command. I think what needs to be done is to duplicate Figure 5 here, showing each service and primitive on the figure.

pg 73, 9.1 request: A request primitive cannot have returned values. Move all args in front of the ||, and delete Status.

pg 73, 9.1 indication: What sets the autosense flag?

pg 76, 9.2.1: many problems here. I know of no LLP that will send a confirmation with a task ID attached. What "response of Command Complete" is referred to? Do you mean confirmation? What do status codes have to do with data delivery to the buffer? Requirements on data ordering are good, but this isn't right.

pg 76, 9.2.2: Data server buffer pointer is on the wrong side of the ||.

pg 77, 9.3: Include identifiers noted previously.

FCP Review Comments Revision 7

#001 (E) Page 1, Section 2

Add FC-AL to the list of documents referenced.

#002 (E) Page 4, Section 4.1

Add a comment on the ability to connect multiple SCSI devices on an arbitrated loop. I feel that this is an important item for potential implementors of the FCP to understand. SCSI on a peripheral device should be a low cost option which is enabled by the loop topology.

#003 (E) Page 4, Section 4.2

What is an FCP I/O Operation? Is this the same as an I/O Process in the SAM wording? If so, please eliminate the use of the term in favor of the SAM wording.

#004 (E) Page 4, Section 4.2, Para 2

"... and is only used for the transfers associated with that FCP I/O Operation and any commands linked to that command." Drop the "and any commands linked to that command." part. This blurs the definition of the I/O Operation. Is the I/O Operation a single command or a group of linked commands? See the confusion? The definition listed states it is the series of linked commands.

#005 (T) Page 4, Section 4.2, Para 3

You state that the data transfer occurs with a XFER_RDY IU followed by a single FCP_DATA IU. Does this have to be the case, or can a target or initiator split the FCP_DATA IU into two or more IUs for the data described in the XFER_RDY? For example, I get a XFER_RDY IU for 8k of data, but I send two 4k FCP_DATA sequences.

#006 (E) Page 4, Section 4.2

You may want to state that this is a non-linked I/O process with data descriptors enabled. As I read this section, I keep saying, "Sure, but what if the command is linked? Then you won't send this sequence."

#007 (E) Page 5, Section 4.2, Para 3

List FC-AL as another supported option. You list classes and fabric stuff, so listing loop may be appropriate.

#008 (E) Page 6, Section 5, Para 3

"SCSI data transfers may be performed by one or more data delivery requests, each one limited to a length no longer than..." This implies that the length of the data delivery request is limited when you meant that the data delivery length is limited. I would suggest the following wording. "SCSI data transfers may be initiated by one or more Data Delivery Requests, each one limiting

the subsequent Data Delivery Responses to a length no longer..."

#009 (E) Page 6, Section 5, Para 3

Last sentence should state, "... using the parameters in the Disconnect/Reconnect Mode Page."

#010 (E) Page 6, Section 5, Para 5

"Some SCSI reset functions..." should be "Some SCSI task management functions..." to be consistent with SAM.

#011 (I) Page 7, Section 5, Para 2 (last sentence)

What does this sentence mean? How is the performance of the SCSI command service primitives managed by completion of the individual sequences?

#012 (E) Page 7, Table 3

Sequences T3, T9 and T10 should have a note that a previous sequence T1, T2, T7 or T8 had to be sent prior to this sequence with the link bit set in the control byte of the CDB. I understand what the intent of these sequences are, but I do not think it is clear to the first time reader.

#013 (E) Page 7, Table 3, Note 1

IUS should be IUs.

#014 (E) Page 8, Table 4

Are sequences that terminate the exchange (I4 and I6) really doing a transfer of initiative or is this field a "don't care" for last sequences in an exchange? My guess is that we don't care since the exchange has ended.

#015 (E) Page 10, Section 5.3.6

Isn't the SEQ_ID a unique identifier for the sequence so that multiple frames in that sequence can be associated with each other? This states that it is more like the SEQ_CNT indicating an order to the frames within a sequence.

#016 (E) Page 11, Section 6.1, Para Last

In the case of a PRLI when one is in effect, what happens to any tasks which are in the device at the time? Are they terminated? Also, I assume that if this affects an initiator/target relationship, that a Unit Attention Condition is created? Please state whether or not this is the case.

#017 (T) Page 13, Section 6.1, Para 3

For command and data mixed, you state, "If either the originator or the responder do not allow the Command and Data to be combined, then all FCP I/O Operations between them shall be performed using only

one Category in each Sequence." I do not agree with this since it also restricts the use of Data/Status combined sequences. This should state that all command and data transfers shall be performed in separate sequences.

#018 (E) Page 13, Section 6.1, Para 4

"Any other combinations of these bits is invalid and will be rejected." Should say "shall be rejected."

#019 (E) Page 13, Section 6.1, Para 6

See #017. For Read/Status, we should not limit all uses of multiple information categories per sequence.

#020 (E) Page 13, Section 6.1, Para 8 and 10

Replace "wills" with "shall".

#021 (T) Page 15, Section 7.1, Para 2

This paragraph should be replaced with something like, "If XFER_RDY IUs are disabled and Command/Data Mixed is allowed, the Sequence carrying the FCP_CMND IU shall contain FCP_DATA IU frames following the FCP_CMND IU. If XFER_RDY IUs are disabled but Command Data Mixed is not allowed, sequence initiative may be transferred."

#022 (I) Page 17, Section 7.1.2, Para 5

Terminate Task presents the same problem that Abort Task presented in that the OXID is a duplicate of one that exists. This means that the interface chip cannot detect duplicate XID creation. Is this an issue that we want to address?

#023 (I) Page 17, Section 7.1.2, Para 6

Clear ACA also presents a problem. If I want to just clear the ACA and do nothing else, I can't since the command gets executed. I think that this type of operation should state that the CDB payload is considered invalid in some manner. Perhaps this should be done for all task management functions.

#024 (E) Page 17, Section 7.1.2, Para 14

Abort Task is no longer defined. This paragraph should be deleted.

#025 (E) Page 17, Section 7.1.4, Para 1

The FCP_DL is actually a count of the number of bytes to be transferred for the SCSI CDB, not the I/O Operation (which I don't think we define).

#026 (E) Page 18, Section 7.2, Para 3

The last sentence has some grammatical problems. "...is expected to have be ready to transmit...".

#027 (E) Page 18, Sections 7.2.1 and 7.2.2

Last sentences need periods at the end

#028 (E) Page 17, Section 7.1.4

Add some wording that indicates that insufficient FCP_DL will cause

a command to be stopped in progress. That is, if a write command for 8k were requested, but the FCP_DL was set to 7k, the command would not complete. Also, what is the expected behavior? Do I write 7k and terminate or just terminate? What about the converse where I have a read for 8k and a FCP_DL of 7k?

#029 (E) Page 21, Section 7.4.3, Para 1

Remove the sentence, "The value is specified in SCSI-2". It has no meaning.

#030 (E) Page 21, Section 7.4.3

Add a sentence that states that the length of this field must be padded to a four byte length due to the restrictions of FC-PH.

#031 (T) Page 22, Section 7.4.5

What is the purpose of the FCP_RSP fields? I see them as additional sense information. Are they considered "soft" error information that presents no reason to retry the command or think that it failed, but gives information useful to the ULP. I see them as a duplication of something already available, namely the sense data.

#032 (E) Page 22, Section 7.4.6

Remove references to SCSI-2. I think that we can just say that there may be sense data at any time and that it is in the format shown in the Request Sense command. There is no reason why I can't give sense data on every command.

#033 (E) Annex B

Not to throw a monkey wrench into this, but what about a RAID controller attached to a RAID controller, attached to a RAID controller.....

I think this a specific example of something that you would like to do, which is valid, but may not be appropriate for an annex. Find out what the RAID guys have to say and update accordingly.

#034 (E) Page 4, Section 4.2

The 3rd paragraph under Device management seems awkward. We suggest:

When the device server for a Data-In or Data-Out task has completed the interpretation of the command and is prepared to transmit or receive some or all of the Data associated with the task, it:

1. optionally sends a Data Descriptor IU containing the FCP_XFER_RDY payload to the SCSI Initiator to indicate which portion of the data is to follow.
2. sends or receives the solicited data sequence containing the FCP_DATA payload

Steps 1 and 2 are repeated until all data described by the SCSI command transferred.

#035 (I) Page 5, Section 4.2

The 2nd to last paragraph has problems:

The design basis for FCP isn't relevant or helpful. The first sentence which describes the wonderful match between FCP and class 2 should be struck. We also believe that class 3 DOES support the FCP with only one possible exception: support of delivery of commands in the order transmitted. Class 3 doesn't exhibit a lack of "confirmed service" support: the passing of Sequence Initiative may count as confirmation for most, if not all of the delivery services.

The paragraph should either be deleted or re-written to say that FCP may be run: class 2-only, mixed class2/class1, or class 3: as determined in a way beyond the scope of this standard (unless you want to suggest that if command comes in class 3, the entire Exchange shall be run in class 3, etc).

If there are specific cases where class 3 does not provide enough confirmation as required by SAM, they should be listed.

#036 (E) Page 6, Section 5

Seems like 4.2 on is the overview. And all of 5 is more detail than overview. Please consider re-titling the section (possibly merge 5 and 6 under 6's title).

#037 (T) Page 16, Section 7.1.2

The "Task Identifier" of SAM is missing, and thus untagged commands can't be issued. This breaks the leverage of existing CAM interfaces unless the FCP SIM does some sort of extra management to synthesize untagged behavior on an otherwise tagged-only FCP. The correct solution seems to be to preserve the ability to pass untagged commands. All that is needed technically is to add the "Task Identifier" bit in the command payload; editorial additions clarifying that an OX_ID tag isn't really a tag if "Task Identifier = Untagged Task Identifier", may also be necessary.

Which of the Task Management Flags should be allowed to be executed concurrently? Our understanding is that SAM presents a single Task Management function per service call paradigm. We suggest that the "flags" should be changed to "functions."

Also, since SCSI NOP isn't mandatory, we suggest there be a bit that says whether a command is valid. Since it isn't always possible to send Terminate Task, or Abort Task, at least these two must be sent using an FCP Link Application look-alike to ABTX: I.e., the OX_ID must be used to identify the task. However, it must be done on a separate Exchange in order to be able to send it asynchronously. We recommend putting all these functions in a separate Exchange and remove the bits from "normal" commands (except that Clear ACA will appear in both normal commands with ACA_Q and as an FC4 FCP extended link application):

Task Management flags -> functions (encoding)

reserved		0
Terminate Task	bit 7	1
Clear ACA	bit 6	2
Target Reset	bit 5	3
Clear Task Set	bit 2	4
Abort Task Set	bit 1	5
reserved		:

In summary, we propose the following re-structuring of the FCP_CNTL field:

bits	Tid	Task Attributes
7 6 5 4 3 2 1 0	-----	-----
MSByte command serial# / res.	0 Untagged	0 Simple_Q
LSByte command serial# / res.	1 Tagged	1 Head_of_Q
reserved		2 Ordered_Q
		3 ACA_Q
	R/W	: reserved
ACA Task Attribute Tid R/W	-----	
7 6 5 4 3 2 1 0	0 reserved	ACA
	1 Read	----
	2 Write	0 = Clear ACA
	3 reserved	1 = ACA unchanged

FCP Link Applications:

Terminate Task
 Clear ACA
 Target Reset
 Clear Task Set
 Abort Task Set

#038 (T) Page 17, Section 7.1.2 (FCP_CNTL: Ordered_Q)

The second sentence describes a procedure which is insufficient to guarantee SAM requirements that commands to a target identifier be received in the order sent. Merely requesting in-order delivery is not sufficient, since guaranteeing in-order delivery involves much more than setting and recognizing the Fabric Login in-order delivery bit.

An alternative is to introduce a serial number in the command: this number would be used to order all commands as appropriate at the device. Such a scheme may require new SCSI errors codes: We haven't researched it (something to indicate: timed-out-waiting for missing command to arrive). For a random access device like a disk, this serial number can probably be ignored unless Ordered Q is supported. For sequential access devices, explicit link-independent ordering of commands may be more useful.

The alternative is waiting for some indication of command confirmation. Class 1 and Class 2 offer ACK as a notification that the command was received with good CRC. Although the ACK, doesn't guarantee that the command will be delivered, it is a reasonably reliable confirmation for practical purposes. Class 3 has no such acknowledgment. The command serial number would improve robustness for class 1 & 2 and enable all classes of services to follow the SAM model.

An unpleasant alternative is to wait for confirmation of command received by the target before sending subsequent commands: this requires some sort of polling and/or notification when a command is acknowledged. I.e., subsequent commands to the same LU and/or device will be blocked until each command is acknowledged.

#039 General

We didn't find anywhere that says under what conditions (if any) a Target may provide the FCP_RSP IU "early". For example, if queue space is limited a Target may choose to go directly to RSP with "Task Set Full" status.

Are there any other restrictions to doing this besides requiring Sequence Initiative? I.e., can it send FCP_RSP immediately after an FCP_XFER_RDY (for a non-zero number of Data-In bytes)?

X3T9.2/93-186r0

Date: Fri, 5 Nov 93 13:32:37 -0800
From: Jim McGrath </G=Jim/S=McGrath/O=QMAILGW/PRMD=QUANTUM/ADMD=MCI/C=US/@qntm.com>
To: SCSI@WichitaKS.NCR.COM
Subject: Forwarding Votes
Extracted-To: SCSI_Reflector

OFFICE MEMO

Subject:
Forwarding Votes

Time: 12:07 PM
Date: 11/5/93

Quantum is voting NO on forwarding the SAM document due to technical issues. Specifically, we share many of the same concerns expressed in Sun's NO vote.

Quantum is voting NO on forwarding GPP, FCP, and SBP due to the inability to vote YES on SAM. Since these standards must be in compliance with SAM, we do not think it would be consistent to vote to forward these standards without voting to forward SAM.

As a comment, we find the serial protocol documents to have fewer technical issues than SAM, leading us to believe that it might be possible to vote in favor of some or all of them if the SAM issues could be resolved. It is also our understanding that any vote to forward at this point might be ineffective in any event given the pending committee reorganization.

We regret these NO votes, and hope the committee can resolve the outstanding SAM issues as soon as possible.

Date: Fri, 5 Nov 93 18:48:54 PST
From: Jeff Stai <stai@dt.wdc.com>
To: scsi@WichitaKS.NCR.COM
Subject: my ballot comments on SBP
Extracted-To: SCSI_Reflector

Isn't it great having all these Gene-clones running around making massive comment lists...?-)

Anyway, here's another from me!

jeff stai
western digital lsi, oem storage products
stai@dt.wdc.com
714-932-7644

=====
Comments on Letter Ballot for SBP
11/05/93 jd stai

This posting constitutes my comments on SBP rev 14 (X3T9.2/992D R14). My vote to forward is YES. Well, a provisional yes. I don't see anything in SAM that is controversial that will affect SBP (or any other proposal). I hope I am right. I have many comments on SBP, but none of them are substantive to cause me to vote no on SBP.

I'm going to choose not to try and identify technical versus editorial. That's not my job...;-) HOWEVER, most of these I think ARE boring editorial changes, until I get to page 32. Happy reading!

pg viii: This Introduction is redundant with the Scope section on the following page. I would delete it.

pg 1-end: I would delete all references to SCSI-2, since I believe this is a SCSI-3 standard and I see no reason to refer to SCSI-2.

pg 1-end: The 1394 standard is referred to in various ways throughout this document. I would propose we standardize on "IEEE 1394" for all references rather than P1394, since it is reasonable to expect that P1394 will be standardized in a similar timeframe.

pg 1-end: Any use of the word "ID" should be changed to "identifier" to match SAM terminology.

pg 1, 3.1: The following terms should also be defined: FIFO (in regards to the idempotence issue; don't write twice to a FIFO); stream identifier; sub-chain.

pg 2, several: Shouldn't the several asynch and isoch FIFOs be defined as command FIFOs?

pg 2, 3.1.15, sp (spelling): "Initiator"

pg 2-3, several: The following should reference SAM: 3.1.15, 3.1.21, 3.1.26, 3.1.29, 3.1.30.

pg 3, 3.2: Abbreviations should be included for "asynch" and "isoch". Also, "async" in several locations should be fixed.

pg 3, 3.1.27: The definition of tap should include that it occurs via a write transaction.

pg 5, 5.0, pgf. 2: All of SBP is new to SCSI. Delete the 1st line through the word "defined".

pg 6, 5.1, several: The words "shall" and "may" should be used in several places. I can share with the editors specifics if necessary.

pg 6, 5.1, pgf. 2: The last sentence should read: "The task attribute is in effect at the time of the first acquisition of any part of the CDS."

pg 6-7, 5.2-5.4: These should be in a task management section, not here. Perhaps part of clause 14. Also, for completeness, TARGET RESET, TERMINATE TASK, and CLEAR ACA should also be defined.

pg 6-7, 5.2-5.4: The various references to "status" in these clauses should be "SBP status".

pg 7, 5.3-5.4: What does "complete any outstanding CDS fetch operation" mean? Why bother?

pg 7, 5.4: This CLEAR TASK SET doesn't work. The initiator can't set the abort task flag of the tasks belonging to other initiators, or even be aware of them if it could have write permission to do so. The procedure must be modified to eliminate this requirement.

pg 7, 5.5.1: The two instances of "target/logical unit" should be "logical unit identifier" as per SAM.

pg 7, 5.5.2: Task management functions should also be mentioned in the 2nd sentence.

pg 7, 5.6: The 2nd pgf can be entirely replaced by the much simpler: "The initiator shall indicate it does not wish the target to send autosense data by setting the CDS sense length to zero."

pg 8, 5.6, pgf 2: "...then the target shall set the sense data flag to one within the..."

{at this point, I moved on to clause 6. I think we all realize clause 5 needs work...;-}

pg 16, fig 3: A box around "TASK SET" and "TASK SET MANAGER" indicating they are defined by SAM would be helpful.

pg 16, 6.1, pgf 1: There should be something in here that says that the information in a 1212 unit dependent directory defines a single SBP target. Also, should we restrict a node to a single SBP unit, with all targets defined below it? Just for simplicity....

pg 16, 6.1, pgf 2: This pgf makes no sense.

pg 17, 6.1, pgf 2: Replace "P1394" with "known". SBP information may be transported over non-1394, but 1212 compliant transports.

pg 17, table 3: The quadlet order shown is not required by 1212. Why are we requiring it? Also, a note that unit software version is a 1212 requirement would be helpful.

pg 17, 6.1, last 2 pgfs: If you don't know that the "vendor ID" is being used for a "unit specifier ID", this is very confusing. Just call it "unit specifier ID", like 1212.

pg 18, table 4: The quadlet order shown is not required by 1212. Why

are we requiring it? Also, the NOTE has the word "proposed". I assume "proposed". By whom?

pg 18, 6.1, pgf 3: What "base address"? It's not defined.

pg 18, 6.1, pgfs 7 & 9: What is the "base of CSR space". Be nice to define it here.

pg 18, 6.2, pgf 2: All 1394 devices do asynch transfers. Better wording would be "A target that transfers SCSI data using IEEE 1394 asynch data transactions shall...".

pg 19, table 5: What is the "Offset" column trying to tell me? Offset from what?

pg 19, 6.2.2-6.2.4: References to "command descriptor blocks" should be "CDSs", since these FIFOs can receive only CDSs.

pg 19, 6.2.2: References to BUSY are inappropriate. More on this when I comment on clauses 9-13.

pg 19, 6.2.2: How do I get more than one asynch normal FIFO for a target?

pg 19, 6.3, pgf 2: Same remark as for 6.2, pgf 2 above. Replace "asynch" with "isoch".

pg 20, table 6: Same remark as for table 5 above.

pg 20, 6.3.2: Same remark regarding BUSY as for 6.2.2 above.

pg 21, 7.1: Everywhere else, the "place" status is put is called a status FIFO (refer to 3.1.25 and 13.0). Here it is called a status block buffer. Change it to FIFO.

pg 21, 7.1, pgf after IMP. NOTE: This is accurate, but hard to follow. Add "... for the associated CDS."

pg 21, 8 tap slot rules: These belong in 14.3.

pg 22, table 9: The descriptions for values 10h and 11h need to be expanded on. When do they occur?

pg 22, 7.2, pgf 1: The last sentence should be a NOTE.

pg 23, 7.2, pgf 7: The data length cannot be limited to FFh and an integral multiple of four. The SCSI REQUEST SENSE command can return up to FFh bytes. FFh and multiple of 4 really means FCh. The limit should be 100h.

pg 24, table 14: The "Identifier" field is not defined in the text.

pg 25, 8.1, pgf 4: The first sentence can be simplified to: "The CDS sense buffer address field contains the address to which the CDS sense data is returned." Also, the missing reference at the end is clause 7.2.

pg 26, table 20: Need a note saying the S100, S200, S400 are defined for the 1394 cable PHY data rates.

pg 26, 8.1.2, pgf 3, line 1: "...boundary which shall not be crossed..."

pg 26, 8.1.2, pgf 4, line 1: "...size of packet that shall be used by..."

pg 27, 8.2, pgf 1: Need forward reference to where M-flag is defined.

pg 28, table 24: All over this document, flag names are nicely spelled out. Except here. I propose the following name changes:

A-flag	abort task flag (as used elsewhere)
L-flag	linked flag
O-flag	data transfer order flag
S-flag	scatter gather flag
M-flag	next CDS address valid flag
SCE-flag	end of sub-chain flag

pg 28, 8.2.1, pgf 1: Last sentence should also say that the abort task flag cannot be cleared until the CDS is released by the target (via status or other action).

pg 28, 8.2.1, pgf 9 to end: These pgfs should be located with the rest of the scatter gather flag definition.

pg 29, 8.3: It isn't clear that the LOGIN CDS can be sent after login to change the slot allocation and sign-in for AE.

pg 29, table 26: The "Number of Slots", "AE Sense Length", and "AE buffer address" fields are not defined in the text.

pg 30, pgfs 8-12: These constitute an example, and should appear as a NOTE. Also, the note in pgf 13 is redundant with pgf 7.

pg 30, table 28: These fields are FIFO addresses, not tap addresses.

pg 31, 8.4: Title should be UNIT MANAGEMENT CDS.

pg 31, table 30: Why are these bits for each function, and not a single code? What happens when I set bit 5 and 7 at the same time? Bit 0 and bit 2?

pg 32, pgf 3: The NOTE following should be replaced with: "The tag value field is valid only for the TERMINATE TASK and ABORT TASK functions." Also, tag value should really be called "task identifier".

pg 32, 9.0 and after: In many places, references are made to specific bus events on the 1394 bus. These events are not visible to SBP, being hidden below the 1394 Transaction Layer. Therefore, I propose that these several references be brought into line with specific 1394 Transaction Layer services. I think this can be done without losing the essential meaning.

I understand why the current wording exists: the Transaction Layer in 1394 was very poorly defined! I have been working to fix that, so I will offer substantive suggestions. The important point here is that when we talking about ack codes and such, we run the danger of redefining 1394. Instead, we should use the service interface supplied in the 1394 standard. If there is anything lacking in that interface for SBP purposes, please let me know and I will work to correct it!

One key point that I keep seeing is the importance of BUSY status. I now have some familiarity with real 1394 implementations, and they all give BUSY (in various forms) rather regularly, for rather short-term busy conditions. These busy conditions almost always clear up on the very next retry. It is silly to bring SBP operations to a halt for what

are HARDWARE transaction operations.

For instance, the data transfer protocols state that if the target receives anything but "complete" or "pending", the data transfer is to be stopped and the command aborted! This would cause every SBP command to be aborted by BUSY on otherwise valid 1394 implementations.

The first change is the remove all references to 1394 "requests" and "responses" and refer to them only as "transactions". An example is pgf 2 under 9.0: "A tap is an IEEE 1394 write transaction with a..."

pg 32, 9.0, pgfs 2-5: New wording using the interface defined in 1394 for all 1394 applications:

"A tap is an IEEE 1394 write transaction. The initiator sends the tap by sending a transaction data request service to its 1394 Transaction Layer, with the destination address equal to the address of one of the target's command FIFOs. A tap contains the first 64 bytes of a CDS.

"The target receives a transaction data indication service from its 1394 Transaction Layer when it receives the tap. If the target accepts the tap without error, it shall respond to the transaction data indication with a resp_complete response code in the transaction data response. If the target does not accept the tap, it shall complete the transaction with a resp_conflict response code in the transaction data response. If the target receives the tap with error, it shall respond to the transaction data indication with the appropriate error response code in the transaction data response.

"If the initiator receives a transaction data confirmation with a request status of COMPLETE, it shall consider the tap accepted by the target. If the initiator receives a transaction data confirmation with a request status of ACKNOWLEDGE_MISSING or TIMEOUT, the tap may have been accepted by the target; the initiator shall not attempt to resend the tap without aborting all tasks in the chain. If the initiator receives a transaction data confirmation with any other request status, the initiator shall consider the tap not accepted by the target."

This is what I have in mind. I will gladly work on new wording for the other sections at the appropriate time. Specific places where rewording needs to be done:

- 10.0, pgfs 2-5
- 11.0, pgfs 2-3
- 11.1, pgfs 1-4
- 11.2, pgfs 1-4
- 12.0, pgfs 2-7
- 13.0, pgfs 1-4

pg 35, 14.0, pgf 2: 1st sentence should start: "Targets shall not respond to..." Also, last sentence: "The target shall respond normally to..."

pg 37, 14.3, pgf 1: Delete the 2nd sentence beginning "Thus...". Begin the 3rd sentence: "A target supporting asynch transfer operation shall be able to..."

pg 37, 14.3, pgf 3: 1st sentence starts: "The target shall not be required...". 2nd sentence: "When an initiator requests tap slots, the target shall report to..."

pg 37, 14.4, pgf 1: How does the initiator "gain release" from a prior sign-in request? There is no provision for this in the LOGIN CDS.

X3T9.2/93-186r0

pg 39, table A-1: Those dangling "y"s are driving me nuts!

pg 47, table D-1: Boy, that REALLY looks like a CDB. A LOT like a CDB. I could swear that 16-byte CDBs were supposed to have their control bytes in the last byte. I presume this is delivered in the SCSI CDS? It doesn't say so. If not, I didn't know we had 16-byte CDSs!

Date: Fri, 5 Nov 1993 23:35:00 +0800
From: Bob Snively <Bob.Snively@eng.sun.com>
To: jlohme@ncr-mpd.ftcollinsco.NCR.COM
Subject: Comments on FCP, X3T9.2/993D, R 007, Bob Snively
Cc: scsi@WichitaKS.NCR.COM
Extracted-To: SCSI_Reflector

To: John Lohmeyer, X3T9.2 Chair
From: Robert Snively
Date: November 5, 1993
Subject: Review of FCP, X3T9.2/993D, Revision 007

Dear Mr. Lohmeyer:

After studying the FCP document, I have decided that it uses the SAM document in a stable manner and does not have any technical errors sufficient to justify a negative vote. I therefore vote affirmatively on the FCP document, but I strongly suggest that the following editorial and minor technical corrections be incorporated in the forwarded document.

E 001 Cover:

Is "editors", s/b "editor"

E 002 Page i

Is "596-3362", s/b "573-3362"

E 003 Page 2, Section 3.1.5

Is "indicates a displacement", s/b "indicates the displacement"

E 004 Page 2, Sections 3.1.9, 3.1.11, 3.1.14

These sections should be deleted because they are not used by the document.

E 005 Page 2, Sections 3.1.16-3.1.19

These sections should be placed as subsections under 3.1.15 in the same order.

E 006 Page 3

The following terms should be defined on page 3, using definitions paraphrased from SAM:

initiator: an SCSI device containing application clients which originate device service requests and task management functions to be processed by a target SCSI device.

SCSI device: a device that originates or services SCSI commands.

target: an SCSI device containing logical units which receive and execute commands from an initiator.

task: an object within the logical unit representing the work associated with a command or group of linked commands.

E 007 Page 3, Section 3.1.24

The logical unit should be redefined as:

an externally addressable entity within a target that implements an SCSI device model.

E 008 Page 3, Section 3.1.26

The logical unit number should be redefined as:

An encoded 64-bit identifier for a logical unit. [SAM] The logical unit...Address (see section 7.1.1).

E 009 Page 3, Section 3.1.31

The target identifier should be redefined as:

address of up to 64-bits by which a target is identified. [SAM] The target identifier is equivalent to the 24-bit Destination_ID used by the Exchange Originator. [FC-PH]

E 010 Page 3, Section 3.1.33

The task identifier should be redefined as:

the information uniquely identifying a task. The identifier contains the initiator identifier, the logical unit identifier, and (optionally) the tag. [SAM] The Fully Qualified Exchange Identifier (FQXID) is the task identifier for the FCP.

E 011 Page 3, Section 3.1.34

task management function should be redefined as:

a peer-to-peer confirmed service provided by a task manager that can be invoked by an application client to affect the execution of one or more tasks. The service consists of a request to manage a task in the logical unit. The response.....

E 012 Page 4, Section 4.2

Third paragraph, is "continue to be transferred until all" s/b "continue until all".

E 013 Page 5, Section 4.2

First paragraph, is "or from the SCSI initiator's memory." s/b "or from the SCSI initiator."

E 014 Page 6, Section 5, Table 1

is "Data Delivery Response" s/b "Data Delivery Action".

E 015 Page 6, Section 5

is "maximum burst length by the FCP using the parameters" s/b "maximum burst length defined by the parameters".

E 016 Page 6, Section 5

fourth paragraph should be:

If required by the FCP Process Login parameters, each inbound and/or outbound FCP_DATA IU shall be preceeded by an FCP_XFER_RDY sequence containing a standard Data Descriptor payload that indicates the exact length of the data delivery. If the FCP Process Login specifies Read XFER_RDY Disabled and/or Write XFER_RDY Disabled. The corresponding FCP_DATA IU's are transmitted without a preceeding FCP_XFER_RDY IU.

E 017 Page 6, Section 5

is "Some SCSI reset functions are.." s/b "Some SCSI Task Management functions are..."

E 018 Page 6, Section 5, Table 2

Add text to indicate that the Task Management Functions are required to be executed in a new Exchange.

T 019 Page 6, Section 5

Add text to indicate that the CDB is ignored when Task Management functions are executed.

E 020 Page 7, Section 5, first paragraph

is "...and the Responder X_ID" s/b "...and optionally the Responder X_ID".

E 021 Page 7, Section 5, last paragraph

is "...managed by the completion of individual Sequences" s/b "...managed by the transfer of individual IU's"

E 022 Page 7, Section 5.1

last note:

s/b "...optional sequence streaming during Write operations."

E 033 Page 9, Section 5.2

correct minor typos and stylistic errors.

E 034 Page 9, Section 5.2

Add a new sentence, next to the last:

Class 3 service is allowed if the management of the FCP ULP is performed using interlocked IU transmissions.

E 035 Page 10, Section 5.3.2 and 5.3.3

Last sentence, is "or striping", s/b "or data striping"

E 036 Page 10, Section 5.3.10

Second and third sentence s/b "The RX_ID shall be assigned by the Exchange Responder and may have any value, including the unassigned value of hexadecimal 'FFFF'. The RX_ID allows the

Exchange Responder to optionally create a private version of the SAM tag."

E 037 Page 10, Section 5.3.11

Next to last, last, and new last sentence s/b "For the Solicited Data Category, the Relative Offset is the SAM application client buffer offset and the Base Address is the beginning address of the application client's buffer, as described by SAM. For all other ... byte of the IU. The Relative_Offset is not required if both N Ports can unambiguously reassemble the transmitted IU's using other FC-PH information and the FCP_XFER_RDY IU."

E 038 Page 11, Section 6

First sentence, second paragraph is "defined as a normative annex" s/b "defined in a normative annex"

E 039 Page 11, Section 6.1

First sentence, first paragraph is "to indicate to" s/b "to identify to"

Fifth paragraph is "for all targets" s/b "for any target"

Last sentence is "from all targets" s/b "from any target"

E 040 Page 12, Table 5

Correct column widths

E 041 Page 13, Section 6.1.1

The word "sequence" should normally be "IU". Correct as appropriate.

The word "will" should normally be "shall". Correct as appropriate.

E 042 Page 13, Section 6.1.1

In both Xfer Rdy Disabled paragraphs, is "FCP_XFER_RDY Sequences will not be used.." s/b "FCP_XFER_RDY IU's may be not used..."

E 043 Page 13, Section 6.2

is "process images specified", s/b "process image pairs specified"

E 044 Page 15, Section 7.1

The FCP_CMND IU is actually used for two functions:

a) Carrying the SCSI command, if and only if no task management function is being requested.

b) Carrying task management requests, leaving CDB invalid

The text should be clarified to indicate this.

E 044 Page 16, Section 7.1.1

Last sentence should become: "An example of the four-layer hierarchical address structure defined for SCSI RAID systems is given in Annex B."

T 045 Page 17, Section 7.1.2

Should an untagged case be explicitly allowed? SAM defines the untagged case as identical in handling to the Simple_Q case.

E 046 Page 17, Section 7.1.2

Clear ACA, third paragraph:

The Clear ACA Task Management function should not be associated with a CDB and is executed in a separate exchange.

T 047 Page 17, Section 7.1.2

Target Reset:

The text should indicate that Target Reset terminates and aborts all outstanding exchanges between the requesting initiator and the receiving target. These events are performed without any FC-PH protocol requirements.

T 048 Page 17, Section 7.1.3

The text should clarify that the CDB is null if a task management function is requested.

T 049 Page 17, Section 7.1.4

The incorrect length cases should be explained. Only the highest addressed burst of data may be truncated to a length other than the specified burst length. Such truncation will always set the FCP_RESID residual count and will always set an appropriate_FCP_RESP code. Any bursts of read data beyond the truncated burst will not be transferred. Write operations may not be truncated by the initiator to less than the FCP_DL.

E 050 Page 18, Section 7.2

Specifications of transfer size redundant with 7.2.2 or other sections should be deleted, including first paragraph second sentence and second paragraph first sentence.

E 051 Page 22, Table 14

Correct column widths.

T 052 Page 27 - 29, Annex B

This section should be rewritten to use the RAID addressing example.

October 29, 1993

John Lohmeyer
Chair ANSI X3T9.2
1635 Aero Plaza Drive
Colorado Springs, CO 80916

Dear John;

Hewlett Packard has decided to cast a NO vote with regard to the forwarding of the GPP document.

Hewlett Packard does not feel that this protocol will become a main stream serial protocol operating over a number of serial and parallel interfaces. We reached this opinion based on the fact that each of the serial interfaces have their own upper level protocols which are being supported as an integral part of the work on the physical interfaces.

Also, in reaching this decision, we felt that the existence of this protocol distracts from the development of the other upper level serial protocols.

Regards,

Jeffrey L. Williams
Principle Member X3T9.2

X3T9.2/93-186r0

To: John Lohmeyer

From: Charles Monia

Date: November 7, 1993

Subject: Review of SCSI-3 Generic Packetized Protocol Working
Draft X3T9.2 991D Revision 7

In my opinion, this standard is not ready for public review, therefore my vote is no.

In my opinion, the following issues need to be addressed before I can vote in the affirmative:

- 001 - Use of the term "Task" is likely to be confused with the task entity defined in SAM.
- 002 - References to explicit sections in SAM must be deleted.
- 004 - Support for untagged tasks by a target is mandatory.
- 006 - A task abort must not be affected by in-transit information.

Comments on Generic Packetized Protocol X3T9.2 991D REV 7

The following remarks address what I consider to be technical issues.

001 Page 5, section 3.1.40

The definition for Task in GPP is different from and at variance with that used by SAM. To avoid confusion. I suggest finding another term.

002 Page 10, section 4.5, fourth bullet, third sentence

I cannot parse the following:

"The mode of operation of a GPP device in a task provides the additional uniqueness to distinguish function."

003 Page 10, section 4.5, fourth bullet, last sentence.

"...see sam 5.6.3.1...."

Explicit references to other documents by section will cause confusion when such documents are revised.

004 Page 15, section 4.7, item 4, last sentence

Support for untagged queueing is not optional for a logical unit.

005 Page 16, section 5.1.2, second paragraph, last sentence.

The requirements in this sentence duplicate those specified in SAM. I suggest you simply make reference to SAM.

006 Page 20, section 5.3.2.1, second sentence.

"With a full-duplex service delivery subsystem, these task management functions may not have the desired effect on all tasks if any information Packets (sic) are in transit when one of the task management functions is processed."

The condition described in the quoted sentence must be handled by the service delivery subsystem. It is the responsibility of the subsystem to flush all data in transit when a task is aborted.

I suspect the only safe way to accomplish this is by requiring that the order in which data is sent from sender to receiver is preserved as noted in item 007. In that case, an ABORT will always come after any data that is in transit for the task. Purging would then be automatic.

007 Page 24, section 6.2, second and third paragraphs.

The specified behavior seems to describe use of multiple paths to stripe data transfers. If so, the behavior must require that ordering of information between the originator and recipient be preserved.

008 Page 32, sections 7.2.1 and 7.2.2. Page 37, sections 7.2.4 and 7.2.5.

These sections duplicate requirements specified in SAM.

To: John Lohmeyer

From: Charles Monia

Date: November 7, 1993

Subject: Review of SCSI-3 Serial Bus Protocol, X3T9.2 992D, revision
14

In my opinion, this standard is not ready for public review for the reasons listed below, therefore my vote is no.

The following problems must be addressed before I can vote in the affirmative.

General comment 1: I believe the number of editorial errors are unacceptable for a document to be submitted for public review.

General Comment 3: Requirements need to be correctly identified throughout the standard.

005 The target requirements for command queue processing need to be defined. From the specification, it is not clear when the target is apt to stop reading a command chain.

010 I found the time stamping description in section 5.7 hard to understand. I assume, proper time stamping by an initiator is essential to insure that command queues for all initiators are serviced fairly. In my opinion, a clear description is required before the document can be reviewed adequately.

I feel that the remaining technical issues ought to be addressed during the public review period.

Comments on Serial Bus Protocol Specification - X3T9.2 992D, Rev 14

General:

1. References to SCSI-2 should be changed to SCSI-3.
2. The text needs a thorough proofreading. There are a lot of editing and spelling errors, along with many incomplete or awkward sentences. I can supply specifics.
3. A number of places seem to be missing a "shall" statement. I'll be happy to point them out.
4. The following protocol-specific responses are not defined as required by SAM:
 - a) Command Complete (SAM R12, pp 38)
 - b) Service Delivery or Target Failure (SAM R12, pp 68)
 - c) Function Complete (SAM R12, pp 68)
 - d) Function Rejected (SAM R12, pp 68)
5. The specification should describe the protocol-specific mapping of SAM service primitives.

6. A concise description of how an initiator specifies the sequence of CDS delivery is essential. This description should include the precedence of each command fifo.

001 Various

The acronyms "isoch" and "asynch" are undefined.

002 Page 4, Section 4.1, third paragraph.

The definitions for "octlet" and "quadlet" should appear in the glossary.

003 Page 6, section 5.2, second paragraph.

"recieved" should be "received".

004 Page 6, section 5.2, second, third and fourth paragraphs.

I was uncertain about the status parameter referred to in this section. Does this refer to SCSI status or the status associated with a completed CDS?

The term "status" should be qualified to distinguish between these two.

005 Page 6, section 5.2, last paragraph.

The note says "... The target must insure that it has the ability to continue forward progress in a chain beyond the CDS which has been aborted."

What is the definition of "forward progress"? I assume this refers to the targets ability to acquire CDS's. Please define the conditions under which a target may suspend CDS acquisition.

I am especially concerned that a task set full condition on one LUN may cause the target to stop fetching CDS's for other LUNS.

006 Page 6, Section 5.3, first paragraph.

The fifo used for the task management tap associated with the Abort task set should be specified.

007 Page 7, Section 5.4, "Clear Task Set"

There is the potential for data corruption that needs to be addressed for this and other scenarios in which a CDS terminates due to an exception condition.

The problem is caused by the fact that, after a command failure, the target may continue to fetch and execute SCSI commands for the logical unit before the initiator has had a chance to intervene.

For Clear Task Set, the problem occurs when:

- a) There is a CDS chain from another initiator containing SCSI commands for that task set and;

- b) The ACA flag is clear in one or more CDBs within that chain.

In that case, when the target acquires the next CDS from the other initiator, the command will not be executed and will be terminated with CHECK CONDITION (due to the unit attention condition from the Clear Task Set). Since ACA was disabled, a subsequent chained CDS for that logical unit will be executed, thus causing the CA and unit attention conditions to be lost. More importantly, because command execution was not stopped after the initial unit attention failure, subsequent commands will execute "out of sequence".

By the time the initiator discovers that a command has failed it will be too late. The target will have already acquired and executed other CDS's in the chain, possibly corrupting data beyond recovery.

At the very least, there needs to be text somewhere warning of this possibility.

There is a similar problem when the initiator is notified of a unit attention condition by means of the AER mechanism. In that case, since there is no ACA, the initiator cannot intervene to stop the target from acquiring and executing additional CDS's, regardless of whether or not ACAs were disabled.

Note that both problems are generic to all non-interlocked SCSI protocols. ie. the problem needs to be addressed in SAM.

By the way, in the case of SIP/SPI, it was always possible for the initiator to intercept an AEN and intervene before more commands were sent.

008 Page 7, section 5.6, second paragraph, last sentence on page

"....to send automatically the...." should be "...
to automatically send...".

009 page 8, section 5.6, second paragraph.

"...the target is required to set...." should be "...
the target shall set..."

010 Page 8, section 5.7

I did not understand the role of the time stamp from this description. From various presentations, I get the idea that the function of the time stamp is to insure some level of fairness in processing command chains. If so, that is not clear from the explanation.

011 General

One picture is worth a thousand words. It would be nice to have a diagram or two showing how CDS queues work.

012 Section 5

The material in this section seems tutorial. Perhaps it belongs in an annex.

013 Page 8, section 5.10

The SAM service primitives should be described along with their mapping to SBP.

014 Page 9, section 5.12

"...by which an initiator can operation..." - seems like this should read "...by which an initiator can initiate an operation ...".

015 Page 10, sections 5.12.2, 5.12.3

Seems like tutorial matter belonging in an annex.

016 Page 12, section 5.12.4

"... the target is allowed to process one and only one...."
- should be changed to "...the target shall not process more than one...".

017 Page 14, section 5.12.5, fifth paragraph

There should be a pointer to the section describing how an initiator is notified of an available tap slot.

018 Page 23, Section 7.3, second paragraph, last sentence.

"...AE reporting is disabled until the initiator performs another login procedure." Should login be changed to sign-in?

019 Page 28, section 8.2.1, second paragraph.

There seems to be a problem insuring that a linked command can be aborted in a timely manner.

Assuming the CDS address corresponds to the tag in SAM, then The task identifier seems to change depending on which linked CDS is being serviced at a particular time. (By the way, SAM assumes that the identifier is fixed for the duration of the task.)

As a result, the initiator has no way of knowing which linked CDS has been acquired, and hence has no way to insure that the "current" CDS in a linked series will be aborted. Note that setting the "A" flag has no effect in this case.

I'd suggest using the address of the last CDS in the series as the "tag", which would then be fixed for the duration of the task. Note that using the address of any other linked CDS prevents that CDS from being reused until the task terminates.

020 Page 28, section 8.2.1, second paragraph.

The initiator has no way to intervene and alter the sequence of linked commands depending on the outcome of a command early in the chain.

021 Page 29, Section 8.3

There should be a statement somewhere explaining the reason for allowing an initiator to request an explicit login identification.

022 Page 30, section 8.3, fifth paragraph.

Reference to "disk drive" seems inappropriate. Aren't these parameters applicable to any device that uses the isochronous facility?

023 Page 36, Section 14.2.1, Asynch Logout.

What is the disposition of chained CDS's upon completion of a logout? I assume they are implicitly returned to the initiator as soon as the logout response is received. If so, that should be stated.

024 Page 37, section 14.4

When superceding an AE buffer as described in the last sentence of paragraph 3, how does the initiator know when it is safe to reuse the previous buffer? Also, if an AE occurs during a replacement, how does the initiator know which buffer has the data?

X3T9.2/93-186r0

To: John Lohmeyer - Chairman, X3T9.2
From: Stephen F. Heil - Panasonic Technologies, Inc.
Date: November 3, 1993
Subject: Comments to X3T9.2 Letter Ballot X3T9.2/93-152r0 Regarding the
Forwarding of the SCSI-3 Architecture Model (SAM) revision 12
(X3T9.2/944D revision 12)

The following lists my reasons for voting "NO" on letter ballot X3T9.2/93-12r0 to forward the SCSI-3 Architecture Model (SAM) revision 12 (X3T9.2/944D revision 12) document to X3T9 for further processing (first public review).

- 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.
- 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.
- 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).

Sincerely,
Stephen F. Heil
Tel: 609-987-3948
Fax: 609-987-0483
Email: sfh@research.panasonic.com

To: John Lohmeyer and X3T9.2 Committee
From: Erich Oetting (Storage Technology)
Re: Reasons for No vote on forwarding motion for SAM Rev 12.
(X3T9.2/994D rev 12)
Date: November 5, 1993

I must vote no on forwarding SAM revision 12 for the following reasons. (Items from 4 onward are minor issues):

1. On page 41 and 42, Tables 2, 3 and 4 mark fields as Reserved that are used in some command sets. As an example Read Position and Locate stream commands use byte 1 in the ten byte CDB. The other fields specified for Logical Block Length, Transfer Length, Parameter List Length and Allocation Length should be removed and replaced with "Parameters". The positions specified are only correct for the block command set.

(A guide to field placement may be appropriate here, but should be clearly marked as informative.)
2. The conflicts between the Task State Management section and the Annex C Queueing Model need to be resolved and Annex C removed.

(Moving the Task Management examples to an annex would help separate the normative rules from informative suggestions.)
3. The document is not ready for public review due to the large number of typographical errors, and formatting problems in the tables. (Most of these errors have already been noted in the comments on the SCSI reflector by Snively, Stai and Penokie.)
4. The small legends inside the figures should use a slightly larger type to make them readable, or be removed.
5. Page 42, remove "(see Table 3)"
6. On page 44, Inquiry and Request sense are also exceptions to the Reservation Conflict rules.
7. On page 48 & 49, the description of Auto Contingent Allegiance and the ACA bit is confusing. This would be easier to describe if we kept Contingent Allegiance around and described how the ACA bit controlled entry into either the CA or ACA condition. Then the explanation of how to exit CA or ACA would be much simpler.

X3T9.2/93-186r0

TO: ANSI X3T9.2
FROM: N.T.Wanamaker, Amdahl Corporation, Compatible Systems
SUBJECT: Letter Ballot Comments on FCP

I recommend a "yes" vote on FCP. I have the following comments on the FCP document:

3.1.10., 3.1.12., 3.1.15., 3.1.28. SAM does not appear to require confirmed service here. I agree that it is appropriate to use confirmed service here.

3.1.15. This refers to the FC_PH service interface, rather than FC_PH services.

4.2., par. 7: intermix should replace mixed.

5., par.2: The number of open exchanges is not defined by FC-PH characteristics, but rather negotiated at login as defined in FC-PH. The number of exchanges is for the pair of N_Ports, not qualified by FC-4 type.

5., Table 2: Functional negotiation isn't in SAM. Login/Logout isn't optional. (note typo in header).

6.2.1. First sentence should refer to "Logout".

7.1., second sentence: "...an.. frames..."

7.2., last sentence of par.3: "...have be ready"

7.2.1. What is the initiator to do if the relative offset of the data frame doesn't match the Transfer Ready?

From ncrhub1!eng.amdahl.com!ntw20 Fri Nov 5 21:47 MST 1993
Received: by ncr-mpd.FtCollinsCO.NCR.COM; 5 Nov 93 21:47:50 MST
Received: from ncrgw1 by ncrhub1.NCR.COM id be28063; 5 Nov 93 19:36 EST
Received: by ncrgw1.NCR.COM; 5 Nov 93 19:31:52 EST
Received: by amdahl.com (/\\==/\\ Smail #25.33)
id <m0ovbZ8-0000HiC@amdahl.com>; Fri, 5 Nov 93 16:31 PST
Received: from sanjacinto.eng.amdahl.com by cliffy.eng.amdahl.com (4.1/SMI-4.1)
id AA12930; Fri, 5 Nov 93 16:30:15 PST
Date: Fri, 5 Nov 93 16:30:15 PST
From: Neil T Wanamaker <ntw20@eng.amdahl.com>
Message-Id: <9311060030.AA12930@cliffy.eng.amdahl.com>
To: john.lohmeyer@FtCollinsCO.NCR.COM
Subject: GPP Letter Ballot
Status: 0

(I will bring the letter ballot + hard-copy of these comments)
I recommend a "yes" vote on GPP, with the following comments:

Appendix B.1 (general) - A rework of these definitions with the non-Fibre-Channel literate reader in mind would be helpful. There are many cyclic references or words used having meanings not described in this section. Connect is particularly gruesome, as I believe it to be used in both the FC sense and SCSI sense in this appendix.

B.1.2., B.1.3., last clause: busy or reject link continue and link responses??

B.1.3., next-to-last clause: Class 2 uses only buffer-to-buffer flow control.

B.1.4. In many implementations, the CRC is calculated as the frame is being transmitted. Whether or not it is is not part of the standard. A reference to

either FC-PH Appendix N or the FDDI MAC specification would seem appropriate here.

B.1.7. The Destination_Identifier is used for frame routing, not identification.

B.1.11, B.1.12: Some mention of Arbitrated Loop Technology or Distributed Fabric Elements would seem appropriate here., especially as B.3, third bullet refers to AL which is otherwise undefined.

B.1.18.: The "other fields" do far more than help determine protocol errors: identify operations, protocol types and logical partitions, quantify data offsets and padding, etc.,

B.3. Third bullet: Arbitrated Loop is not defined.

B.4. "Unit of work" suggests task or operation, not Information Packet.

B.5 There are several paragraphs numbered B.5.1.1, B.5.2.1, B.5.3.1., B.5.4.1.

B.5.1.1. (Effect of Receipt) Between bullets 10 and 11 some mention should be made of flow control and acknowledgement.

B.5.1.1. (Function Parameter usage) Relative Offset: The absence of save/restore pointers (only modify) would appear to limit capabilities. The absence of relative offset means that DMA setup must be done by the FC-4 layer (more likely to be firmware) rather than the FC-2 layer (more likely to be hardware), despite the fact that the capability will need to be built into the FC-2 layer.

X3T9.2/93-186r0

Unitrode comment on GPP (991D Rev 7):

Page 2; The Roadmap should be the same as the SAM roadmap. X3T9.2-994D Rev 12 page 9 is a clear roadmap and should be used on all documents that include a SCSI-III roadmap.

Page 10 paragraph 5; wording very poor. If the CDB block contains an error, the command is not executed. Last sentence report and does not have a space.

Date: Nov,7 1993

X3T9.2/93- r0

To: X3T9.2 Committee (SCSI)

From: George Penokie (IBM)

Subject: Comments on SBP

5.12. "...allocation of tap slots is done for advisory purposes..." This phrase should be removed. Section 5.9 requires the target to remove the slot from the available pool when it is allocated. This is more than an "advisory". (The phrase about not enforcing a given initiator to its allocation can be left in.)

5.12.3. On p. 11, "TAP SLOTS ALLOCATED FOR USE BY AN INITIATOR:" The last two sentences of this paragraph don't seem to make sense. I'm not sure whether they need to be removed or clarified.

7.1. P. 21, last paragraph, "TSA flag": Section 5.12.2, p. 12, "USE OF TAP SLOTS FROM THE GENERAL POOL OF TAP SLOTS" states that there are only "two" groups of slots: Those in the general pool and those tied to initiators. Item 3 refers to tap slots "with an allocation tied to a given initiator", however. Likewise, item 6 refers to tap slots tied to a given initiator. If section 5.12.2 is correct, then I believe that items 3 and 4 could be removed, and item 6 could be restated to say that "The tap slot available notice is given only to the initiator which used the slot."

7.2. "Residue" needs to be defined as the number of bytes which were not transferred for the SCSI command. (Refer to FCP) Also, the "REQUEST SENSE data (bytes 24 to n) should be interchanged with the "Response information" (bytes n+1 to m) to agree with FCP format.

The following comments are intended to make the hardware level retry mechanisms (i.e. Busy_x or Busy_a & Busy_b) which are to be documented in P1394 available to both targets and initiators in an identical manner. They are also intended to describe the result of either the target or the initiator returning a "resp_conflict" rcode.

9. According to my reading of P1394, the appropriate way for a target to reject a tap is to follow item d) of paragraph 5. (To "respond with a 'pending' acknowledge code and then generate a write response packet with 'resp_conflict rcode.") Items a)-c) should be removed. Instead of items a)-c), paragraph 2 could be modified as follows:

A tap shall meet all requirements of the IEEE P1394 standard for block write requests. This includes either the Single-phase or Dual-phase retry mechanisms as described in IEEE P1394.

(Note that these retry mechanisms are not yet fully documented in Draft 6.4v0, but drafts of the procedures were available on the "on line" documentation via ftp.)

10. Paragraph 3 should include the option for the initiator to follow the Single or Dual phase retry procedure of P1394. Paragraph 4 should clarify that if the initiator rejects the fetch WITH A RESP CONFLICT RCODE, then the target shall not fetch the rest of the chain.

11.1. Same comments as #10, above on paragraphs 3 & 4, respectively.

X3T9.2/93-186r0

11.2 Same comments as #10, above on paragraphs 3 & 4, respectively.

12. Same comments as #10, on paragraphs 6 & 7, respectively

13. Same comments as #10-- the initiator must be able to use either the Single or Dual-phase retry mechanisms. If the write is rejected by a resp_conflict rcode, then it could be stated that the target is not obligated to retry the operation.

Date: Nov,7 1993

X3T9.2/93- r0

To: X3T9.2 Committee (SCSI)

From: George Penokie (IBM)

Subject: Comments on FCP

Some comments for FCP revision 007. A small "e" after the number indicates that the comment is purely editorial.

- 1e. Section 4.2 paragraph 2, change UI to IU.
2. Section 6.1 paragraph 1: I believe should not refer to a "command pair." I think this should just say "command."
3. Section 6.1 paragraph 7 refers to a PRLI to an "image pair," while paragraph 1 indicates PRLI goes from an originator N_Port to a Responder N_Port--not an image pair. Perhaps what is said is correct but I think it needs clarification.
4. Section 6.1 should probably state that all SCSI commands associated with a process which logs out are aborted. It seems unclear from reading the document whether SCSI commands are able to span process logout/login cycles.
- 7.1.1 States that the entity address is vendor unique. If possible it would be better to refer to the RAID document on addressing, X3T9/93-161. Annex B, although informative, should also be updated to conform to that document.

X3T9.2/93-186r0

Date: Nov,7 1993

X3T9.2/93- r0

To: X3T9.2 Committee (SCSI)

From: George Penokie (IBM)

Subject: Comments on GPP

It is IBMs postion that another interface protocol is not needed; we have enough already.