From: **Scott Smyers**
Sony US Research Labs

Date: 02/13/2001

1 Proposal for modifications to SBP3 and RBC

This proposal describes a means of utilizing modified RBC commands and the SBP3 protocol to access data on an AVHDD that was recorded isochronously using the AVC command set for the AVHDD subunit.

2 Background

This document describes a proposal for modifications to the SBP3 protocol and the RBC command set standards. This document does not propose any modifications to any AVC standard.

An HDD compliant with this proposal implements, without modification, the AVC command set for the AVHDD subunit. Such an HDD, if connected to a controller capable of recognizing an AVHDD subunit and implementing the corresponding AVC command set, behaves in all respects as a compliant AVHDD subunit. The AVC controller, if using only the AVHDD subunit AVC command set and FCP, has no means of determining that this is anything other than a compliant AVHDD subunit.

The modifications to SBP3 and RBC enable asynchronous access to data that was recorded isochronously using AVC.

3 Details of Proposal

This section describes the technical details of this proposal. Please note that this material was prepared prior to open discussions in the SBP3 working group, and the author intends that this material will be altered over time as a result of those open working group discussions. In particular, this proposal describes new management ORBs; however, the author recognizes that the SBP3 working group has already tentatively decided to define new RBC commands that serve the function of these new management ORBs. It is anticipated that the editor of the RBC standard will review this proposal and construct new RBC commands that perform the function of these management ORBs. These RBC commands then, after complete review of the SBP3 working group, are intended to be proposed to the RBC committee at some point in the future.

New Management ORB:

This document defines a new management ORB called the “Manage Asynchronous Space” ORB. This ORB may only be delivered to a target LUN by a logged-in initiator. This ORB has the following format:

0	Undefined				
4	Undefined				
8	Undefined				
C	Undefined				
10	N	Rq	Reserved	fcn	Sub-function
14	Sub-function dependent				

18	Status FIFO addr
1C	

The first 4 quadlets are not defined.

The next quadlet contains several fields. The n and rq_fmt fields are as defined elsewhere in the SBP2 standard. The function field (abbreviated “fcn” in the table) contains a newly defined value indicating that this is a “Manage Asynch Space” ORB as described here (such a value for the function field is not defined in the current SBP2 standard; this document proposes taking one of the reserved values and assigning for this purpose).

The sub-function field exists for the manage asynchronous space ORB, and the possible values for this field are as defined in the following table:

Value	Meaning	Definition of Sub-Function Dependant field
0	Create asynchronous space	Size request (in LBAs)
1	Delete asynchronous space	Asynchronous space number
2	Query asynchronous space	Not used
3-FF	Reserved values	

Note that the sub-function field may be something other than 16 bits wide and the asynchronous space numbers could be something other than 32 bits wide without affecting the operations defined here.

The “create asynchronous space” sub-function requests the target to create an asynchronous space of the size contained in the sub-function dependent field. This size is expressed in units of Logical Blocks. The response for this sub-function contains the status of the operation (success or failure). If the response block contains the success status, then the response block also contains the number of the newly created asynchronous space. If the response block contains the failure status, then the response block also contains the total number of LBAs which can be assigned to an asynchronous space. The response block may contain additional information. When assigning numbers for asynchronous spaces, the target shall assign them from the same space as that used for AVC defined track numbers. During operation under the SBP2 and FCP/AVC protocols, the target shall assure that there is never an asynchronous space which has the same number as an AVC track, and that there is no AVC track which has the same number as an asynchronous space.

The “delete asynchronous space” sub-function requests the target to delete the asynchronous space whose number is contained in the sub-function dependent field. The target returns a response block indicating the status of the operation (success or failure).

The “query asynchronous space” sub-function requests the target to return the numbers of currently existing asynchronous spaces. The format of the data returned for this request is as follows:

Number of asynchronous spaces
Asynchronous space number
Asynchronous space number
Asynchronous space number
…

The first quadlet contains the number of asynchronous spaces, followed by that many quadlets. Each succeeding quadlet contains the number of an existing asynchronous space.

Modification to command block ORB:

Command block ORBs containing a value of 0 in the rq_fmt field shall operate on the default asynchronous space, which is defined as the asynchronous space with the numerically lowest asynchronous space number. Reads, writes and other operations operate as defined in SBP2 and the relevant command set standard.

Command block ORBs containing a value of 1 in the rq_fmt field contain, in the first quadlet of the command block field, an asynchronous space number. The actual command follows this asynchronous space number field. Such commands operate on the asynchronous space having the specified asynchronous space number, according to the details of operation of the command contained in the ORB following the asynchronous space number.

Operation:

Asynchronous spaces are logically contiguous storage areas that are accessed as a flat address space organized as Logical Blocks. Each asynchronous space begins with logical block address (LBA) 0 and continues in a contiguous flat address space to the final LBA of the asynchronous space. The value of this final LBA is a function of the size of the asynchronous space.

No AVC command shall modify any portion of an asynchronous space. However, the AVC play command may specify a track number which is the same as the number of an asynchronous space. In this case, the AVHDD shall treat that asynchronous space as if it were an AVC track, and it shall attempt to play this asynchronous space according to the rules of the AVC play command. The initiator that stored the content contained in such an asynchronous space is responsible for assuring that the data contained there is of a format that, if played, results in a meaningful isochronous stream.

An SBP2 initiator may direct read or write commands to an AVC track. To do so, the initiator places a valid AVC track number in the asynchronous space field of an ORB containing a value of 1 in the rq_fmt field. In this case, the AVHDD shall use the LBA in the command in the ORB and address the AVC track as if it were an asynchronous space. While no asynchronous space shall be modified by an AVC command, an AVC track which is modified by SBP2 delivered commands shall retain its ability to be played or modified using AVC commands.

In no event shall the AVHDD permit SBP2 commands to access, in any way, tracks which were recorded via AVC as copy protected tracks.