Date: Feb 18, 1993 X3T9.2/93-035 rev 0

To: X3T9.2 Committee (SCSI)

From: Scott Smyers (Apple)

Subject: SBP Editor’s Meeting

These minutes document the SBP working group meeting which took place in Austin, TX on Thursday February 18, 1993.

Attendance:

Greg Floryance, IBM Rochester
Jeff Stai, WD
Jerry Marazas, IBM Boca Raton
Ed Gardner, DEC
John Lohmeyer, NCR
Larry Lamers, Maxtor
Scott Smyers, Apple Computer
Charles Monia, DEC
Mr. Drouot, Bull SA

As a point of administration, Ed Gardner was appointed "head" editor, meaning that he accepted the responsibility to fold all changes into the master copy.

We started the meeting by going over the specific feedback that Greg and others had on the text of the document. Below are some of the specific issues that came up. All input was transcribed onto a master marked up copy:

Section 3.1.1, and a general comment on the document - remove the word "intend", and replace the softer terms (such as "intend" with statements of requirement).

Do not mention that command and other FIFO’s "hold" things, and instead state that FIFOs are addresses to which things are written - for example, a FIFO is "a data structure within a target to which a tap packet is addressed" is a suggested wording that Greg Floryance provided for section 3.1.13.

Greg had another question regarding exactly what things can be written to each of the FIFOs in the standard. This brought us to a general discussion about the distinction between the normal and urgent command FIFOs. Jerry described this distinction, helped out by Ed and Scott, saying that tap messages sent to the urgent FIFO require immediate attention, in other words, the target must fetch commands for which a tap message is written to the urgent FIFO. This proved to be an item of confusion for several people present, so Ed launched into a chalkboard discussion on the subject.

Ed presented an "object model" of a Target/LUN. The initiator talks to a LUN through a target. The initiator addresses a LUN in parallel SCSI by sending an IDENTIFY message. In SBP, the tap messages are notifications to the target, while the contents of the command block identify the LUN thereby providing sufficient information for the target to enter the command into the appropriate task set. "Fetching" in SBP is defined as reading the command block from initiator space and entering it into the appropriate task set. The fetching engine is the target, the execution engine is the LUN. The urgent versus normal FIFOs affect the operation of the fetching engine (the target is obligated to fetch commands for which a tap is written to the urgent FIFO before fetching any available commands for which taps have already been written to the normal FIFO). The execution engine then uses the QueueType field in the command block to decide on execution order.

There was some additional discussion on this point, but the distinction between a fetch engine and an execution engine proved to be a useful tool. Further discussion brought the notion of sub-chains into view. The fetch engine has requirements placed on it for the order in which it fetches commands. Unfortunately, a less than fully documented element of the protocol surfaced. Namely, what happens if someone sets the end sub-chain bit in an urgent chain? We kicked this one around, and the simplest suggestion came from Ed - if the end of sub-chain bit is set, the target goes on to its fetch order policy which includes the normal chains. Greg liked this one because he felt he could use the same hardware for fetching from the urgent FIFO as from all other FIFOs, except that all commands in an urgent chain up to one with the end sub-chain bit set must be fetched on an expedited basis. We then encountered the question of how tap slots are allocated.

The feeling was put forth that in addition to the general pool of tap slots from which allocations are made to initiators, there needs to be one urgent tap slot reserved for each initiator, and one tap slot for ACA commands. This opened up the issue for further discussion. After some time on this point, we were led to question the need to have an identifier for the initiator. Ed argued that for reserve/release, we needed an initiator ID. All present, with the exception of Jerry, agreed that an initiator ID was necessary and that login was necessary for all initiators in all environments. Furthermore, we agreed that there will be an additional encoding of the reg_type field in the command FIFO address that directs taps against the general pool of tap slots, instead of one of the tap slots dedicated to a given initiator. Now, bit 7 of the FIFO address determines if the tap slot comes from the reserved slots or not, and bit 6 determines if this is a normal or urgent tap. Greg will take the action item to write up what we agreed to as a proposal for inclusion into the document. Jerry will do some more thinking about an alternative to the login protocol which could provide a simplification for single initiator environments.

Back to editorial issues, a general comment was made that the ESC bit should be renamed because of the confusion with the normal use of that abbreviation (i.e., ESCape). ESC was changed to SCE (Sub Chain End).

There was some discussion on section 5, but it was observed that this section needs to be reworked to include much of the concepts presented in the SCSI-3 queuing model. Greg did, however, cite some examples of sentences which were simply inaccurate. Greg brought attention to those areas for emphasis. Jeff noted a confusion over how the SCE and M bits are treated when the L bit was set. An action item was taken to make the information on this subject consistent with the description of it in the section on bit definitions. On a general note, Jerry had previously volunteered to take responsibility for this section (the device model). Therefore, Jerry took an action item to rework this section keeping in mind the input received at this meeting.

Greg had some additional editorial changes which he fed to the editors. Most of them had to do with loose wording (instead of statements of requirement) and inconsistent use of terminology.

Charles had a question of when the initiator can reuse the resources occupied by a command following an abort. The abort mechanism was reviewed, and it was suggested that we, at some later time, revisit this protocol to come up with a way to release the data buffers referenced by an aborted command before actually returning status indicating that the command itself was encountered
and aborted. The second event would mark the point in time at which the command block and sense buffer could be reused.

Charles also asked what happens when a clear queue occurs - does the target return status for all aborted commands, or does it simply acknowledge the clear queue and make a gentleman’s agreement that the initiator can reuse the command blocks? This is an area that is not completely documented in the standard, and Ed offered to focus on this issue for a future meeting.

Action items:

Jerry - Rework and enhance the model section (section 5 in rev 8)
Jerry - write a proposal for a default initiator ID which does not require log in and draws only from the general pool of tap slots
Greg - create some text describing the principles of operation and concepts of command FIFOs and tap slots; this will be a candidate for eventual inclusion in the SBP document
Ed - write up details of the clear queue operation
Ed will become the "clearing house" for submissions to the SBP standard