To: X3T9.2 Membership X3T9.2/93-084r0

From: SMYERS.S@APPLELINK.APPLE.COM

Date: April 22, 1993

Subject: Minutes of the 4/22/93 SBP working group

Serial Bus Protocol Meeting (4/22/93) minutes

The Serial Bus Protocol meeting took place on Thursday 4/22/93 at the Tradewinds Hotel in St. Petersburg Florida. The meeting convened at 9AM.

Attendance
Dave Webber, Cirrus Logic
Alan Wetzel, TI
Jerry Marazas, IBM
Jeff Stai, WD
Greg Floryance, IBM
George Penokie, IBM
Jim McGrath, Quantum
Larry Lamers, Maxtor
John Lohmeyer, NCR
Scott Smyers, Apple Computer
Brian Bell, TI
Erich Oetting, Storage Tech

Document Distribution
Revision 10i of the SBP document was distributed at the meeting. The " i" in the revision number indicates that this is an intermediate release. Jerry Marazas was the only one of the three editors who was able to work on the document since last time. Jerry actively solicited input from the other two editors and did his best to incorporate that input into this latest release. However, there were some contributions from Scott Smyers and some contributions from Ed Gardner which there was not time to incorporate into this latest revision, hence the "i" designation.

High Level Overview
Jerry began the meeting with a high level overview of the protocol. Starting with the configuration ROM, the config ROM contains some information about the device. For the purposes of the SBP protocol, there is one entry in the ROM containing the 64 bit address of the "log in" command FIFO. The initiator parses the config ROM to find that address. Using that address, the initiator issues a command tap containing either an asynchronous login request command block, or an isochronous login request command block. In response to the login request, the target returns: For the asynchronous login request: Address of the normal command FIFO Address of the urgent command FIFO Address of the ACA command FIFO Initiator ID value (8 bits) For the isochronous login request: Address of the isochronous command FIFO Address of the Isochronous Control Register Stream ID value (8 bits) This information is all that the initiator needs to deliver commands and receive status from the target, including error handling. Depending on target implementation, the target could assign different addresses for each of the command FIFOs for each logged in initiator. The target could also assign the same address for all command FIFOs. Jim McGrath asked if the target is required to verify that the initiator is writing the correct commands to the correct command FIFOs. The answer is that the command block itself contains a field which identifies it as an isochronous, normal, ACA etc., command block, so the address to which a command block is written is duplicated by the settings of the fields in the command block itself. The SBP standard places requirements on the initiator regarding the usage of the different command FIFOs, but the standard does not require the target to verify that the initiator is acting correctly in this regard. For example, a target could reject an

isochronous command written to the ACA command FIFO, but if the command block is unambiguous, the target could also accept it and execute it as normal. Both options are intended to be spelled out in the standard, if they are not already, and both are valid target implementations.

Jerry next described the format of the command block. The command block contains: 8 bytes - Address of the next command in the chain, if any 8 bytes - Address of this command block 4 bytes - LUN/initiator ID or stream ID/reserved - 16 bits are reserved for the LUN, 8 bits are reserved for the initiator ID or stream ID and 8 bits are reserved 4 bytes - codes/flags - contains the command block type, priority and function, sub-chain end, more flag, etc., etc. 16 bytes - CDB 4 bytes - transfer length 4 bytes - Xfer control/reserved/sense buffer length 8 bytes - data buffer address 8 bytes - status FIFO address 8 bytes - sense buffer address.

Jim McGrath asked if 8 bits was enough for the initiator ID. Those present felt that it was enough. His points included that because a single initiator can log in more than once, IDs could get gobbled up fairly quickly. We left it at 8 bits, but Jerry noted that to the left of the initiator ID field is an 8 bit reserved field, so as of now, the initiator ID could grow to 16 bits if necessary without too much difficulty. Jim also asked about the location of the "next" and "this" command addresses. He maintained that his hardware would be simpler if the next command address were placed at the end because he could retain that in his hardware and walk the chain more efficiently.

After some discussion, we took a straw pole to gauge people’s opinions on the location of the next command address. Next command address should be the first 8 bytes of the command block, as it is now - 4 in favor Next command address should be the last 8 bytes of the command block - 3 in favor Larry noted that Ed Gardner preferred having the next command address at the top, but unfortunately, Ed wasn’t here. The group concluded that there was no consensus. Jim was asked to write up his arguments for putting the next command address at the end. Jim is to write this up, fax it to Scott Smyers who will put the argument on the SCSI reflector for discussion and in preparation for the next SBP meeting.

Jerry next did a quick overview of other parts of the standard that have changed since last time. One of these changes addresses target behavior when its resources are exhausted and it can’t take a tap packet. The current draft allows for ’pending/response conflict’ or busy in order to reject a packet. After some discussion, the group agreed that the spec should permit 2 types of implementations: 1. Target always issues a pending followed by a success or response conflict write response (it may need to do a busy in some cases) 2. Target always acks complete or busy and never issues a pending By extension, a target could implement option 1, but sometimes issue an ack complete (i.e., not always issue a pending). Note that these elements of 1394 are referenced in the SBP document as acceptable methods to reject a tap packet when necessary. The question was raised of how many of the other 1394 elements the SBP document must reference (ack = data CRC error, for example). The goal of the SBP document is to avoid restating the 1394 standard, but this question will have to be kept in mind as we proceed. This concluded the high level overview of the things that have changed.

Document review

Jerry made substantial changes to the SBP document since the last time to reflect the agreements reached at the last SBP working group meeting. The first point was raised by Jim McGrath, who had a problem with Jerry’s reference to a target allocating tap slots. Jerry confirmed that this did not indicate a requirement for a target to allocate and enforce its resource of tap slots, but that some management tools are available in this standard. In a minimal target implementation, enforcement of tap slot allocation would be placed on the initiators cooperating and not exceeding their tap slot allocation. However, in a more elaborate target implementation, the target would have to enforce the initiator allocations. It was noted that the tap slot management tools were put in to satisfy the needs of those who felt that target enforcement of the division of resources among multiple initiators was a necessity. Jim McGrath suggested that we take out the tap slot management stuff in favor of the proposal that Ed is writing for tap slot management, which appears to be headed in a different direction from what’s in SBP today. We took a straw poll to test the opinion of those present on this subject. Who doesn’t want tap slot management in the SBP standard?

8 don’t like it, 0 like it, 5 didn’t vote.

The straw poll reflected that the consensus was to remove tap slot management from SBP. It was suggested that any proposal for managing queuing resources in the target among multiple initiators be presented to SAM and not SBP. Since Ed is known to be a significant proponent of tap slot management, the group decided to hold off a final decision until he is available to present his case. This topic spawned a discussion of the importance of the timeliness of closure of this standard. There was consensus that this working group should work toward coming up with an SBP document that is suitable for acceptance as a baseline document at the July plenary. The deadline for the mailing for the July plenary is June 10. This was set as a goal for the working group.

On a different subject, Jim McGrath observed the "shall"s in the document and suggested that we indicate who has the burden of enforcing the requirements that the shall’s imply. Jim suggested that we include an indication of who is obligated to obey a rule, and who is responsible for enforcing it or reporting it as an error, if anyone. Future editing will be done with this suggestion in mind. Isochronous control register, Jim and Jeff had some input on this. The primary difficulty that they had was that this was an 8 byte register, which is larger than all other CSR registers (which are all 4 bytes). Also, there was no mention of whether this is a read register, and Jim had a particular problem with write only registers. Jeff suggested that we either make this a read/write register, or we make it a message. Making this a message would require us to turn the ICR writes into 64 byte packets, if we follow the convention of the rest of the SBP standard. Out of this discussion, the following recommendations/suggestions materialized:

- o There will be one unique ICR per isochronous stream
- o The ICR should be set of registers, not one 8 byte register. The multiple registers would facilitate the separation of the mostly static elements (like channel number) into one quadlet register, and the action fields (start, pause, etc.) in another quadlet register. Note that the stream ID field would no longer be needed because there is now a requirement that there be one ICR register set per stream.
- o Make the cycle number into 32 bits, not just 20 bits After some additional considerations and discussion, Scott took the action item to seek input from people at Apple and craft a proposal to SBP, or come back with some additional information for discussion. Meeting schedule

The following is the proposed 1394 and SBP meeting schedule:

Location	1394	SBP
----------------	------	------
St. Petersburg	4/21	4/22
Santa Clara	5/6-7	
Santa Fe, NM	5/20	see below
Irvine	6/2-3	
Minneapolis	6/23-24	
Santa Clara	7/14-15	
Manchester NH	7/20	7/20 or 21
Colorado Springs	8/18	8/19
Poughkeepsie	9/14	9/15
Ft Lauderdale	10/13	10/14
Colorado Springs	11/9	11/10

For the Santa Fe SBP meeting, the group voted on several options:
1. Friday 5/21, 9 to 2PM - 3 yes
2. Tuesday 5/18 evening 7 to 10PM - 6 yes
3. Tuesday and Wednesday 5/18 and 5/19 7 to 10PM - 1 yes The consensus was to shoot for a meeting on Tuesday evening, assuming we can find a meeting room.

In case we can’t get a room, the group decided on the following fallback alternatives:

1. Thursday night - 5 yes
2. Friday morning - 3 yes

The fallback, then, would be to have an SBP meeting on Thursday. Jerry will look into meeting room availability in Santa Fe and put a note on the reflector.